Bharatiya Vidya Bhavan's
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Autonomous Institute Affiliated to University of Mumbai)
Munshi Nagar, Andheri (W), Mumbai – 400 058.

Experiment No. 7 - Experiment on Web Security

Aim: To identify, exploit and mitigate the common web application vulnerabilities

----

Part A — Setup & baseline

1. Show DVWA running: screenshot of login page and the "Create/Reset Database" page.
2. Change and note the security level settings (low/medium/high) and explain what the setting changes in code or behavior (short answer).

Part B — Basic vulnerabilities

For each: (a) identify vulnerable page, (b) exploit (screenshot + short reproduction steps), (c) explain root cause, (d) propose a fix.

1. **SQL Injection (SQLi)** vulnerabilities/sqli
   - Demonstrate retrieving another user's password or dumping a table.
2. **Reflected XSS** — vulnerabilities/xss_r
   - Craft a payload that displays an alert and show impact (cookie theft discussion).
3. **Stored XSS** — vulnerabilities/xss_s
   - Post a persistent payload and demonstrate page rendering it.

Part C — Auth / session / logic problems (intermediate)

1. **Brute force / password strength** — examine DVWA login protections; demonstrate a simple brute force (rate-limited, controlled).
2. **CSRF** — vulnerabilities/csrf
   - Build a proof-of-concept HTML page that triggers a state change.
3. **Insecure direct object references (IDOR)** — access resources by ID and show unauthorized access.

Part D — File/functionality exploitation

1. **File upload vulnerability** — vulnerabilities/upload
   - Upload an allowed file and attempt to upload a web shell (document how DVWA blocks/permits).
2. **Command injection** — vulnerabilities/exec
   - Execute system command via vulnerable parameter (show output).
3. **Remote code execution / File inclusion** — vulnerabilities/fi and vulnerabilities/command
   - Demonstrate local file inclusion or remote file include vectors if possible at chosen security level.

Part E — Defense & remediation

For **three** vulnerabilities you exploited:

- Implement fixes (or pseudo-fixes if full changes are invasive) and demonstrate mitigation.

- Examples: prepared statements for SQLi, proper output encoding for XSS, CSRF tokens for CSRF, file validation/whitelisting for uploads.

Part F — Report & reflection

Deliver a report (PDF) that contains:

- Executive summary (1 paragraph)
- Setup notes (commands, IPs masked)
- For each vulnerability: steps, evidence (screenshots), explanation, remediation, and CVSS-like risk rating (Low/Med/High)
- Lessons learned and recommended hardening checklist for a LAMP web app

Submission:

1. Single PDF report (see Part F).
2. A document with evidence/ containing labeled screenshots and short shell commands used (or a terminal transcript).
3. A Git repo link with:
    1. For at least two vulnerabilities: a short video (2–4 mins each) demonstrating exploit and fix (optional if bandwidth limited; screenshots acceptable).
    2. Code snippets/patches applied to DVWA (if fixes implemented).