Vulnerability Assessment Report

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

Prepared by: Veera Venkata Vinay Yamala

Date: 13 Feb 2026

1. Introduction

This report presents the results of a basic vulnerability assessment performed on a public test website. The purpose of this assessment is to identify common security issues using only passive and ethical methods. No exploitation or harmful actions were performed during this assessment.

2. Scope & Ethics

Scope of Testing:

- Only public-facing pages were tested

- Only passive scanning and configuration checks were performed

- No login bypass, exploitation, or attack was attempted

This assessment follows ethical guidelines and is intended only for learning and security improvement purposes.

3. Tools Used

- Nmap – For basic port and service scanning

- Browser Developer Tools – For checking HTTP headers

- OWASP ZAP (Passive Scan) – For identifying security misconfigurations

- Google Docs / Word – For report writing

4. Target Information

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

Type: Public test website

Testing Type: Read-only / Passive analysis

5. Summary of Findings

- Website uses HTTP instead of HTTPS – Medium Risk

- Outdated PHP Version – High Risk

- Content Security Policy Header Not Set – Medium Risk

- Missing Anti-Clickjacking Header – Medium Risk

- Absence of Anti-CSRF Tokens – Medium Risk

- Server Leaks Information via Headers – Low Risk

6. Detailed Findings

Finding 1: Website Uses HTTP (Not HTTPS)

Risk Level: Medium

What is it?

The website is accessible using HTTP, which does not encrypt data between the user and the server.

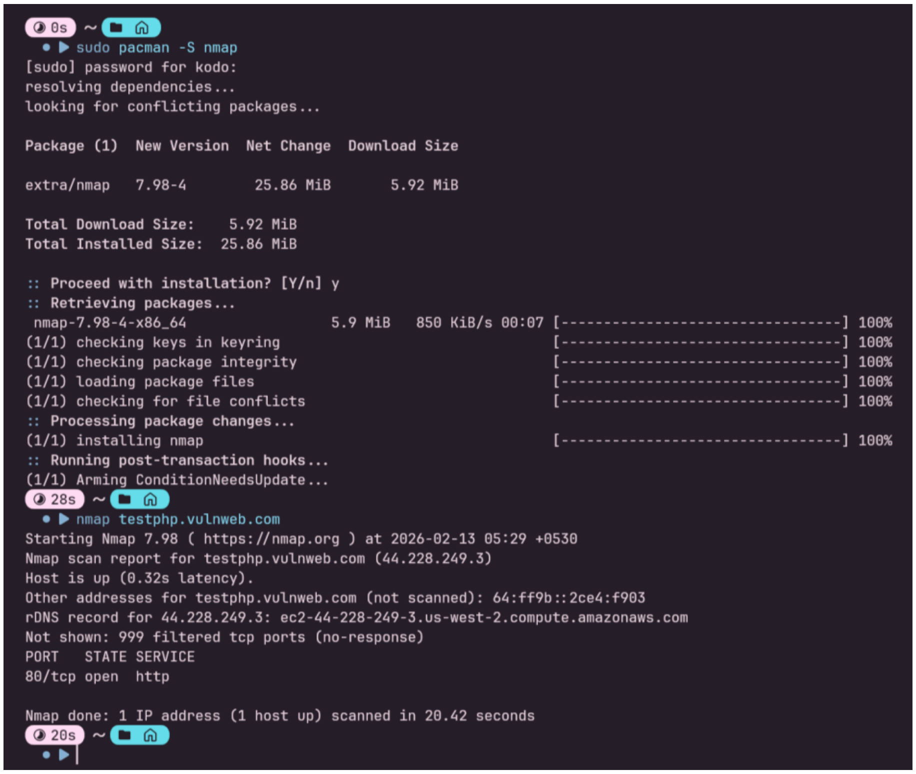
Why does it matter?

Without encryption, attackers on the network can read or modify the data being sent.

How to fix it?

Enable HTTPS using an SSL/TLS certificate and redirect all HTTP traffic to HTTPS.

Evidence:



```
◑ 0s ~ ■ ⌂
  ● ▶ sudo pacman -S nmap
[sudo] password for kodo:
resolving dependencies...
looking for conflicting packages...

Package (1)  New Version  Net Change  Download Size

extra/nmap   7.98-4        25.86 MiB       5.92 MiB

Total Download Size:    5.92 MiB
Total Installed Size:  25.86 MiB

:: Proceed with installation? [Y/n] y
:: Retrieving packages...
 nmap-7.98-4-x86_64                5.9 MiB   850 KiB/s 00:07 [----------------------------------] 100%
(1/1) checking keys in keyring                               [----------------------------------] 100%
(1/1) checking package integrity                             [----------------------------------] 100%
(1/1) loading package files                                  [----------------------------------] 100%
(1/1) checking for file conflicts                            [----------------------------------] 100%
:: Processing package changes...
(1/1) installing nmap                                        [----------------------------------] 100%
:: Running post-transaction hooks...
(1/1) Arming ConditionNeedsUpdate...
◑ 28s ~ ■ ⌂
  ● ▶ nmap testphp.vulnweb.com
Starting Nmap 7.98 ( https://nmap.org ) at 2026-02-13 05:29 +0530
Nmap scan report for testphp.vulnweb.com (44.228.249.3)
Host is up (0.32s latency).
Other addresses for testphp.vulnweb.com (not scanned): 64:ff9b::2ce4:f903
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
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 20.42 seconds
◑ 20s ~ ■ ⌂
  ● ▶
```

Finding 2: Outdated PHP Version

Risk Level: High

What is it?

The website is using an old PHP version (5.6.40), which is no longer supported.

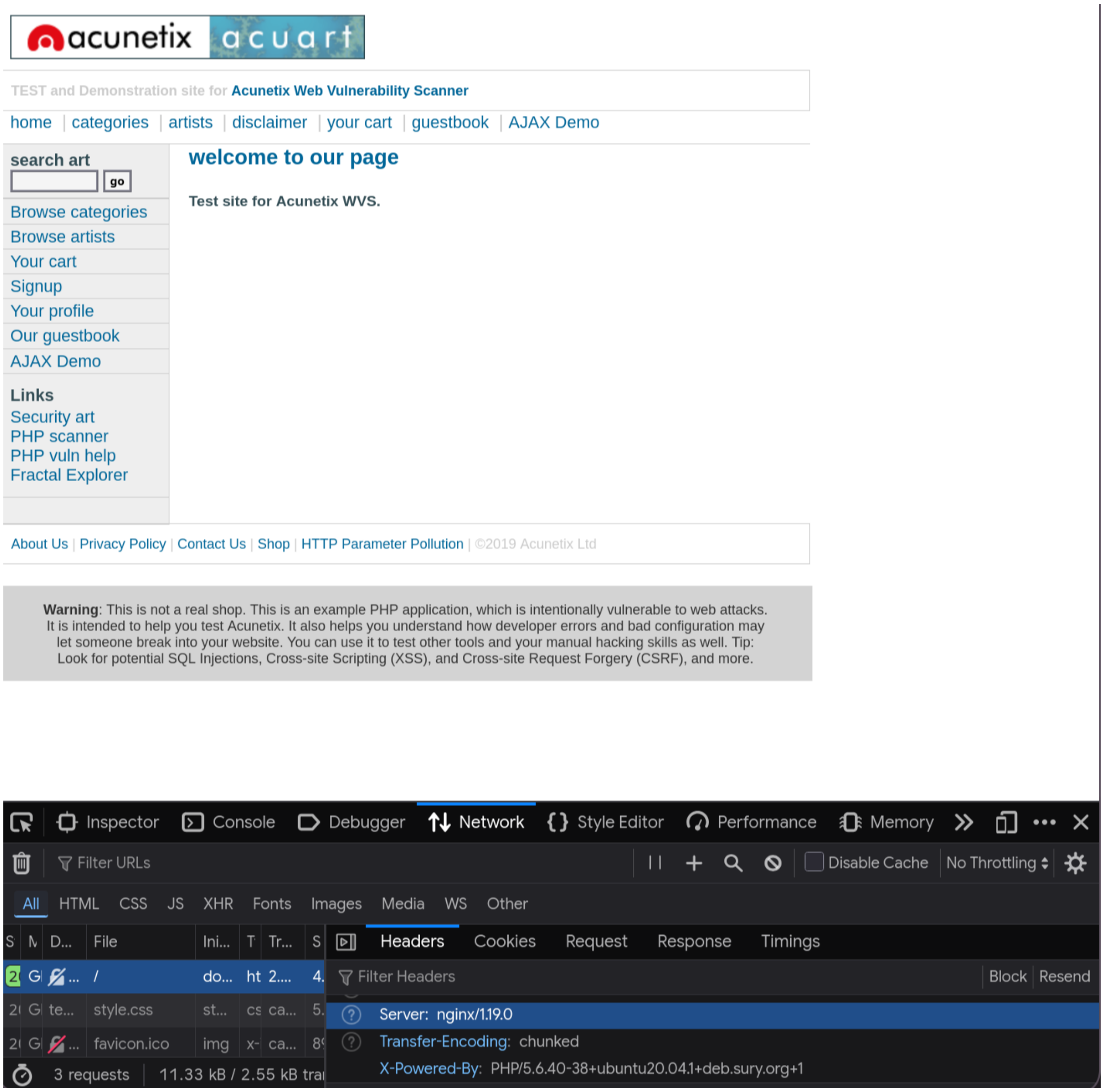
Why does it matter?

Old software versions may contain known security vulnerabilities.

How to fix it?

Upgrade PHP to a supported and secure version and keep it updated regularly.

Evidence:



TEST and Demonstration site for Acunetix Web Vulnerability Scanner

welcome to our page

Test site for Acunetix WVS.

©2019 Acunetix Ltd

Warning: This is not a real shop. This is an example PHP application, which is intentionally vulnerable to web attacks. It is intended to help you test Acunetix. It also helps you understand how developer errors and bad configuration may let someone break into your website. You can use it to test other tools and your manual hacking skills as well. Tip: Look for potential SQL Injections, Cross-site Scripting (XSS), and Cross-site Request Forgery (CSRF), and more.

```
S  M  D...  File         Ini...  T   Tr...  S
2( G  ...   /            do...   ht  2....  4.
2( G  te... style.css    st...   cs  ca...  5.
2( G  ...   favicon.ico  img     x-  ca...  8!
3 requests | 11.33 kB / 2.55 kB trai
```

```
Server: nginx/1.19.0
Transfer-Encoding: chunked
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
```

Finding 3: Content Security Policy Header Not Set

Risk Level: Medium

What is it?

The website does not define a Content Security Policy to control what content can run in the browser.

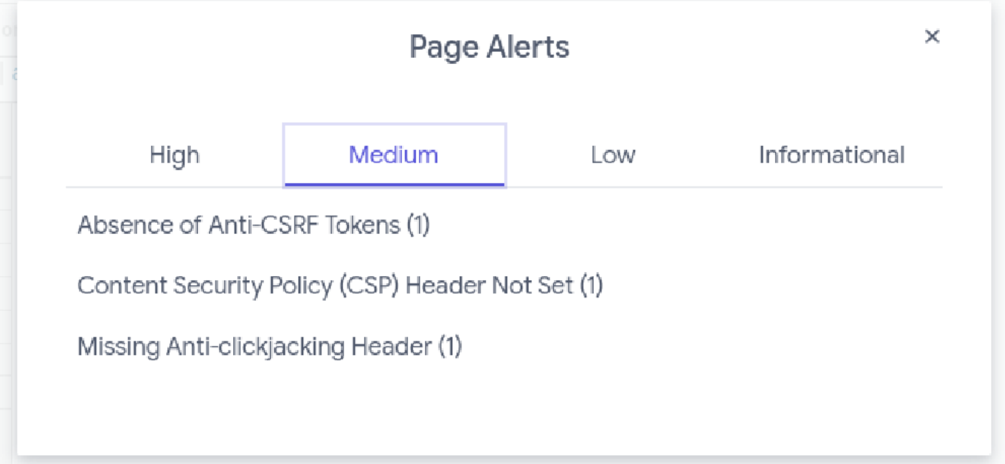
Why does it matter?

This increases the risk of content injection and cross-site scripting attacks.

How to fix it?

Configure the web server to add a proper Content-Security-Policy header.

Evidence:



Absence of Anti-CSRF Tokens (1)

Content Security Policy (CSP) Header Not Set (1)

Missing Anti-clickjacking Header (1)

Finding 4: Missing Anti-Clickjacking Header

Risk Level: Medium

What is it?

The website does not use headers to prevent being embedded inside other websites.

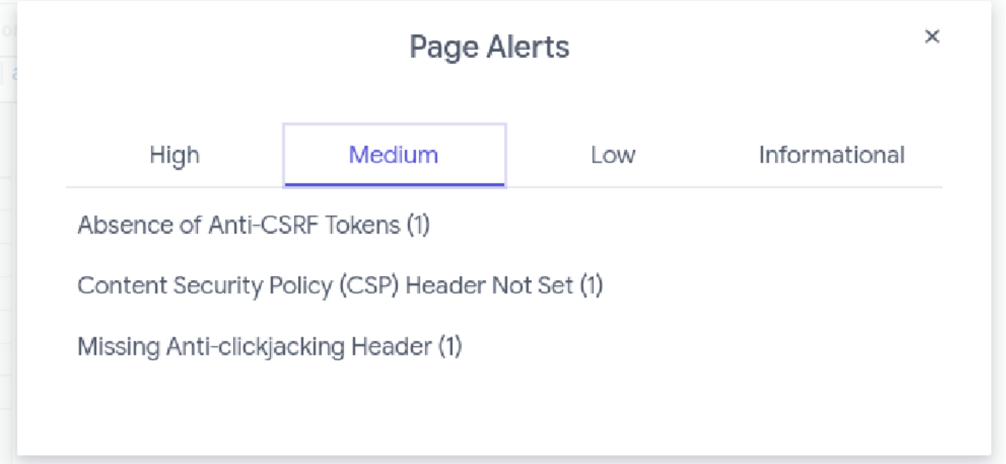
Why does it matter?

Attackers can trick users by placing the website inside a fake page (clickjacking).

How to fix it?

Add the X-Frame-Options header or frame protection settings.

Evidence:


Page Alerts

Absence of Anti-CSRF Tokens (1)

Content Security Policy (CSP) Header Not Set (1)

Missing Anti-clickjacking Header (1)

Finding 5: Absence of Anti-CSRF Tokens

Risk Level: Medium

What is it?

The website does not use CSRF tokens to protect forms from unauthorized requests.

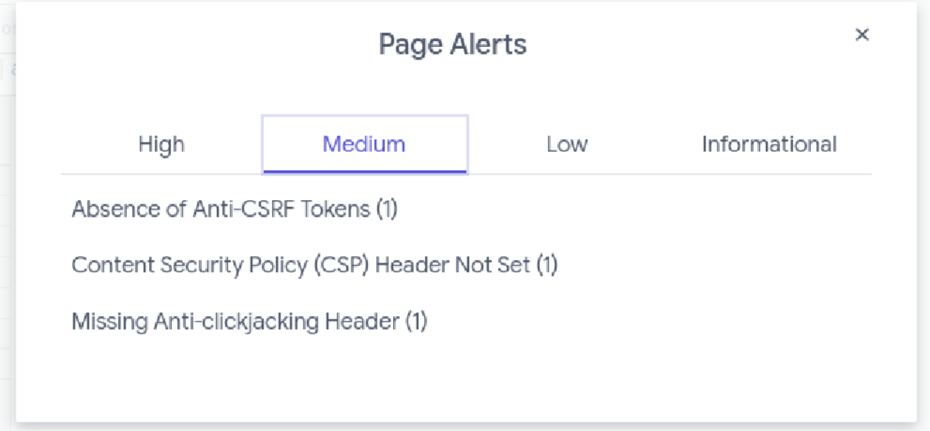
Why does it matter?

Attackers can trick users into performing actions without their knowledge.

How to fix it?

Implement CSRF tokens in all sensitive forms and requests.

Evidence:


Page Alerts

Absence of Anti-CSRF Tokens (1)

Content Security Policy (CSP) Header Not Set (1)

Missing Anti-clickjacking Header (1)

Finding 6: Server Leaks Information via Headers

Risk Level: Low

What is it?

The server reveals technology and version information in HTTP headers.

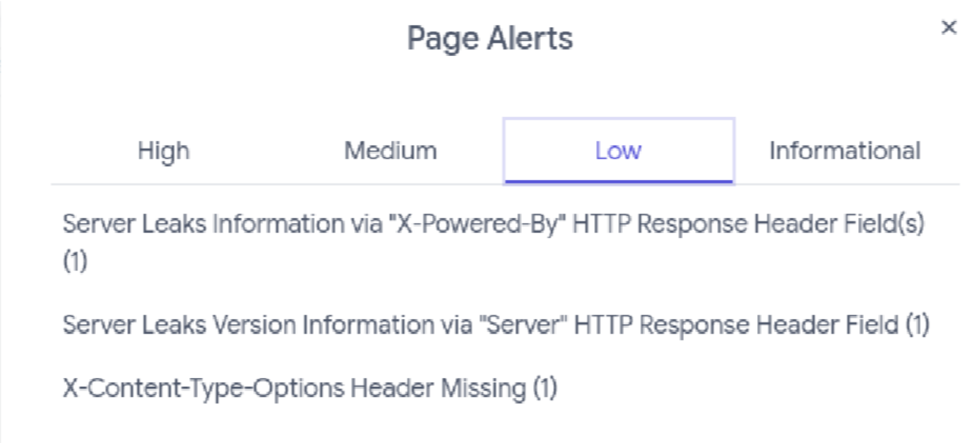
Why does it matter?

Attackers can use this information to target known vulnerabilities.

How to fix it?

Hide or minimize server and technology information in HTTP headers.

Evidence:



Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s) (1)

Server Leaks Version Information via "Server" HTTP Response Header Field (1)

X-Content-Type-Options Header Missing (1)

7. Conclusion & Recommendations

The assessment identified several security misconfigurations and outdated components. It is recommended to enable HTTPS, update outdated software, and apply proper security headers to improve the overall security posture of the website.