VULNERABILITY ASSESSMENT REPORT

Target: http://testphp.vulnweb.com
Assessment Type: Passive Security Analysis
Date:19/02/2026
Prepared By: Shreesh Kumar

Introduction

In today’s digital environment, website security is critical for maintaining customer trust, protecting sensitive information, and ensuring uninterrupted business operations. Even minor misconfigurations can expose a website to significant risks. This report presents a structured vulnerability assessment of the public website http://testphp.vulnweb.com. The assessment was conducted strictly within a read-only scope, meaning no exploitation, brute force attempts, login bypass, or disruptive testing methods were performed.

The purpose of this assessment is to identify visible security weaknesses that could potentially impact confidentiality, integrity, or availability. The review focuses on public-facing configurations and response behaviors observable through passive analysis techniques.

Three primary security issues were identified during this evaluation: absence of HTTPS encryption, missing security headers, and server version disclosure. Each issue has been analyzed in terms of technical impact, business risk, and recommended remediation.

This report aims to provide clear, actionable recommendations in non-technical language suitable for decision-makers and business stakeholders.

Scope and Methodology

This assessment was conducted under strict ethical and professional guidelines. Only publicly accessible pages were reviewed, and no attempts were made to access restricted areas or modify application data. The evaluation was entirely passive and observational in nature.

The tools used during the assessment included Nmap for service visibility analysis, browser developer tools for header inspection, and general manual inspection of HTTP responses. No exploitation frameworks, attack payloads, or active intrusion techniques were utilized.

The scope included:

- Examination of connection security (HTTP vs HTTPS)
- Analysis of response headers
- Identification of server technology exposure

The following activities were explicitly excluded:

- Login bypass attempts
- SQL injection testing
- Cross-site scripting exploitation
- Denial-of-service testing
- Data manipulation

The objective was to assess configuration-level weaknesses that are visible to any internet user. The findings represent publicly exposed security misconfigurations rather than confirmed exploit paths.

This methodology reflects standard industry practices for preliminary security posture evaluations.

Overview

During the assessment, three significant configuration-related vulnerabilities were identified. These weaknesses do not require advanced exploitation techniques to observe, making them particularly important from a risk management perspective.

The first vulnerability involves the absence of HTTPS encryption. The website operates over HTTP, meaning communication between users and the server is not encrypted. This can expose transmitted data to interception risks.

The second vulnerability relates to missing security headers. Modern web security relies heavily on browser-enforced protections delivered through HTTP response headers. Several recommended headers were not present, increasing the risk of client-side attacks.

The third vulnerability involves information disclosure through server response headers. The server reveals specific software and version details, which could assist attackers in identifying known vulnerabilities associated with those versions.

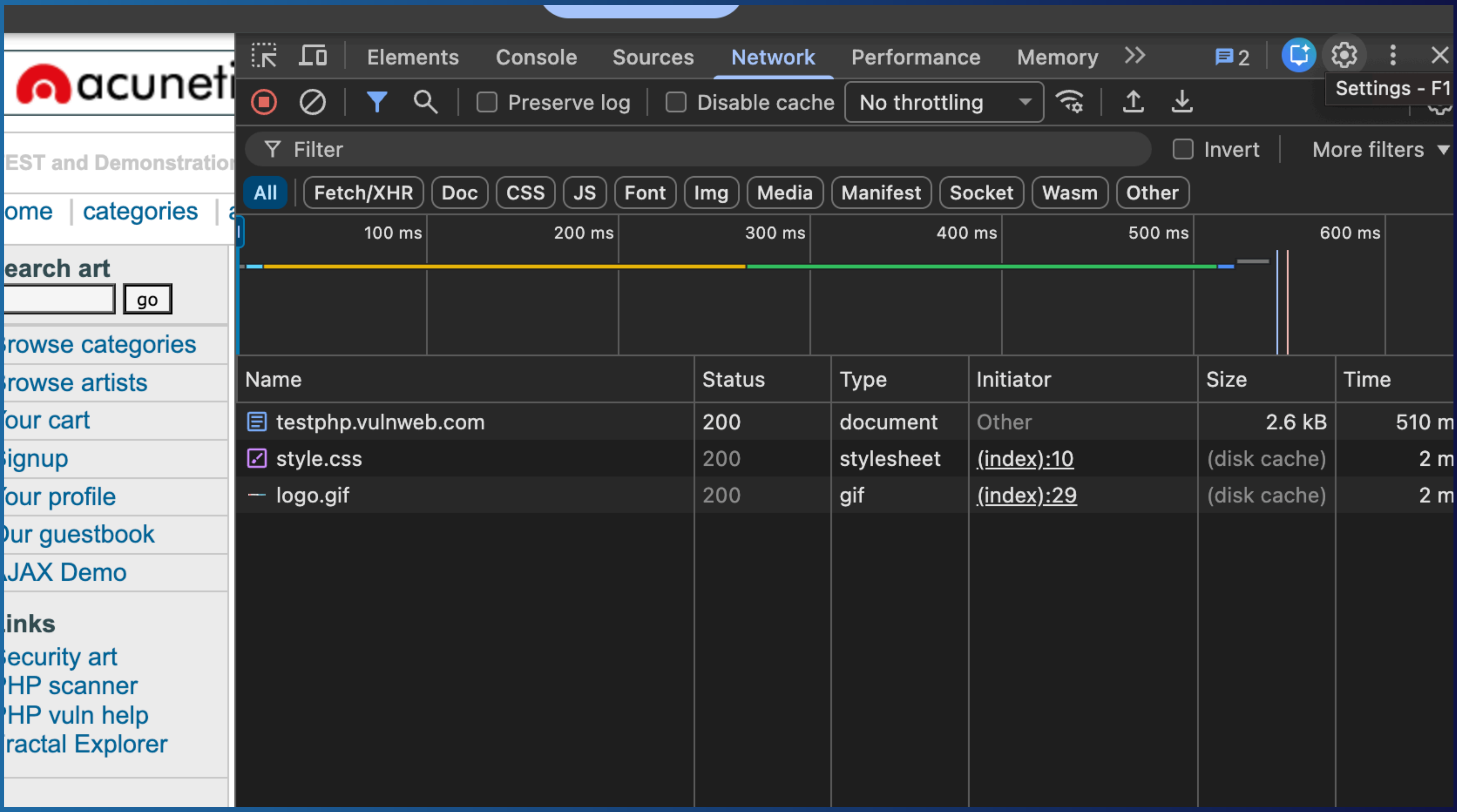
Each of these issues increases exposure in different ways. While none represent active exploitation during this assessment, they significantly impact the overall security posture and should be addressed proactively.

Vulnerability 1: Insecure Communication (No HTTPS)

The website operates entirely over HTTP rather than HTTPS. This means that data transmitted between the user’s browser and the web server is not encrypted. When encryption is absent, data packets travel across networks in plain text and may be intercepted by malicious actors.
Without HTTPS, sensitive information such as login credentials, form submissions, session identifiers, and browsing activity can potentially be captured through man-in-the-middle attacks. Even if the website does not currently process payment information, the absence of encryption significantly weakens overall security trust.
Modern browsers mark HTTP sites as “Not Secure,” which can damage user confidence and business credibility. Additionally, search engines prioritize HTTPS-enabled websites, meaning this issue may also impact visibility and reputation.

This vulnerability was confirmed by observing the URL structure and the absence of a secure connection indicator in the browser.
Risk Level: High

The absence of HTTPS is considered a critical security weakness because it affects all users accessing the website.



Name	Status	Type	Initiator	Size	Time
testphp.vulnweb.com	200	document	Other	2.6 kB	510 m
style.css	200	stylesheet	(index):10	(disk cache)	2 m
logo.gif	200	gif	(index):29	(disk cache)	2 m

The screenshot shows that the website is operating over HTTP instead of HTTPS, as indicated by the “Not Secure” warning in the browser’s address bar. This means the connection between the user and the server is not encrypted. Without SSL/TLS encryption, data such as login credentials, session cookies, and form inputs can be intercepted or modified during transmission. This exposes users to risks like man-in-the-middle attacks and data theft. Modern security standards require HTTPS to protect confidentiality and maintain user trust. Implementing a valid SSL certificate and enforcing HTTPS redirection would significantly enhance the website’s security posture.

Vulnerability 2: Missing Security Headers

Security headers provide browser-level protections against common web-based attacks. During the assessment, several recommended headers were missing from server responses.

Missing headers include:

- X-Frame-Options
- Content-Security-Policy
- X-Content-Type-Options
- Strict-Transport-Security
- X-XSS-Protection

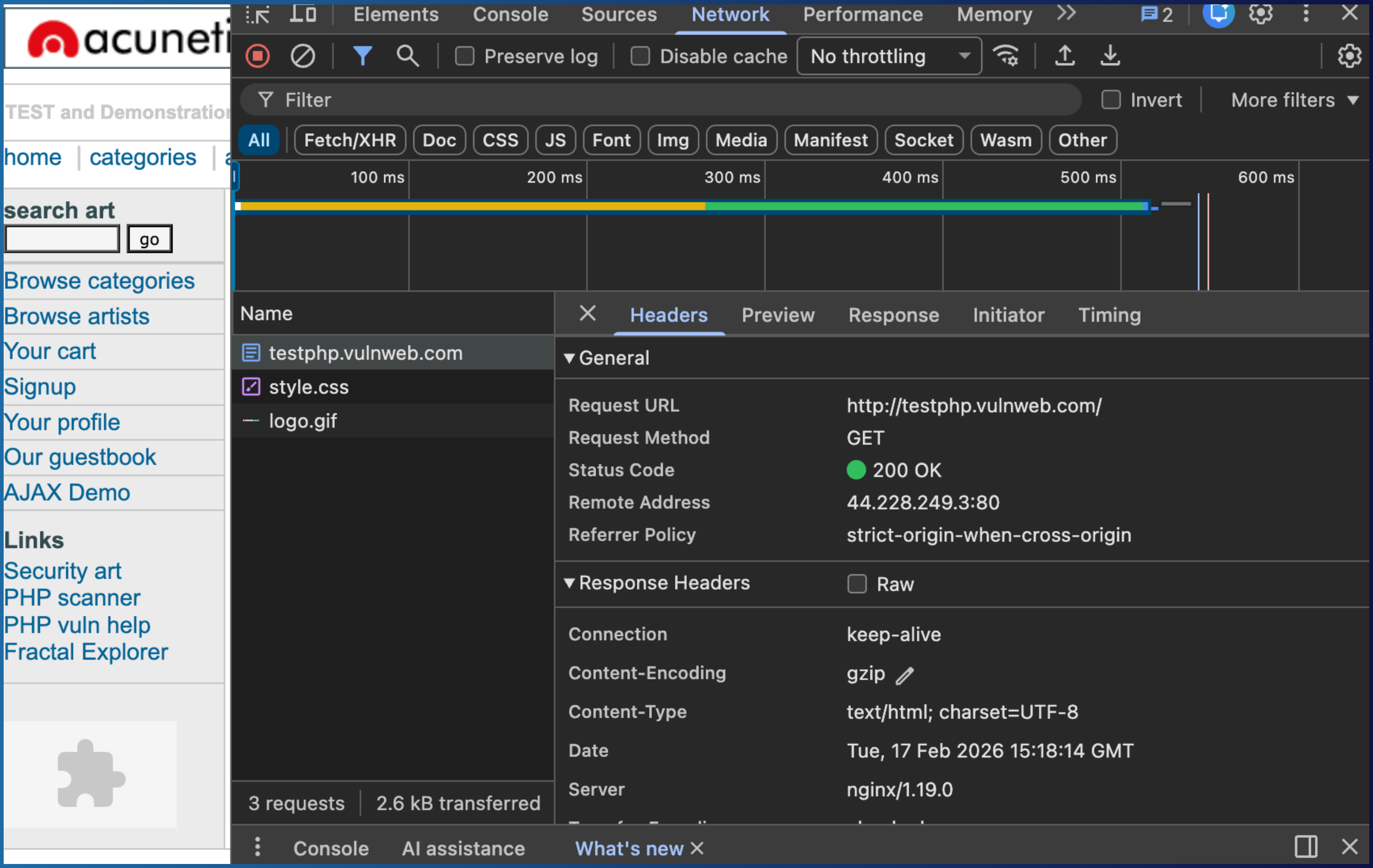
These headers instruct browsers how to handle content securely. Without them, the website is more vulnerable to clickjacking, cross-site scripting, MIME-type confusion, and other client-side threats.

For example, the absence of X-Frame-Options allows attackers to embed the website within malicious frames. Similarly, missing Content-Security-Policy increases the risk of unauthorized script execution.

These weaknesses were identified through inspection of response headers using browser developer tools.

Risk Level: Medium

While not immediately exploitable without additional weaknesses, missing headers reduce layered defense mechanisms.



▼General

Request URL	http://testphp.vulnweb.com/
Request Method	GET
Status Code	● 200 OK
Remote Address	44.228.249.3:80
Referrer Policy	strict-origin-when-cross-origin

▼Response Headers □ Raw

Connection	keep-alive
Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Tue, 17 Feb 2026 15:18:14 GMT
Server	nginx/1.19.0

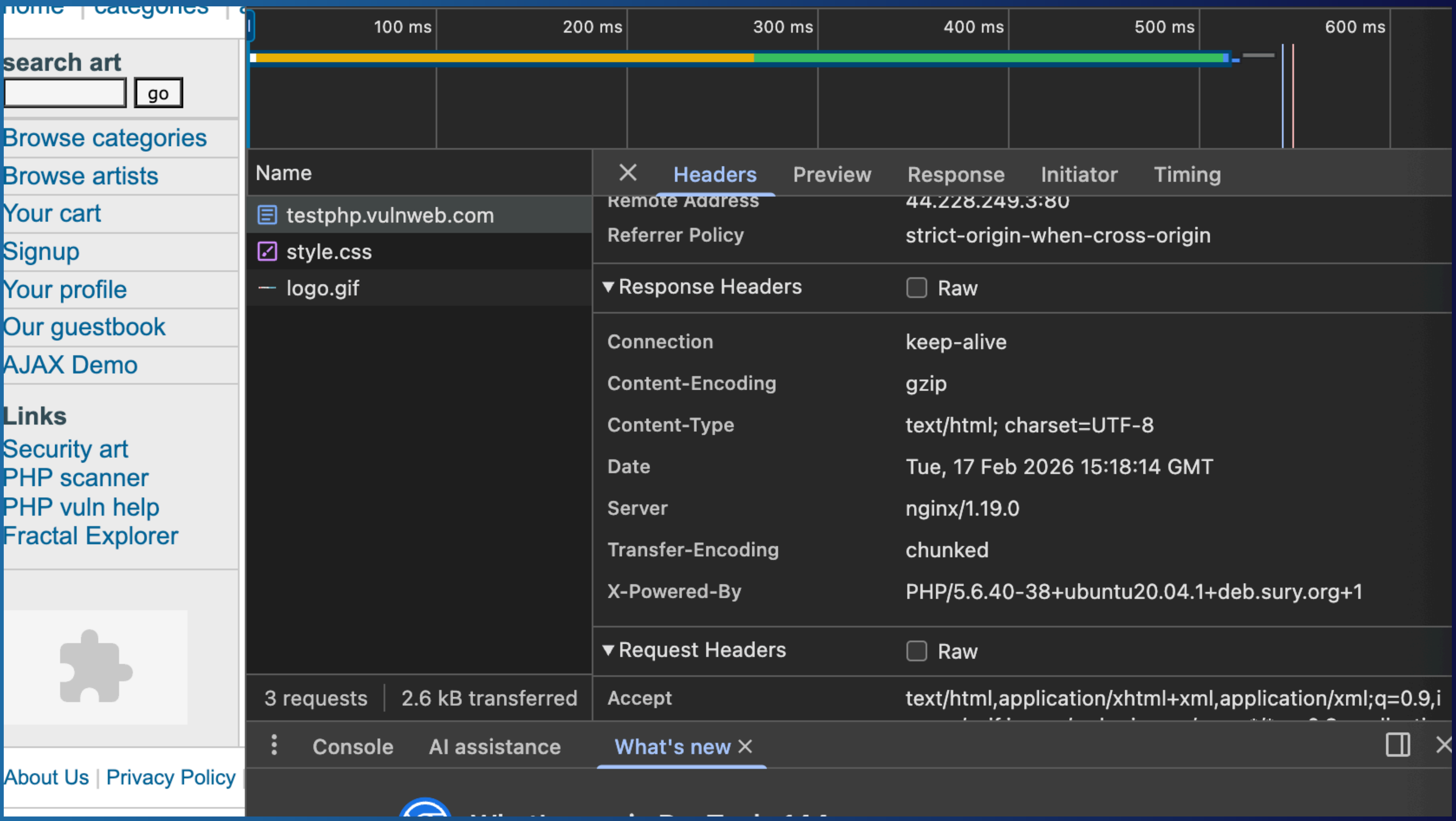
The screenshot shows the HTTP response headers of the website viewed through the browser's Network tab. While basic headers such as Content-Type, Server, and Connection are present, important security headers are missing. Headers like X-Frame-Options, Content-Security-Policy, X-Content-Type-Options, and Strict-Transport-Security are not visible in the response. These headers play a critical role in protecting websites against attacks such as clickjacking, cross-site scripting (XSS), and MIME-type sniffing. Without these protective headers, the browser does not enforce additional security controls, increasing exposure to client-side attacks. Although this issue does not directly exploit the system, it weakens the overall security posture of the website. Implementing these headers would significantly enhance browser-level protection and reduce security risks.

Vulnerability 3: Server & Version Disclosure

The web server exposes technology and version details in response headers. Specifically, server type and underlying software versions are visible to any user inspecting network responses.
This information was observed through response headers and confirmed using passive service detection methods.
Revealing exact software versions provides attackers with valuable intelligence. They can search public vulnerability databases for known weaknesses affecting those versions. This increases the risk of targeted attacks.

Although version disclosure alone does not constitute an exploit, it simplifies reconnaissance efforts and reduces the effort required to identify attack vectors.
Risk Level: Medium

Information disclosure issues are often overlooked but play a crucial role in the attack lifecycle



testphp.vulnweb.com
style.css
logo.gif

3 requests | 2.6 kB transferred

Remote Address	44.228.249.3:80
Referrer Policy	strict-origin-when-cross-origin

▾Response Headers □ Raw

Connection	keep-alive
Content-Encoding	gzip
Content-Type	text/html; charset=UTF-8
Date	Tue, 17 Feb 2026 15:18:14 GMT
Server	nginx/1.19.0
Transfer-Encoding	chunked
X-Powered-By	PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1

▾Request Headers □ Raw

Accept	text/html,application/xhtml+xml,application/xml;q=0.9,i

The screenshot shows the HTTP response headers revealing detailed server and backend technology information. Specifically, the headers disclose nginx/1.19.0 as the web server and PHP/5.6.40 as the backend scripting engine. This version information is publicly visible to any user inspecting network traffic. Exposing such technical details is considered a security misconfiguration because it aids attackers during reconnaissance. By knowing the exact software versions, malicious actors can search vulnerability databases for known exploits targeting those versions. Notably, PHP 5.6 is an end-of-life version and no longer receives security updates, increasing potential risk. Suppressing server tokens and removing the X-Powered-By header would reduce unnecessary exposure and strengthen the security posture.

Risk Classification Summary

The vulnerabilities identified during this assessment vary in severity and impact. A risk-based classification helps prioritize remediation efforts.

High Risk:

- Absence of HTTPS encryption.

Medium Risk:

- Missing security headers.
- Server version disclosure.

The highest priority should be implementation of HTTPS, as it directly affects all user communications. Security header configuration and information disclosure mitigation should follow as immediate secondary improvements.

The overall security posture can be described as moderate risk with critical encryption gaps. None of the vulnerabilities require advanced exploitation to observe, which increases their importance.

Addressing these configuration weaknesses would significantly improve the website's defensive resilience and trustworthiness.

Overall Security Posture Assessment

The website demonstrates functional availability and accessibility but lacks modern security hardening practices. The identified vulnerabilities are primarily configuration-based rather than application-logic flaws.

This suggests that improving server-level security settings could substantially enhance protection without major architectural changes.

The absence of encryption represents the most urgent issue. Missing headers and information disclosure further indicate that security hardening measures were not fully implemented.

The overall posture can be categorized as:

Security Maturity Level: Basic

Immediate Action Required: Yes (HTTPS deployment)

Long-Term Recommendation: Implement security configuration baseline

Proactive remediation would reduce exposure to interception, injection, and reconnaissance-based attacks.

Conclusion

This read-only vulnerability assessment identified three significant security weaknesses affecting the public website. The findings demonstrate that even without active exploitation, meaningful security risks can be observed through passive analysis.

The most critical issue is the absence of HTTPS encryption, which exposes all user communication. Missing security headers reduce browser-enforced protections, and server version disclosure increases reconnaissance risk.

None of the identified vulnerabilities require advanced attack techniques to detect. This emphasizes the importance of proper configuration and security hygiene.

By implementing recommended remediation steps, the organization can significantly improve confidentiality, integrity, and overall trustworthiness.

This assessment highlights the value of proactive security consulting and regular configuration reviews in maintaining a strong digital security posture.