Future Interns

Vulnerability Assessment Report for a Live Website

Conducted using industry-standard security testing tools

Target Website:
http://testphp.vulnweb.com

Assessment Type:
Read-Only Web Application Vulnerability Assessment

Assessment Date:
31/01/2026

Prepared By:
Neeraj Mudunuru (FIT/JAN26/CS5927)
Cybersecurity Student | Aspiring Security Analyst

Tools Used:
Nmap, OWASP ZAP (Passive Scan), Browser Developer Tools

Confidentiality Notice

Executive Summary

A non-intrusive security assessment was performed on the target website to identify common security weaknesses that could impact data protection and user trust. The assessment was conducted in a **read-only manner**, without exploiting vulnerabilities or disrupting services.

The review identified **one high-risk vulnerability** related to unencrypted communication, along with **multiple medium and low-risk issues** primarily involving security misconfigurations and client-side weaknesses.

If exploited, these issues could allow attackers to intercept user data, inject malicious scripts, or gain insight into the technologies used by the application. While no active compromise was observed, these findings indicate areas where the website's security posture can be strengthened.

All identified issues are **remediable using standard security best practices**. Addressing the recommendations outlined in this report will significantly reduce risk and improve the overall security resilience of the application.

Scope & Methodology

Assessment Scope

The scope of this security assessment was limited to a **non-intrusive, read-only review** of the target website. The objective was to identify **commonly occurring security weaknesses** without performing exploitation or causing any impact to system availability or data integrity.

The assessment included:

- Review of publicly accessible web pages
- Analysis of HTTP response headers
- Passive vulnerability identification
- Network-level visibility checks

The following activities were **explicitly excluded** from this assessment:

- Authentication bypass attempts
- Data extraction or modification
- Denial-of-Service (DoS) testing
- Brute-force or exploitation techniques

Assessment Methodology

The assessment was conducted using **industry-standard security tools and techniques**, focusing on passive observation and configuration analysis. The methodology aligns with common best practices used in preliminary web security reviews.

Tools and techniques used during the assessment included:

- **Nmap** for network-level service identification
- **OWASP ZAP (Passive Scan)** for application-level vulnerability detection
- **Browser Developer Tools** for client-side inspection and header analysis

All findings documented in this report are based on **observable behavior and tool-generated evidence** at the time of assessment.

Risk Summary

The table below provides a **high-level overview** of the security issues identified during the assessment, categorized by risk severity. This summary allows stakeholders to quickly understand the overall risk posture without reviewing technical details.

Risk Level	Number of Findings	Description
🔴 High	1	Critical issue that could directly expose sensitive data or communications
🟠 Medium	3	Issues that increase attack surface or enable client-side attacks
🟡 Low / Informational	1	Issues related to information exposure and security hardening
Total	5	—

Summary of Identified Issues

- **High Risk**
  - Lack of HTTPS Encryption
- **Medium Risk**
  - Missing HTTP Security Headers
  - Exposure of HTTP Service on Port 80
  - Cross-Site Scripting (DOM-Based)
- **Low / Medium Risk**
  - Server and Technology Information Disclosure

Business Impact Overview

The presence of high and medium-risk vulnerabilities indicates that the website is exposed to **avoidable security risks**. While no exploitation was performed, attackers could potentially leverage these weaknesses to intercept data, manipulate content, or gain insight into backend technologies.

Addressing the high-risk issue should be treated as a **priority**, followed by remediation of medium-risk findings to improve the overall security posture.

Finding 1: Lack of HTTPS Encryption

Risk Level: 🔴High

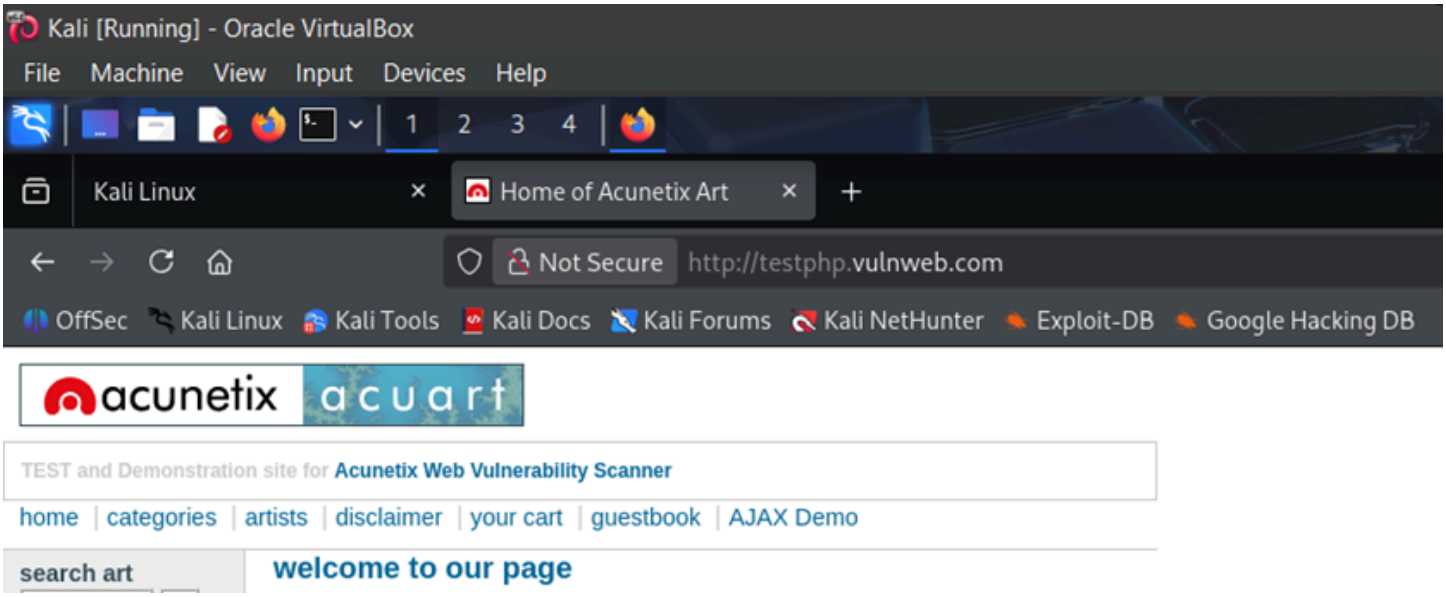
Description:

The website is accessible over the **HTTP protocol**, which does not provide encryption for data transmitted between the user's browser and the server. When accessing the site, modern browsers indicate the connection as **"Not Secure"**, confirming that communication is not protected by SSL/TLS.

Observation (Evidence):

During the assessment, the website was observed to be reachable via http://, and the browser displayed a security warning indicating an unencrypted connection.

Evidence Reference:



Business Impact:

Without HTTPS encryption, sensitive information transmitted by users—such as login details, form data, or session identifiers—can be:

- Intercepted by attackers on the same network
- Modified during transmission
- Stolen through man-in-the-middle attacks

This can lead to **loss of user trust**, potential **data breaches**, and **reputational damage** to the organization.

Recommendation:

It is strongly recommended to:

- Implement a valid **SSL/TLS certificate** on the web server
- Redirect all HTTP traffic to HTTPS
- Enable **HTTP Strict Transport Security (HSTS)** to enforce secure connections

Addressing this issue should be treated as a **top priority**, as it significantly reduces the risk of data interception.

Finding 2: Missing HTTP Security Headers

Risk Level: 🟠Medium

Description:

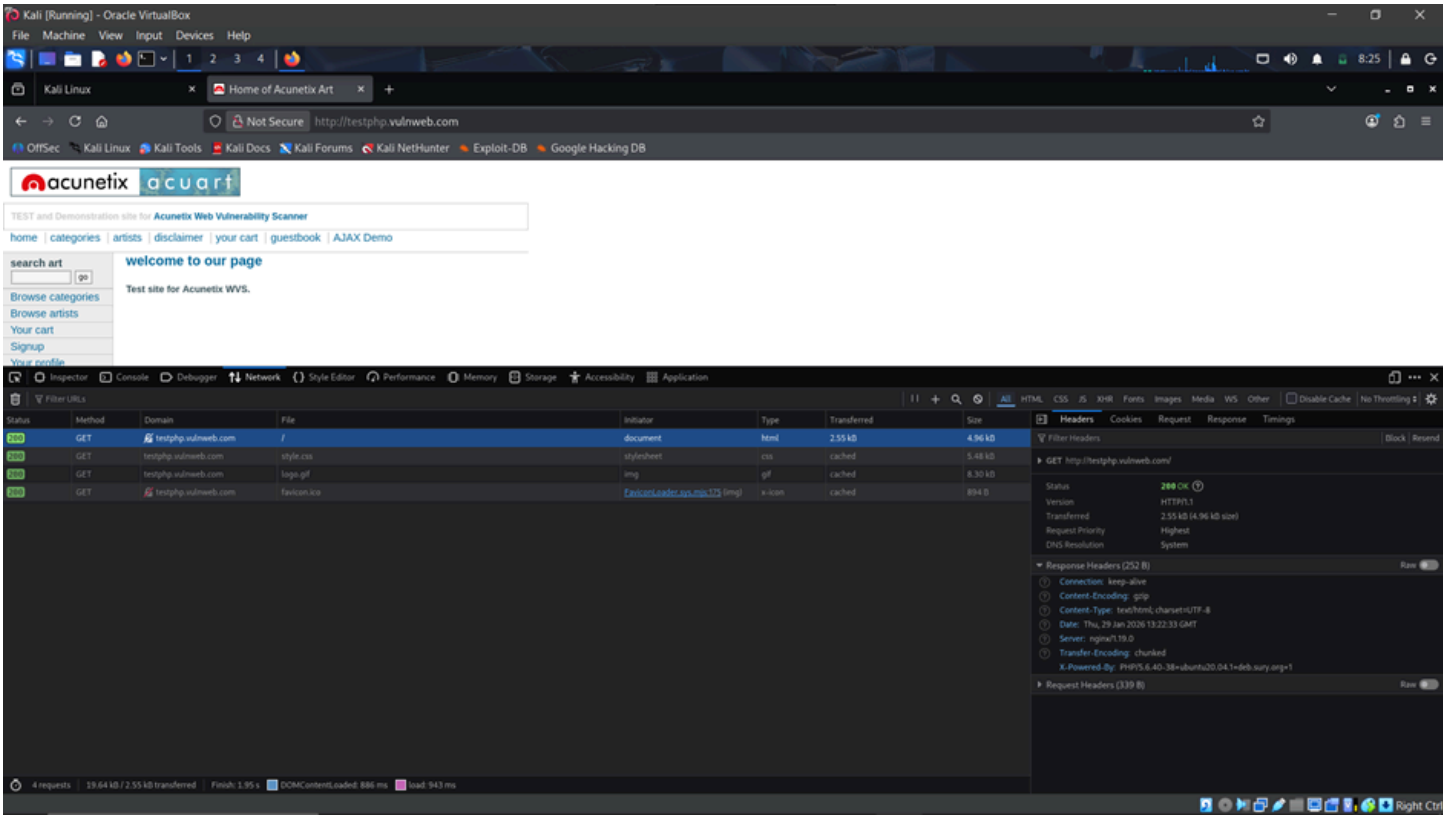
The website's HTTP responses do not include several **recommended security headers** that are designed to provide additional protection at the browser level. These headers help prevent common client-side attacks and enforce safer browser behavior.

Observation (Evidence):

Analysis of the HTTP response headers revealed that the following security headers were **not present**:

- Content-Security-Policy
- X-Frame-Options
- X-Content-Type-Options
- Strict-Transport-Security
- Missing Anti-Clickjacking Header
- Content Security Policy (CSP) Header Not Set

Evidence Reference:



Business Impact:

Missing security headers reduce the browser's ability to protect users against common web-based attacks. As a result:

- Users may be vulnerable to **clickjacking attacks**
- Malicious scripts may be executed more easily
- Browsers may incorrectly interpret content types, increasing exploitation risk

While this issue does not typically lead to immediate compromise on its own, it **weakens the overall security posture** of the application and increases the likelihood of successful attacks when combined with other vulnerabilities.

Recommendation:

It is recommended to:

- Implement industry-standard HTTP security headers
- Apply secure default configurations supported by the web server or framework
- Validate header implementation after deployment using security testing tools

Properly configured security headers provide an additional layer of defense and significantly improve client-side protection.

Finding 3: Exposure of HTTP Service on Port 80

Risk Level: 🟠Medium

Description:

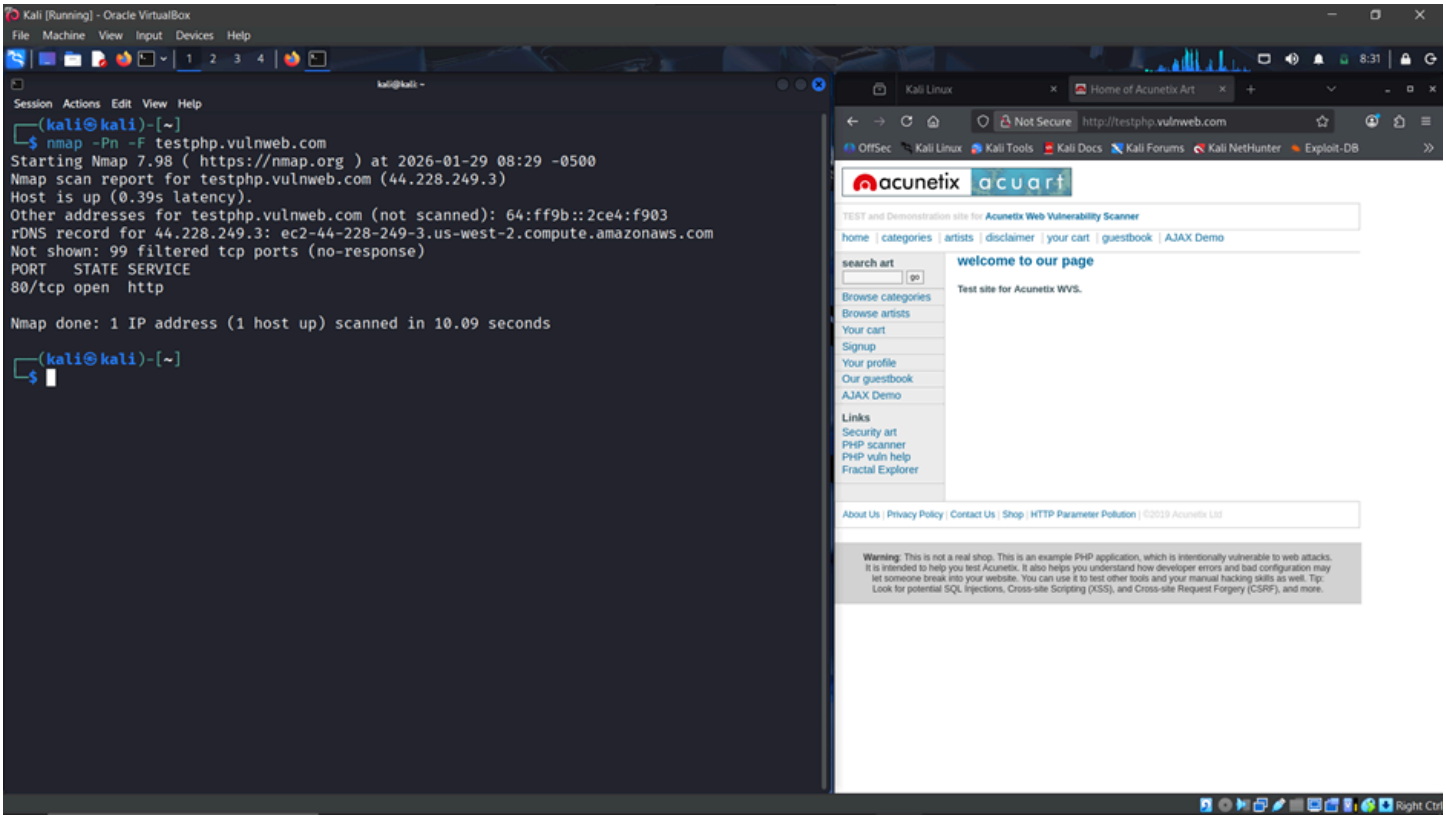
The application exposes an HTTP service over TCP port 80, which is publicly accessible.

Observation (Evidence)

A passive Nmap scan identified the following open service:

80/tcp open http

Evidence Reference:



Business Impact:

An exposed HTTP service:

- Transmits data without encryption
- Can be intercepted or manipulated by attackers
- Increases network-level exposure

This adds to the overall risk when combined with missing HTTPS enforcement.

Recommendation:

It is recommended to:

- Disable direct access to HTTP where possible
- Redirect all HTTP traffic to HTTPS
- Restrict unnecessary exposed services

Minimizing exposed services reduces attack surface.

Finding 4: Cross-Site Scripting (DOM-Based)

Risk Level: 🟠Medium

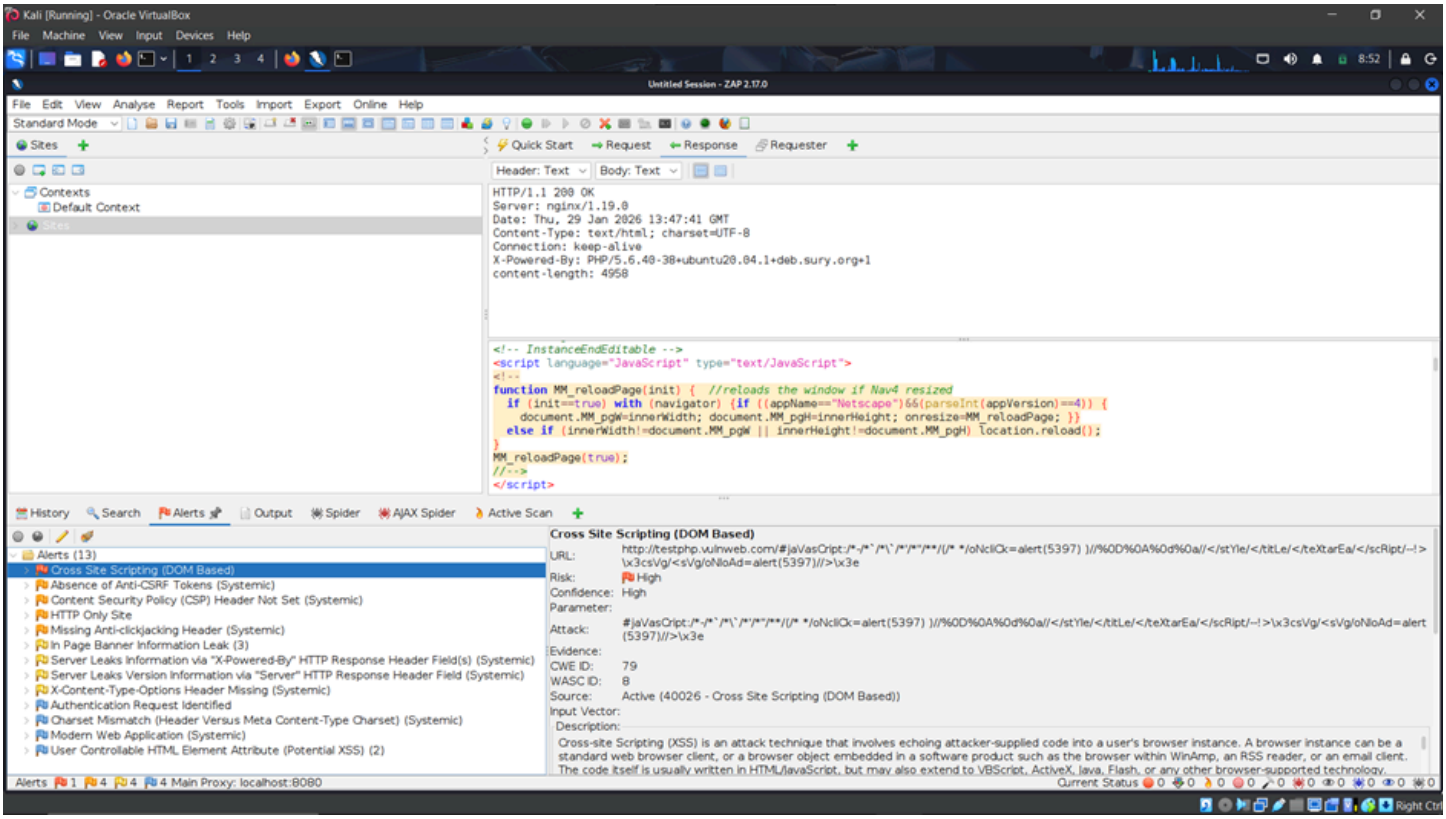
Description:

Passive scanning detected potential DOM-based Cross-Site Scripting (XSS) issues where user-controlled data may be dynamically processed by client-side scripts.

Observation (Evidence):

OWASP ZAP passive scan reported that certain inputs may be reflected in the DOM without proper sanitization.
No exploitation or active testing was performed.

Evidence Reference:



Business Impact:

If exploited, DOM-based XSS can:

- Steal user session information
- Modify page content
- Perform actions on behalf of users

This can negatively impact user trust and application integrity.

Recommendation:

It is recommended to:

- Validate and sanitize all user-controlled inputs
- Use secure JavaScript methods when handling dynamic content
- Implement a strict Content Security Policy (CSP)

Proper input handling reduces the risk of client-side attacks.

Finding 5: Insufficient Input Validation and Request Protection

Risk Level: 🟠Medium

Description:

The application does not adequately protect user-driven requests and client-side input handling. Passive analysis indicates the absence of Anti-CSRF protection and the presence of user-controllable HTML element attributes, which together increase the risk of unauthorized or unintended actions.

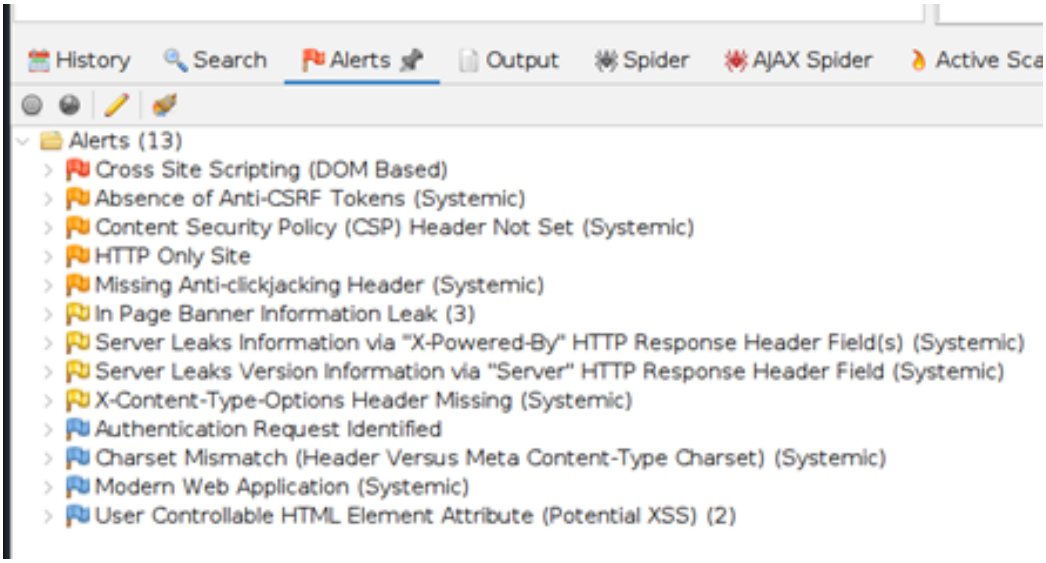
Observation (Evidence):

OWASP ZAP passive scan identified:

- Absence of Anti-CSRF tokens on state-changing requests
- User-controllable input being reflected in HTML element attributes without sufficient validation

No exploitation or active testing was performed.

Evidence Reference:



- Absence of Anti-CSRF Tokens
- User Controllable HTML Element Attribute

Business Impact:

Insufficient request and input protection can:

- Allow attackers to trick users into performing unintended actions
- Increase exposure to client-side manipulation

- Reduce user trust in application security

While no immediate breach was observed, these weaknesses lower the overall security maturity of the application.

Recommendation:

It is recommended to:

- Implement Anti-CSRF tokens for all state-changing requests
- Validate and sanitize all user-controlled inputs
- Avoid directly embedding user input into HTML attributes
- Use secure development frameworks with built-in protections

Strengthening input validation and request controls reduces the risk of unauthorized actions and improves application reliability.

Finding 6: Server and Technology Information Disclosure

Risk Level: 🟡Low to Medium

Description

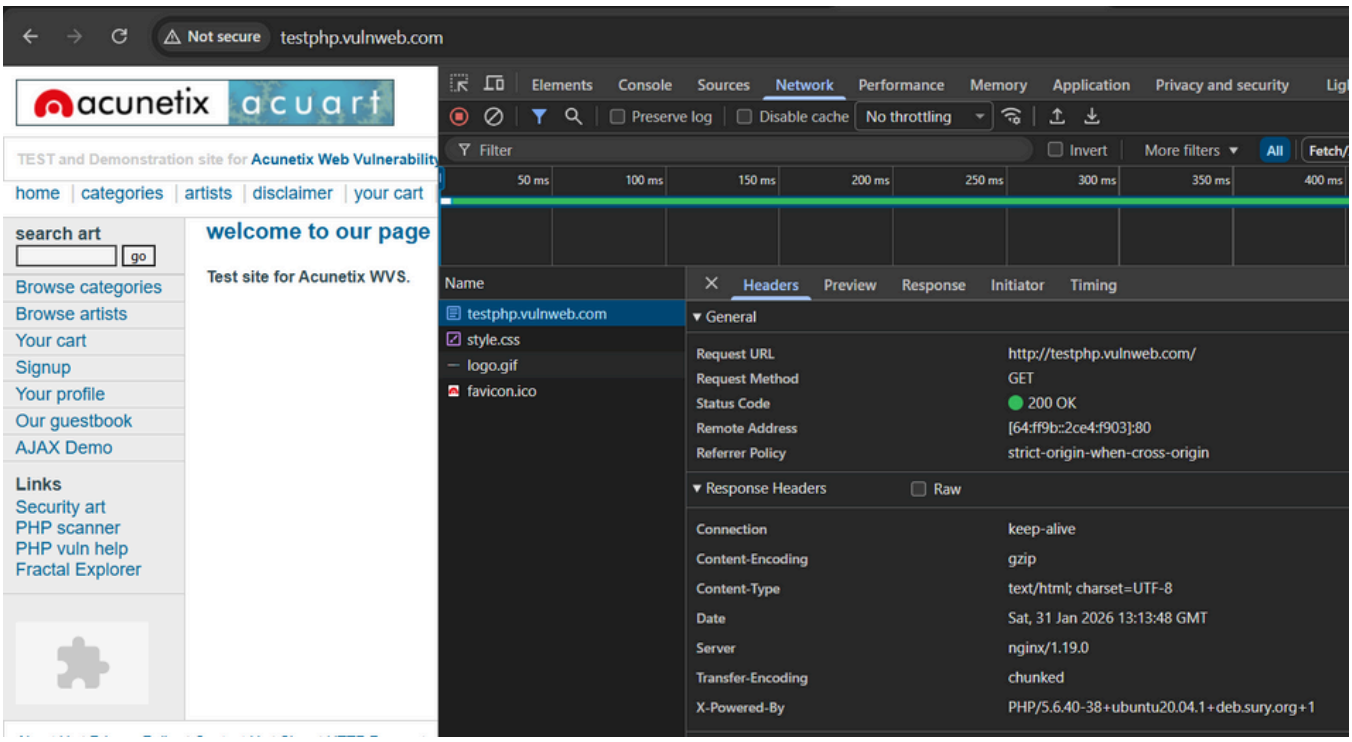
The web server discloses **internal technology and version information** through HTTP response headers. This includes details about the web server software and backend technologies in use.

Observation (Evidence):

During header analysis, the following information was observed to be exposed:

- Web server software and version (e.g., nginx/1.19.0)
- Backend technology details (e.g., PHP/5.6.40)

Evidence Reference:



Business Impact:

While this issue does not directly result in a security breach, exposing internal technology details can:

- Help attackers identify **outdated or vulnerable components**
- Enable targeted attacks based on known vulnerabilities
- Increase the efficiency of reconnaissance activities

Such information disclosure lowers the effort required for an attacker to plan more effective attacks against the application.

Recommendation:

It is recommended to:

- Remove or obfuscate the **Server** and **X-Powered-By** HTTP headers
- Configure the web server to limit the exposure of internal technology details
- Regularly update server and application components to supported versions

Reducing information disclosure improves the overall security posture and limits attacker reconnaissance.

Summary of Findings Table

Sl.No	Finding	Risk
1	Lack of HTTPS Encryption	🔴 High
2	Missing Security Headers	🟠 Medium
3	HTTP Service Exposure (Port 80)	🟠 Medium
4	DOM-Based XSS	🟠 Medium
5	Insufficient Input Validation & Request Protection	🟠 Medium
6	Information Disclosure via Headers	🟡 Low–Medium

Overall Recommendations

Strategic Recommendations

- Enforce HTTPS across all services
- Implement standard security headers
- Reduce information disclosure
- Strengthen input validation and request protections
- Regularly review configurations and dependencies

Business Value

Implementing these recommendations will:

- Improve user trust
- Reduce attack surface
- Align with industry security best practices
- Lower long-term security risk

Limitations & Disclaimer

Assessment Limitations

- Passive testing only
- No exploitation conducted
- Results limited to publicly visible components

Disclaimer

This assessment reflects the security posture at the time of testing. Future changes to the application may alter risk exposure.

Conclusion

The assessment identified several security weaknesses that, if addressed, will significantly improve the application’s security posture. While no immediate critical exploitation was observed, proactive remediation is strongly recommended.

This report demonstrates a preventive security approach, focused on risk reduction rather than exploitation.