Final Security Assessment Report

# 1. Executive Summary

This report presents a security assessment of the Damn Vulnerable Web Application (DVWA), a purposefully vulnerable PHP/MySQL web application. The assessment focused on identifying common web application vulnerabilities using manual and semi-automated testing techniques. The findings demonstrate the existence of several critical security flaws that could be exploited by malicious attackers. Each vulnerability was tested and verified, and mitigation recommendations have been provided.

# 2. Scope and Tools

- Target: DVWA running in Docker on Kali Linux (localhost:4280)  
- Tools: Browser (Firefox), Terminal, Python HTTP server  
- Methodology: Manual testing of OWASP Top 10 vulnerabilities

# 3. Vulnerabilities Discovered

## 3.1 SQL Injection

The DVWA 'SQL Injection' module is vulnerable to both boolean-based and union-based injection techniques.

Payloads used:

* 1. ' OR '1'='1
* 2. 1' UNION SELECT null, version() #

Impact: Authentication bypass, database extraction.  
Risk: High

Recommendation: Use prepared statements (e.g., PDO, bind\_param in mysqli).

## 3.2 Reflected Cross-Site Scripting (XSS)

The input field in the 'XSS (Reflected)' module is vulnerable to JavaScript injection.

Payloads used:

* <script>alert('XSS')</script>
* <script>window.location='http://evil.com'</script>

Impact: Code execution in the user’s browser.  
Risk: High

Recommendation: Use output encoding and Content Security Policy (CSP).

## 3.3 Command Injection

The DVWA 'Command Injection' module allows OS-level command execution.

Payloads used:

* 127.0.0.1; whoami
* 127.0.0.1; ls -l

Impact: Remote command execution.  
Risk: Critical

Recommendation: Sanitize input and avoid unsanitized command execution.

## 3.4 Cross-Site Request Forgery (CSRF)

DVWA's password change form can be exploited using CSRF by auto-submitting a hidden form.

Payload crafted as a fake HTML page with auto-submit behavior.

Impact: Account takeover if user is logged in.  
Risk: High

Recommendation: Implement CSRF tokens, check Referer headers.

## 3.5 Insecure File Upload

The 'File Upload' module allows uploading of arbitrary PHP scripts and executing them.

Payload uploaded: shell.php (contained: <?php echo "Hacked by me!"; ?>)

Impact: Remote Code Execution.  
Risk: Critical

Recommendation: Validate file types and block execution of uploaded files.

# 4. Conclusion

DVWA is intentionally insecure, but this assessment shows how easily real-world applications can fall to OWASP Top 10 vulnerabilities. Developers must implement secure coding practices, input validation, output encoding, and proper session management to mitigate these risks.