

VULNERABILITY ASSESSMENT REPORT

Date: [Monday 3rd March, 2025]

Assessed By: [Ganiyu Adeola]

Target System/Website: [<http://testphp.vulnweb.com/>]

Assessment Tool Used: [OWASP ZAP]

1. Introduction

This report provides a detailed assessment of the target system to identify vulnerabilities and evaluate the security posture. The goal is to highlight weaknesses and recommend solutions to mitigate risks.

Objective: [The assessment was conducted to identify common web application vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Security Misconfigurations using OWASP ZAP. The findings aim to provide recommendations to improve the overall security of the web application.]

2. Scope of Assessment




- Target System/Website: [<http://testphp.vulnweb.com/>]
 - Type of Assessment: [Automated]
 - Tools Used: [OWASP ZAP]
-

3. Methodology

The vulnerability assessment followed these steps:




1. Information Gathering: The target website URL was selected and basic information was collected to understand the structure of the website.
 2. Automated Scanning: OWASP ZAP was used to perform an automated scan on the target website to identify security vulnerabilities.
 3. Vulnerability Detection: The tool scanned the web application for common vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), and Security Misconfigurations.
 4. Result Analysis: The scan results were analyzed to identify critical, medium, and low-level vulnerabilities.
 5. Reporting: All identified vulnerabilities were documented with their severity levels and recommended solutions.
-

4. Vulnerabilities Identified

Vulnerability Name	Severity	Description	Risk Level	Recommendation
SQL Injection	High 	Allows attackers to manipulate database queries	High	Use parameterized queries and input validation
Cross-Site Scripting (XSS)	High 	Injects malicious scripts into web pages	High	Validate and sanitize user input
Missing Security Headers	Medium 	No X-Frame-Options header found	Medium	Add security headers in HTTP responses

5. Risk Analysis

The assessment identified 3 vulnerabilities with the following risk distribution:

Severity	Count	Risk Explanation
High 	2	High-risk vulnerabilities like SQL Injection and Cross-Site Scripting (XSS) could allow attackers to compromise sensitive data or take control of the system. Immediate action is required.
Medium 	1	The Missing Security Headers vulnerability can expose the application to Clickjacking attacks, potentially compromising user sessions or sensitive information.
Low 	0	No low-risk vulnerabilities were identified during the scan.

6. Recommendations

Based on the findings, the following recommendations are made:

- Immediate Action: Fix high-risk vulnerabilities like SQL Injection and Cross-Site Scripting by implementing input validation, parameterized queries, and escaping special characters.
- Security Headers: Configure X-Frame-Options, Content Security Policy (CSP), and X-XSS-Protection headers to protect against clickjacking and XSS attacks.
- Regular Updates: Keep all software, plugins, and libraries up-to-date to reduce the risk of exploitation through outdated components.
- Input Validation: Implement strict input validation and sanitization across all user input fields.
- Periodic Vulnerability Assessments: Perform regular vulnerability scans and penetration testing to identify and mitigate security weaknesses.

7. Conclusion

The vulnerability assessment conducted on <http://testphp.vulnweb.com/> using OWASP ZAP identified 3 vulnerabilities, with 2 high-risk vulnerabilities and 1 medium-risk vulnerability. The presence of SQL Injection and Cross-Site Scripting (XSS) poses a significant threat to the web application's security and could lead to unauthorized access or data compromise. Addressing these vulnerabilities promptly will help enhance the overall security posture of the system. It is recommended that the organization implements the suggested mitigation strategies, conducts regular vulnerability assessments, and adopts security best practices to minimize future risks.

Scenario Description

A suspicious email was received from what appeared to be a legitimate financial institution, requesting the recipient to update their account details due to security reasons. The email contained:

- An urgent message prompting immediate action.
- A hyperlink redirecting the recipient to a login page resembling the official bank website.
- Instructions to enter login credentials and credit card information.

Key Indicators of Compromise (IoCs)

Indicator	Description	How to Detect It
Suspicious Email Address	The sender's email address is slightly misspelled or uses a free email service (e.g., support@bankk.com instead of support@bank.com).	Check the email header and domain name carefully.
Urgency in the Message	The email uses urgent language like "Immediate Action Required" or "Your Account Will Be Suspended."	Analyze the email body for pressure tactics.
Generic Greetings	Phrases like "Dear Customer" instead of your actual name.	Review how the sender addresses you.
Malicious Links	Hyperlinks that, when hovered over, display a suspicious URL different from the one written.	Hover over links without clicking. Use URL scanners like VirusTotal.
Attachments with Malware	Suspicious attachments (.zip, .exe, .doc) that may contain malware.	Use antivirus software to scan attachments before opening.
Typos and Grammatical Errors	Professional organizations rarely send emails with multiple spelling or grammar mistakes.	Proofread the message for errors.
HTTPS Missing	The fake website uses HTTP instead of HTTPS.	Always check for secure padlock symbols in the URL bar.
Spoofed Domain Names	Website URLs that slightly alter official domains (e.g., www.bank-login.com instead of www.bank.com).	Verify the URL and domain name spelling.