Vulnerability Assessment Report

Target website: testphp.vulnweb.com

Assessment type: Read only/passive

Prepared By: Harish Saud

Role: Cybersecurity Intern

Date: January 20, 2026

1. Executive Summary

This Vulnerability Assessment was conducted to evaluate the security posture of the publicly accessible website **testphp.vulnweb.com** using **passive and non-intrusive testing techniques**. The objective of the assessment was to identify common configuration weaknesses, assess their potential business impact, and provide clear remediation guidance in a manner suitable for non-technical stakeholders.

The assessment was performed strictly within a **read-only scope**, focusing only on public-facing components of the application. No exploitation, authentication bypass, brute-force attempts, or denial-of-service activities were conducted during the engagement. The analysis relied on passive traffic inspection, configuration review, and service exposure analysis using industry-recognized tools.

During the assessment, a total of **seven security findings** were identified. These findings primarily relate to **missing security headers, absence of protective controls, and information disclosure through server configuration**. No **high-risk or critical vulnerabilities** were observed as part of this assessment; however, several **medium-risk issues** were identified that could increase the application's exposure to common web-based attacks if left unaddressed.

The most notable issues include the absence of a **Content Security Policy (CSP)**, missing **anti-clickjacking protections**, lack of **anti-CSRF mechanisms**, and disclosure of server software and version information. While these issues do not represent an immediate compromise of the system, they weaken the overall security posture and may assist attackers in executing client-side attacks or conducting targeted reconnaissance.

From a business perspective, these weaknesses may lead to **reduced user trust, increased attack surface, and higher risk of browser-based attacks**, such as clickjacking or malicious script execution. The identified issues are largely **configuration-level weaknesses** and can be mitigated through well-established security best practices with minimal operational impact.

It is strongly recommended that the organization prioritize the implementation of missing security headers, reduce unnecessary information disclosure, and periodically review server configurations to align with modern web security standards. Addressing these findings will significantly enhance the website's resilience against common threats and demonstrate a proactive approach to cybersecurity risk management.

2. Scope of Assessment

The scope of this assessment was limited to the **publicly accessible components** of the website **testphp.vulnweb.com**. The assessment was conducted under a **read-only scope**, ensuring no impact on system availability, integrity, or confidentiality.

In Scope

- Public-facing web pages
- HTTP response headers and configurations
- Client-side behavior observable via passive analysis
- Service exposure and basic network visibility

Out of Scope

- Authentication and authorization testing
- Login bypass or credential-based attacks
- Exploitation of vulnerabilities
- Brute-force attacks, fuzzing, or DoS testing
- Access to administrative or restricted resources

3. Methodology & Tools Used

Methodology

The assessment followed a structured approach:

1. Target identification and scope definition
2. Passive vulnerability detection
3. Security header and configuration analysis
4. Service exposure analysis
5. Risk classification and reporting

Tools Used

- **OWASP ZAP (Passive Scan)** – OWASP Zed Attack Proxy (ZAP) is an open-source web application security tool used to identify common security issues by passively analyzing HTTP and HTTPS traffic between a browser and a web application. It helps detect misconfigurations such as missing security headers, insecure cookies, and information disclosure without actively attacking the target, making it suitable for ethical, read-only security assessments.
- **Nmap** – Nmap (Network Mapper) is an open-source network scanning tool used to identify publicly exposed services and open ports on a target system. In this assessment, Nmap was used in a non-intrusive manner to detect accessible services and server version information, helping evaluate basic network exposure without attempting exploitation or disrupting the target system.
- **Browser Developer Tools** – Browser Developer Tools are built-in features of modern web browsers that allow security analysts to inspect client-side behavior of web applications. In this assessment, they were used to review HTTP response headers, cookies, and security configurations directly from the browser, helping identify missing security controls and misconfigurations without interacting with or altering the application.

4. Risk Rating Criteria

Risk Level	Description
High	Issues that may directly lead to data compromise or system abuse
Medium	Security weaknesses that increase attack surface
Low	Information disclosure or best-practice gaps
Informational	Observations with no direct security impact

5. Summary of Findings

Vulnerability	Risk Level
Absence of Anti-CSRF Tokens	Medium
Content Security Policy (CSP) Not Set	Medium
Missing Anti-Clickjacking Header	Medium
Server Version Disclosure	Low
X-Content-Type-Options Header Missing	Low
X-Powered-By Header Disclosure	Low
Charset Mismatch	Low

6. Detailed Findings

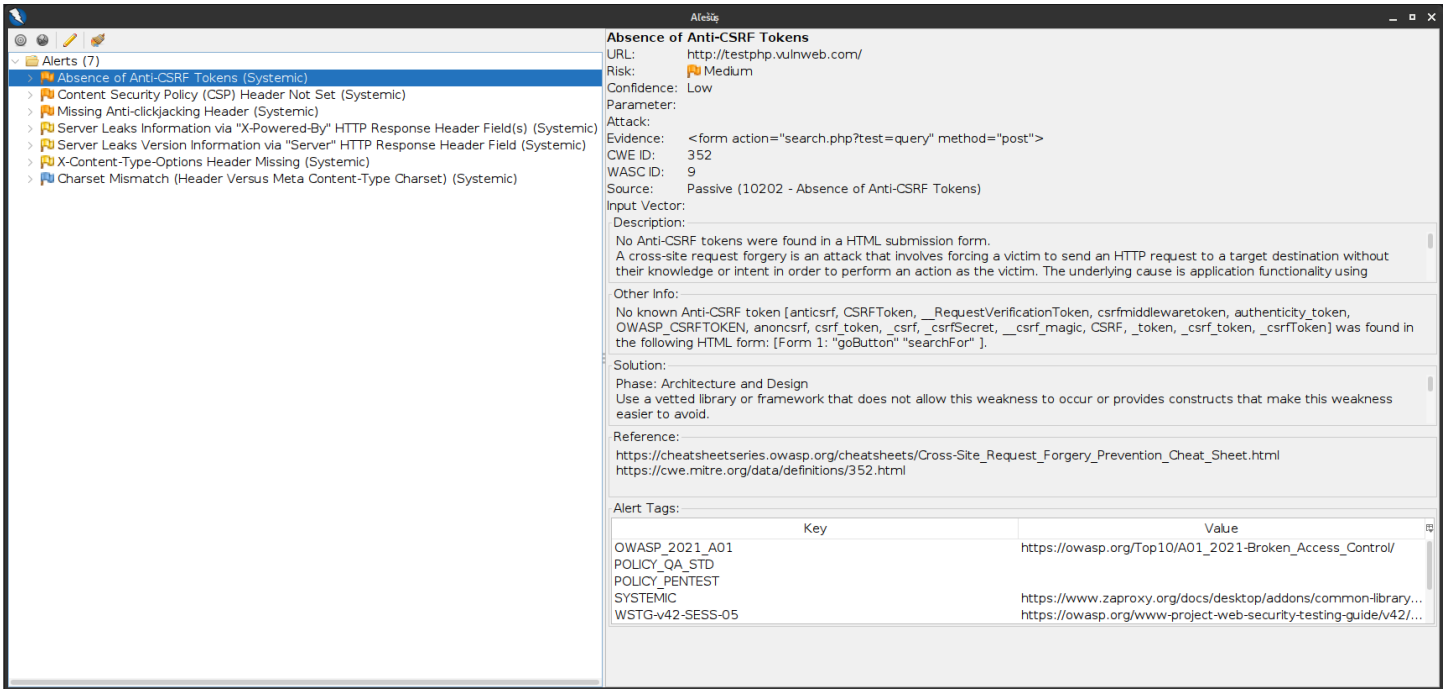
Finding 1: Absence of Anti-CSRF Tokens

Risk Level: Medium
Source: OWASP ZAP

Description:
The application does not implement Anti-Cross-Site Request Forgery (CSRF) tokens to protect against unauthorized requests initiated from external websites. Without CSRF protection, an attacker may be able to trick authenticated users into performing unintended actions, which could impact application integrity and user trust.

Business Impact:
The absence of Anti-CSRF tokens can allow attackers to manipulate user actions without their knowledge, potentially leading to unauthorized changes, misuse of application features, and loss of user confidence. This may result in reputational damage, customer dissatisfaction, and increased security risk for the business.

Evidence:



Recommendation:
Implement Anti-CSRF protection by generating unique, unpredictable tokens for all state-changing requests. Ensure these tokens are securely associated with user sessions and validated on the server side. Apply CSRF protection consistently across all forms and sensitive endpoints. Regularly review and test CSRF controls to ensure they remain effective against evolving threats.

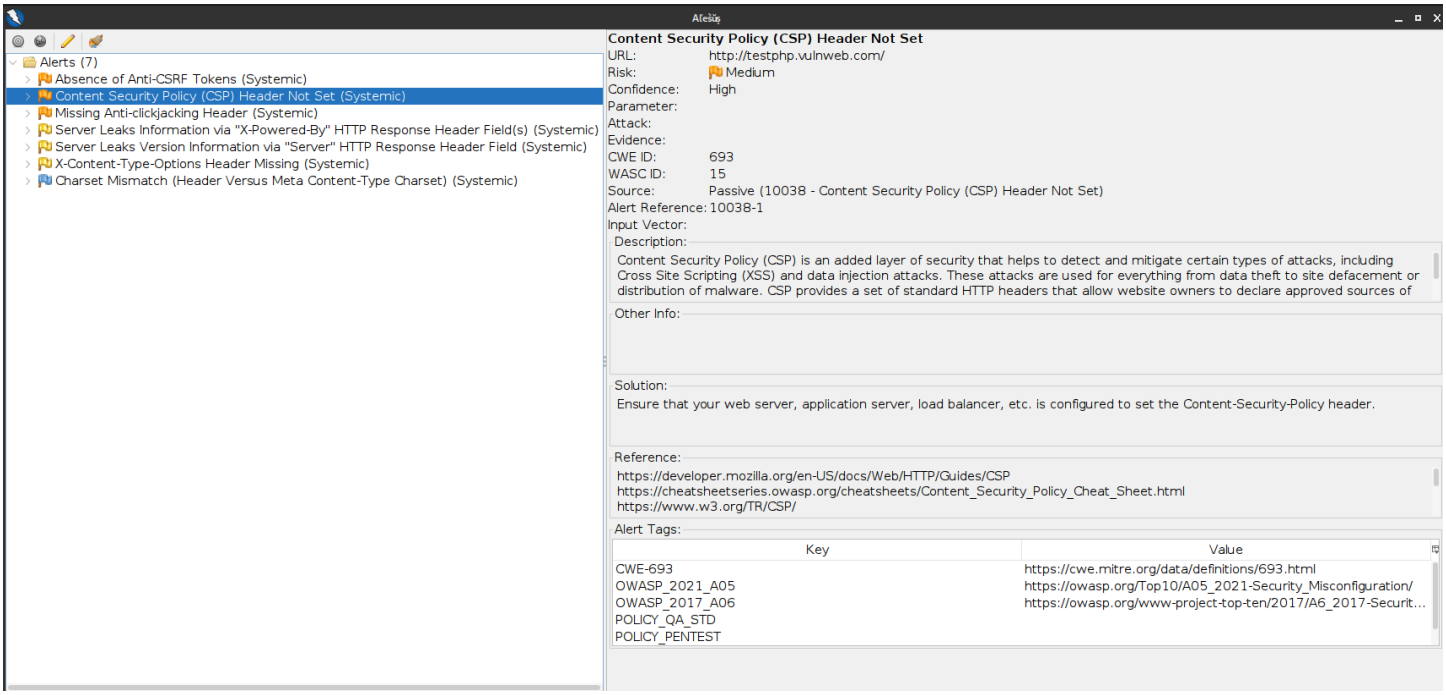
Finding 2: Content Security Policy (CSP) Header Not Set

Risk Level: Medium
Source: OWASP ZAP

Description:
The website does not define a Content Security Policy (CSP) header to control which external resources the browser is allowed to load. Without CSP, the browser has no restrictions on executing scripts or loading content from untrusted sources. This increases the likelihood of client-side attacks such as malicious script injection and data manipulation.

Business Impact:
The absence of a Content Security Policy (CSP) header increases the risk of malicious scripts being executed in users' browsers. This can lead to data theft, session compromise, and reduced trust in the website. For the business, such attacks may result in reputational damage, potential compliance issues, and loss of customer confidence.

Evidence:



Recommendation:
Implement a strong Content Security Policy (CSP) header to restrict the execution of scripts, styles, and other resources to trusted sources only.Define allowed domains explicitly and block inline and unauthorized scripts wherever possible.Test the CSP configuration in report-only mode before full enforcement to avoid breaking functionality.Regularly review and update the policy as the application evolves.

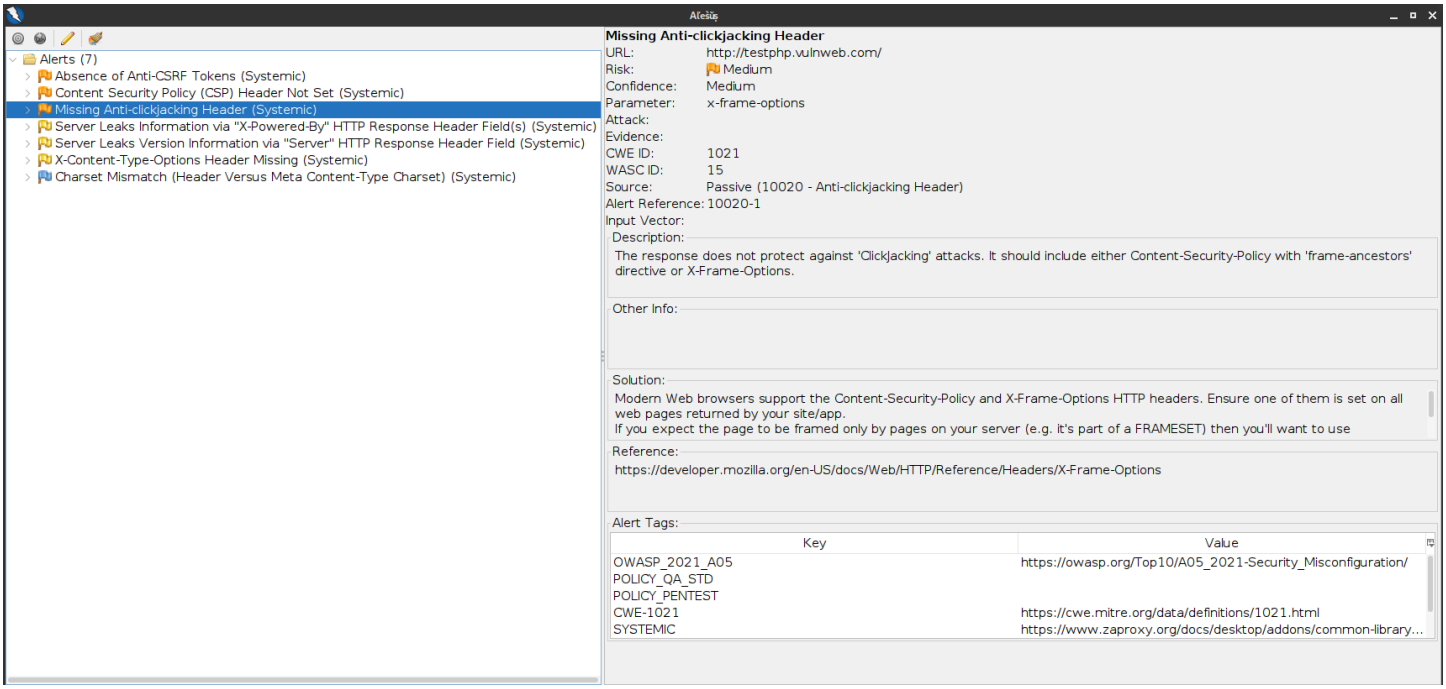
Finding 3: Missing Anti-Clickjacking Header

Risk Level: Medium
Source: OWASP ZAP

Description:
The website does not include an anti-clickjacking protection header, such as **X-Frame-Options** or the equivalent **frame-ancestors** directive in Content Security Policy. Without this protection, the application can be embedded within malicious iframes, potentially tricking users into interacting with hidden or disguised content.

Business Impact:
The absence of anti-clickjacking protection can allow attackers to deceive users into performing unintended actions on the website. This may lead to unauthorized transactions, misuse of application features, and erosion of user trust. For the business, such incidents can result in reputational damage and increased security risk.

Evidence:



Recommendation:
Configure the **X-Frame-Options** header to restrict how the website can be embedded in iframes. Alternatively, implement the **frame-ancestors** directive within a Content Security Policy for more flexible control. Apply this protection consistently across all pages to prevent clickjacking attacks. Regularly test and review header configurations to ensure continued effectiveness.

Finding 4: Server Version Disclosure

Risk Level: Low

Source: Nmap & OWASP ZAP

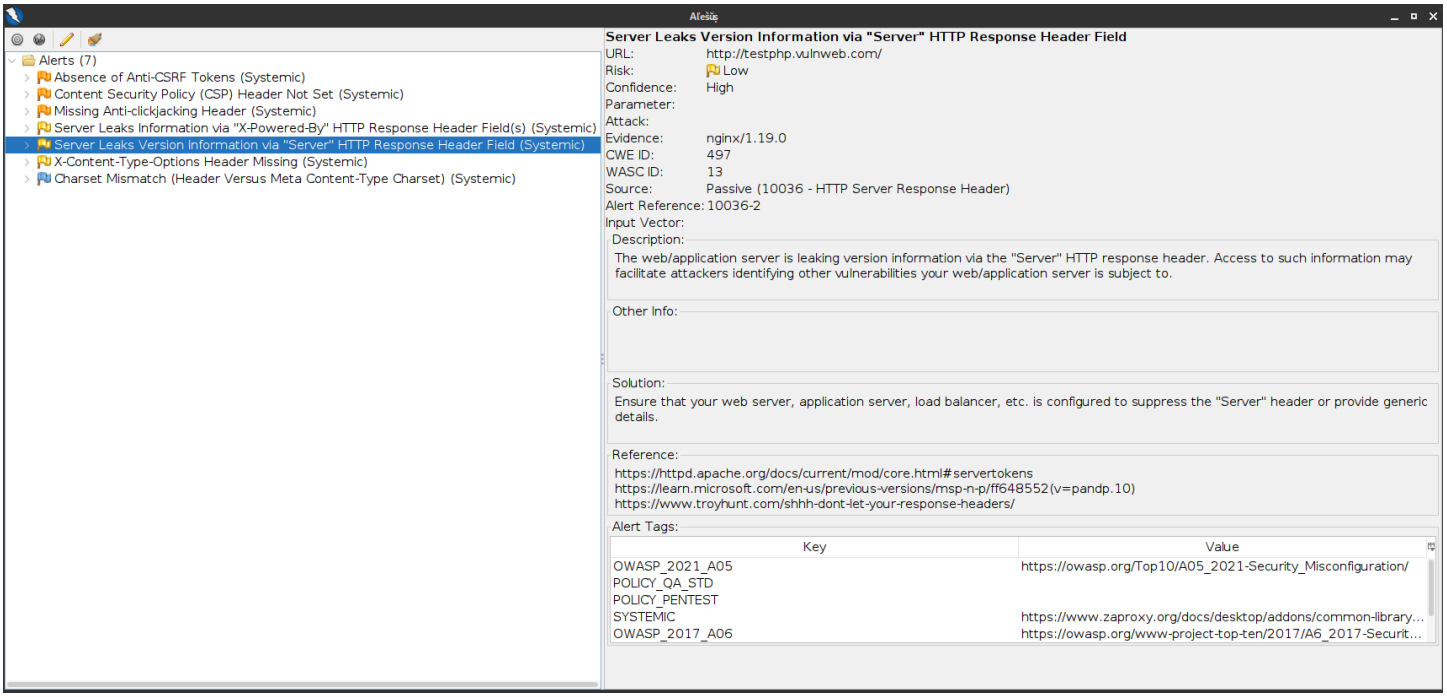
Description:

The server discloses its software type and version (nginx 1.19.0).The web server discloses detailed version information through HTTP response headers, such as the **Server** or **X-Powered-By** fields. Exposing this information allows attackers to identify the underlying server technology and version, which may help them target known vulnerabilities associated with that software.

Business Impact:

The disclosure of server software type and version information makes it easier for attackers to identify known vulnerabilities specific to the disclosed technology. This increases the risk of targeted attacks and reduces the overall security posture of the website. For the business, such exposure can lead to higher chances of exploitation and potential service disruption.

Evidence:



Recommendation:

Configure the web server to suppress or remove detailed version information from HTTP response headers. Ensure that headers such as **Server** and **X-Powered-By** reveal minimal or no technology details. Regularly update and patch the server software to reduce exposure to known vulnerabilities. Review server configuration periodically to maintain secure information disclosure practices.

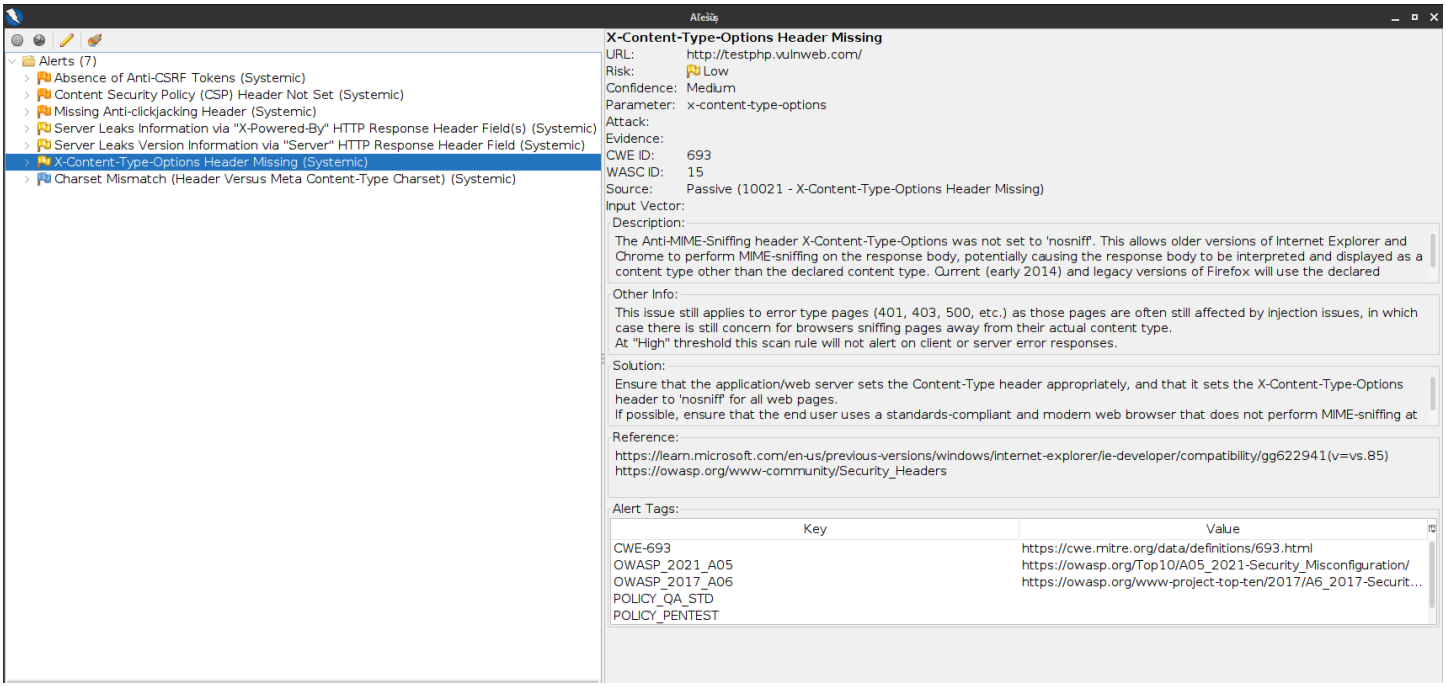
Finding 5: X-Content-Type-Options Header Missing

Risk Level: Low
Source: OWASP ZAP

Description:
The website does not include the **X-Content-Type-Options** security header in its HTTP responses. Without this header, browsers may attempt to interpret files as a different content type than intended, which can increase the risk of certain client-side attacks such as MIME-type sniffing.

Business Impact:
The absence of the X-Content-Type-Options header increases the risk of malicious content being incorrectly interpreted by users' browsers. This can potentially lead to script execution, data exposure, and a reduced level of trust in the website. For the business, such risks may result in security incidents and reputational damage.

Evidence:



Recommendation:
Configure the web server to include the **X-Content-Type-Options: nosniff** header in all HTTP responses. This prevents browsers from performing MIME-type sniffing and enforces proper content handling. Apply the header consistently across all application pages and resources. Regularly verify security headers as part of routine security reviews.

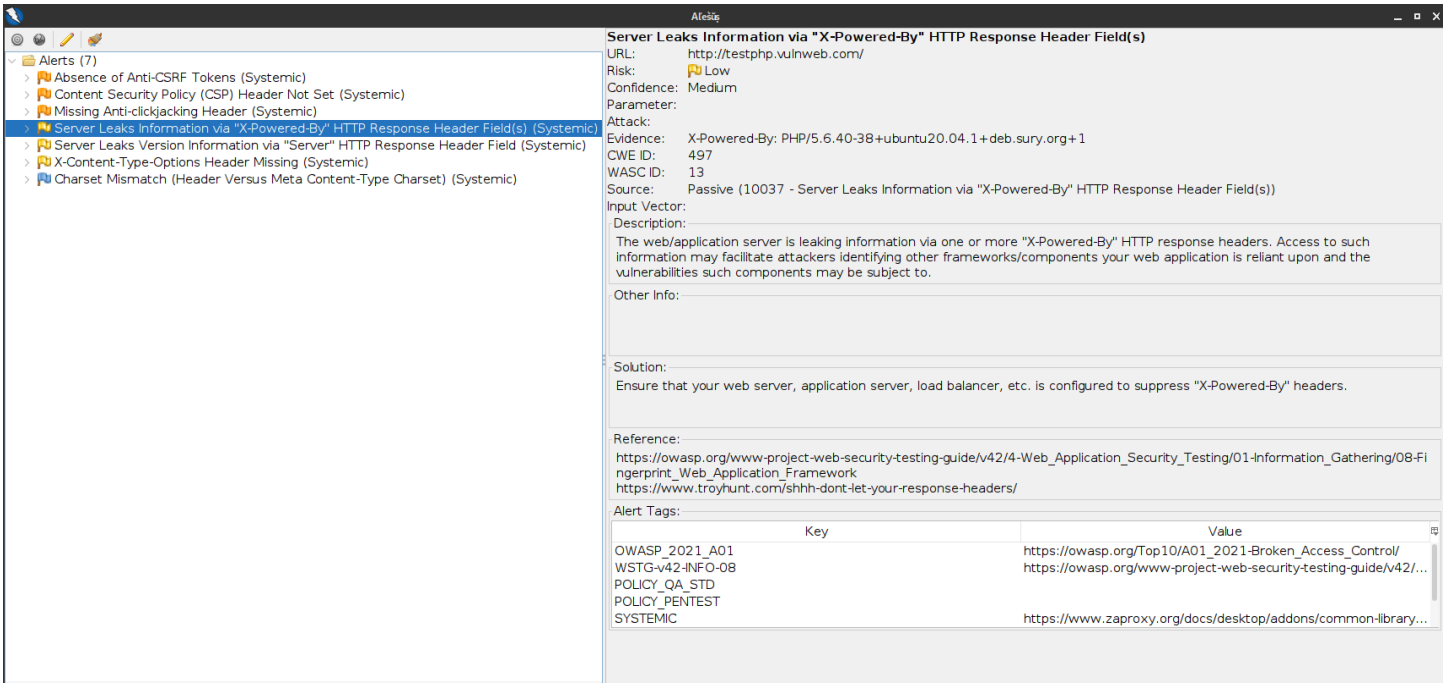
Finding 6: X-Powered-By Header Disclosure

Risk Level: Low
Source: OWASP ZAP

Description:
The website exposes the **X-Powered-By** HTTP response header, which reveals information about the underlying technology used by the application. Disclosing such implementation details provides unnecessary insight into the technology stack, which attackers can leverage to identify potential weaknesses or known vulnerabilities.

Business Impact:
The disclosure of technology details through the X-Powered-By header increases the likelihood of targeted attacks against known vulnerabilities in the underlying platform. This exposure weakens the overall security posture of the website and may lead to increased risk of exploitation. For the business, it can result in security incidents and loss of user confidence.

Evidence:



Recommendation:
Disable or remove the **X-Powered-By** header from server responses to limit exposure of internal technology details. Ensure only necessary information is included in HTTP headers. Keep all underlying technologies updated and securely configured.Periodically review response headers to prevent unintended information disclosure.

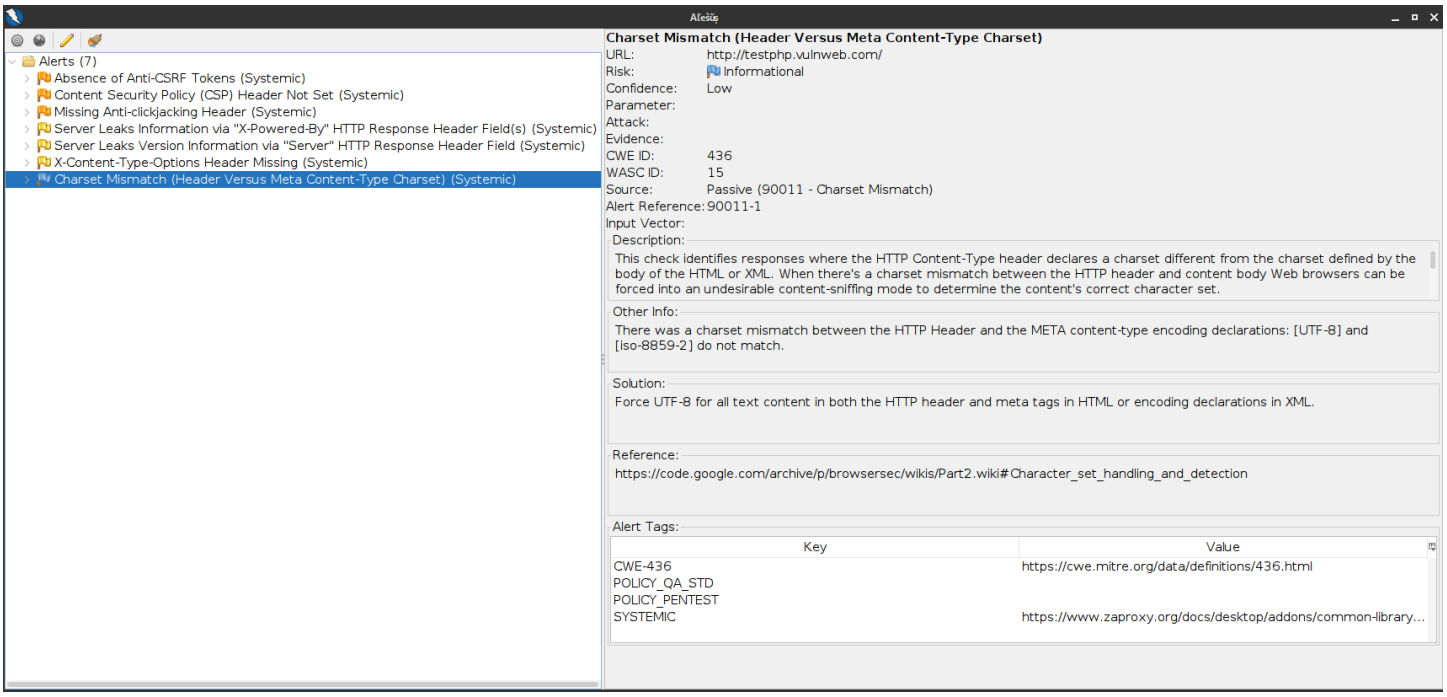
Finding 7: Charset Mismatch

Risk Level: Low
Source: OWASP ZAP

Description:
A charset mismatch was identified between the HTTP response header and the HTML meta Content-Type declaration. When character encoding is inconsistently defined, browsers may incorrectly interpret page content, which can lead to unexpected behavior and potential security issues such as improper content rendering or injection risks.

Business Impact:
A charset mismatch can cause browsers to misinterpret page content, leading to display issues and inconsistent user experience. In some cases, it may increase the risk of client-side security issues, such as improper handling of user input. For the business, this can reduce application reliability and user trust.

Evidence:



Recommendation:
Ensure that the character encoding defined in the HTTP response header matches the encoding specified in the HTML meta Content-Type tag.Standardize the application to use a single, secure charset such as UTF-8. Apply consistent encoding settings across all pages and resources. Regularly validate response headers and page content for encoding consistency.

7. Network Exposure Analysis (Nmap)

Open Ports Identified:

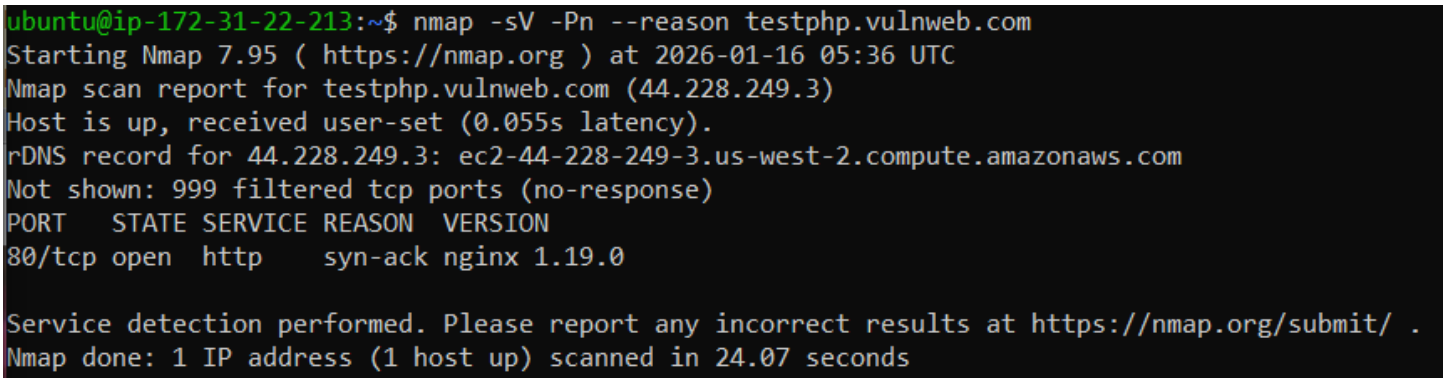
- Port 80 (HTTP)

Service Detected:

- nginx 1.19.0

Risk Level: Low

Evidence:



8. Overall Recommendations

Overall, it is recommended that the website improve its security posture by implementing all missing HTTP security headers to enhance protection against common client-side attacks. Unnecessary disclosure of server and technology information should be minimized to reduce the risk of targeted attacks. Security configurations should be applied consistently across all public-facing pages and reviewed regularly to ensure they remain effective. The server and underlying technologies should be kept up to date with the latest security patches and best practices. Additionally, periodic vulnerability assessments and security reviews should be conducted to proactively identify and address emerging risks, helping maintain system reliability, protect user data, and preserve business trust.

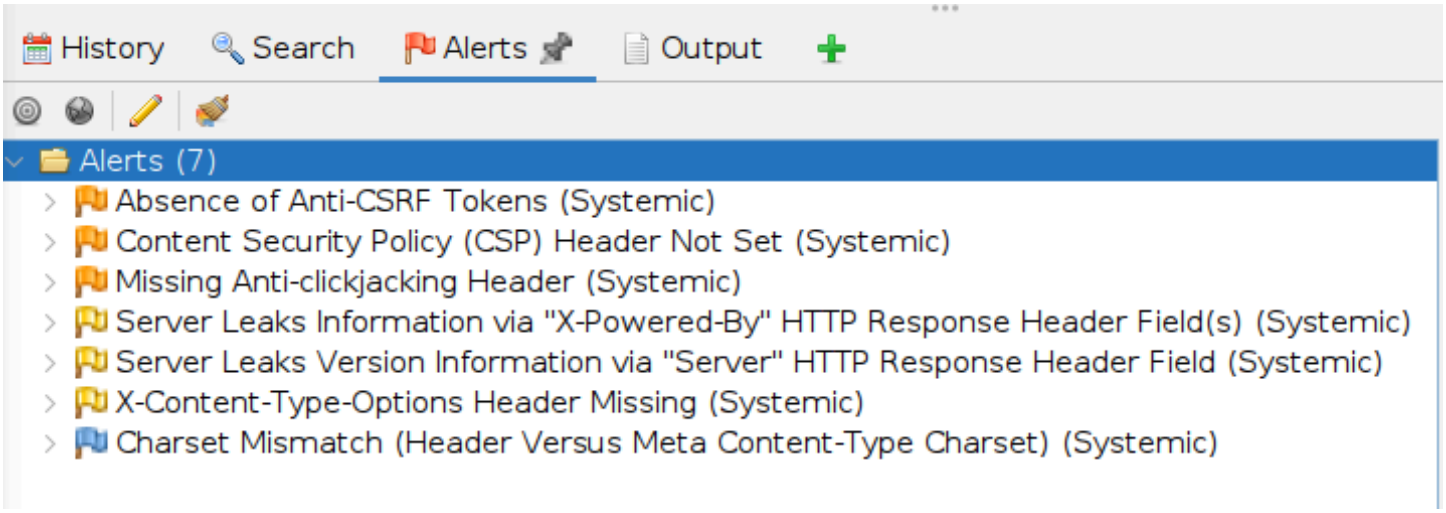
9. Conclusion

This vulnerability assessment was conducted to evaluate the security posture of the target website using passive and non-intrusive techniques. The assessment identified several security misconfigurations, primarily related to missing HTTP security headers and information disclosure through server responses. While no critical vulnerabilities requiring immediate exploitation were observed, these issues may increase the risk of client-side attacks and targeted threats if left unaddressed.

Implementing the recommended security controls will help strengthen the website's overall security, reduce exposure to common web-based risks, and improve resilience against potential attacks. Regular security reviews and vulnerability assessments are advised to ensure continued protection as the application evolves. This assessment demonstrates the importance of proactive security measures in maintaining user trust and supporting long-term business objectives.

10. Appendix (Evidence)

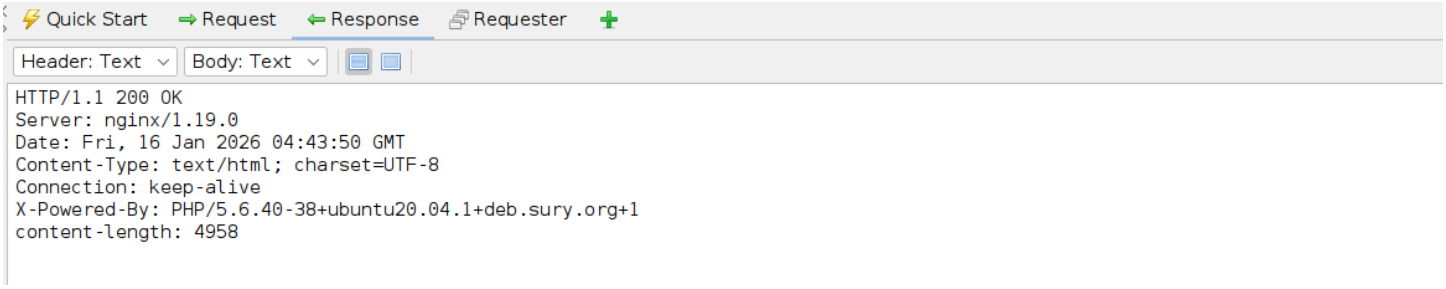
- OWASP ZAP Alerts Summary – During the passive security assessment conducted using OWASP ZAP, multiple security misconfigurations were identified related to missing or improperly configured HTTP security headers and information disclosure. The findings primarily indicate weaknesses in browser-side security controls, which may increase the risk of client-side attacks such as clickjacking, cross-site request forgery, and malicious script execution. No active exploitation was performed, and all alerts were generated through passive analysis of server responses. Overall, the identified issues suggest opportunities to strengthen the website's security posture through proper header configuration and secure server settings.



- Individual ZAP Findings – The OWASP ZAP passive scan identified several individual security findings related to HTTP response headers and application configuration. Each finding represents a specific area where security controls are either missing or not optimally configured. While none of the issues indicate immediate exploitation, they collectively increase the website's exposure to client-side attacks and information disclosure. The findings have been assessed individually based on potential impact and likelihood, with clear recommendations provided to help improve the overall security posture of the application.

- Request:

- Response:



- Zap Sites Tree: