DVWA Penetration Test Report

Project Title: Web Application Penetration Test — DVWA (Damn Vulnerable Web Application)
Analyst: Ramjith M – Group-6
Date: 22/06/2025
Target: Local DVWA instance
Security Level: Low

---

1. Summary

This report documents a web application penetration test conducted on DVWA. The assessment simulated common web attacks mapped to the OWASP Top 10 vulnerabilities. Multiple vulnerabilities were identified, some exploited successfully, and proof-of-concept (PoC) evidence was collected. Remediation guidance is provided to mitigate these risks.

---

2. Scope

- **Target:** DVWA running in a controlled lab environment.
- **Test Type:** Manual exploitation with Burp Suite and browser-based payloads.
- **Focus:**
  - SQL Injection
  - Command Injection
  - Cross-Site Scripting (Reflected & Stored)
  - Brute Force
  - CSRF
  - Sensitive Data Exposure
  - Broken Access Control (attempted)
  - Security Misconfiguration

---

3. Methodology

Testing followed industry-standard practice:

- Reconnaissance using Burp Suite proxy
- Manual injection and payload testing for SQLi, XSS, Command Injection.
- Brute force performed using Burp Intruder.
- CSRF tested using custom HTML forms and direct requests.
- Traffic inspection for sensitive data exposure.
- Attempts to bypass access controls directly by URLs.

Vulnerability	Typical CVSS v3.1 Score	Risk Level
SQL Injection	**9.8 (Critical)**	Critical impact, allows DB compromise
Command Injection	**9.0 (Critical)**	Remote code execution, full system compromise
Reflected XSS	**6.1 (Medium)**	User session hijack, limited scope
Stored XSS	**7.4 (High)**	Persistent user compromise, broader impact
Brute Force (Weak Authentication)	**6.5 (Medium)**	Credential compromise, unauthorized access
CSRF	**6.8 (Medium)**	Unauthorized state changes
Sensitive Data Exposure (HTTP)	**7.5 (High)**	User credentials exposed over network
Broken Access Control	*(Attempted, no bypass)* — Not rated, as not exploitable	
Security Misconfiguration (Low Security Level)	**4.3 (Low)**	Configuration issue, no direct exploit

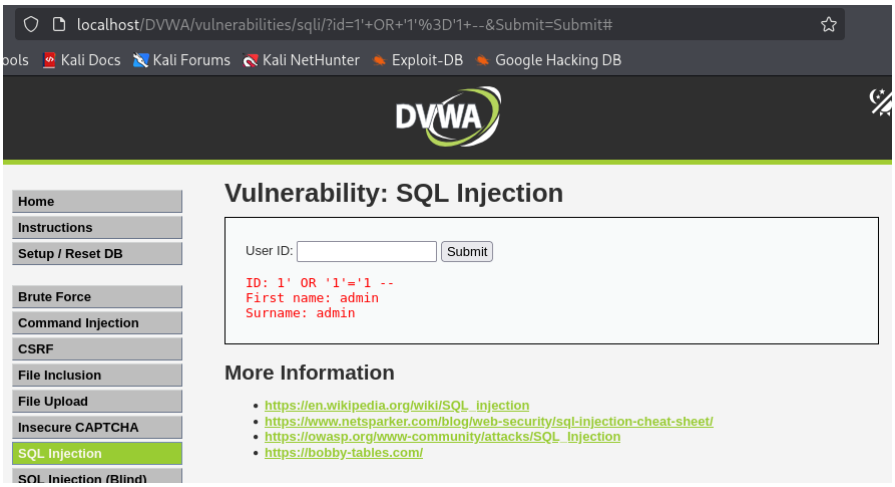
4. Findings

4.1 SQL Injection

- **Status:** Successfully exploited.
- **Technique:** `'OR'1'='1--` bypassed query logic, dumping user table.
- **Impact:** Full read access to backend database.
- **Evidence:**

- SQLi input –

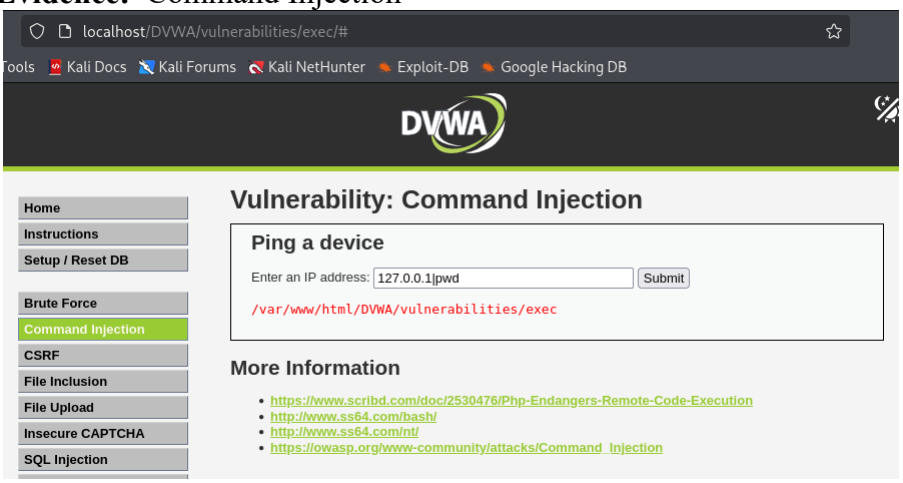
• SQLi output –



- **Remediation:** Use parameterized queries, enforce input validation.

4.2 Command Injection

- **Status:** Successfully exploited.
- **Technique:** `127.0.0.1|pwd` Executed OS command.
- **Impact:** Revealed Print Working Directory.
- **Evidence:** Command Injection

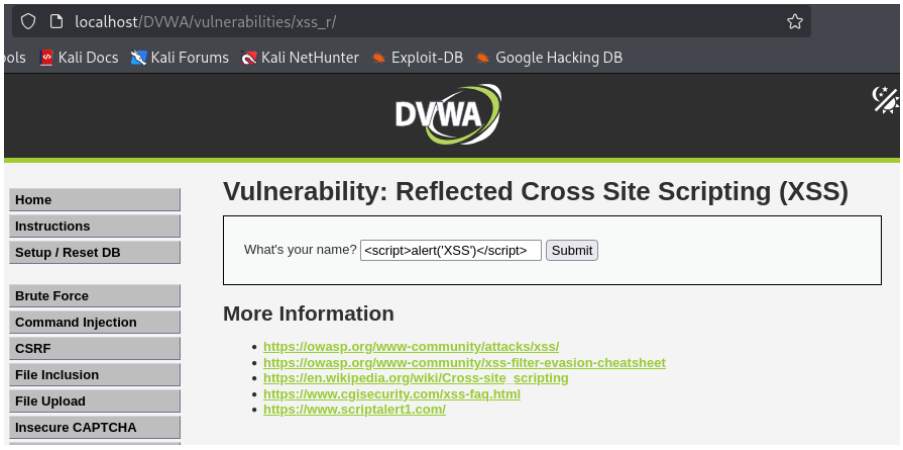


- **Remediation:** Sanitize input, use safe API calls.

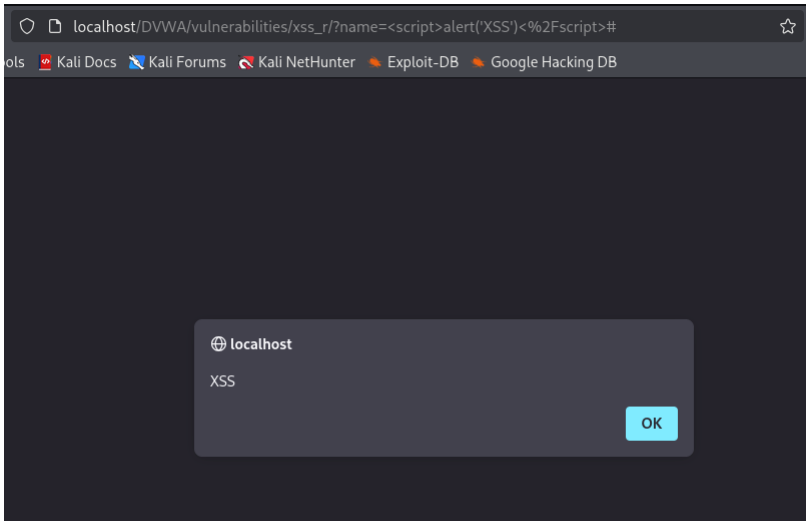
4.3 Cross-Site Scripting (Reflected)

- **Status:** Successfully exploited.
- **Technique:** `<script>alert('XSS')</script>`
- **Impact:** Arbitrary script execution in user's browser.
- **Evidence:**

XSS Reflected input



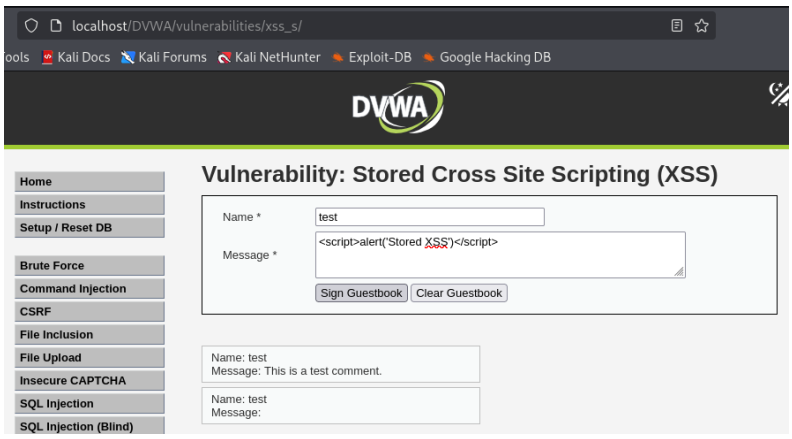
XSS Reflected output.

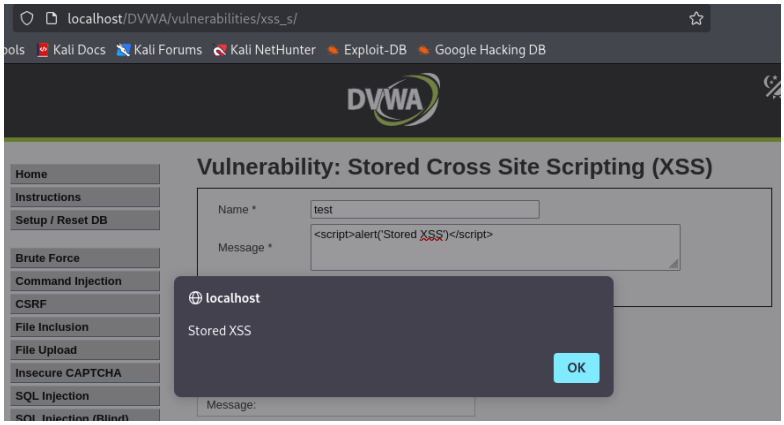


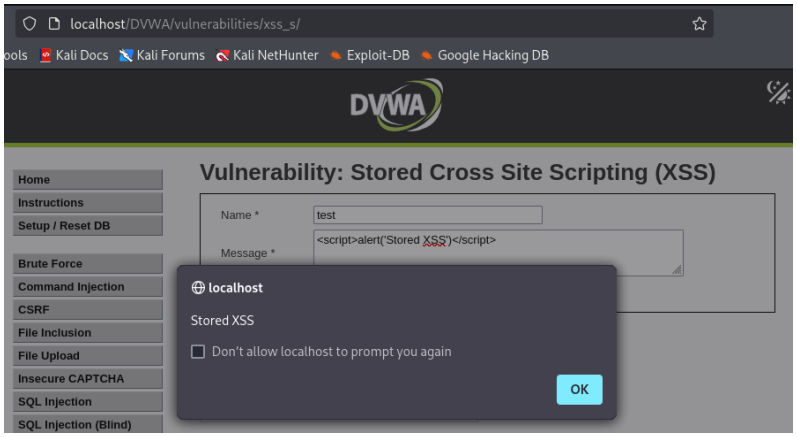
- **Remediation:** Encode output, sanitize input.

4.4 Cross-Site Scripting (Stored)

- **Status:** Successfully exploited.
- **Technique:** `<script>alert('Stored XSS')</script>` in guestbook.
- **Impact:** Persistent client-side code execution.
- **Evidence:**



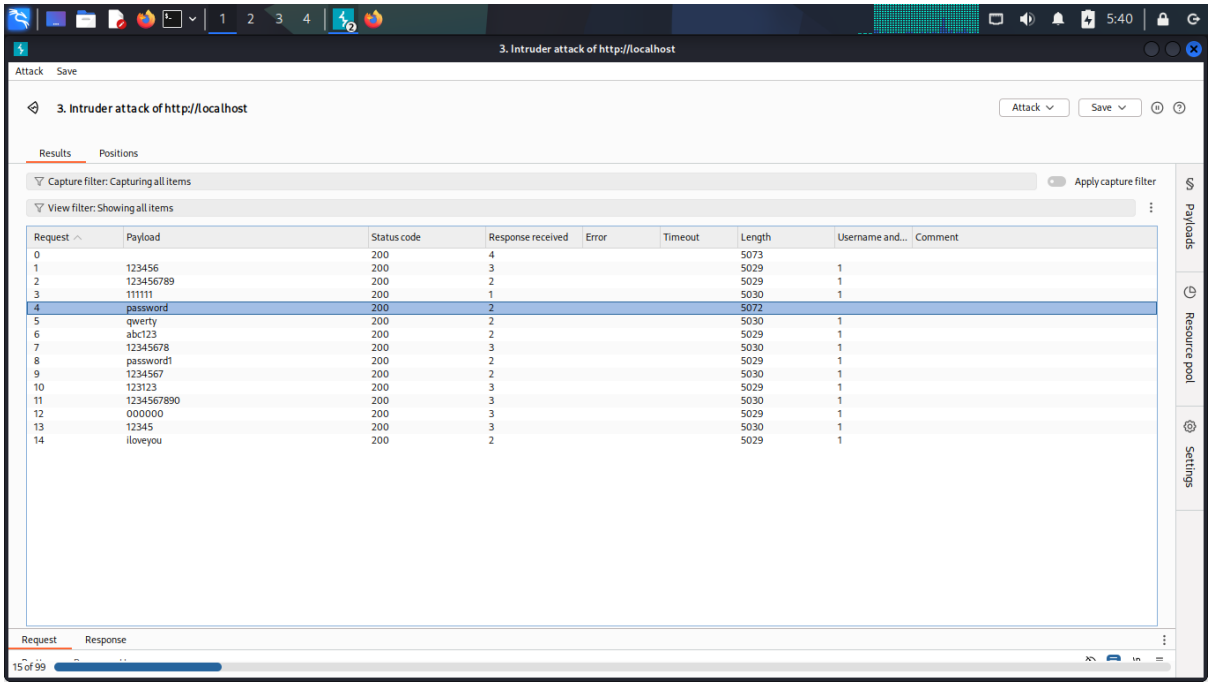




- **Remediation:** Apply input sanitization and output encoding.

4.5 Brute Force

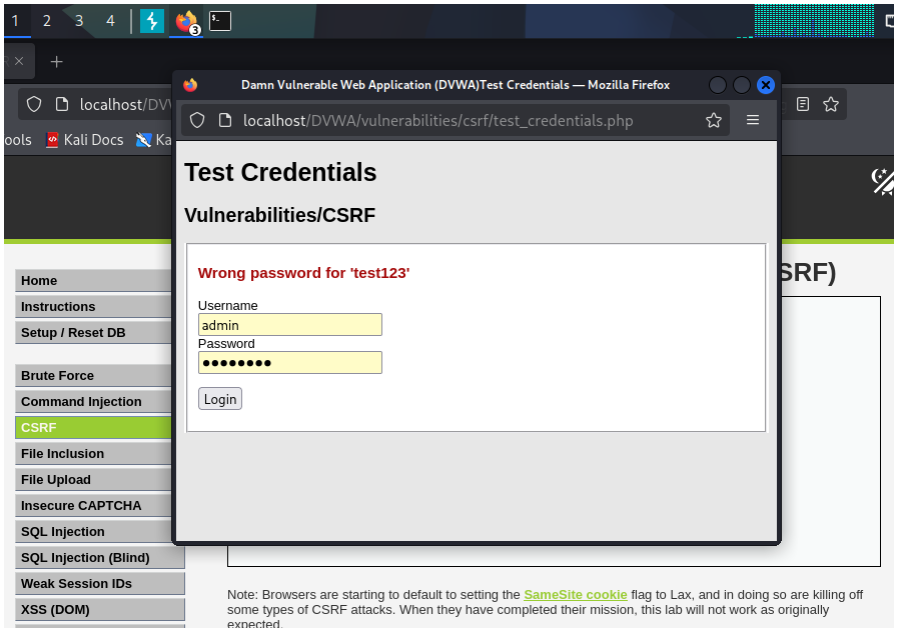
- **Status:** Successfully exploited.
- **Technique:** Burp Intruder attack against login page.
- **Impact:** Weak password policy allows guessing.
- **Evidence:**

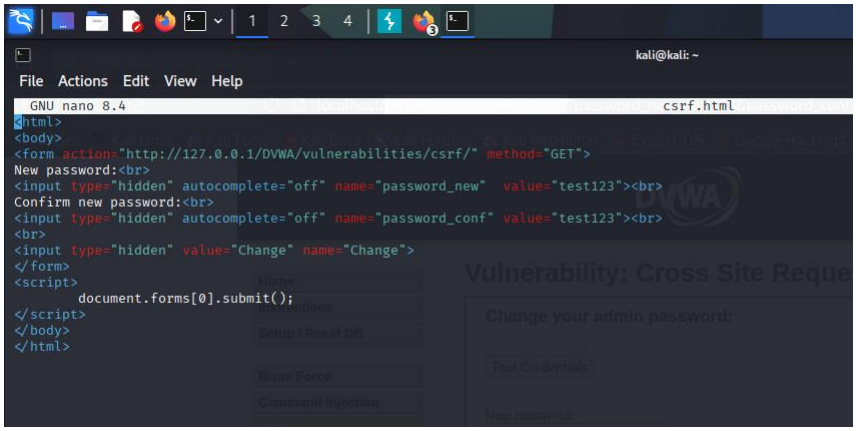


- **Remediation:** Enforce strong password policy, implement lockouts.

4.6 CSRF

- **Status:** Attempted. Limited due to browser security; tested via crafted HTML and Burp.
- **Evidence:**



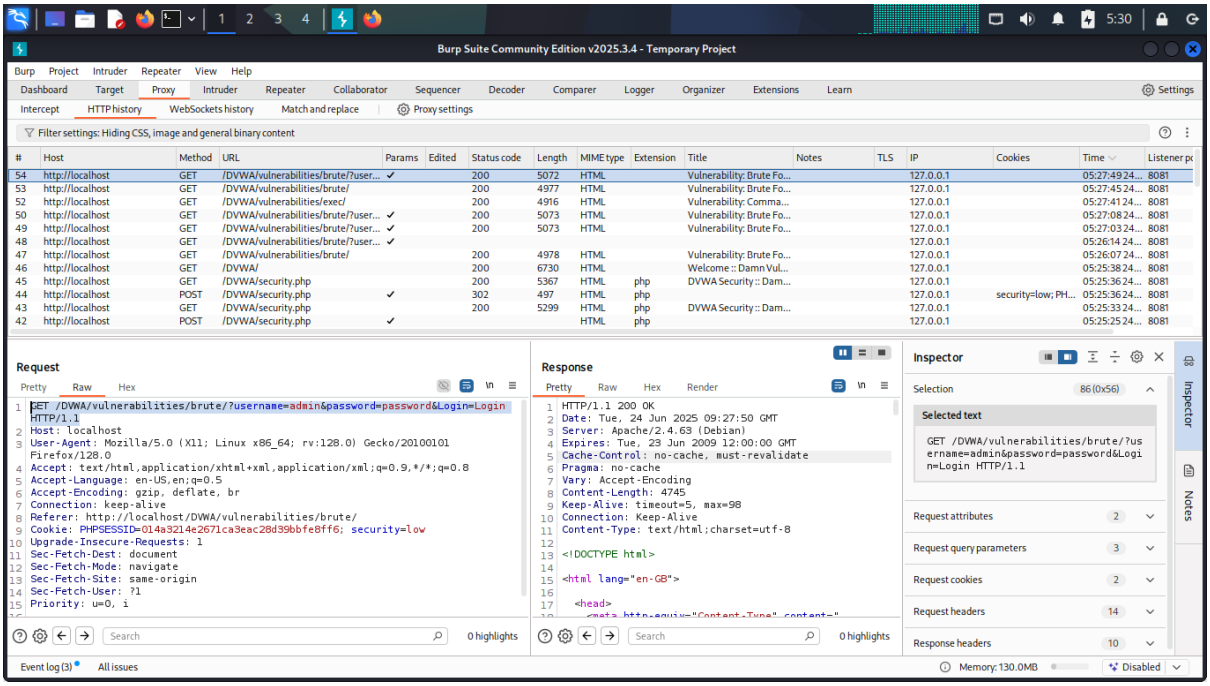


- **Remediation:** Implement anti-CSRF tokens for state-changing requests.

4.7 Sensitive Data Exposure

- **Status:** Confirmed.
- **Technique:** Plaintext credentials intercepted.
- **Impact:** Usernames and passwords visible over HTTP.

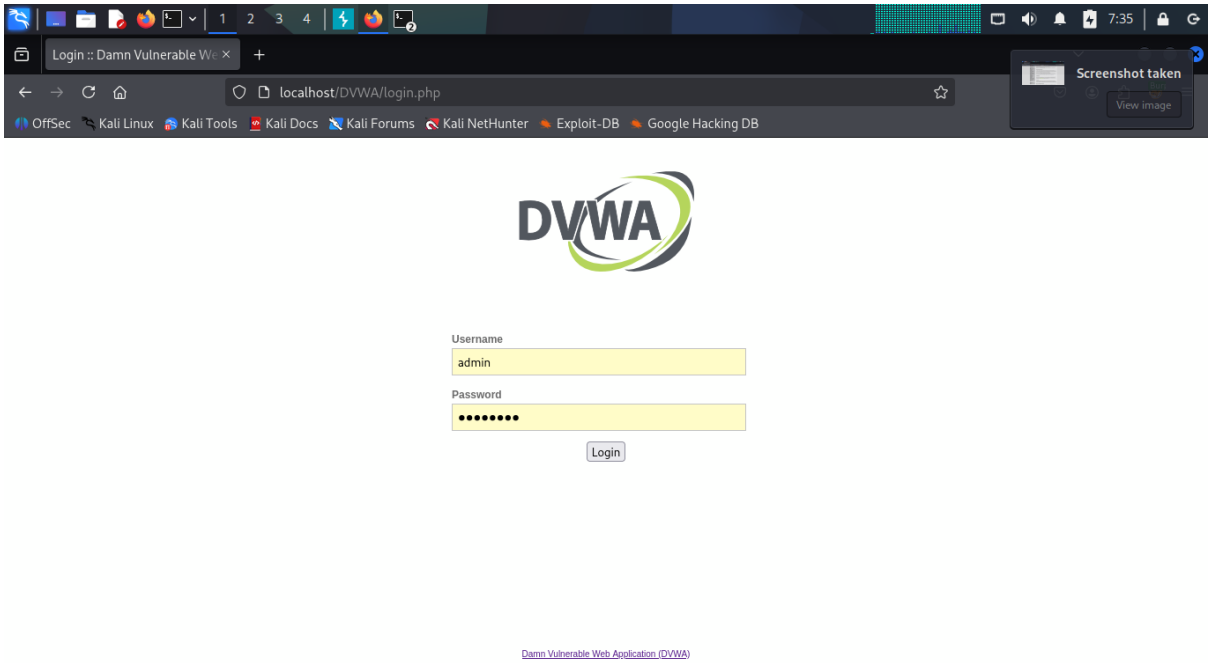
- **Evidence:**



- **Remediation:** Use HTTPS/TLS for all authentication endpoints.

4.8 Broken Access Control

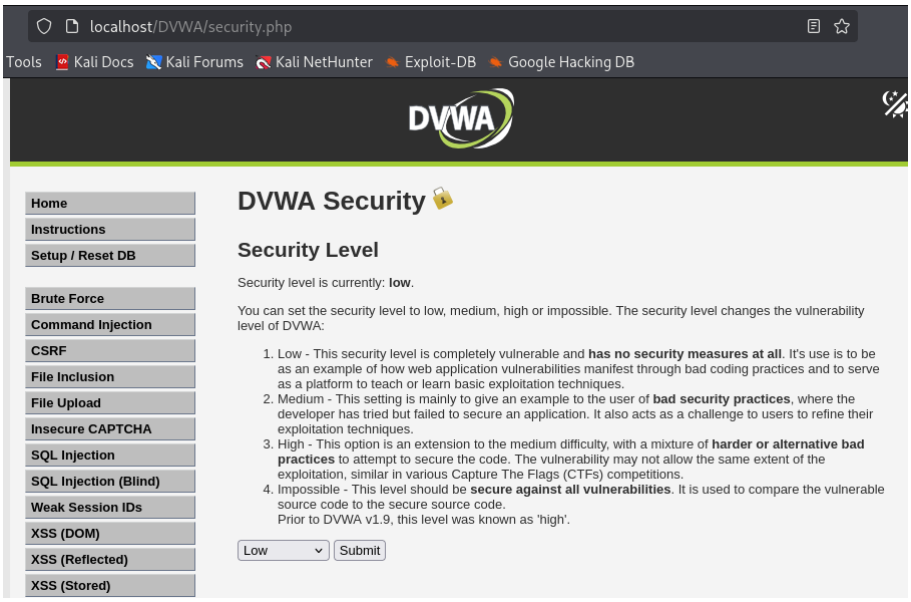
- **Status:** Attempted. Direct URL access redirects to login; no bypass found.
- **Evidence:** `broken access control not working.png`

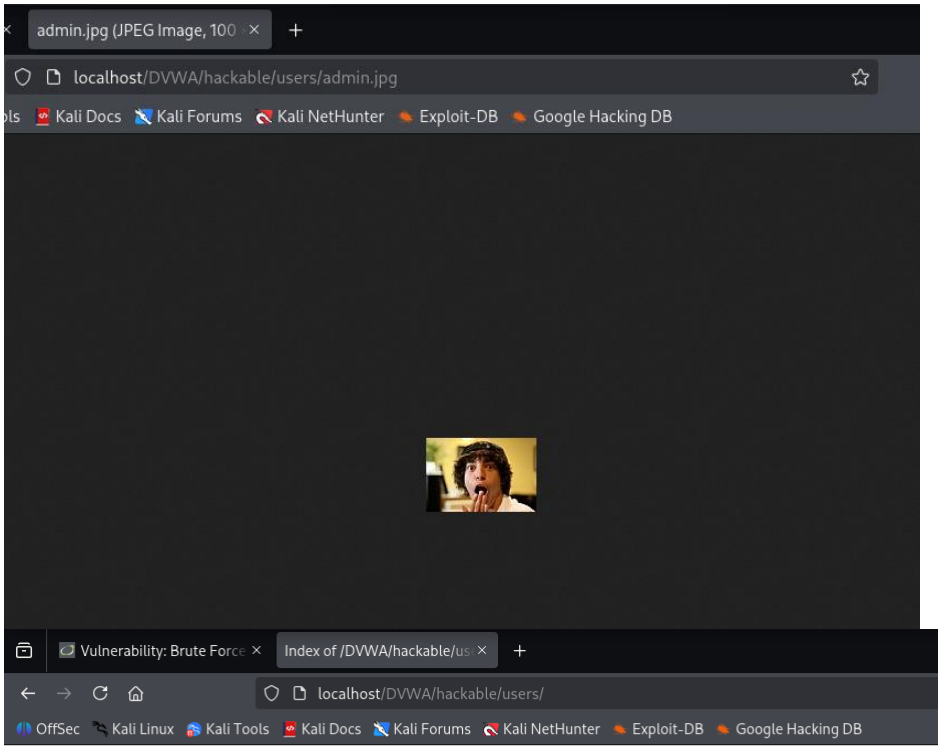


- **Remediation:** Keep session checks enforced server-side.

4.9 Security Misconfiguration

- **Status:** Confirmed.
- **Finding:** DVWA security level set to “Low”.
- **Evidence:**





Index of /DVWA/hackable/users

Name	Last modified	Size	Description
Parent Directory		-	
1337.jpg	2025-06-16 19:36	3.6K	
admin.jpg	2025-06-16 19:36	3.5K	
gordonb.jpg	2025-06-16 19:36	3.0K	
pablo.jpg	2025-06-16 19:36	2.9K	
smithy.jpg	2025-06-16 19:36	4.3K	

Apache/2.4.63 (Debian) Server at localhost Port 80

- **Remediation:** Use secure configurations in production; remove debug/low settings.

Conclusion

The DVWA instance demonstrates common web vulnerabilities when misconfigured. Critical issues like SQLi, XSS, and Command Injection were exploited successfully. Remediation requires secure coding, strict input validation, proper authentication controls, and encryption of sensitive data in transit.
