

# Cyber Security Project – OWASP ZAP Vulnerability Assessment

## 1. Aim

To perform security testing of a web application using OWASP ZAP tool and identify common vulnerabilities and security misconfigurations.

## 2. Objectives

- Understand cyber security concepts and CIA Triad
- Learn about attack surface and threats
- Perform automated scan using OWASP ZAP
- Analyze vulnerabilities and suggest fixes

## 3. Theory

**CIA Triad:** Confidentiality, Integrity and Availability are the core principles of cyber security. Confidentiality protects data from unauthorized access, Integrity ensures data is not modified illegally, Availability ensures services are accessible to authorized users.

**Attack Surface:** The total number of entry points where an attacker can try to enter data or extract data from a system such as web apps, APIs, network ports and user inputs.

## 4. Tools Used

- OWASP ZAP
- Web Browser
- Local web application

## 5. Steps Performed

1. Installed and launched OWASP ZAP
2. Entered target URL
3. Performed automated scan
4. Reviewed alerts and evidence
5. Generated HTML report

## 6. Findings

Common vulnerabilities detected:

- Missing X-Frame-Options Header – Risk: Medium
- Cookie without HttpOnly flag – Risk: Low
- Content Security Policy missing – Risk: Medium

## 7. Recommendations

- Implement security headers
- Use HttpOnly and Secure cookies
- Apply Content Security Policy

## 8. Conclusion

The assessment helped to understand practical web vulnerabilities and how OWASP ZAP can be used to detect and remediate them.



# Vulnerability Assessment Report using OWASP ZAP

This report contains the vulnerability assessment and security analysis of the target web application performed using OWASP ZAP tool. The scan was conducted to identify security weaknesses such as missing security headers, insecure cookies, and misconfigurations. The report includes risk levels, evidence, impact, and recommended remediation steps.

**Site:** <http://testphp.vulnweb.com>

**Generated on** Thu, 15 Jan 2026 16:55:13

**ZAP Version:** 2.17.0

ZAP by [Checkmarx](#)

## Summary of Alerts

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

## Insights

Level	Reason	Site	Description	Statistic
Medium	Exceeded High		Percentage of network failures	56 %
Low	Warning		ZAP errors logged - see the zap.log file for details	12
Low	Warning		ZAP warnings logged - see the zap.log file for details	2,736
Low	Exceeded High	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of slow responses	100 %
Info	Informational	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of responses with status code 2xx	90 %
Info	Informational	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of responses with status code 3xx	3 %
Info	Exceeded Low	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of responses with status code 4xx	6 %
Info	Informational	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of endpoints with content type application/octet-stream	1 %
Info	Informational	<a href="http://testphp.vulnweb.com">http://testphp.vulnweb.com</a>	Percentage of endpoints with content type application/x-shockwave-flash	1 %

Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type image/gif	2 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type image/jpeg	5 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type image/x-icon	1 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type text/css	3 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type text/html	76 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with content type text/xml	8 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with method GET	91 %
Info	Informational	http://testphp.vulnweb.com	Percentage of endpoints with method POST	8 %
Info	Informational	http://testphp.vulnweb.com	Count of total endpoints	85
Info	Informational	https://firefox-settings-attachments.cdn.mozilla.net	Percentage of responses with status code 2xx	100 %
Info	Informational	https://firefox-settings-attachments.cdn.mozilla.net	Percentage of endpoints with content type text/plain	100 %
Info	Informational	https://firefox-settings-attachments.cdn.mozilla.net	Percentage of endpoints with method GET	100 %
Info	Informational	https://firefox-settings-attachments.cdn.mozilla.net	Count of total endpoints	2
Info	Informational	https://firefox-settings-attachments.cdn.mozilla.net	Percentage of slow responses	80 %
Info	Informational	https://firefox.settings.services.mozilla.com	Percentage of responses with status code 2xx	100 %
Info	Informational	https://firefox.settings.services.mozilla.com	Percentage of endpoints with content type application/json	100 %
Info	Informational	https://firefox.settings.services.mozilla.com	Percentage of endpoints with method GET	100 %
Info	Informational	https://firefox.settings.services.mozilla.com	Count of total endpoints	1
Info	Informational	https://firefox.settings.services.mozilla.com	Percentage of slow responses	100 %
Info	Informational	https://www.acunetix.com	Percentage of responses with status code 2xx	100 %
Info	Informational	https://www.acunetix.com	Percentage of slow responses	100 %

## Summary of Sequences

For each step: result (Pass/Fail) - risk (of highest alert(s) for the step, if any).

## Alerts

Name

Risk Level

Number of  
Instances

## Alert Detail

## Sequence Details

With the associated active scan results.

The screenshot shows the OWASP ZAP interface with the 'Alerts' tab selected. The left sidebar lists various alerts, with 'Modern Web Application (Systemic)' highlighted. The main pane displays the details for this alert:

- URL:** http://testphp.vulnweb.com/AJAX/index.php
- Risk:** Informational
- Confidence:** Medium
- Parameter:**
- Attack:**
- Evidence:** <a href="" if onclick=""loadSomething('files.php')>files</a>
- CWE ID:**
- WASC ID:**
- Source:** Passive (10109 - Modern Web Application)
- Input Vector:**
- Description:** The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.
- Other Info:** Links have been found that do not have traditional href attributes, which is an indication that this is a modern web application.
- Solution:** This is an informational alert and so no changes are required.
- Reference:**
- Alert Tags:**

Key	Value
POLICY_QA_STD	
POLICY_PENTEST	
SYSTEMIC	https://www.zaproxy.org/docs/desktop/addons/common-library/alerttags/systemic
POLICY_DEV_STD	

The screenshot shows the OWASP ZAP interface with the 'Alerts' tab selected. The left sidebar lists various alerts, with 'User Controllable HTML Element Attribute (Potential XSS)' highlighted. The main pane displays the details for this alert:

- URL:** http://testphp.vulnweb.com/search.php?test=query
- Risk:** Informational
- Confidence:** Low
- Parameter:** goButton
- Attack:**
- Evidence:**
- CWE ID:** 20
- WASC ID:** 20
- Source:** Passive (10031 - User Controllable HTML Element Attribute (Potential XSS))
- Input Vector:**
- Description:** This check looks at user-supplied input in query string parameters and POST data to identify where certain HTML attribute values might be controlled. This provides hot-spot detection for XSS (cross-site scripting) that will require further review by a security analyst to determine exploitability.
- Other Info:** User-controlled HTML attribute values were found. Try injecting special characters to see if XSS might be possible. The page at the following URL: http://testphp.vulnweb.com/search.php?test=query
- Solution:** Validate all input and sanitize output if before writing to any HTML attributes.
- Reference:** https://cheatsheetseries.owasp.org/cheatsheets/Input\_Validation\_Cheat\_Sheet.html
- Alert Tags:**

Key	Value
OWASP_2021_A03	https://owasp.org/Top10/A03_2021-Injection/
CWE-20	https://cwe.mitre.org/data/definitions/20.html
POLICY_PENTEST	
OWASP_2017_A01	https://owasp.org/www-project-top-ten/2017/A1_2017-Injection.html

The screenshot shows the OWASP ZAP interface with a vulnerability alert titled "Server Leaks Version Information via 'Server' HTTP Response Header Field". The alert is categorized as "Low" risk and "High" confidence. The URL is `http://testphp.vulnweb.com`. The attack is passive, triggered by the "10036 - HTTP Server Response Header" source. The alert description states: "The web/application server is leaking version information via the 'Server' HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to." The solution suggests ensuring that the web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details. The alert tags include "POLICY\_PENTEST", "SYSTEMIC", "OWASP\_2017\_A06", "WSTG-v42-INFO-02", and "CWE-497". The alert tags table shows the following values:

Key	Value
POLICY_PENTEST	
SYSTEMIC	<a href="https://www.zaproxy.org/docs/desktop/addons/common-library/alerttags/#systemic">https://www.zaproxy.org/docs/desktop/addons/common-library/alerttags/#systemic</a>
OWASP_2017_A06	<a href="https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html">https://owasp.org/www-project-top-ten/2017/A6_2017-Security_Misconfiguration.html</a>
WSTG-v42-INFO-02	<a href="https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework.html">https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/01-Information_Gathering/08-Fingerprint_Web_Application_Framework.html</a>
CWE-497	<a href="https://cwe.mitre.org/data/definitions/497.html">https://cwe.mitre.org/data/definitions/497.html</a>

The screenshot shows the OWASP ZAP interface with a vulnerability alert titled "Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)". The alert is categorized as "Low" risk and "Medium" confidence. The URL is `http://testphp.vulnweb.com`. The attack is passive, triggered by the "10037 - Server Leaks Information via 'X-Powered-By' HTTP Response Header Field(s)" source. The alert description states: "The web/application server is leaking information via one or more 'X-Powered-By' HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to." The solution suggests ensuring that the web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers. The alert tags include "POLICY\_QA\_STD", "POLICY\_PENTEST", "SYSTEMIC", "OWASP\_2017\_A03", and "CWE-497". The alert tags table shows the following values:

Key	Value
POLICY_QA_STD	
POLICY_PENTEST	
SYSTEMIC	<a href="https://www.zaproxy.org/docs/desktop/addons/common-library/alerttags/#systemic">https://www.zaproxy.org/docs/desktop/addons/common-library/alerttags/#systemic</a>
OWASP_2017_A03	<a href="https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html">https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure.html</a>
CWE-497	<a href="https://cwe.mitre.org/data/definitions/497.html">https://cwe.mitre.org/data/definitions/497.html</a>