Security Testing Summary

Week 5 – Security Testing & Vulnerability Analysis

Tested By: Areeba Khalid
Test Date: January 17, 2026
Application Tested: OWASP Mutillidae II v2.12.5
Testing Environment:

- Localhost (XAMPP)
- URL: http://127.0.0.1/mutillidae/
- Browser: Mozilla Firefox 147.0
- Tool Used: OWASP ZAP 2.17.0

1. Introduction

The purpose of this security testing activity was to identify common web application vulnerabilities based on the **OWASP Top 10 (2021)**. The testing focused on detecting weaknesses in input validation, authentication mechanisms, session management, and security configuration using both **manual testing techniques** and **automated scanning**.

The target application, OWASP Mutillidae II, is an intentionally vulnerable web application designed for security testing and educational purposes.

2. Scope of Testing

The following areas were included in the security testing scope:

- User input fields (login, lookup, echo pages)
- Authentication and session handling
- Logout and session invalidation
- Passive and active vulnerability scanning
- HTTP headers and security misconfigurations

Out of Scope:

- Denial of Service (DoS) attacks
- Brute-force attacks
- Production or real-world systems

3. Testing Methodology

The testing methodology followed a structured QA security approach:

❖ **Manual Testing**

- Input validation testing using special characters and payloads
- SQL Injection testing via user input fields
- Cross-Site Scripting (XSS) payload execution
- Session handling checks (logout, cookie reuse)

❖ **Automated Testing**

- Passive scanning using OWASP ZAP
- Alert review and severity classification
- Verification of detected vulnerabilities

All testing was performed ethically on a controlled environment.

4. Summary of Findings

A total of **4 significant vulnerabilities** were identified during the assessment.

Vulnerability ID	Vulnerability Name	Severity
VULN-001	SQL Injection – User Lookup	CRITICAL
VULN-002	Reflected Cross-Site Scripting (XSS)	HIGH
VULN-003	Session Not Invalidated on Logout	MEDIUM
VULN-004	Cross-Domain Misconfiguration (CORS)	MEDIUM

5. Key Observations

- The application lacks proper input validation and parameterized queries, leading to SQL Injection.
- User input is reflected without encoding, resulting in XSS vulnerabilities.
- Sessions remain active after logout, exposing users to session hijacking risks.
- Insecure CORS configuration allows cross-domain data access.
- Sensitive data such as passwords and client secrets are exposed in plaintext.

6. Risk Assessment

The overall security posture of the application is **HIGH RISK**, primarily due to:

- Critical SQL Injection vulnerability
- Exposure of authentication credentials
- Weak session management
- Missing defensive security headers

If exploited in a real-world environment, these issues could result in complete system compromise, data breaches, and legal consequences.

7. Recommendations

The following actions are strongly recommended:

- Implement prepared statements and parameterized queries
- Apply output encoding to all user-generated content
- Enforce proper session destruction on logout
- Restrict CORS policies to trusted domains only
- Use secure password hashing (bcrypt / Argon2)
- Add security testing into the CI/CD pipeline

8. Conclusion

This security testing exercise successfully demonstrated how common OWASP Top 10 vulnerabilities can be identified using standard QA security practices. The findings highlight the importance of secure coding, proper authentication handling, and regular security assessments.

By addressing the identified vulnerabilities, the application’s security posture can be significantly improved.