Pentest Report

Executive Summary:

This assessment evaluated the security of the DVWA web application hosted at 192.168.74.16. Testing focused on common web vulnerabilities, including SQL injection, cross-site scripting, and weak authentication controls. Several high-risk issues were identified that could allow attackers to access or modify application data and compromise used accounts

The most critical findings were a SQL injection vulnerability in the id parameter, insecure session cookie handling, and reflected XSS in user-supplied input fields. If exploited, these issues could lead to database disclosure, account takeover, and full application compromise. Immediate remediation and follow-up testing are recommended

Scope & Methodology:

Scope included the DVWA instance at 192.168.74.16 and its web modules. No other hosts or services were tested.
The engagement followed the Penetration Testing Execution Standard (PTES): pre-engagement interactions.

- Intelligence gathering
- Threat modeling
- Vulnerability analysis
- Exploitation
- Post-exploitation
- Reporting

Tools Allowed: Nmap, Burp Suite, OWASP ZAP, Sqlmap, Metasploit.

Finding Table:

ID	Vulnerability	Severity	CVSS	Affected URL/Host	Status
F001	SQL Injection in id	Critical	9.1	/dvwa/vulnerabilities/sqli/?id=	Exploited

F002	Reflected XSS in name	High	8.0	/dvwa/vulnerabilities/xss_r/?name=	Exploited
F003	Weak Session Cookie	High	7.5	PHPSESSID (no HttpOnly/Secure flags)	Verified
F004	Outdated Components	Medium	6.5	Apache/PHP versions on DVWA host	Detected

SQL Injection:

Location:

http://192.168.74.16/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit

Description:

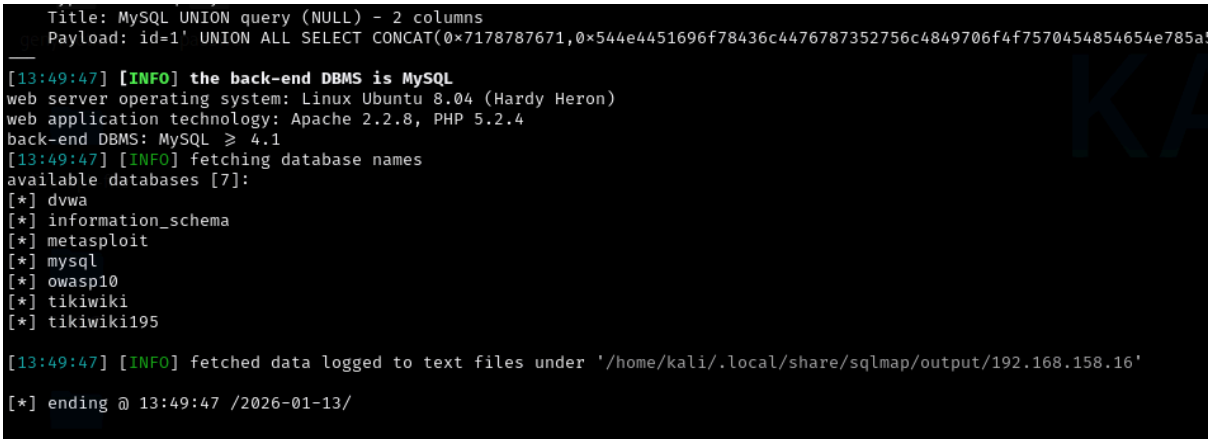
The id parameter in the SQL Injection module is directly concatenated into a SQL query without proper input validation or parameterization. Supplying crafted input allows modification of the underlying query.

Impact:

An unauthenticated attacker can enumerate databases and dump the dvwa.users table, including usernames and password hashes. This enables credential reuse, offline cracking, and full compromise of application accounts.

Evidence:

sqlmap -u "http://192.168.74.16/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" \
  --cookie="PHPSESSID=...; security=low" -D dvwa -T users --dump --batch



Remediation:

- Implement parameterized queries using prepared statements for all database access.
- Avoid string concatenation with untrusted input.
- Apply server-side input validation and enforce least-privilege database accounts.

Xss Injection:

Location:

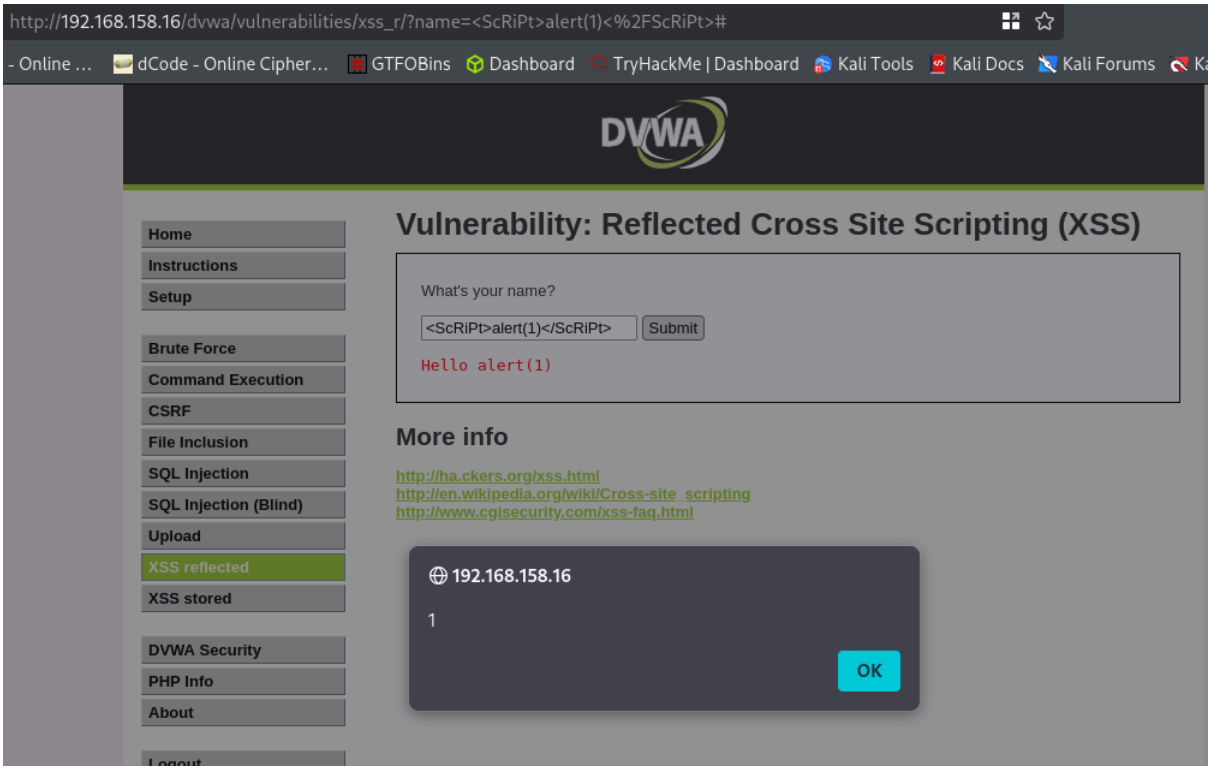
http://192.168.74.16/dvwa/vulnerabilities/xss_r/?name=

Description:

Cross-Site Scripting (XSS) occurs when an attacker injects malicious JavaScript into a web application that is then executed in other users' browsers. This happens when user input is reflected back (reflected XSS) or stored in the database (stored/persistent XSS) without proper encoding

Evidence:

Search parameters: /name?=<script>alert('XSS')</script>



Remediation:

1. Using Context-Aware Output Encoding
2. Implementing Content Security Policy (CSP)
3. Enabling HttpOnly + Secure Cookies: which prevent JS cookie access
4. Input Validation: Whitelist allowed characters

Non-Technical Summary:

A security review performed on the targeted web application(192.168.158.16). Which identifies serious risks that could affect real production systems. Attackers could inject malicious code to steal user login sessions, extract the entire database including passwords, and upload dangerous files for server takeover. These vulnerabilities would allow unauthorized access to sensitive data and full system compromise.

Priority Actions:

1. Database Protection - Use secure coding to prevent data leaks
2. User Safety - Stop malicious scripts from running in browsers
3. File Security - Block dangerous file uploads

Immediate fixes plus quarterly security testing recommended to prevent exploitation.

Risk Level: Critical - Action required within 30 days.