# VULNERABILITY ASSESSMENT REPORT

## 1. Overview

Tools Used:

* OWASP ZAP
* Burp Suite
* Nmap

**Objective:** The objective of this assessment is to identify security weaknesses in the Damn Vulnerable Web Application (DVWA), evaluate potential risks, and recommend security enhancements. DVWA is intentionally designed for security testing and learning about common web vulnerabilities.

## 2. Methodology

The assessment followed a structured 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 | Found injectable parameters in login and search fields. |
| Cross-Site Scripting (XSS) | High | Identified reflected and stored XSS vulnerabilities in user input fields. |
| Security Misconfigurations | Medium | Missing HTTP security headers detected. |
| Insecure Direct Object References (IDOR) | Medium | Users can access unauthorized data 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 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 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 like 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 has 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.

### Next Steps:

✔ Immediate remediation of high-risk vulnerabilities.  
✔ Continuous monitoring using automated security tools.  
✔ Security training for developers and IT staff.

Prepared By: PAVAN KUMAR CHUNCHU  
Date: 20-03-2025