FUTURE INTERNS

VULUNEBBLITY ASSESSMENT REPORT FOR A LIVE WEBSITE

- Target Website : testphp.vulnweb.com
- Assessment Type: Vulenability Assessment
- Scope : Read-only(No explotation, no data modification)
- Testing Method : Non-intrusive / Ethical Testing

Total Vulenability Identified : 09

Submitted By : Sachin Kumar

Table of Contents

Executive Summary

-This report presents the results of a read-only vulnerability assessment conducted on the publicly accessible website testphp.vulnweb.com. The purpose of this assessment was to identify common and visible security weaknesses without performing any exploitation or intrusive activity.

The assessment identified several low to medium-risk security issues mainly related to configuration weaknesses and missing security best practices. No active attacks, login bypass, or data manipulation was performed during this assessment.

Overall, the website functions correctly; however, certain security improvements are recommended to reduce potential risks and improve its security posture.

About

- **Purpose of This Assessment**

The purpose of this vulnerability assessment is to identify common security weaknesses in the target website that could expose the business to potential risks. This assessment focuses on publicly accessible areas of the website using non-intrusive and read-only techniques. The goal is not to exploit vulnerabilities, but to understand the current security posture and highlight areas that need improvement.

- **Importance of Vulnerability Assessment**

A vulnerability assessment helps organizations discover security gaps before attackers can take advantage of them. Even simple issues like missing security headers or outdated configurations can lead to data leakage, loss of customer trust, or financial damage. Regular assessments allow businesses to reduce risk, strengthen defenses, and maintain a secure online presence.

- **Client / Business Point of View**

From a business perspective, this assessment provides clear visibility into the website’s security health. It helps decision-makers understand which issues require immediate attention and which can be addressed later. The report is written in simple, business-friendly language so that clients can easily understand the risks and take informed actions to protect their brand, customers, and digital assets.

Scope & Ethics

Scope of Testing

- This vulnerability assessment was conducted on the target website testphp.vulnweb.com and was strictly limited to read-only analysis.
- The assessment covered only publicly accessible web pages using passive and non-intrusive testing methods.

No actions were performed that could modify data or affect website availability.

Out of Scope

- The following activities were explicitly excluded from this assessment:
- Authentication or login-based testing
- Vulnerability exploitation
- Denial-of-Service (DoS) attacks
- Brute-force or intrusive attack techniques

All testing was performed in compliance/accordance with ethical security assessment standards.

Tool & Purpose

Tools	Purpose
Nmap	Port & service visibility
OWASP ZAP	Passive vulnerability detection
Browser DevTools	Headers, cookies, client-side review
Canva	Professional report design

- # Nmap

- **Nmap** (Network Mapper) is a security scanning tool used to identify:
  - Open ports
  - Running services
  - Basic exposure of a website/server

In this task, we use only basic and safe scanning, because the scope is Read-Only / Ethical Assessment. No exploitation, no aggressive scan.

Port	Service	Version	Status	Risk Level
80	HTTP	nginx 1.19.0	Open	Medium

- **OWASP ZAP**

OWASP ZAP (Zed Attack Proxy) is a free, open-source web application security testing tool developed by the OWASP Foundation.

It is used to find security vulnerabilities in web applications by acting as a man-in-the-middle proxy between the user and the web application.

- **Browser DevTools:**

It is the use of built-in browser tools to observe web application behavior by inspecting normal traffic only. No attacks are performed and no data is changed, making it a safe and ethical method for read-only security analysis and awareness reports.

- **Canva**

Canva is an online graphic design platform used to create professional documents, presentations, reports, posters, and infographics using ready-made templates and drag-and-drop tools.

It is widely used for project reports, internships, and security awareness documentation.

Assessment Methodology

The vulnerability assessment was conducted using a structured, ethical, and non-intrusive approach. All testing activities were read-only, ensuring that no data was modified and no harm was caused to the target website.

Assessment Steps

1. T**arget Identification**

The target website http://testphp.vulnweb.com/ was identified for assessment.
Scope and testing limitations were clearly defined before starting.

2. **Passive Scanning**

Passive scanning was conducted using OWASP ZAP to identify vulnerabilities without sending harmful requests.

3. **Header Analysis**

HTTP security headers were reviewed using Browser Developer Tools to check for missing or misconfigured headers.

4. **Exposure Review**

Publicly accessible information such as server details, parameters, and responses were analyzed for potential exposure.

5. **Risk Classification**

Identified issues were classified into Low, Medium, or High risk based on their impact and likelihood.

6. **Documentation**

All findings were properly documented with description, impact, and recommendations in a professional report format.

Target web identification

Information	Details
Target URL	http://testphp.vulnweb.com/
Domain	testphp.vulnweb.com
Protocol	HTTP
Server	nginx 1.19.0
Hosting Platform	Amazon Web Services (AWS)
Status	Open
Technology	PHP 5.6.40
Access Level	Public
Application Type	Vulnerable web app
Testing Type	Passive / Read-only
Assessment Date	Jan 2026

Vulnerability Breakdown

Identified Vulnerabilities

1. Absence of Anti-CSRF Tokens

Summary

The application does not have Anti-CSRF protection on its forms. Because of this, an attacker can trick a logged-in user into unknowingly sending unauthorized requests. This happens because the application does not verify whether a request is genuine or intentionally made by the user. Unlike XSS, which abuses user trust, CSRF abuses the application's trust in authenticated users.

Impact

If exploited, an attacker can perform actions on behalf of a logged-in user, such as changing data, submitting forms, or triggering sensitive operations. The risk is higher when the user is already authenticated or when the application is also vulnerable to XSS, which can help attackers bypass CSRF defenses.

Recommendation

Implement Anti-CSRF tokens and use secure frameworks that provide built-in CSRF protection.

Solution

The application should use unique and unpredictable CSRF tokens for all forms and validate them on the server. State-changing actions should avoid GET requests and use secure methods. Protection against XSS is also necessary, as XSS can bypass CSRF defenses. For sensitive actions, user confirmation should be added. Optional checks like Referer validation may be used carefully.

Other Info

No known Anti-CSRF token [token, csrfToken, csrf-token] was found in the following HTML form: [Form 1: "name"].

References

- https://cheatsheetseries.owasp.org/cheatsheets/Cross-Site_Request_Forgery_Prevention_Cheat_Sheet.html
- https://cwe.mitre.org/data/definitions/352.html

Details	
Alert ID	10202
Alert Type	Passive
Status	release
Risk	Medium
CWE	352
WASC	9
Technologies Targeted	All
Tags	CWE-352 OWASP_2017_A05 OWASP_2021_A01 POLICY_DEV_STD POLICY_PENTEST POLICY_QA_STD SYSTEMIC WSTG-V42-SESS-05
More Info	Scan Rule Help

2. Content Security Policy (CSP) Header Not Set

Summary

Content Security Policy (CSP) is a security mechanism that helps protect web applications from common attacks such as Cross-Site Scripting (XSS) and data injection. It works by allowing website owners to define trusted sources from which content can be loaded by the browser. By restricting the execution of untrusted scripts and resources, CSP reduces the risk of data theft, website defacement, and malware distribution.

Impact : script injection, data theft, and malware attacks.
Recommendation : Apply a strict CSP allowing content only from trusted sources.

Solution

Configure the web server or application to include the Content-Security-Policy HTTP response header on all pages.

References

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/CSP
- https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
- https://www.w3.org/TR/CSP/
- https://w3c.github.io/webappsec-csp/
- https://web.dev/articles/csp
- https://caniuse.com/#feat=contentsecuritypolicy
- https://content-security-policy.com/

Details	
Alert ID	10038-1
Alert Type	Passive
Status	release
Risk	Medium
CWE	693
WASC	15
Technologies Targeted	All
Tags	CWE-693 OWASP_2017_A06 OWASP_2021_A05 POLICY_PENTEST POLICY_QA_STD SYSTEMIC
More Info	Scan Rule Help

3. Missing Anti-clickjacking Header

Summary

The application does not include protection against Clickjacking attacks, as required security headers such as Content-Security-Policy with frame-ancestors or X-Frame-Options are missing.

Impact : Clickjacking attacks

Recommendation : Clickjacking protection should be enforced on all pages using modern browser-supported security headers.

Solution

Configure the server to set X-Frame-Options as DENY or SAMEORIGIN, or implement Content Security Policy with the frame-ancestors directive.

References

- https://developer.mozilla.org/en-US/docs/Web/HTTP/Reference/Headers/X-Frame-Options

Details	
Alert ID	10020-1
Alert Type	Passive
Status	release
Risk	Medium
CWE	1021
WASC	15
Technologies Targeted	All
Tags	CWE-1021 OWASP_2017_A06 OWASP_2021_A05 POLICY_PENTEST POLICY_QA_STD SYSTEMIC WSTG-V42-CLNT-09
More Info	Scan Rule Help

4. In Page Banner Information Leak

Summary

The server returned a version banner string in the response content. Such information leaks may allow attackers to further target specific issues impacting the product and version in use.

Impact

Exposed server version information can help attackers identify known vulnerabilities specific to the server software and version, increasing the risk of targeted attacks.

Recommendation

Disable server version banners and error disclosures by configuring server settings to hide product and version details in all responses.

Solution

Configure the server to prevent such information leaks. For example: Under Tomcat this is done via the "server" directive and implementation of custom error pages. Under Apache this is done via the "ServerSignature" and "ServerTokens" directives.

Other Info

There is a chance that the highlight in the finding is on a value in the headers, versus the actual matched string in the response body.

References

- https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/

Details	
Alert ID	10009
Alert Type	Passive
Status	release
Risk	Low
CWE	497
WASC	13
Technologies Targeted	All
Tags	CWE-497 OWASP_2017_A06 OWASP_2021_A05 POLICY_PENTEST POLICY_QA_STD SYSTEMIC WSTG-V42-INFO-02
More Info	Scan Rule Help

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

Summary

The web/application server is leaking information via one or more “X-Powered-By” HTTP response headers. Access to such information may facilitate attackers identifying other frameworks/components your web application is reliant upon and the vulnerabilities such components may be subject to.

Impact: Technology disclosure risk

Recommendation : Suppress or remove the X-Powered-By header from all server responses.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to suppress "X-Powered-By" headers.

References

- https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/
- https://www.troyhunt.com/shhh-dont-let-your-response-headers/

Details	
Alert ID	10037
Alert Type	Passive
Status	release
Risk	Low
CWE	497
WASC	13
Technologies Targeted	All
Tags	CWE-497 OWASP_2017_A03 OWASP_2021_A01 POLICY_PENTEST POLICY_QA_STD SYSTEMIC WSTG-V42-INFO-08
More Info	Scan Rule Help

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

Summary

The web/application server is leaking version information via the “Server” HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.

Impact:

Version disclosure can help attackers identify and exploit known server vulnerabilities.

Recommendation:

Avoid revealing exact server software and version information.

Solution

Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

Other Info:

The version information was observed in the HTTP response headers and may vary depending on server configuration, intermediary devices, or default framework settings.

References

- https://httpd.apache.org/docs/current/mod/core.html#servertokens
- https://learn.microsoft.com/en-us/previous-versions/msp-n-p/ff648552(v=pandp.10)
- https://www.troyhunt.com/shhh-dont-let-your-response-headers/

Details	
Alert ID	10036-2
Alert Type	Passive
Status	release
Risk	Low
CWE	497
WASC	13
Technologies Targeted	All
Tags	CWE-497 OWASP_2017_A06 OWASP_2021_A05 POLICY_PENTEST POLICY_QA_STD SYSTEMIC WSTG-V42-INFO-02
More Info	Scan Rule Help

7. X-Content-Type-Options Header Missing

Summary

The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Impact

Missing X-Content-Type-Options may allow browsers to execute malicious content.

Recommendation

Enable X-Content-Type-Options: nosniff on all server responses.

Solution

Configure the web server or application to set the X-Content-Type-Options header to nosniff on all responses, including error pages, and ensure correct Content-Type headers are used.

Other Info

Error pages (401, 403, 500) may also be affected by MIME sniffing, though high scan thresholds may not report these responses.

References

- https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85)
- https://owasp.org/www-community/Security_Headers

Details	
Alert ID	10021
Alert Type	Passive
Status	release
Risk	Low
CWE	693
WASC	15
Technologies Targeted	All
Tags	CWE-693 OWASP_2017_A06 OWASP_2021_A05 POLICY_PENTEST POLICY_QA_STD SYSTEMIC
More Info	Scan Rule Help

8. Charset Mismatch (Header Versus Meta Content-Type Charset)

Summary

The application returns responses where the character encoding defined in the HTTP Content-Type header does not match the encoding declared in the HTML or XML body. This mismatch may cause browsers to perform content sniffing to determine the correct charset, which can lead to improper content interpretation.

Impact

Charset mismatch may allow attackers to manipulate content parsing and inject malicious scripts.

Recommendation

Ensure a single consistent charset (UTF-8) is used in both HTTP headers and page content.

Solution

Force UTF-8 for all text content in both the HTTP header and meta tags in HTML or encoding declarations in XML.

Other Info

There was a charset mismatch between the HTTP Header and the META content-type encoding declarations: [UTF-8] and [ISO-123] do not match.

References

- https://code.google.com/archive/p/browsersec/wikis/Part2.wiki#Character_set_handling_and_detection

Details	
Alert ID	90011-1
Alert Type	Passive
Status	release
Risk	Informational
CWE	436
WASC	15
Technologies Targeted	All
Tags	CWE-436 POLICY_PENTEST POLICY_QA_STD SYSTEMIC
More Info	Scan Rule Help

9. **Modern Web Application**

Summary

The application appears to be a modern web application. If you need to explore it automatically then the Ajax Spider may well be more effective than the standard one.

Impact

This is an informational finding and does not represent a security risk to the application.

Recommendation

No remediation is required. Use the Ajax Spider for more effective automated testing of modern web applications.

Solution

This is an informational alert and so no changes are required.

Other Info

Links using the _self target are commonly used by modern frameworks to trigger full page reloads and are not considered a security issue.

Details	
Alert ID	10109
Alert Type	Passive
Status	release
Risk	Informational
CWE	
WASC	
Technologies Targeted	All
Tags	POLICY_DEV_STD POLICY_PENTEST POLICY_QA_STD SYSTEMIC
More Info	Scan Rule Help

Conclusion

This vulnerability assessment was carried out on the publicly accessible website testphp.vulnweb.com using a strictly read-only and non-intrusive approach. The purpose of this assessment was to identify visible security weaknesses without performing any exploitation or intrusive testing. During the assessment, several low to medium risk issues were identified, mainly related to missing security controls and weak security configurations.

No active attacks, authentication bypass, data manipulation, or service disruption was performed at any stage of the assessment. While the website is operational and functions as expected, the identified issues could be misused by attackers if left unaddressed. Overall, the current security posture can be improved by implementing standard security best practices.