Web Application Security Testing Report

Prepared by: Washington Adiado

Date: March 30, 2025

Abstract

This report presents the findings of a web application security assessment conducted on the Damn Vulnerable Web Application (DVWA). The primary objective of this assessment was to identify security vulnerabilities such as SQL Injection (SQLi), Cross-Site Scripting (XSS), and authentication flaws. A structured penetration testing methodology was employed, utilizing tools like OWASP ZAP, Burp Suite, and SQLMap. The results revealed critical security risks, including SQL injection, session hijacking, and weak authentication mechanisms. This report provides detailed findings, risk assessments, and remediation recommendations to mitigate the identified vulnerabilities effectively.

1. Introduction

Web application security is crucial for protecting sensitive data and preventing unauthorized access. This assessment aimed to identify and analyze vulnerabilities within DVWA using ethical hacking techniques. The testing process followed a structured methodology, incorporating reconnaissance, vulnerability scanning, exploitation, and risk evaluation. The tools used for this assessment included OWASP ZAP, Burp Suite, and SQLMap.

2. Methodology

The security assessment followed a systematic approach:

- **Reconnaissance & Information Gathering –** Identified exposed services and gathered available system information.
- **Automated Scanning –** Used OWASP ZAP and Burp Suite to detect vulnerabilities.
- **Manual Testing & Exploitation –** Conducted targeted penetration testing to validate vulnerabilities.
- **Risk Evaluation –** Assessed the likelihood and impact of each vulnerability.
- **Remediation Recommendations –** Provided actionable solutions to mitigate risks.

3. Findings & Analysis

3.1 OWASP ZAP Scan Results

Vulnerability	Risk Level	Description
SQL Injection	High	Injectable parameters found in login and search fields.
Cross-Site Scripting (XSS)	High	Identified reflected and stored XSS vulnerabilities.
Security Misconfigurations	Medium	Missing HTTP security headers.
Insecure Direct Object References (IDOR)	Medium	Unauthorized data access by modifying URLs.

3.2 Burp Suite Analysis

- **Session Hijacking:** Weak session management exposes users to hijacking risks.
- **Broken Authentication:** Lack of account lockout mechanisms makes the login page vulnerable to brute-force attacks.
- **Unvalidated Redirects:** Open redirects could be exploited for phishing attacks.

3.3 SQLMap & Nmap Scan Results

Port	Service	Vulnerability
22 (SSH)	Open	Risk of brute-force attacks.
80 (HTTP)	Open	No security headers, making it vulnerable to MITM attacks.
3306 (MySQL)	Open	Direct database exposure increases security risks.

4. Risk Assessment & Impact

Risk	Impact
SQL Injection	Attackers can manipulate database queries, steal data, or gain system access.
Cross-Site Scripting (XSS)	Can lead to credential theft and session hijacking.
Security Misconfigurations	Increases exposure to unauthorized access and exploitation.
Insecure Authentication	Higher chances of account takeovers.

5. Remediation Recommendations

5.1 Application-Level Fixes

- Use parameterized queries and Object-Relational Mapping (ORM) to prevent SQL Injection.
- Implement input validation and output encoding to mitigate XSS risks.
- Strengthen session security by enabling HTTPOnly, Secure, and SameSite flags.
- Enforce strong authentication policies, including Multi-Factor Authentication (MFA) and account lockout mechanisms.

5.2 Network & Server Security

- Restrict database access using firewall rules and VPN.
- Disable unused ports and services to reduce the attack surface.
- Implement a Web Application Firewall (WAF) to detect and block malicious requests.

5.3 Security Best Practices

- Conduct regular security assessments and penetration testing.
- Set up continuous monitoring and logging to detect threats in real time.
- Train employees on secure coding practices and phishing awareness.

6. Conclusion

This assessment identified several critical security vulnerabilities in DVWA that could be exploited by attackers. By adopting secure coding practices, implementing robust authentication mechanisms, and maintaining continuous security monitoring, these risks can be significantly mitigated.

7. Next Steps

✓ Immediate remediation of high-risk vulnerabilities.

✓ Continuous monitoring using automated security tools.

✓ Security training for developers and IT staff.