

# Web Application Vulnerability Assessment Report

## Internship Project - Cybersecurity | Future Interns

**Project Title:** Web Application Security Testing on DVWA (Damn Vulnerable Web Application)

**Name:** Yatin Annam

**Platform Used:** Kali Linux (via VirtualBox)

**Target Application:** DVWA (Docker Container)

**Tools Used:** Manual Browser Input, Docker, Kali Linux

### - Objective

To conduct a security assessment of a sample web application (DVWA) by identifying at least 3-5 real-world vulnerabilities that map to the OWASP Top 10 threats. The vulnerabilities tested include SQL Injection, Cross-Site Scripting (XSS), Broken Authentication, and CSRF.

### - Setup Summary

- **Operating System:** Kali Linux (running in VirtualBox)
- **DVWA Installation:** Launched via Docker (vulnerables/web-dvwa image)
- **Database:** Auto-configured within DVWA Docker container
- **Browser:** Firefox (Kali default)
- **Security Level in DVWA:** Set to "Low" to allow easy testing of vulnerabilities

### - Vulnerability 1: SQL Injection

**Tested Page:** DVWA → SQL Injection

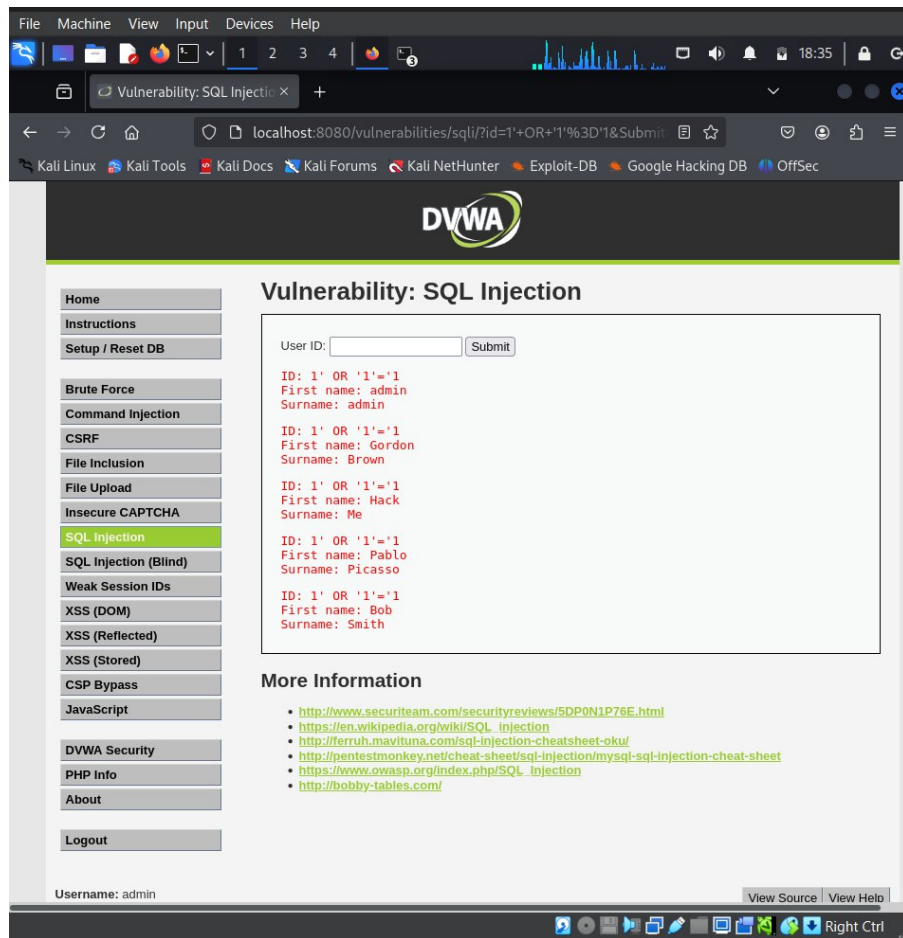
**Payload Used:** 1' OR '1'='1

**Result:** Successfully retrieved admin user data by injecting into the "User ID" field. This proves that input validation is absent and database queries are vulnerable to injection.

**OWASP Mapping:** A1: Injection

**Impact:** High

**Mitigation:** Implement parameterized queries and input sanitization.



## - Vulnerability 2: Reflected XSS

**Tested Page:** DVWA → Reflected XSS

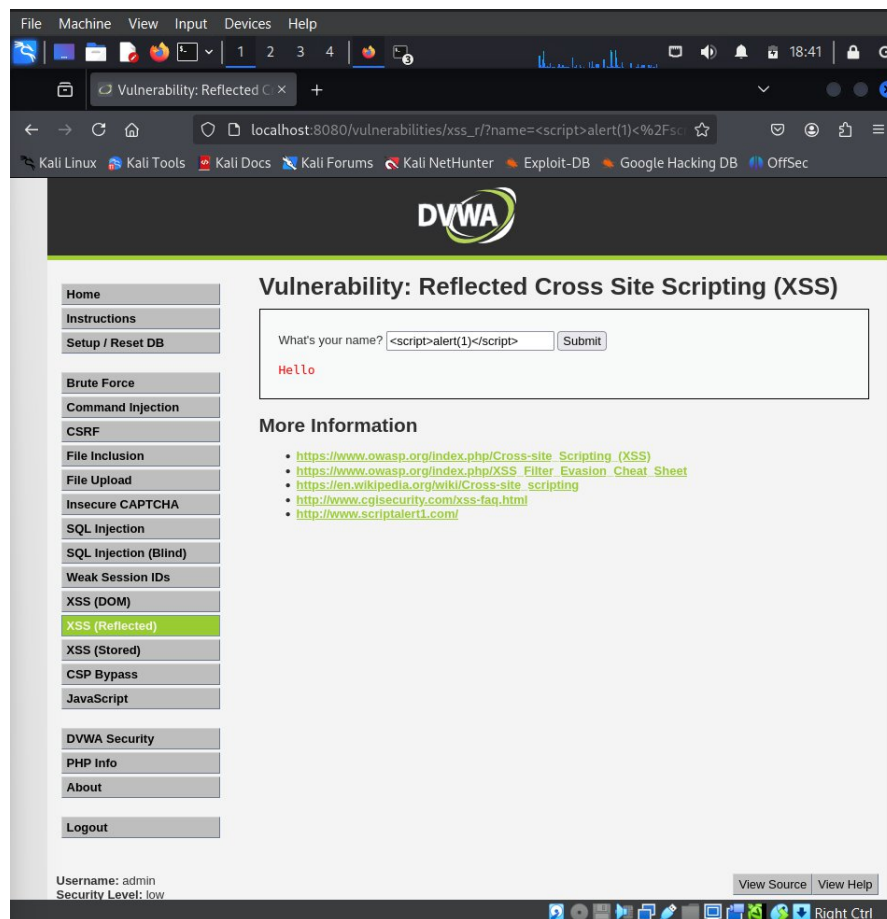
**Payload Used:** <script>alert('Hello')</script>

**Result:** JavaScript executed in browser alert, proving XSS vulnerability via user input. Reflected payload was not properly encoded.

**OWASP Mapping:** A7: Cross-Site Scripting

**Impact:** Medium to High

**Mitigation:** Escape all user inputs on output; implement Content Security Policy (CSP).



### - Vulnerability 3: Broken Authentication

**Tested Page:** DVWA → Brute Force

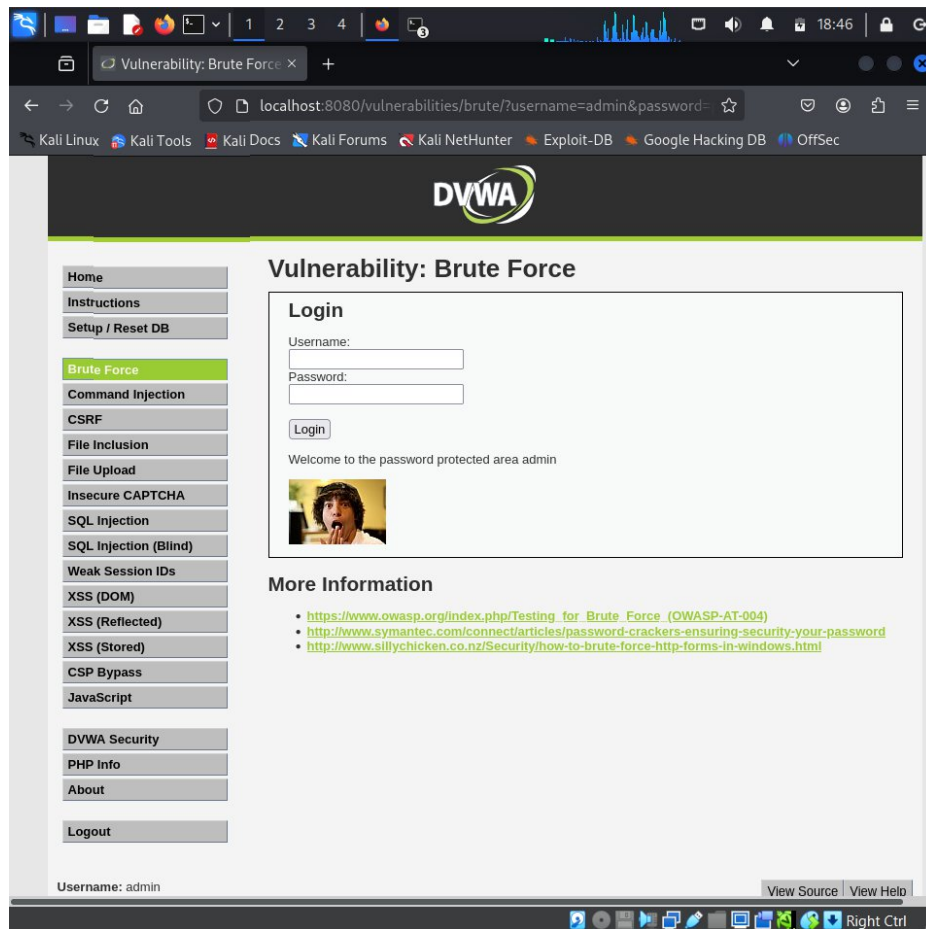
**Test Attempt:** Username: admin | Password: password

**Result:** Successfully logged into admin panel without rate limiting or CAPTCHA, confirming weak credentials were accepted.

**OWASP Mapping:** A2: Broken Authentication

**Impact:** High

**Mitigation:** Enforce strong password policies and login rate-limiting mechanisms.



## - Vulnerability 4: CSRF (Cross-Site Request Forgery)

**Tested Page:** DVWA → CSRF

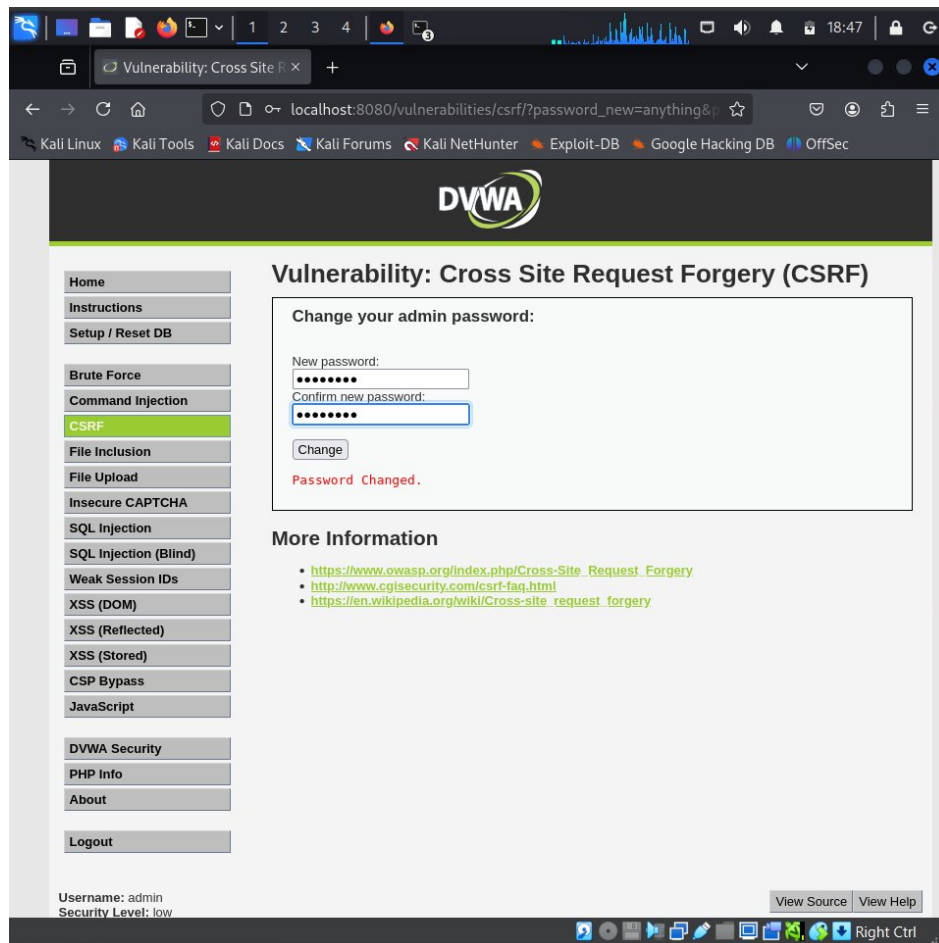
**Test:** Changed admin password without confirming old password or session validation.

**Result:** Password changed silently; no CSRF token or authentication checks.

**OWASP Mapping:** A5: Broken Access Control / Insecure Design

**Impact:** High

**Mitigation:** Implement CSRF tokens and validate session identity for critical requests.



## - Conclusion

The above assessment successfully demonstrated 4 core web application vulnerabilities using DVWA, mapped directly to the OWASP Top 10. This hands-on experience helped reinforce understanding of web application attack vectors and defense strategies. Each issue found includes screenshots, technical impact, and recommended mitigations.

*Report Prepared by: Yatin Annam*  
*Cybersecurity Intern – Future Interns*