Web Application Penetration Testing Report

1. Executive Summary

This report presents the results of a web application security assessment conducted using OWASP ZAP (Zed Attack Proxy). The target application was scanned to identify potential security vulnerabilities that could be exploited by attackers. The assessment identified multiple high, medium, low, and informational-level vulnerabilities.

Overall, the application demonstrates **critical security weaknesses**, particularly related to injection and cross-site scripting (XSS). Immediate remediation is recommended to reduce the risk of exploitation.

Target URL: http://testphp.vulnweb.com
Tool Used: OWASP ZAP v2.16.1
Scan Date: 30 January 2026

---

2. Scope and Methodology

2.1 Scope

- Target: http://testphp.vulnweb.com
- Testing Type: Automated vulnerability scanning
- Tools:
- OWASP ZAP (Active & Passive Scan)

2.2 Methodology

The assessment followed OWASP testing principles and included: - Crawling and spidering the application - Passive scanning for misconfigurations and security headers - Active scanning for injection and client-side vulnerabilities - Risk classification based on OWASP standards

---

3. Risk Summary

A total of **18 unique alerts** were identified.

Risk Level	Number of Findings	Percentage
High	5	27.8%
Medium	4	22.2%
Low	3	16.7%

Risk Level	Number of Findings	Percentage
Informational	6	33.3%
Total	**18**	**100%**

---

4. Key Findings

4.1 High-Risk Vulnerabilities

4.1.1 Cross-Site Scripting (DOM-Based)

- **Risk Level:** High
- **Description:** DOM-based XSS allows attackers to inject malicious JavaScript into the client-side execution context. This can lead to session hijacking, data theft, or malicious redirection.
- **Impact:**
- Account compromise
- Sensitive data exposure
- Client-side attacks
- **Evidence:** Malicious payload executed via manipulated URL parameters.
- **Recommendation:**
- Implement proper input validation and output encoding.
- Use Content Security Policy (CSP).
- Avoid unsafe JavaScript functions such as `eval()`.

4.1.2 SQL Injection (MySQL, Oracle, SQLite)

- **Risk Level:** High
- **Description:** SQL Injection vulnerabilities allow attackers to manipulate database queries.
- **Impact:**
- Unauthorized data access
- Database modification or deletion
- Full system compromise
- **Recommendation:**
- Use prepared statements and parameterized queries.
- Apply strict input validation.
- Implement least privilege database access.

4.1.3 Reflected Cross-Site Scripting (XSS)

- **Risk Level:** High
- **Description:** Reflected XSS occurs when user input is reflected in HTTP responses without proper sanitization.
- **Recommendation:**
- Encode user input before rendering.
- Use secure frameworks and libraries.

4.2 Medium-Risk Vulnerabilities

4.2.1 Absence of Anti-CSRF Tokens

• **Impact:** Attackers can perform unauthorized actions on behalf of authenticated users.
• **Recommendation:**
• Implement CSRF tokens in all state-changing requests.

4.2.2 Missing Security Headers

• Content Security Policy (CSP) not set
• Clickjacking protection missing

Recommendation: - Add the following headers: - Content-Security-Policy - X-Frame-Options - X-XSS-Protection

4.2.3 XSLT Injection

• **Impact:** Potential server-side code execution.
• **Recommendation:**
• Validate and sanitize XML/XSLT inputs.

4.3 Low-Risk Vulnerabilities

4.3.1 Information Disclosure via HTTP Headers

• Server and technology details exposed (e.g., Server, X-Powered-By).
• **Recommendation:**
• Remove or obfuscate sensitive headers.

4.3.2 Missing X-Content-Type-Options Header

• **Recommendation:**
• Set `X-Content-Type-Options: nosniff`.

4.4 Informational Findings

• Authentication endpoints detected
• Charset mismatch
• GET used instead of POST
• User agent fuzzing results

These findings do not pose immediate risks but indicate areas for improvement.

---

5. Overall Security Assessment

Based on the findings, the application is classified as:

Security Risk Level: HIGH

The presence of multiple high-risk vulnerabilities indicates that the application is vulnerable to real-world attacks. Immediate remediation and re-testing are strongly recommended.

---

6. Recommendations and Mitigation Strategy

6.1 Short-Term Actions

- Fix SQL Injection and XSS vulnerabilities immediately.
- Implement secure coding practices.
- Apply security headers.

6.2 Long-Term Actions

- Conduct regular penetration testing.
- Implement Secure SDLC (Software Development Life Cycle).
- Perform code reviews and security audits.

---

7. Conclusion

This penetration testing assessment identified critical vulnerabilities that could significantly impact the confidentiality, integrity, and availability of the application. By implementing the recommended security controls and remediation steps, the organization can significantly improve its security posture.

---

8. Appendix (Tools & References)

- OWASP ZAP
- OWASP Top 10 (2021)
- CWE (Common Weakness Enumeration)