# Mini Pentest Report: DVWA + Mutillidae

**Target IP:** 192.168.168.129

---

## 1. Executive Summary

- **Objective:** Perform a controlled mini penetration test.
- **Target systems:** DVWA and Mutillidae vulnerable web applications.
- **Scope:** Web application vulnerabilities, reconnaissance, scanning, and controlled exploitation.
- **Key findings:** SQL Injection, Command Injection, File Upload vulnerabilities, Reflected & Stored XSS, Sensitive information exposure, and misconfigurations.

---

## 2. Technical Findings

### a) DVWA

- **Recon & Scanning:**
- Nmap: Apache 2.4.65, HTTP open, ports 443/8080 closed.
- Gobuster: directories found: `/database`, `/config`, `/tests`, `.git/`.
- Nikto: Missing security headers (X-Frame-Options, X-Content-Type-Options), directory indexing, Git information leakage.
- **Vulnerabilities Exploited:**
- SQL Injection (Reflected & Blind).
- Command Injection.
- File Upload.
- Reflected & Stored XSS.
- **Evidence:**
- Screenshots and detailed outputs are shared in the GitHub repository.

### b) Mutillidae

- **Recon & Scanning:**
- Nmap: Apache 2.4.65, ports 443/8080 closed.
- Gobuster: `.git/HEAD`, `/passwords`, `/includes/`, `/data/` directories found.
- Nikto: Exposed `phpinfo.php`, directory indexing, HTTP headers issues, cookies without HttpOnly flag.

• **Vulnerabilities Exploited:**

• SQL Injection.
• Stored XSS.
• DNS Lookup information disclosure.
• Authentication bypass with low security.

• Sensitive data exposure.

• **Evidence:**

• Screenshots and detailed outputs are shared in the GitHub repository.

---

## 3. Vulnerability Comparison

| Vulnerability Type | DVWA | Mutillidae | Severity | Exploitability |
|---|---|---|---|---|
| SQL Injection | Yes (Reflected, Blind) | Yes (User Info) | High | High |
| Command Injection | Yes | No | High | Medium |
| File Upload | Yes | Limited/Not Tested | Medium | Medium |
| XSS (Reflected/Stored) | Yes | Yes | Medium | Medium |
| Sensitive Info Exposure | .git, config files | Passwords, phpinfo | High | High |
| Misconfigurations | Missing headers | Missing headers, cookie flags | Medium | Medium |

---

## 4. Impact Analysis

• **Database compromise:** SQLi can allow reading/modifying user data.
• **Remote command execution:** Command injection allows system-level access as `www-data`.
• **Client-side attacks:** XSS can steal session cookies, perform phishing.
• **Information leakage:** Exposed `.git`, `phpinfo.php`, passwords, API keys.

---

## 5. Remediation Recommendations

• Implement input validation and prepared statements to prevent SQLi.
• Sanitize all user input and escape outputs to prevent XSS.
• Secure file upload with proper checks and storage outside web root.

- Restrict directory indexing and prevent access to `.git`, `.env`, and other sensitive files.
- Apply security headers: X-Frame-Options, X-Content-Type-Options, HSTS, HttpOnly cookies.
- Enforce strong passwords, session security, and least privilege.

---

## 6. Appendices

- Scan outputs (Gobuster, Nikto, Nmap) included in main findings.
- Screenshots and detailed evidence are shared in the GitHub repository.

---

**End of Report**