Vulnerability Assessment Report - VulnHub Login Website including identified risks and recommendations to mitigate them

Generated by: OWASP ZAP 2.16.0

Target: http://testphp.vulnweb.com

Date: February 20, 2025

Summary of Findings

Risk Level	Number of Alerts
High	0
Medium	3
Low	3
Informational	5

Identified Vulnerabilities & Recommended Mitigation

1. Absence of Anti-CSRF Tokens (Medium)

Description:

Cross-Site Request Forgery (CSRF) vulnerabilities exist due to missing anti-CSRF tokens in HTML forms. Attackers could exploit this to perform unauthorized actions on behalf of users.

Mitigation Recommendations:

- Implement anti-CSRF tokens for all forms that modify server-side data.
- Use a CSRF protection framework such as OWASP CSRFGuard.
- Validate CSRF tokens on the server before processing requests.
- Avoid using GET requests for actions that modify server state.

2. Content Security Policy (CSP) Header Not Set (Medium)

Description:

The lack of a CSP header makes the application vulnerable to Cross-Site Scripting (XSS) and data injection attacks.

Mitigation Recommendations:

- Configure the web server to include a CSP header that restricts script execution to trusted sources.
- Use the following header as a starting point:

  Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted-cdn.com;
- Regularly review and update CSP policies to minimize exposure.

3. Missing Anti-clickjacking Header (Medium)

Description:

The application does not include headers that protect against clickjacking attacks.

Mitigation Recommendations:

- Implement X-Frame-Options in the HTTP response headers:

  X-Frame-Options: DENY

  or

  X-Frame-Options: SAMEORIGIN
- Alternatively, use the frame-ancestors directive in CSP:

  Content-Security-Policy: frame-ancestors 'none';

4. Server Leaks Information via "X-Powered-By" Header (Low)

Description:

The "X-Powered-By" header exposes server framework details.

Mitigation Recommendations:

- Disable the X-Powered-By header in the web server configuration.
- Apache: Header unset X-Powered-By
- Nginx: server_tokens off;

5. Server Leaks Version Information via "Server" Header (Low)

Description:

The "Server" header reveals version information.

Mitigation Recommendations:

- Disable the Server header or configure it to display generic information.
- Apache: ServerSignature Off, ServerTokens Prod
- Nginx: server_tokens off;

6. X-Content-Type-Options Header Missing (Low)

Description:

The absence of this header allows MIME-type sniffing attacks.

Mitigation Recommendations:

- Set the following HTTP response header:
  X-Content-Type-Options: nosniff
- Ensure all static files are served with correct MIME types.

Conclusion:

This assessment highlights security weaknesses that need immediate remediation to prevent exploitation. Implementing the recommended mitigations will improve the website's resilience against attacks.

Next Steps:

- Prioritize implementing fixes for medium-severity issues.
- Regularly perform vulnerability assessments.
- Keep software and frameworks up to date.
- Conduct security awareness training for developers.

-Vaibhav Dhonde