📄 WEB APPLICATION PENETRATION TESTING REPORT

(OWASP Top 10 Based)

Conducted by

Name: Prajwal V
Branch: Computer Science and Engineering (CSE)
College: PES University

Tools & Technologies Used

- Kali Linux (Virtual Machine)
- Damn Vulnerable Web Application (DVWA)
- Apache Web Server
- MariaDB (MySQL)
- PHP
- Web Browser (Firefox)

Methodology Followed

- OWASP Top 10 Web Application Security Risks
- Manual testing in a controlled lab environment

Disclaimer

This penetration testing activity was conducted strictly for **educational purposes** in a **controlled local lab environment** using a deliberately vulnerable application (DVWA). No unauthorized systems or real-world applications were tested.

🔍 1. Introduction

Web applications are widely used in modern systems and often store sensitive user and business data. Due to improper input validation, weak authentication mechanisms, and insecure configurations, many web applications become vulnerable to attacks.

This project focuses on performing a **Web Application Penetration Test** using **OWASP Top 10 methodology** on a deliberately vulnerable application. The objective is to identify, exploit, and document common web application vulnerabilities in order to understand their impact and recommended mitigations.

This project also builds upon **network reconnaissance skills** demonstrated in a previous project, where exposed services were identified before testing the application layer.

🎯 2. Scope of Testing

In Scope

- Web application hosted locally (DVWA)
- Application-level vulnerabilities
- Authentication, input handling, and configuration issues

Out of Scope

- Real-world websites
- Denial-of-service attacks
- Exploitation beyond proof-of-concept

⚙️ 3. Test Environment

Component	Details
Operating System	Kali Linux (VM)
Web Server	Apache
Database	MariaDB
Application	DVWA
Security Level	LOW

🧪 4. Vulnerability Findings

4.1 SQL Injection

OWASP Category

Injection

Description

The application fails to properly sanitize user input before using it in SQL queries. This allows attackers to manipulate database queries.

Proof of Concept

Input used:

1' OR '1'='1

Result

The application returned **multiple user records**, demonstrating unauthorized access to database contents.

Impact

- Disclosure of sensitive user data
- Authentication bypass
- Potential full database compromise

Recommendation

- Use prepared statements and parameterized queries
- Validate and sanitize all user input

4.2 Reflected Cross-Site Scripting (XSS)

OWASP Category

Cross-Site Scripting

Description

User input is reflected back in the HTTP response without proper output encoding, allowing JavaScript execution.

Proof of Concept

<script>alert(1)</script>

Result

A JavaScript alert popup was executed in the browser.

Impact

- Session hijacking
- Cookie theft
- Malicious redirection

Recommendation

- Encode output before rendering
- Implement Content Security Policy (CSP)

4.3 Broken Authentication

OWASP Category

Identification and Authentication Failures

Description

The application allows login using weak and default credentials and lacks account lockout or rate-limiting mechanisms.

Proof of Concept

Username: admin

Password: password

Impact

- Unauthorized account access
- Privilege escalation

Recommendation

- Enforce strong password policies
- Implement account lockout and rate limiting
- Enable multi-factor authentication (MFA)

4.4 Security Misconfiguration

OWASP Category

Security Misconfiguration

Description

Sensitive server configuration information is publicly accessible through the PHP Info page.

Evidence

- PHP configuration details
- Server paths
- Enabled modules

Impact

- Information disclosure
- Easier exploitation through system fingerprinting

Recommendation

- Disable PHP Info pages in production
- Restrict access to sensitive endpoints

4.5 Vulnerable & Outdated Components

OWASP Category

Vulnerable and Outdated Components

Description

The application discloses software component versions such as the web server.

Evidence

Apache/2.4.66

Impact

- Attackers can search for known CVEs
- Increased risk of targeted attacks

Recommendation

- Hide version banners
- Regularly update and patch components
- Monitor CVE advisories

4.6 Cross-Site Request Forgery (CSRF)

OWASP Category

Cross-Site Request Forgery

Description

Sensitive actions such as password changes can be performed without verifying the authenticity of the request.

Proof of Concept

Password changed without CSRF token validation.

Impact

- Unauthorized actions performed on behalf of users
- Account compromise

Recommendation

- Implement CSRF tokens
- Require re-authentication for sensitive actions
- Use SameSite cookie attributes

📊 5. Risk Summary

Severity	Count
High	SQL Injection, Broken Authentication
Medium	XSS, Security Misconfiguration
Low	Version Disclosure

🏁 6. Conclusion

This penetration testing assessment identified **multiple critical and medium-risk vulnerabilities** within the web application. Exploiting these issues could lead to unauthorized data access, account compromise, and information disclosure.

The project demonstrates a complete understanding of:

- Web application attack surfaces
- OWASP Top 10 vulnerabilities
- Ethical penetration testing methodology
- Professional security reporting

Addressing these vulnerabilities through secure coding practices and proper configuration would significantly improve the application's security posture.

📌 7. Future Enhancements

- Testing higher DVWA security levels
- Mapping vulnerabilities to CVEs
- Integrating Burp Suite for advanced interception
- Automating report generation

✅ END OF REPORT