DVWA Authentication Security Testing Report

Title:
DVWA Authentication Security Testing Report

Prepared By:
Risha Gupta

Tools Used:

- **DVWA (Damn Vulnerable Web Application)**
  - **XAMPP** (Apache + MySQL)
  - **Browser Developer Tools** (Chrome)

Testing Environment:
Localhost — http://localhost/dvwa/

1. Introduction

This report documents the authentication-related vulnerabilities identified in DVWA (Damn Vulnerable Web Application).
Testing was performed on the local DVWA instance running on:

```
http://localhost/dvwa/
```

The goal was to analyze weaknesses in:

- Credential security
- Authentication mechanism
- Session management
- Session cookie security

---

2. Vulnerability Summary

Vulnerability	Status	Risk Level
Weak / Default Credentials	Confirmed	High
Brute Force Possible	Confirmed	High
Session Not Destroyed on Logout	Confirmed	High
Session Fixation	Confirmed	High
Weak Session Cookies	Confirmed	High
Missing Cookie Flags (HttpOnly, Secure, SameSite)	Confirmed	High

3. Vulnerability Details

3.1 Weak / Default Credentials

✔ Description

DVWA allows logging in using the default credentials:

```
Username: admin
Password: password
```

No password complexity & no forced password change is implemented.

✔ Impact

- Attackers can easily guess default credentials
- Leads to full application compromise

✔ Evidence (What I observed)

- Entering `admin/password` → Login successful.

✔ Recommendation

- Enforce strong password policy
- Remove default credentials
- Implement account lockout on multiple failed attempts

3.2 Brute Force Attack (Using Burp Suite)

✔ Description

DVWA does not have any brute-force protection:

- No captcha
- No account lockout
- No rate limiting

✔ Proof

Using Burp Intruder:

- Password field was brute-forced successfully
- Server allowed unlimited login attempts

✔ Impact

Attacker can guess password through automated attacks.

✔ Recommendation

- Implement rate limiting & lockout
- Add CAPTCHA
- Add failed-attempt warnings

3.3 Session Hijacking & Broken Logout

✔ Description

When testing session behavior:

Observations:

State	PHPSESSID
Before Login	rsk0kip2igbiqajpj4v4gt91g8
After Login	rsk0kip2igbiqajpj4v4gt91g8
After Logout	rsk0kip2igbiqajpj4v4gt91g8

Session ID remained **same** in all 3 conditions.

✔ Impact

- Logout does NOT destroy session
- Attacker can reuse same PHPSESSID
- Leads to full session hijacking

✔ Proof

When I manually pasted old session ID after logout →
I still got logged-in dashboard.

This confirms:

- Broken Authentication
- Session Fixation Vulnerability
- Session Hijacking is possible

✔ Recommendation

- Invalidate session on logout
- Regenerate session ID on login
- Delete session cookie properly

3.4 Weak Session Cookies

✔ Description

Session cookie (`PHPSESSID`) is missing important security flags.

Observed Cookie Flags:

Flag	Status
HttpOnly	□ Disabled
Secure	□ Disabled
SameSite	□ Not Set
Path	/
Expires	Session Cookie

✔ Impact

- Cookie can be stolen using XSS
- Cookie is sent over HTTP → network sniffing risk
- Vulnerable to CSRF

✔ Recommendation

Use secure cookie:

```
setcookie('PHPSESSID', session_id(), [
  'secure'   => true,
  'httponly' => true,
  'samesite' => 'Strict'
]);
```

---

4. Overall Risk Rating

HIGH

DVWA’s authentication system is completely insecure and vulnerable to:

- Account compromise
- Credential guessing
- Session hijacking
- Unauthorized access

---

5. Conclusion

The authentication system of DVWA is insecure and contains multiple high-risk vulnerabilities.
If this were a real application, attackers could easily:

- Bypass authentication
- Steal session cookies
- Hijack accounts
- Perform unauthorized actions

All vulnerabilities listed require urgent remediation in production environments.