

# DVWA – OWASP Top 10 Penetration Testing Report (2025 Edition)

Prepared By: [Your Name]

Date: [Insert Date]

Environment: Kali Linux (MySQL + Apache)

Target Application: DVWA hosted at <http://127.0.0.1/DVWA>

## 1. Executive Summary

This report documents a full penetration test conducted on the Damn Vulnerable Web Application (DVWA) environment to understand, demonstrate, and evaluate the OWASP Top 10 (2025) vulnerabilities. The objective of this assessment was to practice real-world exploitation techniques in a safe lab environment, analyze security weaknesses, and propose actionable mitigation strategies.

All attacks performed were strictly within the authorized DVWA environment running locally on Kali Linux.

The results reveal exploitable vulnerabilities across all categories of the OWASP Top 10, each demonstrated with practical testing steps, evidence (screenshots), and detailed remediation guidelines.

## 2. Scope of Assessment

### In-Scope Components

DVWA Web Application (<http://127.0.0.1/DVWA>)

MySQL Database (local)

Apache2 Web Server (local)

### Out-of-Scope

Any system outside local DVWA environment

External networks

Production systems

Testing Mode

DVWA Security Level: LOW

Tools Used:

Burp Suite Community

Browser Developer Tools

SQLMap (where applicable)

cURL/Nmap where relevant

### 3. Testing Methodology

Phase 1 — Reconnaissance

Enumerated application features

Observed request/response behavior

Identified parameters & input vectors

Phase 2 — Manual Exploitation

Each OWASP Top 10 category tested manually

Payloads crafted based on vulnerability type

Phase 3 — Proof of Concept (PoC)

Each exploit validated

Screenshot taken as evidence (insert where indicated)

Phase 4 — Risk Analysis & CVSS

Impact evaluated using CVSS 3.1

Phase 5 — Reporting & Recommendations

Technical and business-level remediation provided

### 4. OWASP Top 10 Testing & Findings

A01:2025 – Broken Access Control

Description

DVWA allows unauthorized access to restricted functionalities without verifying session privileges.

### Steps Performed (As Me)

Logged in as a low-privilege user.

Navigated directly to admin-only pages by manipulating URLs.

Observed missing access validation.

### Proof of Concept

(Insert screenshot here: Direct URL access without admin role)

### Impact

Attackers can perform actions outside their authorization scope.

### Mitigation

Implement server-side role verification

Enforce access control checks on every sensitive function

Use deny-by-default approach

CVSS Score: 8.0 (High)

## A02:2025 – Cryptographic Failures

### Description

DVWA stores passwords using weak hash algorithms and insecure transport.

### Steps Performed

Captured login request — observed no HTTPS.

Retrieved password hash from DB (weak MD5).

Cracked hash using dictionary attack.

### Proof of Concept

(Insert screenshot: MD5 hash cracked)

### Impact

Credentials can be stolen and reused in credential stuffing attacks.

### Mitigation

Enforce HTTPS with TLS 1.3

Use bcrypt/Argon2 hashing

Implement HSTS headers

CVSS Score: 7.5 (High)

#### A03:2025 – Injection (SQL Injection)

##### Description

The login and user ID search forms are vulnerable to SQL Injection.

##### Steps Performed

Navigated to SQL Injection module.

Injected payload:

1' OR '1'='1

Retrieved full user list from database.

##### Proof of Concept

(Insert screenshot: SQLi result dumping users)

##### Impact

Unauthorized access to database contents, privilege escalation, data modification.

##### Mitigation

Use prepared statements (PDO/MySQLi)

Server-side input validation

Disable detailed error messages in production

CVSS Score: 9.8 (Critical)

#### A04:2025 – Insecure Design

##### Description

DVWA lacks secure design patterns such as input validation frameworks, multi-layered access rules, and secure workflows.

##### Examples Observed

User workflows assume trusted input

No rate limiting

Several modules designed intentionally insecure

Mitigation

Adopt threat modeling (STRIDE)

Apply secure design patterns (input sanitization layer, validation middleware)

Enforce least privilege design

#### A05:2025 – Security Misconfiguration

Description

The DVWA lab demonstrates multiple configuration weaknesses.

Examples

Default credentials (admin/password)

Displaying verbose error messages

Directory listing enabled

Debug mode active

Proof of Concept

(Insert screenshot: PHP warning or debug output)

Mitigation

Disable directory indexing

Disable debug mode in production

Use secure and unique credentials

Harden server configuration (Apache, MySQL)

CVSS Score: 6.8 (Medium)

#### A06:2025 – Vulnerable and Outdated Components

Description

DVWA intentionally runs outdated PHP, MySQL, and Apache versions.

#### Steps Performed

Checked PHP version via /dvwa/phpinfo.php (if enabled)

Observed EOL versions vulnerable to public exploits

#### Impact

Outdated components increase attack surface.

#### Mitigation

Regular patch management

Remove unsupported components

Implement dependency scanning

### A07:2025 – Identification and Authentication Failures

#### Description

Weak password policies and predictable session IDs.

#### Steps Performed

Bruteforced login credentials with common wordlist

Verified no CAPTCHA or account lockout

Observed predictable PHPSESSID value

#### Proof of Concept

(Insert screenshot: successful brute force login)

#### Mitigation

Enforce password strength policy

Add login throttling and account lockouts

Use secure session generation

CVSS Score: 7.1 (High)

### A08:2025 – Software and Data Integrity Failures

#### Description

DVWA does not validate integrity of user-uploaded files, nor perform code-signing checks.

## Steps Performed

Uploaded PHP file in File Upload module.

Bypassed file type checks.

Executed shell commands via uploaded payload.

## Proof of Concept

(Insert screenshot: Web shell command output)

## Impact

Leads to remote code execution (RCE).

## Mitigation

Strict file-type validation

Store uploads outside web root

Block execution permissions on uploads directory

CVSS Score: 9.9 (Critical)

## A09:2025 – Security Logging and Monitoring Failures

### Description

DVWA does not log authentication attempts or suspicious activities.

### Findings

No logs for failed logins

No alerting mechanisms

No audit trail

### Mitigation

Enable centralized logging

Implement SIEM or WAF alerts

Retain logs securely

## A10:2025 – Server-Side Request Forgery (SSRF)

### Description

DVWA features allow fetching external URLs without validation.

#### Steps Performed

Entered internal URL `http://127.0.0.1/etc/passwd`

Successfully retrieved local resource (if module enabled)

#### Proof of Concept

(Insert screenshot: Internal resource output)

#### Mitigation

Block internal/private IP ranges

Whitelist allowed domains

Validate URL schemas

CVSS Score: 8.6 (High)

### 5. Overall Risk Summary

#### OWASP Category Status Risk Level

A01 Broken Access Control Exploited High

A02 Cryptographic Failures Exploited High

A03 Injection Exploited Critical

A04 Insecure Design Observed High

A05 Security Misconfiguration Exploited Medium

A06 Outdated Components Observed Medium

A07 Auth Failures Exploited High

A08 Integrity Failures Exploited Critical

A09 Logging Failures Observed Medium

A10 SSRF Exploited High

### 6. Conclusion

This assessment shows that DVWA—by design—contains severe vulnerabilities across the OWASP Top 10. Successfully exploiting these vulnerabilities demonstrates a strong understanding of modern web exploitation techniques, secure coding practices, and real-world attack surfaces.

This report and corresponding screenshots form a complete professional penetration test deliverable suitable for internship evaluation and recruiter review.

## 7. Appendix

Tools Used

Kali Linux

Burp Suite Community Edition

SQLMap

Firefox/Chrome DevTools

DVWA

Screenshot Placements

Place screenshot after each PoC heading

Use labels such as:

Figure 1: SQL Injection Dump

Figure 2: Web Shell Execution

END OF REPORT