**Web Application Security Testing Report**

Submitted By: Atif Mehmood

Course: Cyber Security

Instructor: Future Interns

Submission Date: 20 July 2025

# Project Task

Task: Conduct security testing on a sample web application to identify vulnerabilities like SQL injection, XSS, and authentication flaws.

Skills Gained: Web application security, ethical hacking, penetration testing.

Tools Used: OWASP ZAP, Burp Suite, SQLMap

Deliverable: A detailed security report with identified vulnerabilities and mitigation strategies.

# Scope of Testing

Target Web App: DVWA (Damn Vulnerable Web Application) - hosted locally on XAMPP

Testing Window: 3 Days

Tested Areas: Login form, comment section, search bar, user profile settings

**Tools & Environment** Tools Used:

* Burp Suite: Intercepting, modifying requests, scanning
* SQLMap: Automating SQL Injection testing
* OWASP ZAP: Passive and active scanning

System Used:

* OS: Kali Linux (VM)
* Browser: Firefox Developer Edition
* Target App: DVWA (Security Level: Low & Medium)

# Methodology

1. Reconnaissance & Scanning: Crawled website using OWASP ZAP; mapped out all forms and parameters
2. Manual Testing: Used Burp Suite for modifying requests; identified injection points
3. Automated Testing: Used SQLMap and payloads for deeper detection

# Vulnerabilities Found

1. SQL Injection - Login form bypassed using `' OR '1'='1` (High)
2. Reflected XSS - Search bar executed JS with <script> payload (Medium)
3. Weak Authentication - No brute-force protection or account lockout (High)
4. Predictable Session IDs - Poor randomness in session tokens (Medium)

# Mitigation Strategies

SQL Injection: Use parameterized queries (e.g., PreparedStatements)

XSS: Sanitize inputs, use CSP headers

Weak Authentication: Implement account lockout, enable MFA

Session Weakness: Use cryptographically secure random tokens, enable secure cookie flags

# Skills Gained

* Identified OWASP Top 10 vulnerabilities in a live app
* Used Burp Suite & OWASP ZAP effectively
* Ran SQLMap to automate SQLi detection
* Reported vulnerabilities professionally

# Conclusion

Testing successfully identified high-risk vulnerabilities.

The report provides mitigation strategies for secure application development.

Ethical hacking tools proved essential in analyzing web app weaknesses.