

# **VULNERABILITY ASSESSMENT REPORT**

**READ-ONLY SECURITY REVIEW OF A PUBLIC  
WEB APPLICATION**

**PREPARED BY: DHANUSH KUMAR**

**INTERNSHIP PROGRAM: FUTURE INTERNS - CYBER SECURITY (2026)**

**ASSESSMENT DATE: JANUARY 2026**

**TARGET WEBSITE: TESTPHP.VULNWEB.COM**

# Executive Summary

*This Vulnerability Assessment Report documents the results of a read-only security review conducted on a publicly accessible web application. The purpose of this assessment was to identify common security weaknesses and configuration gaps that could expose the application to potential cyber risks.*

*The assessment was performed using passive analysis techniques only. No exploitation, authentication bypass, data modification, or denial-of-service activities were conducted. All findings are based on publicly observable information such as HTTP response headers, network exposure, and visible application behavior.*

*The review identified multiple low to medium risk security issues, primarily related to missing HTTP security headers, insecure transport configuration, and server information disclosure. While no critical vulnerabilities were observed, addressing the identified issues would significantly strengthen the website's security posture, reduce attack surface, and improve user trust.*

# Scope of Assessment

## In Scope

- *Public-facing web pages*
- *Passive traffic observation*
- *HTTP header analysis*
- *Network service exposure review*

## Out of Scope

- *Login or authentication testing*
- *Exploitation of vulnerabilities*
- *Brute-force attacks*
- *Denial-of-Service (DoS) testing*
- *Any activity that could disrupt the target website*

# METHODOLOGY

THE ASSESSMENT FOLLOWED A STRUCTURED, ETHICAL APPROACH ALIGNED WITH INDUSTRY-STANDARD SECURITY AUDITING PRACTICES:

## **RECONNAISSANCE (PASSIVE)**

*IDENTIFICATION OF EXPOSED SERVICES AND BASIC NETWORK CONFIGURATION USING NON-INTRUSIVE SCANNING.*

## **CONFIGURATION ANALYSIS**

*REVIEW OF HTTP RESPONSE HEADERS AND TRANSPORT SECURITY SETTINGS.*

## **CLIENT-SIDE INSPECTION**

*ANALYSIS OF BROWSER-VISIBLE SECURITY CONTROLS USING DEVELOPER TOOLS.*

## **RISK CLASSIFICATION**

*EACH FINDING WAS CLASSIFIED AS LOW, MEDIUM, OR HIGH BASED ON POTENTIAL BUSINESS IMPACT AND LIKELIHOOD.*

## **REPORTING**

*FINDINGS WERE DOCUMENTED WITH CLEAR EXPLANATIONS AND PRACTICAL REMEDIATION STEPS SUITABLE FOR NON-TECHNICAL STAKEHOLDERS.*

# TOOLS USED

- *NMAP: BASIC PORT AND SERVICE EXPOSURE ANALYSIS*
- *BROWSER DEVELOPER TOOLS: INSPECTION OF HTTP HEADERS AND CLIENT-SIDE BEHAVIOR*
- *MOZILLA HTTP OBSERVATORY: SECURITY HEADER AND CONFIGURATION ANALYSIS*
- *CANVA: PROFESSIONAL REPORT DESIGN AND PRESENTATION*

# FINDINGS AND RISK ANALYSIS

## ***Finding 1: Missing Content Security Policy (CSP)***

**Risk Level: Medium**

### **Description:**

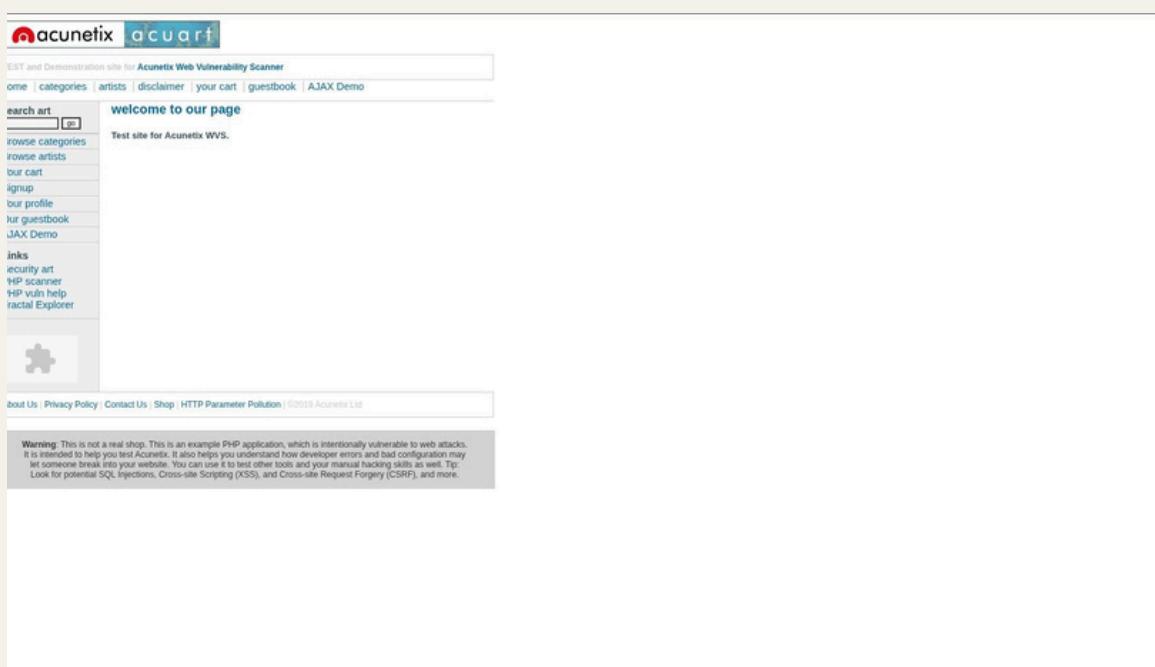
*The application does not implement a Content Security Policy (CSP). CSP is a security mechanism that helps prevent cross-site scripting (XSS) and other code injection attacks by defining which sources of content are trusted.*

### **Business Impact:**

*Without CSP, an attacker may be able to inject malicious scripts that execute in users' browsers, potentially leading to session hijacking, data theft, or defacement.*

### **Recommendation:**

*Implement a strong Content Security Policy header that restricts script, style, and resource loading to trusted sources only.*



## **Finding 2: Missing X-Frame-Options Header**

### **Risk Level: Medium**

#### **Description:**

*The application does not include the X-Frame-Options header, which protects against clickjacking attacks by preventing the site from being embedded in malicious frames.*

#### **Business Impact:**

*Attackers could trick users into clicking on hidden elements, potentially leading to unauthorized actions or data exposure.*

#### **Recommendation:**

*Configure the server to include the X-Frame-Options header (DENY or SAMEORIGIN) or define equivalent protections using CSP frame-ancestors.*

```
(kali㉿kali)-[~]
$ nmap -sV -Pn testphp.vulnweb.com

Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-28 12:23 +0530
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.28s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http    nginx 1.19.0

Service detection performed. Please report any incorrect results at https://nma
Nmap done: 1 IP address (1 host up) scanned in 41.42 seconds
```

# Finding 3: Missing X-Content-Type-Options Header

## Risk Level: Low

### Description:

The application does not set the X-Content-Type-Options header, allowing browsers to perform MIME-type sniffing.

### Business Impact:

This could allow attackers to disguise malicious content as safe file types, increasing the risk of script execution.

### Recommendation:

Add the header X-Content-Type-Options: nosniff to all server responses.

The screenshot displays a browser window with two main panes. On the left is a web application interface for 'Acunetix acuart'. It features a sidebar with links like 'search art', 'Browse categories', 'Your cart', 'Signup', 'Your profile', 'Our guestbook', 'AJAX Demo', 'Links', 'Security art', 'PHP scanner', 'PHP vuln help', and 'Fractal Explorer'. The main content area says 'welcome to our page' and 'Test site for Acunetix WVS.' Below this is a 'Warning' message about the application being intentionally vulnerable. On the right is a screenshot of the Chrome developer tools Network tab. It shows a list of requests for files like 'logo.gif' and 'favicon.ico'. The 'Response' column for the 'logo.gif' request shows the following headers:

Name	Value
Content-Type	image/gif
Content-Encoding	gzip
Date	Wed, 28 Jan 2026 07:04:34 GMT
Server	nginx/1.10.0
Transfer-Encoding	chunked
X-Powered-By	PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1

The 'Request' column shows the following headers:

Name	Value
Accept	text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
Accept-Encoding	gzip, deflate
Accept-Language	en-US,en;q=0.5

## **Finding 4: Insecure Transport Configuration (No HTTPS Enforcement / HSTS)**

**Risk Level: Medium**

### **Description:**

*The application does not enforce HTTPS connections or implement HTTP Strict Transport Security (HSTS).*

### **Business Impact:**

*Users may be vulnerable to man-in-the-middle attacks, where attackers intercept or manipulate traffic between the browser and server.*

### **Recommendation:**

*Enforce HTTPS across the application and enable HSTS to ensure secure connections at all times.*

MDN

HTML ▾ CSS ▾ JavaScript ▾ Web APIs ▾ All ▾ Learn ▾ Tools ▾ About ▾ Blog

HTTP Observatory > Report

Theme English (US)

HTTP Observatory Report

Scan summary: testphp.vulnweb.com

Score: 10 / 100

Scan Time: Just now

Tests Passed: 5 / 10

Wait a minute to rescan

Scan another website

Report Feedback

HTTP Observatory

Tests & Scoring

FAQ

Scan results

Scoring	CSP analysis	Cookies	Raw server headers	Scan history	Benchmark comparison
Test	Score	Reason	Recommendation		
Content Security Policy (CSP)	-25 ✗	Content Security Policy (CSP) header not implemented	Implement one, see MDN's Content Security Policy (CSP) documentation.		
Cookies	-	No cookies detected	None		
Cross-Origin Resource Sharing (CORS)	0 ✓	Content is not visible via cross-origin resource sharing (CORS) files or headers.	None		
Redirection	-20 ✗	Does not redirect to an HTTPS site.	Redirect to the same host on HTTPS first, then redirect to the final host on HTTPS.		

## Finding 5: Server Information Disclosure

### Risk Level: Low

#### Description:

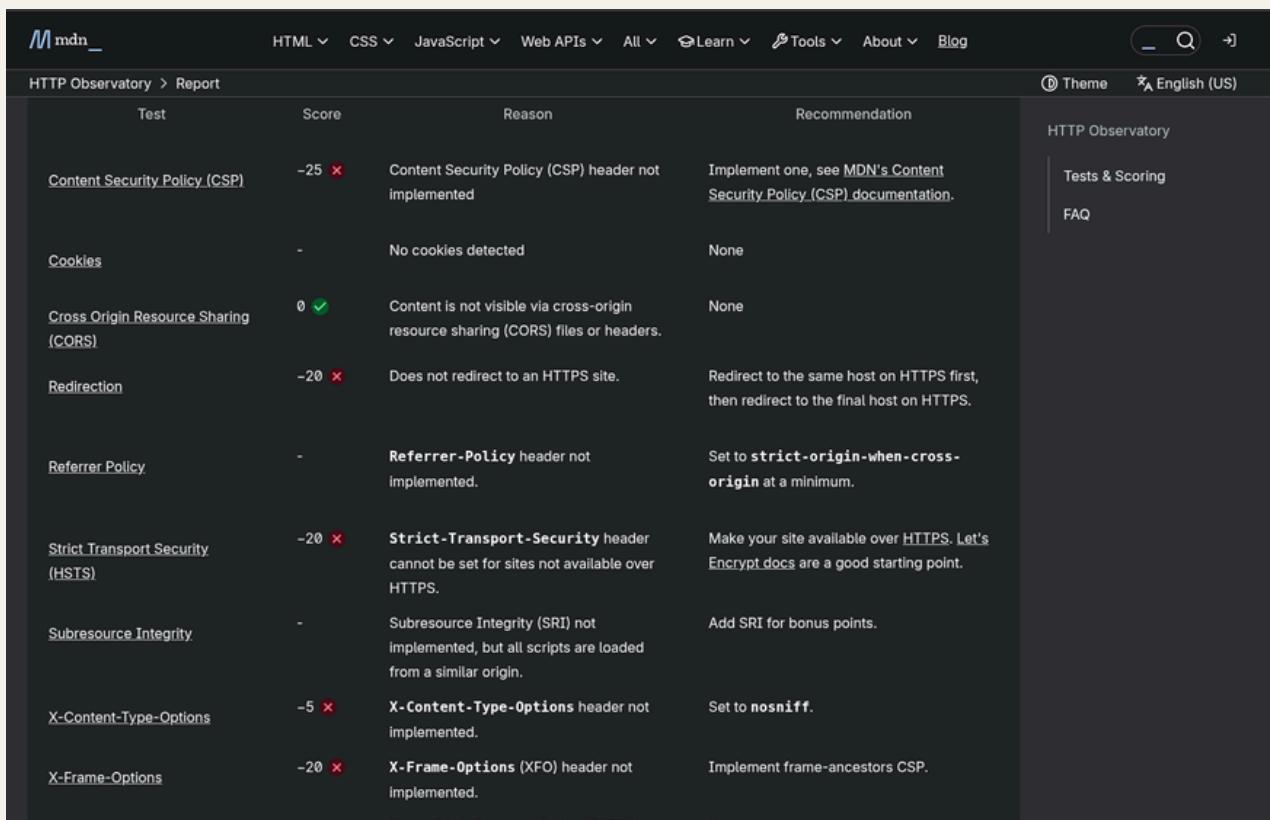
Server headers reveal information about the underlying technology stack.

#### Business Impact:

Exposed server details can assist attackers during reconnaissance and increase the likelihood of targeted attacks.

#### Recommendation:

Configure the web server to suppress or generalize server version information in HTTP responses.



The screenshot shows a dark-themed browser window displaying the MDN Web Docs HTTP Observatory report. The top navigation bar includes links for HTML, CSS, JavaScript, Web APIs, All, Learn, Tools, About, and Blog. A search bar and a theme switcher are also present. The main content area is titled "HTTP Observatory > Report". It contains a table with the following data:

Test	Score	Reason	Recommendation
Content Security Policy (CSP)	-25 ✗	Content Security Policy (CSP) header not implemented	Implement one, see MDN's Content Security Policy (CSP) documentation.
Cookies	-	No cookies detected	None
Cross Origin Resource Sharing (CORS)	0 ✓	Content is not visible via cross-origin resource sharing (CORS) files or headers.	None
Redirection	-20 ✗	Does not redirect to an HTTPS site.	Redirect to the same host on HTTPS first, then redirect to the final host on HTTPS.
Referrer Policy	-	Referrer-Policy header not implemented.	Set to strict-origin-when-cross-origin at a minimum.
Strict Transport Security (HSTS)	-20 ✗	Strict-Transport-Security header cannot be set for sites not available over HTTPS.	Make your site available over HTTPS. Let's Encrypt docs are a good starting point.
Subresource Integrity	-	Subresource Integrity (SRI) not implemented, but all scripts are loaded from a similar origin.	Add SRI for bonus points.
X-Content-Type-Options	-5 ✗	X-Content-Type-Options header not implemented.	Set to nosniff.
X-Frame-Options	-20 ✗	X-Frame-Options (XFO) header not implemented.	Implement frame-ancestors CSP.

A sidebar on the right provides links to "HTTP Observatory", "Tests & Scoring", and "FAQ".

# **Risk Summary Table**

Finding	Risk Level	Potential Impact
Missing Content Security Policy	Medium	XSS, script injection
Missing X-Frame-Options	Medium	Clickjacking
Missing X-Content-Type-Options	Low	MIME sniffing
No HTTPS / HSTS	Medium	Man-in-the-middle attacks, data interception
Server Information Disclosure	Low	Reconnaissance and targeted attacks

# Conclusion

*The vulnerability assessment identified several configuration-level security weaknesses that are commonly found in public web applications. While no critical vulnerabilities were discovered, the identified issues could be exploited if left unaddressed.*

*By implementing the recommended remediation steps, the organization can significantly reduce its attack surface, enhance user trust, and align the application with modern web security best practices.*

## Disclaimer

*This assessment was conducted for educational and training purposes as part of the Future Interns Cyber Security Internship Program. All testing was limited to passive, read-only techniques on a publicly available demo website. No unauthorized access or harmful activity was performed.*