VULNERABILITY ASSESSMENT REPORT

Task 1 – Web Application Security Assessment

Target Application:
http://testphp.vulnweb.com
Assessment Type:
Passive Web Application Vulnerability Assessment

Project ID:
FI-CS-2026-T1
Prepared By:
Ankit
Cyber Security Internship – Future Interns
Assessment Date:
14 February 2026
Version:
1.0

Table of Contents

Executive Summary

This report presents the results of a passive vulnerability assessment conducted against the publicly accessible web application http://testphp.vulnweb.com.

The objective of this assessment was to identify security misconfigurations and potential weaknesses using industry-standard tools including Nmap and OWASP ZAP in passive mode.

The assessment identified multiple medium and low-severity security issues primarily related to missing HTTP security headers and information disclosure.

No active exploitation was performed as per internship scope.

Overall Security Posture: Moderate Risk

Immediate implementation of recommended security hardening controls is advised before production deployment.

Disclaimer

This vulnerability assessment was conducted strictly for educational purposes as part of the Cyber Security Internship Program.

The assessment was performed in passive mode only. No active exploitation, brute-force attempts, SQL injection attacks, or intrusive techniques were conducted during testing.

The target application (http://testphp.vulnweb.com) is a publicly accessible intentionally vulnerable test environment used for security learning and research.

This report represents a snapshot of the security posture of the application at the time of assessment. Security conditions may change over time due to configuration updates or system modifications.

The findings and recommendations provided in this report are intended to improve security awareness and hardening practices.

Assessment Overview

This vulnerability assessment was conducted to evaluate the security posture of the publicly accessible web application http://testphp.vulnweb.com.

The objective of this assessment was to identify security misconfigurations and potential weaknesses using industry-standard security tools in passive mode.

The assessment was performed following structured testing principles aligned with:

- OWASP Testing Guide v4
- NIST SP 800-115 Technical Guide to Information Security Testing

Testing Activities Included:

1. Reconnaissance and Service Enumeration
2. Passive Web Application Analysis
3. HTTP Response Header Inspection
4. Vulnerability Identification and Documentation

No active exploitation techniques were performed during this assessment as per the defined internship scope.

Scope of assessment

The scope of this assessment was limited to the publicly accessible web application:

Target:
http://testphp.vulnweb.com

Type of Testing:
Passive Web Application Vulnerability Assessment

Included in Scope:

- Service enumeration using Nmap
- Passive vulnerability scanning using OWASP ZAP
- HTTP response header analysis
- Manual web application browsing and inspection

Excluded from Scope:

- Active exploitation of identified vulnerabilities
- SQL injection exploitation attempts
- Brute-force attacks
- Denial-of-Service (DoS) testing
- Any intrusive or destructive testing methods

The assessment was conducted strictly in passive mode in accordance with the internship requirements.

Testing methodology

The vulnerability assessment was conducted using a structured and systematic approach to ensure accurate identification of potential security weaknesses.

1. Reconnaissance and Service Enumeration

Network-level reconnaissance was performed using Nmap to identify open ports and running services on the target host.

Command Used:
nmap -sV testphp.vulnweb.com

The scan identified port 80 (HTTP) as open and detected the web server as **nginx version 1.19.0.**

2. Web Application Exploration

The target application was manually browsed using a web browser to generate HTTP traffic. This allowed observation of application behavior without performing any intrusive actions.

3. Passive Vulnerability Scanning

OWASP ZAP was configured as a local proxy (127.0.0.1:8080) to intercept and analyze HTTP requests and responses.

The passive scan module was used to identify:

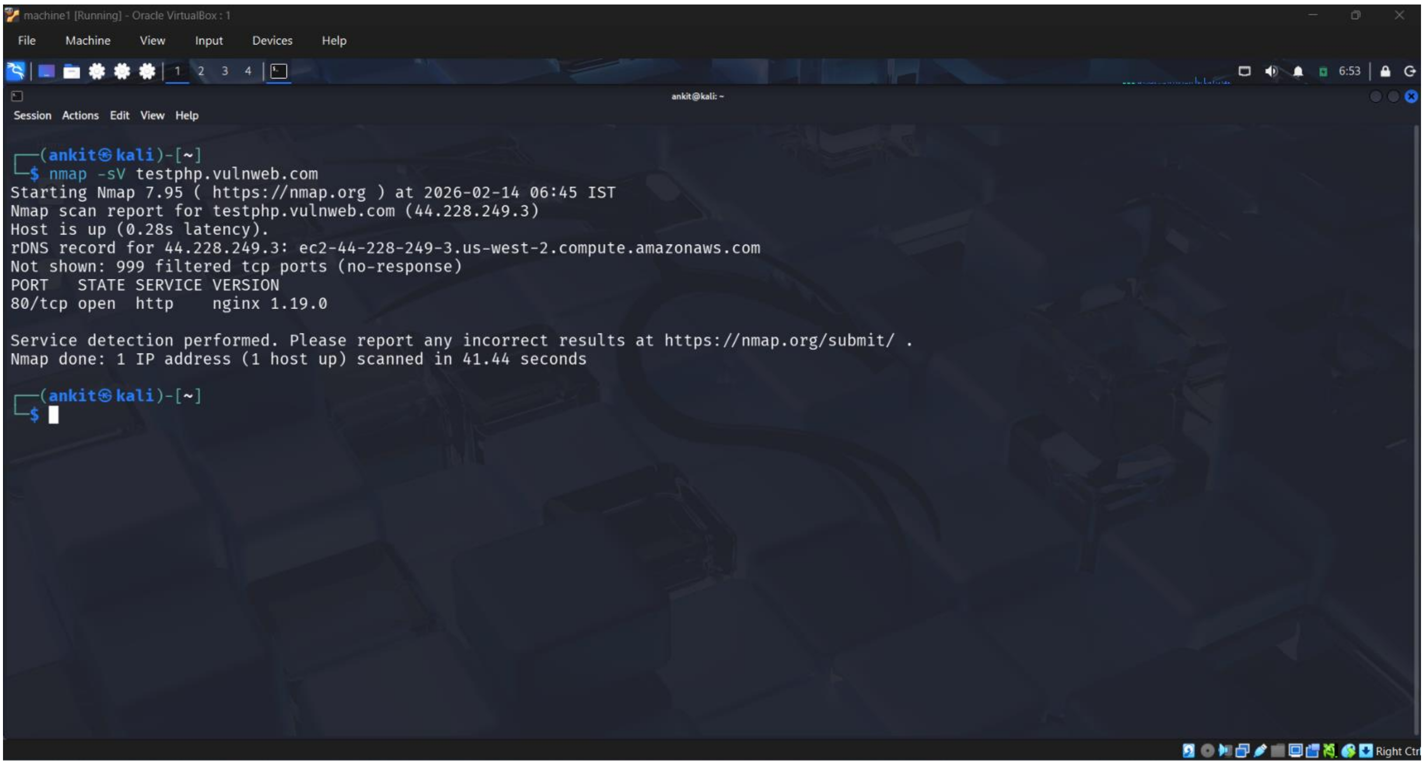
- Missing security headers
- Information disclosure issues
- Configuration weaknesses
- Potential client-side security risks

4. HTTP Header Analysis

HTTP response headers were manually inspected using browser developer tools to verify the presence or absence of important security controls such as:

- Content-Security-Policy (CSP)
- X-Frame-Options
- X-Content-Type-Options
- X-Powered-By
- Server header

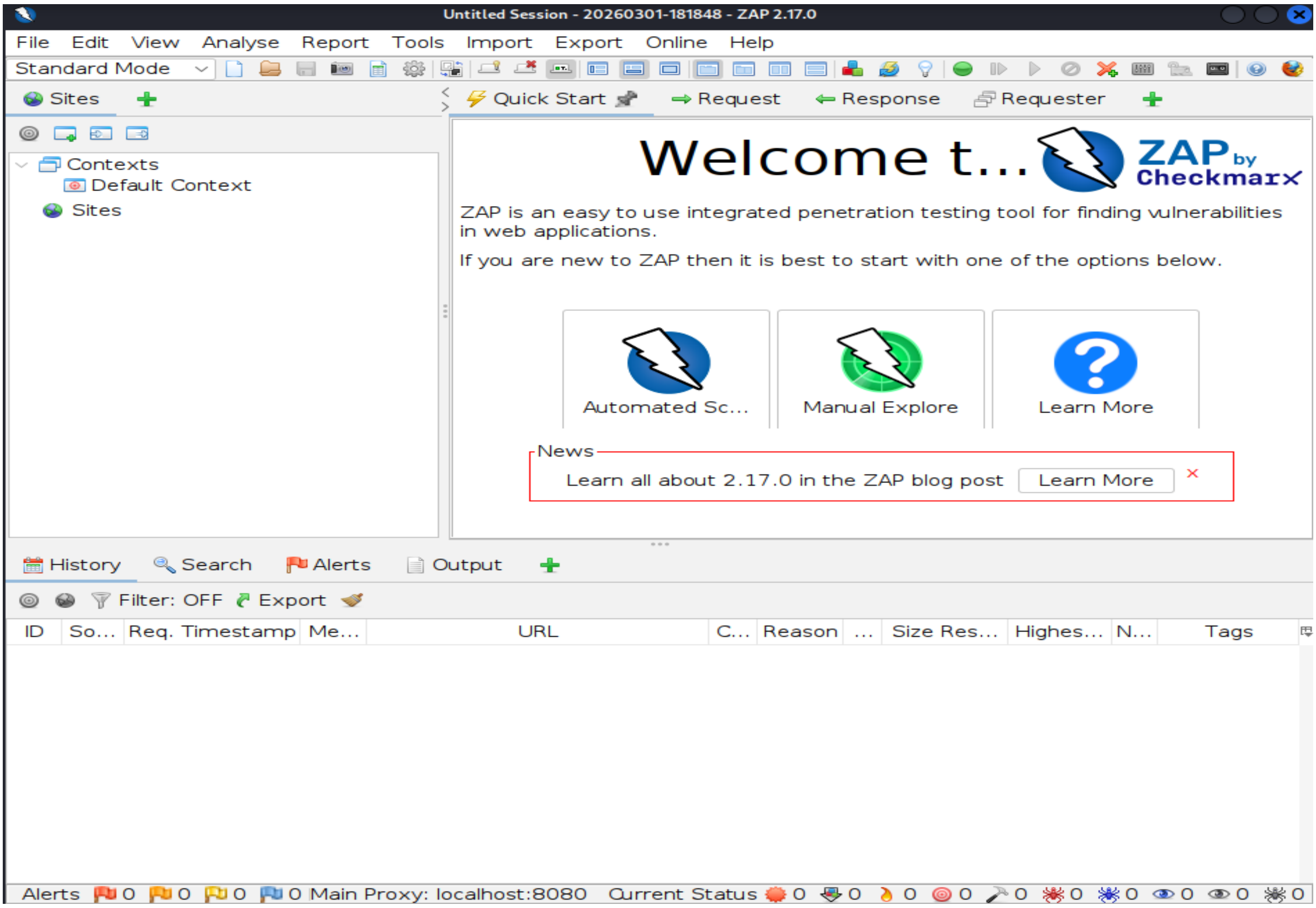
All findings were documented and categorized based on severity level.



```
┌──(ankit㉿kali)-[~]
└─$ nmap -sV testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-14 06:45 IST
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://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 41.44 seconds

┌──(ankit㉿kali)-[~]
└─$ 
```



ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

If you are new to ZAP then it is best to start with one of the options below.

Learn all about 2.17.0 in the ZAP blog post

Tools used

The following tools were used during the passive vulnerability assessment:

1. Nmap

Purpose:
Network scanning and service version detection.

Usage:
Used to perform service enumeration and identify open ports and running services on the target host using the command:

```
nmap -sV testphp.vulnweb.com
```

2. OWASP ZAP (Zed Attack Proxy)

Purpose:
Passive web application vulnerability scanning and HTTP response analysis.

Usage:
Configured as a local proxy (127.0.0.1:8080) to intercept and analyze HTTP requests and responses. The passive scanning module was used to detect security misconfigurations such as missing security headers and information disclosure issues.

3. Web Browser (Google Chrome)

Purpose:
Manual web application exploration and header inspection.

Usage:
Used for browsing the application and inspecting HTTP response headers via Developer Tools.

4. Kali Linux

Purpose:
Security testing environment.

Usage:
All security testing activities were performed within a controlled Kali Linux environment.

5. Canva

Purpose:
Report design and documentation formatting.

Usage:
Used to create and structure the final vulnerability assessment report.

Severity classification framework

The identified vulnerabilities were categorized based on industry-standard risk rating principles aligned with the Common Vulnerability Scoring System (CVSS v3.1).

The severity levels are defined as follows:

Critical (9.0 – 10.0)
Vulnerabilities that can result in complete system compromise, data breach, or unauthorized administrative access. Immediate remediation is required.

High (7.0 – 8.9)
Vulnerabilities that may allow significant impact such as privilege escalation, sensitive data exposure, or service disruption. Prompt remediation is recommended.

Medium (4.0 – 6.9)
Security weaknesses that may increase exposure to attack under certain conditions. Remediation should be planned in a timely manner.

Low (0.1 – 3.9)
Minor security misconfigurations or information disclosure issues with limited direct impact.

Informational
Observations that do not pose direct security risk but may improve overall security posture if addressed.

In this assessment, vulnerabilities were primarily categorized as Medium, Low, and Informational based on passive analysis findings.

Security Level	CVSS Range	Risk Description
Critical	9.0-10.0	Immediate system compromise risk
High	7.0-8.9	Significant security impact
Medium	4.0-6.9	Moderate security weakness
Low	0.1-3.9	Minor misconfiguration
Informational	N/A	Observation only

Attack Surface Overview

The attack surface represents the externally exposed components of the target application that may be accessible to potential attackers.

Based on reconnaissance and passive analysis, the following exposure points were identified:

Open Port:
- 80 (HTTP)

Web Server:
- nginx 1.19.0

Backend Technology:
- PHP (identified via X-Powered-By header)

Transport Security:
- HTTPS not enabled (application accessible over HTTP only)

Application Exposure:
- Publicly accessible web application

Observations:

The absence of HTTPS encryption and the disclosure of server information increase reconnaissance visibility. Although no critical vulnerabilities were actively exploited, these exposures contribute to the overall attack surface of the application.

Summary of findings

The passive vulnerability assessment identified multiple security misconfigurations primarily related to missing HTTP security headers and information disclosure.

A total of seven findings were identified and categorized as follows:

- Medium Severity Issues: 2
- Low Severity Issues: 4
- Informational Issues: 1

The identified weaknesses do not represent immediate critical compromise; however, they increase the overall attack surface and may facilitate client-side attacks or reconnaissance-based exploitation if left unaddressed.

No	Vulnerability	Severity
1	Content Security Policy (CSP) Header Not Set	Medium
2	Missing Anti-Clickjacking Header	Medium
3	Server Version Disclosure	Low
4	X-Powered-By Information Disclosure	Low
5	X-Content-Type-Options Header Missing	Low
6	Absence of Anti-CSRF Tokens	Low
7	Charset Mismatch	Informational

Detailed findings

Finding 1: Content Security Policy Not Set

Finding ID: WF-01

Severity: Medium

CVSS v3.1: 5.3

Confidence: High

Description:

The application does not implement a Content-Security-Policy (CSP) header. CSP is a security mechanism designed to mitigate client-side attacks such as Cross-Site Scripting (XSS) by restricting approved content sources.

Impact:

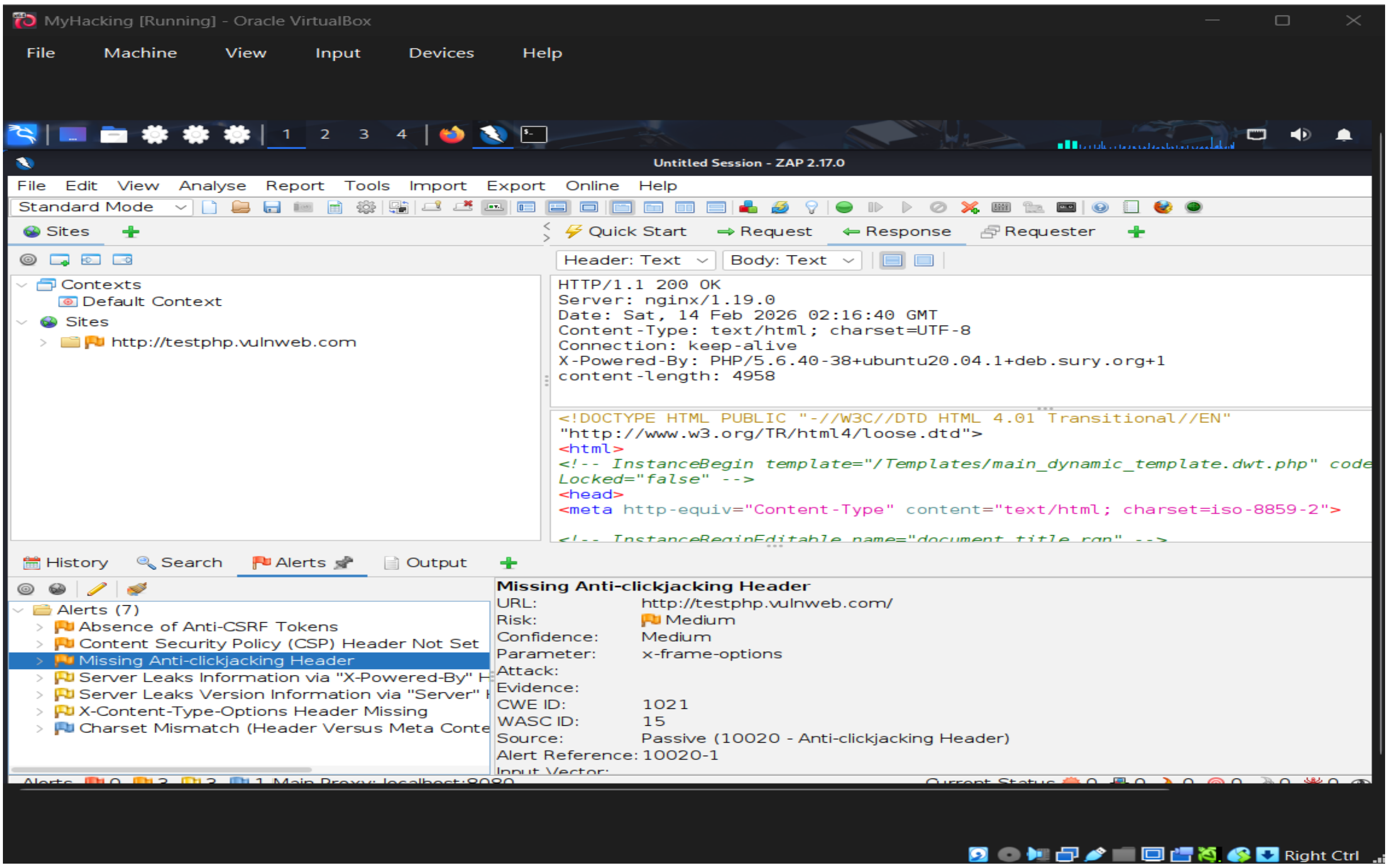
The absence of CSP increases exposure to malicious script injection if an injection vulnerability exists elsewhere in the application.

Affected URL:

http://testphp.vulnweb.com

Evidence:

OWASP ZAP Passive Scan detected absence of the "Content-Security-Policy" header in HTTP response.



MyHacking [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.17.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
History
Search
Alerts
Output
Right Ctrl

Contexts
Default Context
Sites
http://testphp.vulnweb.com

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 14 Feb 2026 02:16:40 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" code
Locked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

<!-- InstanceBeginEditable name="document_title_rgn" -->
```

Alerts (7)
- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Missing Anti-clickjacking Header
- Server Leaks Information via "X-Powered-By" H
- Server Leaks Version Information via "Server" I
- X-Content-Type-Options Header Missing
- Charset Mismatch (Header Versus Meta Conte

Missing Anti-clickjacking Header

URL:	http://testphp.vulnweb.com/
Risk:	Medium
Confidence:	Medium
Parameter:	x-frame-options
Attack:	
Evidence:	
CWE ID:	1021
WASC ID:	15
Source:	Passive (10020 - Anti-clickjacking Header)
Alert Reference:	10020-1
Input Vector:	

Alerts 0 3 3 1 Main Proxy: localhost:8080 Current Status 0 0 0 0 0 0

Recommendation:

Configure a strict Content-Security-Policy header in the web server configuration to restrict trusted sources for scripts, styles, and other resources.

Finding 2: missing anti-clickjacking header

Finding ID: WF-02

Severity: Medium

CVSS v3.1: 4.6

Confidence: Medium

Description:

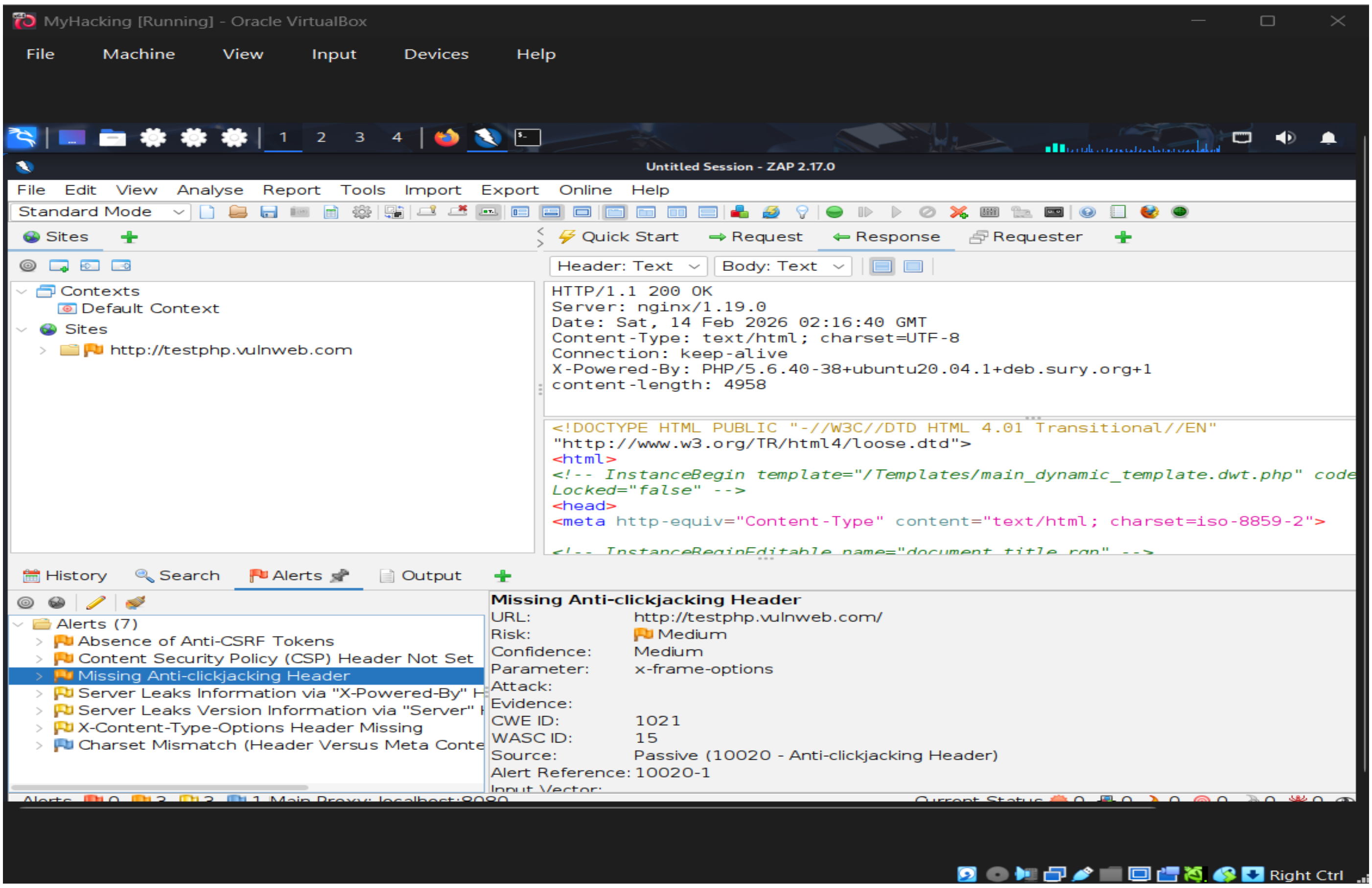
The application does not include the X-Frame-Options or Content-Security-Policy frame-ancestors directive to prevent clickjacking.

Impact:

Attackers may embed the application inside malicious iframes and trick users into performing unintended actions.

Evidence:

OWASP ZAP detected absence of X-Frame-Options header.


MyHacking [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Untitled Session - ZAP 2.17.0
File Edit View Analyse Report Tools Import Export Online Help
Standard Mode
Sites
Quick Start
Request
Response
Requester
Header: Text
Body: Text
Contexts
Default Context
Sites
http://testphp.vulnweb.com
History
Search
Alerts
Output
Right Ctrl

```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 14 Feb 2026 02:16:40 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" code
Locked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

<!-- InstanceBeginEditable name="document_title_rgn" -->
```

Alerts (7)
- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Missing Anti-clickjacking Header
- Server Leaks Information via "X-Powered-By" H
- Server Leaks Version Information via "Server" I
- X-Content-Type-Options Header Missing
- Charset Mismatch (Header Versus Meta Conte

Missing Anti-clickjacking Header

URL:	http://testphp.vulnweb.com/
Risk:	Medium
Confidence:	Medium
Parameter:	x-frame-options
Attack:	
Evidence:	
CWE ID:	1021
WASC ID:	15
Source:	Passive (10020 - Anti-clickjacking Header)
Alert Reference:	10020-1
Input Vector:	

Recommendation:

Add one of the following headers:
X-Frame-Options: DENY
OR
X-Frame-Options: SAMEORIGIN

Finding 3: Server Version Disclosure

Finding ID: WF-03

Severity: Low

CVSS v3.1: 3.1

Confidence: High

Description:

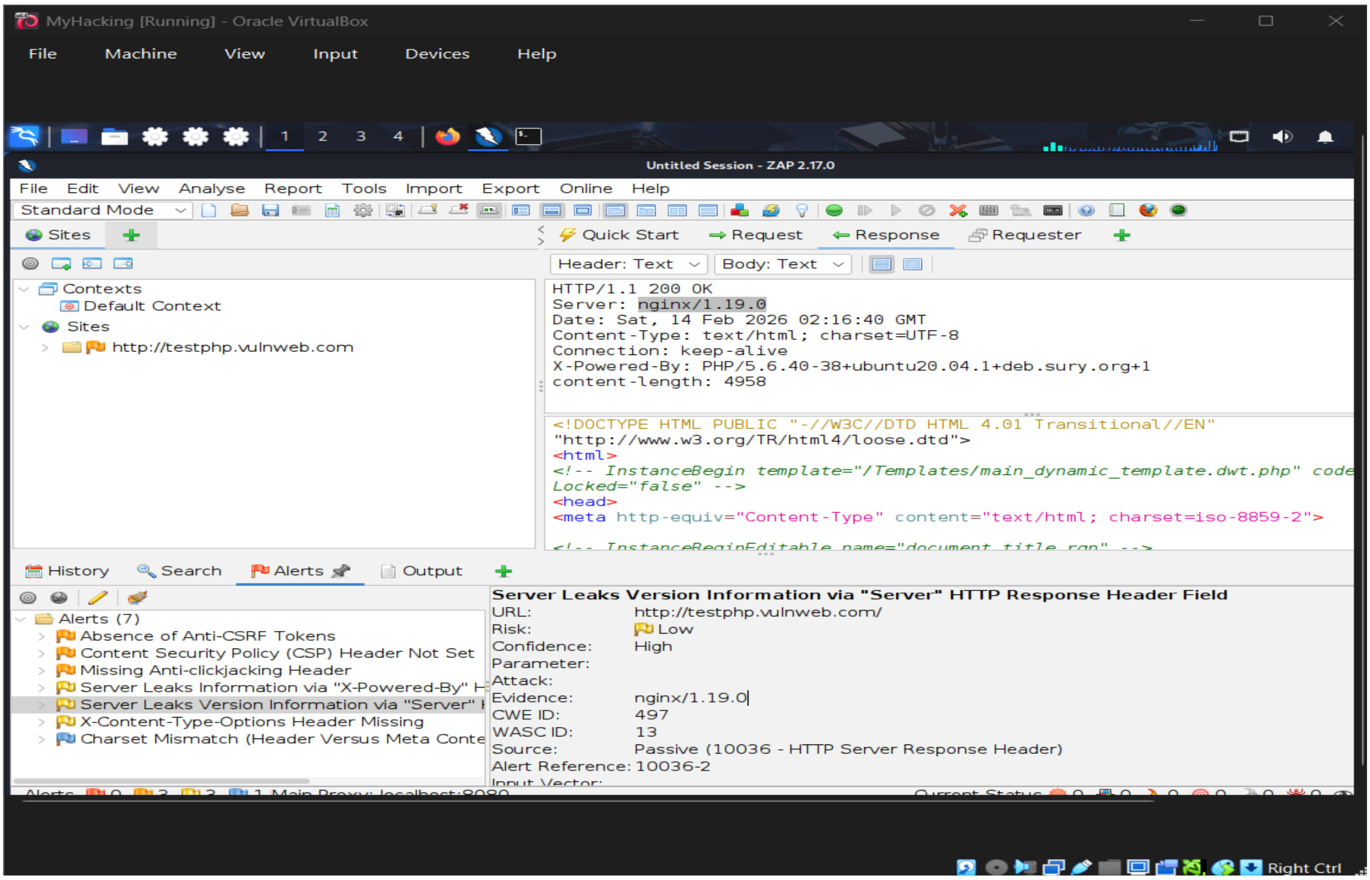
The HTTP response header reveals server version information: nginx/1.19.0.

Impact:

Server version disclosure may assist attackers during reconnaissance.

Evidence:

Server header visible in HTTP response.



```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 14 Feb 2026 02:16:40 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" code
Locked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
```

Alerts (7)
- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Missing Anti-clickjacking Header
- Server Leaks Information via "X-Powered-By" H
- Server Leaks Version Information via "Server" I
- X-Content-Type-Options Header Missing
- Charset Mismatch (Header Versus Meta Conte

Server Leaks Version Information via "Server" HTTP Response Header Field

URL:	http://testphp.vulnweb.com/
Risk:	Low
Confidence:	High
Parameter:	
Attack:	
Evidence:	nginx/1.19.0
CWE ID:	497
WASC ID:	13
Source:	Passive (10036 - HTTP Server Response Header)
Alert Reference:	10036-2
Input Vector:	

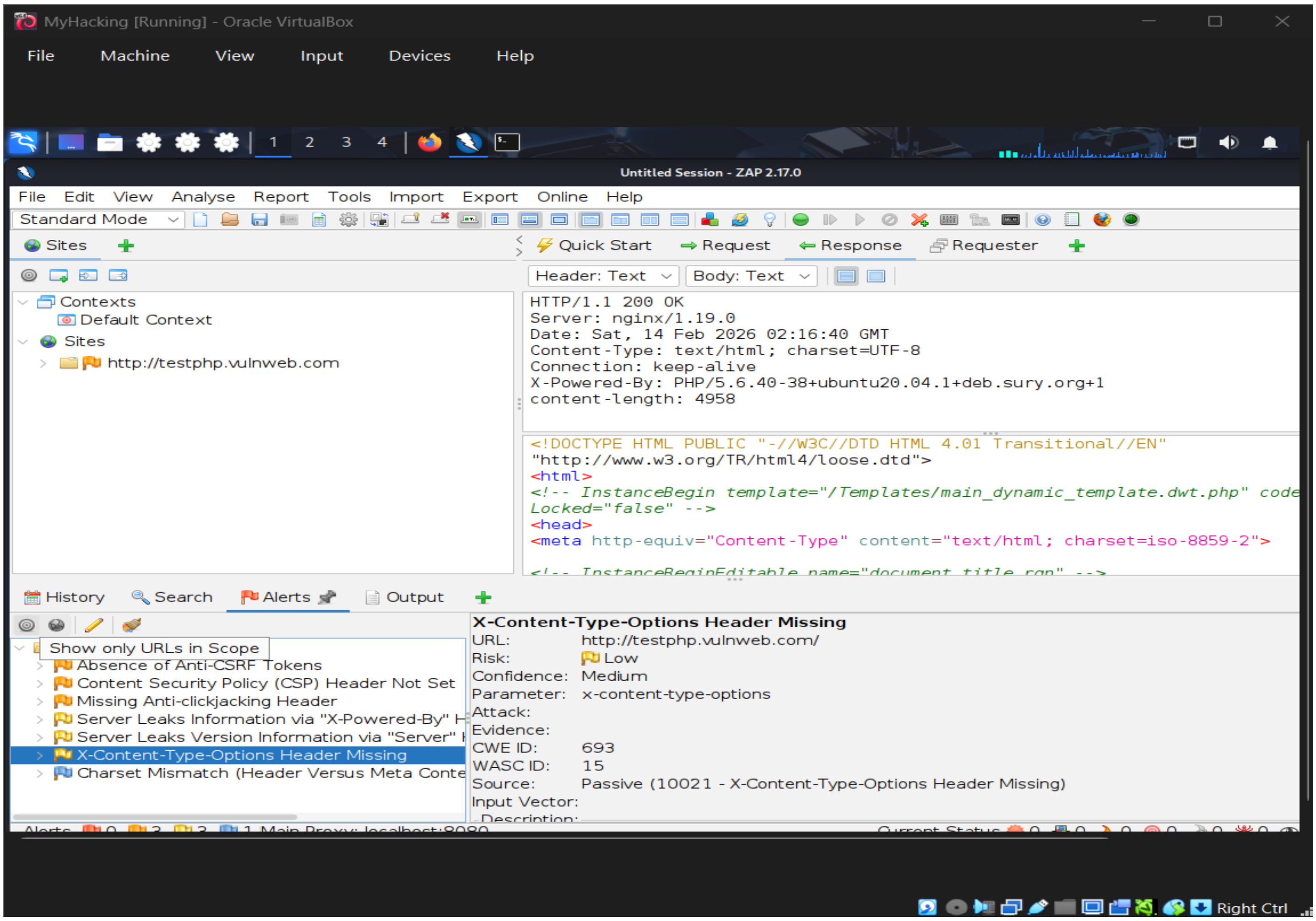
Recommendation:

Disable server version disclosure in web server configuration.

Finding 4: X-Content-Type-Options Header Missing

Severity: Low
CVSS: 2.9

Evidence:
Header not present



```
HTTP/1.1 200 OK
Server: nginx/1.19.0
Date: Sat, 14 Feb 2026 02:16:40 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/5.6.40-38+ubuntu20.04.1+deb.sury.org+1
content-length: 4958
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<!-- InstanceBegin template="/Templates/main_dynamic_template.dwt.php" code
Locked="false" -->
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">

<!-- InstanceBeginEditable name="document_title_rgn" -->
```

- Absence of Anti-CSRF Tokens
- Content Security Policy (CSP) Header Not Set
- Missing Anti-clickjacking Header
- Server Leaks Information via "X-Powered-By" H
- Server Leaks Version Information via "Server" I
- X-Content-Type-Options Header Missing
- Charset Mismatch (Header Versus Meta Conte

X-Content-Type-Options Header Missing

URL:	http://testphp.vulnweb.com/
Risk:	Low
Confidence:	Medium
Parameter:	x-content-type-options
Attack:	
Evidence:	
CWE ID:	693
WASC ID:	15
Source:	Passive (10021 - X-Content-Type-Options Header Missing)
Input Vector:	

Description:

Alerts 0 3 3 1 Main Proxy: localhost:8080

Current Status 0 0 0 0 0 0

Risk classification summary

Based on passive analysis findings, vulnerabilities were categorized according to severity level and potential impact.

Medium Severity Issues: 2

Low Severity Issues: 2

No Critical or High severity vulnerabilities were identified during this passive assessment.

The overall security posture of the application is classified as Moderate Risk due to missing security hardening controls.

Remediation priority

The identified vulnerabilities should be addressed in the following priority order:

Priority 1 – Immediate Attention (Medium Severity)

1. Implement Content Security Policy (CSP)

2. Enable Anti-Clickjacking Protection (X-Frame-Options)

Priority 2 – Scheduled Remediation (Low Severity)

3. Disable Server Version Disclosure

4. Add X-Content-Type-Options Header

Addressing the Medium severity issues will significantly reduce exposure to client-side attacks, while Low severity issues will reduce reconnaissance-based risk.

Priority	Vulnerability	Severity	Action Required
1	CSP Missing	Medium	Immediate
1	Missing Anti-Clickjacking	Medium	Immediate
2	Server Version Disclosure	Low	Scheduled
2	X-Content-Type Missing	Low	Scheduled

Business impact analysis

Although no critical vulnerabilities were identified during this passive assessment, the presence of medium and low-severity security weaknesses may increase overall organizational risk if left unaddressed.

The absence of important security headers such as Content-Security-Policy and X-Frame-Options increases exposure to client-side attacks including Cross-Site Scripting (XSS) and Clickjacking. If exploited in a real production environment, such attacks could lead to:

- Unauthorized user actions
- Session hijacking
- Credential theft
- Data manipulation

Server version disclosure and missing security hardening controls may assist attackers during reconnaissance, making targeted attacks easier to plan.

Additionally, lack of transport security (HTTP without HTTPS) may expose sensitive data to interception in real-world scenarios.

From a business perspective, these weaknesses may result in:

- Loss of user trust
- Reputational damage
- Regulatory compliance concerns
- Increased remediation cost if exploited

Implementing the recommended security controls will significantly reduce exposure and improve overall security posture.

Conclusion

The passive vulnerability assessment conducted on http://testphp.vulnweb.com identified multiple security misconfigurations primarily related to missing HTTP security headers and information disclosure.

No critical or high-severity vulnerabilities were discovered during the assessment. However, the presence of medium-severity issues such as missing Content Security Policy (CSP) and Anti-Clickjacking protection indicates insufficient security hardening.

The overall security posture of the application is classified as Moderate Risk.

While the identified issues may not immediately compromise the system, they increase the attack surface and may facilitate client-side exploitation or reconnaissance-based attacks if deployed in a real production environment.

Timely implementation of the recommended security controls will significantly enhance the resilience and defensive posture of the application.

This assessment demonstrates the importance of secure configuration practices and proactive vulnerability management.

References

1. OWASP Testing Guide v4

Open Web Application Security Project (OWASP)

https://owasp.org

2. OWASP Top 10 – 2021

Open Web Application Security Risks

https://owasp.org/www-project-top-ten/

3. NIST SP 800-115

Technical Guide to Information Security Testing and Assessment

National Institute of Standards and Technology

4. CVSS v3.1 Specification

Common Vulnerability Scoring System

https://www.first.org/cvss/

5. Nmap Official Documentation

https://nmap.org

6. OWASP ZAP Documentation

https://www.zaproxy.org