Web Application Vulnerability Assessment & Penetration Test (VAPT) Report

Client : DVWA

https://rezv.me

Executive Summary

This report documents the results of a Vulnerability Assessment and Penetration Test (VAPT) conducted against a deliberately vulnerable web application (DVWA) hosted on Metasploitable 2. The objective of this assessment was to identify, validate, and analyze security vulnerabilities in a controlled lab environment following OWASP Top 10 methodology.

During the assessment, a Stored Cross-Site Scripting (XSS) vulnerability was identified and successfully exploited. This vulnerability allows an attacker to inject malicious JavaScript code that is persistently stored on the server and executed in the browsers of users accessing the affected page.

If present in a real-world application, this vulnerability could lead to session hijacking, credential theft, and unauthorized actions on behalf of legitimate users.

Scope of Assesment

Asset	Description
Target Application	Damn Vulnerable Web Application (DVWA)
Target Host	Metasploitable 2
Target IP	172.26.22.135
Protocol	HTTP

Out-of-Scope

- Any external or production systems
- Denial-of-Service (DoS) attacks
- Brute-force attacks

Environment Details

Components	Details
Attacker Machine	Kali Linux (VMware)
Target Machine	Metasploitable 2
Webserver	Apache
Backend	PHP & MySQL
Application URL	172.26.22.135/DVWA

Tools Used

Tool	Purpose
BurpSuite	HTTP request interception and analysis
Firefox Browser	Manual Test
Kali Linux Toolset	Penetration Testing Environment

Methodology

The assessment followed a structured methodology aligned with OWASP Testing Guide principles:

1. Environment Validation

- verified network connectivity between Kali Linux & the target system
- Confirmed application accessibility

2. Reconnaissance

- Manual browsing of application modules
- Identification of user input points

3. Vulnerability Identification

- Manual testing of input Malicious JavaScript payload
- Interception on HTTP requests using Burp Suite

4. Exploitation

- Injection of malicious JavaScript payload
- Verification of persistent execution

5. Risk Analysis

- Evaluation of impact and likelihood
- Severity classification

6. Documentation & Reporting

- Evaluation of impact and likelihood
- Severity classification

Vulnerablity Findings

OWASP Category: A07 – Cross-Site Scripting (XSS)

Severity: High

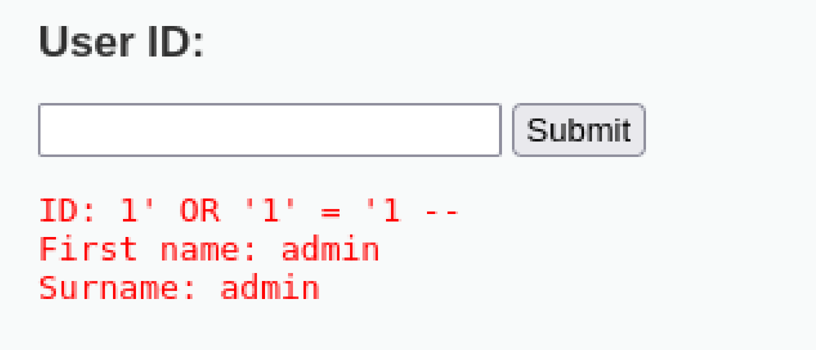
Description

The application stores user-supplied input without proper validation or output encoding. Malicious JavaScript injected into input fields is persistently stored in the backend database and executed whenever the affected page is loaded.

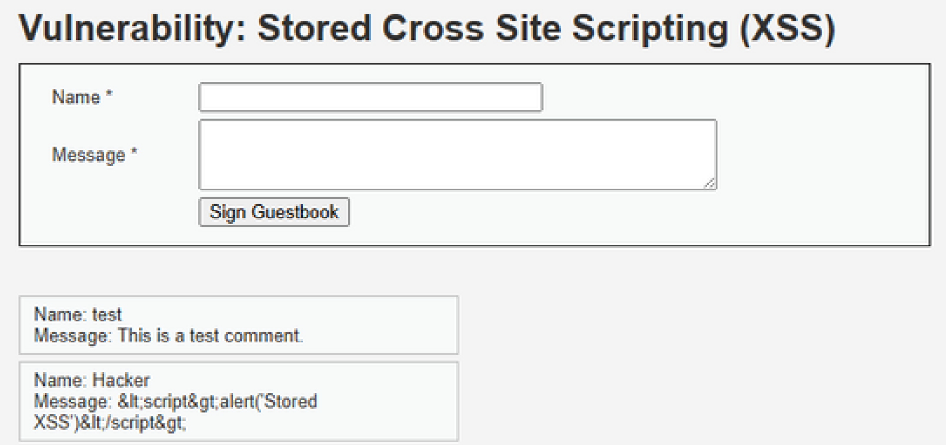
Proof Of concept

Injected Payloads:

```
<script>alert('Stored XSS')</script>
```



```
ID: 1' OR '1' = '1 --
First name: admin
Surname: admin
```


Vulnerability: Stored Cross Site Scripting (XSS)

Name: test
Message: This is a test comment.

Name: Hacker
Message: <script>alert('Stored XSS')</script>

Result:

- JavaScript executed immediately after submission
- Script executed again upon page refresh
- Script persisted for all users accessing the page

Impact

If exploited in a real-world scenario, this vulnerability could allow an attacker to:

- Hijack user sessions
- Steal authentication cookies
- Perform actions on behalf of victims
- Conduct phishing or defacement attacks

Likelihood

- No authentication bypass required
- Minimal technical skill needed
- Easily exploitable

Likelihood: High

Risk Factor Analysis

Factor	**Assesment**
Attack Complexity	Low
Priviliges Required	None
User Interaction	Not Required
Impact	High
Likelihood	High
Overall Risk	High

Remediation

To mitigate Stored XSS vulnerabilities, the following actions are recommended::

1. Input Validation
- Reject or Sanitize user input containing HTML or JavaScript

2. Output Encoding
- Encode all user-supplied data before rendering in HTML
- Use context-aware encoding methods

3. Secure Coding Practices
- Perform server-side validation
- Avoid directly rendering untrusted input

4. Security Headers
- Implement Context Security Policy (CSP)
- Set HttpOnly and Secure flags on cookies

5. Post-Fix Testing
- Retest affected input fields after remediation
- perform regression testing

Conclusion

This assessment demonstrated the presence and exploitability of a Stored Cross-Site Scripting vulnerability within the tested web application. While the target environment was intentionally vulnerable for learning purposes, similar issues frequently occur in real-world applications.

Proper input validation, output encoding, and secure development practices are critical to preventing such vulnerabilities. Regular security testing and adherence to OWASP guidelines significantly reduce the risk of client-side attacks.

Disclaimer

This assessment was conducted in a controlled laboratory environment for educational purposes only. No testing was performed against live, production, or unauthorized systems.