# Web Application Security Assessment Report

**Target:** Damn Vulnerable Web Application (DVWA)
**Assessment Type:** Web Application Penetration Testing
**Tester Role:** VAPT Intern
**Tools Used:** Burp Suite, sqlmap, Draw.io, google docs
**Date:** 16-01-2026

## Executive Summary

This assessment was conducted to identify common web application vulnerabilities in a controlled DVWA lab environment, aligned with the OWASP Top 10 framework. The testing revealed critical security weaknesses, including SQL Injection and Reflected Cross-Site Scripting (XSS), which could allow attackers to compromise sensitive data and user sessions. Manual and automated testing techniques were used to validate findings. Immediate remediation is recommended to reduce the risk of data leakage, unauthorized access, and potential system compromise.

## Scope and Methodology

### Scope

- **Target Application:** DVWA
- **Target URL:** http://192.168.0.9/dvwa
- **Environment:** Local lab setup (DVWA VM + Kali Linux)

### Methodology

Testing followed a black-box approach using:

- Automated scanning (sqlmap)
- Manual request interception (Burp Suite)
- OWASP Top 10 testing techniques
- Session handling analysis

## Technical Findings

### Finding F001: SQL Injection

- **Severity:** Critical
- **CVSS Score:** 9.1
- **Tool Used:** sqlmap
- **Affected Module:** DVWA SQLI

**Description:**
A SQL Injection vulnerability was identified in the DVWA login functionality. By intercepting the HTTP request using Burp Suite and saving it as **request.txt** file, the request was replayed using sqlmap, allowing enumeration of backend databases and tables.

**Commands Used:**

- *sqlmap -r request.txt --dbs --batch*
- *sqlmap -r request.txt -D dvwa --tables*

**Impact:**
An attacker can extract sensitive database information, bypass authentication, and potentially escalate to full system compromise.

### Finding F002: Reflected Cross-Site Scripting (XSS)

- **Severity:** Medium
- **CVSS Score:** 6.1
- **Affected Module:** DVWA Reflected XSS

**Description:**
User input was reflected back to the browser without proper sanitization. Injecting a simple JavaScript payload triggered execution in the victim's browser.

**Payload Used:**
*<script>alert(1)</script>*

**Impact:**
Attackers can steal session cookies, perform phishing attacks, or execute malicious scripts in a victim's browser.

### Finding F003: Session Token Hijacking (Simulated)

- **Severity:** High
- **Technique:** Session Fixation / Hijacking
- **Tool Used:** Burp Suite

**Description:**
Login requests were captured in Burp Suite's HTTP history. The session cookie was copied and reused in Burp Repeater, successfully simulating session hijacking.

**Impact:**
An attacker with access to a valid session cookie can impersonate legitimate users without credentials.

## Vulnerability Findings Log

| Finding ID | Vulnerability | Severity/CVSS | Target URL | Remediation |
|---|---|---|---|---|
| F001 | SQL Injection | Critical/9.1 | http://192.168.0.9/dvwa/vulnerabilities/sqli | Input Validation |
| F002 | Reflected XSS | Medium/6.1 | http://192.168.0.9/dvwa/vulnerabilities/xss_r | Output encoding |
| F003 | Session Hijacking | High/8.2 | http://192.168.0.9/login | Secure cookie handling |

## Network Visualization

A network attack path diagram was created using **Draw.io** to illustrate:

- Attacker (Kali Linux)
- DVWA Web Server
- Intercepted traffic via Burp Suite
- Attack flow from request interception to exploitation

## Remediation Plan

- Implement server-side input validation and parameterized queries
- Encode and sanitize all user-supplied input
- Set cookies with **HttpOnly** and **Secure** flags
- Enforce proper session expiration and regeneration
- Conduct regular security testing and code reviews

## Web Application Test Summary

The DVWA application was tested for common OWASP Top 10 vulnerabilities. Critical SQL injection allowed full database enumeration using sqlmap, while reflected XSS was confirmed through manual payload injection. Session hijacking was simulated by reusing stolen cookies, demonstrating weak session management controls.

## Non-Technical Summary for Management

The security assessment of the DVWA application revealed multiple high-risk vulnerabilities that could allow attackers to access sensitive data and hijack user sessions. A critical SQL Injection flaw enabled full database access, while Cross-Site Scripting and weak session handling increased the risk of user impersonation. These issues, if present in a real-world application, could lead to data breaches and reputational damage. Addressing these weaknesses through secure coding practices, input validation, and regular security testing will significantly reduce organizational risk and strengthen the application’s security posture