Automated Security Testing Report : OWASP ZAP

[25/02/25]

1 Introduction

This report presents the findings from an automated security assessment conducted using **OWASP ZAP** on the target web application. The objective of this test was to identify vulnerabilities and provide remediation steps to enhance the application's security.

2 Tools Used

- **OWASP ZAP v2.16.0** – Automated security scanner for identifying vulnerabilities.
- **Target URL**: `http://testphp.vulnweb.com` .

3 Summary of Findings

The scan identified multiple security vulnerabilities categorized by risk levels:

Risk Level	Number of Issues
Medium	3
Low	3
Informational	5

Risk	Confidence				Total
	User Confirmed	High	Medium	Low	
High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
Medium	0 (0.0%)	1 (9.1%)	1 (9.1%)	1 (9.1%)	3 (27.3%)
Low	0 (0.0%)	1 (9.1%)	2 (18.2%)	0 (0.0%)	3 (27.3%)
Informational	0 (0.0%)	0 (0.0%)	1 (9.1%)	4 (36.4%)	5 (45.5%)
Total	0 (0.0%)	2 (18.2%)	4 (36.4%)	5 (45.5%)	11 (100%)

4 Identified Vulnerabilities and Remediation Steps

4.1 Content Security Policy (CSP) Header Not Set

Risk Level: Medium
Description: The application lacks a Content Security Policy (CSP) header, making it vulnerable to Cross-Site Scripting (XSS) and data injection attacks.
Remediation: Configure the web server to include a CSP header:

```
Content-Security-Policy: default-src 'self';
```

Adjust CSP directives based on application requirements.

4.2 Missing Anti-clickjacking Header

Risk Level: Medium
Description: The response does not contain 'X-Frame-Options' or 'frame-ancestors', leaving the application vulnerable to clickjacking attacks.
Remediation: Set the 'X-Frame-Options' or 'Content-Security-Policy' header:

```
X-Frame-Options: DENY
```

OR

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

4.3 Absence of Anti-CSRF Tokens

Risk Level: Medium
Description: Forms do not include CSRF protection, making them vulnerable to cross-site request forgery attacks.
Remediation: Implement CSRF tokens in all form submissions.

```
<input type="hidden" name="csrf_token" value="random_token_value">
```

Ensure backend validation of the CSRF token before processing requests.

4.4 Server Leaks Version Information via "Server" HTTP Header

Risk Level: Low
Description: The server exposes version information in HTTP response headers, which can help attackers identify vulnerabilities.
Remediation: Hide server version details:

```
# For Apache
ServerSignature Off
ServerTokens Prod

# For Nginx
server_tokens off;
```

5 Other Informational Findings

- Authentication Request Identified – Ensure authentication mechanisms are properly secured.
- Charset Mismatch – Align 'Content-Type' charset in headers and meta tags.
- Information Disclosure - Suspicious Comments – Remove sensitive developer comments.
- Modern Web Application Detection – Review security best practices for modern web frameworks.
- User Controllable HTML Element Attribute (Potential XSS) – Validate and sanitize user input.

6 Conclusion

This security assessment identified key vulnerabilities that should be addressed to improve the application's security posture. Implementing the recommended fixes will reduce the risk of attacks such as **Cross-Site Scripting (XSS), Clickjacking, CSRF, and Information Disclosure**. A follow-up scan is recommended after applying these fixes to ensure all vulnerabilities have been mitigated.

Issued by: [Moumoni Roy]
Date: [25/02/2025]