Comprehensive Security Evaluation of Web Login Services in Academic Digital Libraries: Leveraging OWASP ZAP and Arachni

Hanugra Aulia Sidharta1, a), Muhammad Zakiy Alfaris2, b), and Diah Risqiwati 3, c)

Author Affiliations

1 Computer Science Department, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia   
 2,3 Department of Informatics, Universitas Muhammadiyah Malang, Malang, Indonesia

Author Emails

a) hanugra.sidharta@binus.ac.id

b) zakiyalfaris02@webmail.umm.ac.idc) Corresponding author: risqiwati@umm.ac.id

**Abstract.** **T**he evolving security threat landscape increasingly targets web applications, with attackers exploiting vulnerabilities in data validation and error handling to gain unauthorized access, steal sensitive information, or disrupt services. In response to these growing risks, this research aims to assess the security posture of the University of Muhammadiyah Malang (UMM) public digital library platform, focusing specifically on its web login services, which are critical entry points for users and potential targets for attackers. To conduct a thorough vulnerability assessment, Dynamic Application Security Testing (DAST) tools, OWASP ZAP and Arachni were employed, leveraging their complementary capabilities to identify and analyze potential security weaknesses. The assessment was guided by the OWASP Top 10 2021 framework, widely recognized as a comprehensive and authoritative set of security best practices. To maintain focus and reduce complexity, the study concentrated on vulnerabilities related to Data Validation and Error Handling failures, narrowing the scope from ten to four key observations. Initial automated scans revealed 19 potential risks using ZAP and 5 using Arachni, highlighting areas of concern that warranted further investigation. Subsequent analysis followed the OWASP Web Security Testing Guide (WSTG), incorporating in-depth testing techniques such as Cross-Site Scripting (XSS), SQL Injection (SQLi), and manual evaluation of error handling mechanisms. Based on the calculated Likelihood and Impact Factors, the platform was assessed to have a moderate risk level from ZAP’s perspective, while Arachni’s analysis indicated a lower risk level. These findings underscore the importance of continuous, multi-faceted security testing to protect critical web services against evolving threats.

# introduction

The digital security landscape is evolving rapidly, driven by cybercriminals leveraging artificial intelligence to automate malware mutation, phishing content generation, and defense evasion. By 2025, 60% of IT professionals worldwide have identified AI-driven threats as their primary concern. Deepfake-based social engineering attacks have increased by 550% between 2019 and 2023 and are projected to reach 8 million incidents by 2025. Credential-related risks remain predominant, with Trend Micro reporting identity and access misconfigurations as the leading risk event for 2024–2025, exacerbated by the rise of remote work and cloud migration[1][2].

In 2024, 65% of enterprises inadvertently exposed sensitive data due to cloud storage misconfigurations. Trend Micro identified non-compliant AWS S3 buckets, such as those with public access settings—as the most common misconfiguration, which has led to significant data breaches, including Schneider Electric’s 1.5TB leak[3]. Additionally, the OWASP Top 10 for 2021 introduced "Insecure Design" (A04) as a new category, highlighting vulnerabilities such as Insecure Direct Object Reference (IDOR). A notable example is the First American Financial breach, which exposed 885 million records due to IDOR flaws in website design[4].

Enhancing web security requires a combination of proactive measures and continuous monitoring. A key strategy is conducting regular vulnerability scans to identify security weaknesses before they can be exploited by attackers. For example, Dann’s research focuses on the security of open source software by performing vulnerability scans on 7,024 Java projects developed at SAP, uncovering potential security issues[5]. Additionally, Ferda is developing a dashboard designed to visualize attack patterns, enabling more detailed observation and mitigation of vulnerabilities[6].

Researchers conduct security assessments by utilizing prominent tools, such as OWASP ZAP, which has demonstrated reliability and effectiveness across numerous real-world cases[7]. OWASP ZAP is robust because it offers comprehensive, automated vulnerability scanning combined with manual testing tools, making it effective for identifying a wide range of web application security issues. In addition to OWASP ZAP, another valuable tool is Arachni, which is widely recognized for its thorough and scalable web application security scanning. Arachni offers advanced detection features and supports seamless integration into existing development workflows, facilitating continuous security testing and early vulnerability detection [8]. Its scalability and adaptability make it particularly suitable for large and complex web applications, ensuring that security risks are identified and addressed efficiently.

# related work

The Open Web Application Security Project (OWASP) has been a leading authority in web application security since its establishment in 2001. It provides widely recognized resources that guide vulnerability management and risk mitigation efforts. Among these, the OWASP Top 10 stands out as a critical framework that identifies the most significant security threats to web applications. This resource has become a de facto industry standard and is extensively utilized by researchers and practitioners due to its broad applicability across various security scenarios. Complementing this, the OWASP Web Security Testing Guide (WSTG) offers in-depth methodologies for detecting a wide range of attack vectors. The WSTG has proven effective in diverse applications such as web server office management[9], network penetration testing[10], and assessing security risks in mobile health applications[11], supported by secure coding standards that enhance overall software security[12].

In addition to its documentation, OWASP provides practical tools like the OWASP Zed Attack Proxy (ZAP), which is highly versatile and easily customizable for vulnerability scanning and attack prevention. This tool has been successfully employed to enhance the security posture of government websites k[14] and other critical infrastructures[13]. OWASP’s extensive knowledge base is continuously updated to reflect the latest threat patterns and security research, ensuring its ongoing relevance in the rapidly evolving cybersecurity landscape. The organization’s commitment to maintaining up-to-date resources, including recent updates as of this year[15], underscores its vital role in supporting secure web application development and defense strategies worldwide.

# methodology

In this research is composed of three important key, such as vulnerability framework, DAST tools selection, and scoring system that been used to measure vulnerability.

## Vulnerability Framework

Vulnerability frameworks offer essential methodologies for identifying, categorizing, and mitigating security weaknesses in software systems. Notable frameworks include the NIST Cybersecurity Framework[16], MITRE ATT&CK[17], and OWASP Top 10, each with distinct focuses: NIST provides a broad risk management approach, MITRE ATT&CK emphasizes adversary tactics, and OWASP Top 10 targets the most critical web application risks. OWASP’s specialization in web security, driven by community input and data, helps organizations prioritize security efforts, improve development practices, and effectively reduce attack surfaces. The OWASP Top 10 2021 edition, recognized as a leading standard, reflects the evolving threat landscape by introducing new categories like "Insecure Design" and refining others such as "Broken Access Control," ensuring relevance through empirical data and expert consensus.

This research adopts the OWASP Top 10 2021 as its primary risk taxonomy, focusing specifically on vulnerabilities related to Data Validation and Error Handling failures. These areas are critical because improper data validation can lead to injection attacks and cross-site scripting, while poor error handling may expose sensitive information exploitable by attackers. Together, they account for approximately 80% of common web application security issues. This focus aligns with the OWASP Web Security Testing Guide (WSTG) version 4.1 as define on Table. 1, particularly sections on Data Validation Testing (WSTG-INPV) and Error Handling Analysis (WSTG-ERRH), which provide best practices to prevent malicious input and secure error management. Given that these failures are responsible for many detected vulnerabilities, including SQL injection and information disclosure, concentrating on these aspects offers a comprehensive approach to mitigating prevalent and severe web security risks.

**TABLE 1.** OWASP mapping category aligning with data validation testing dan error handling goals.

|  |  |  |
| --- | --- | --- |
| **OWASP Category** | **WSTG Test ID** | **Relevance** |
| A03: Injection | WSTG-INPV-01 to  WSTG-INPV-19 | Core focus: SQLi, XSS, Command Injection via malicious input fuzzing. |
| A05: Security Misconfiguration | WSTG-ERRH-01 | Critical: Detection of stack traces, debug errors, or system details leakage. |
| A01: Broken Access Control | WSTG-ATHZ-02,04 | Secondary: IDOR via parameter manipulation during input validation. |
| A10: SSRF | WSTG-INPV-12 | High-priority: Input validation flaws enabling internal network access. |

## DAST Tools Selection

In this study, we selected two prominent open-source Dynamic Application Security Testing (DAST) tools—OWASP ZAP (v2.12.0) and Arachni (v2.0.0)—for their complementary strengths in dynamic security analysis. Both tools enhance detection of vulnerabilities related to data validation and error handling, which are central to this research. OWASP ZAP excels in data validation testing with its advanced fuzzing engine featuring over 800 built-in payloads and support for custom Python and JavaScript scripting. It enables extensive input manipulation, including real-time parameter tampering on AJAX and API endpoints via OpenAPI and SOAP integration. For error handling, ZAP scans for leakage of sensitive debug information and analyzes HTTP headers to detect missing security controls like Content Security Policy (CSP) and HTTP Strict Transport Security (HSTS).

Conversely, Arachni offers strengths in data validation through framework-specific payloads tailored for Rails, PHP, and .NET, enabling deep injection testing customized to the target stack. Its machine-learning-driven false-positive reduction improves detection accuracy and efficiency. In error handling, Arachni passively identifies misconfigured cookies and headers by analyzing response fingerprints and provides high-accuracy Server-Side Request Forgery (SSRF) detection using sophisticated out-of-band techniques. Together, OWASP ZAP and Arachni deliver a balanced and thorough dynamic security testing approach, with each tool addressing different aspects of data validation and error handling vulnerabilities. This dual-tool strategy broadens coverage and enhances the reliability and depth of the vulnerability analysis in this research.

## Scoring System

Structured scoring system is employed to evaluate vulnerabilities based on two primary dimensions: Likelihood Factors and Impact Factors. Likelihood factors in vulnerability assessment serve to estimate the probability of a successful exploitation by potential threat actors, often considering worst-case scenarios to ensure comprehensive risk evaluation. These factors encompass attributes related to the attacker, including the level of skill required, motivation, access to necessary resources, and the size of the attacker population. Complementing these are vulnerability-specific factors such as ease of discovery, ease of exploitation, attacker awareness, and detectability. Each element is assigned a weighted score to systematically quantify the likelihood dimension, thereby enabling a nuanced and data-driven appraisal of exploitation potential.

Impact factors assess the ramifications of a successful exploit from both technical and business perspectives. The technical impact dimension evaluates the extent of damage to confidentiality, integrity, availability, and traceability of systems and data, ranging from minimal exposure to total compromise. Business impact considers broader organizational consequences, including financial loss, reputational damage, regulatory non-compliance, and privacy breaches. These impacts are similarly weighted to facilitate prioritization, with an emphasis on business impact as advocated by OWASP, ensuring that security assessments align with strategic organizational objectives.

Together, the likelihood and impact factors constitute a robust scoring framework that balances the probability of exploitation against the severity of potential outcomes. This dual-faceted approach equips security practitioners with a systematic methodology to prioritize vulnerabilities effectively, directing remediation efforts toward those weaknesses that pose the greatest overall risk to both technical assets and business operations.

By integrating these weighted factors into a cohesive scoring model, organizations can make informed, risk-based decisions regarding vulnerability management and resource allocation. This ensures that security initiatives are both efficient and aligned with the organization’s risk tolerance and business priorities, ultimately enhancing the resilience and security posture of the enterprise.

# result and discussion

This research analyzes vulnerabilities related to Data Validation and Error Handling failures using the OWASP Top 10 2021 as the primary risk taxonomy. The study focuses on assessing the security of the University of Muhammadiyah Malang (UMM) public digital library platform, especially its web login services. To achieve this, two complementary open-source Dynamic Application Security Testing (DAST) tools. The vulnerability assessment follows a two-stage process combining automated scanning and active penetration testing. OWASP ZAP uses an advanced fuzzing engine and real-time parameter manipulation to inspect AJAX and API endpoints, while detecting error handling flaws such as debug information leaks and missing security headers. Arachni complements this with framework-specific payloads, machine-learning-based false positive reduction, and out-of-band techniques for complex vulnerabilities like Server-Side Request Forgery (SSRF). A combined scoring system based on Likelihood and Impact Factors guides prioritization of vulnerabilities by evaluating their exploitability and potential technical and business consequences. This comprehensive approach ensures a thorough and accurate security evaluation, supporting effective risk management and improvements for the UMM digital library’s web login services.

## DAST Scanning Analysis

The results of the DAST scanning, summarized in Table 2, illustrate the distribution of identified vulnerabilities by severity risk across both OWASP ZAP and Arachni tools. OWASP ZAP detected a higher number of vulnerabilities overall, with 4 high-risk, 6 medium-risk, and 9 low-risk findings, whereas Arachni reported fewer issues, identifying 1 high-risk, 3 medium-risk, and 2 low-risk vulnerabilities. This disparity highlights the complementary nature of the tools, with ZAP providing more extensive coverage, particularly in lower-severity categories. The greater detection capability of ZAP can be attributed to its advanced fuzzing engine, extensive payload library, and real-time parameter manipulation, which enable it to probe a wider range of inputs and application behaviors. Additionally, ZAP’s active scanning features are highly effective at uncovering common misconfigurations and error-handling issues that often manifest as low- to medium-severity vulnerabilities.

Despite detecting fewer vulnerabilities, Arachni remains an essential component of the testing suite due to its specialized strengths. Its framework-specific payloads and machine-learning-driven false-positive reduction enhance its precision, particularly for complex vulnerabilities that may be missed by more generic scanners. Arachni’s passive detection methods and out-of-band techniques also allow it to identify subtle issues such as Server-Side Request Forgery (SSRF) and cookie misconfigurations with high accuracy. Therefore, while ZAP excels in breadth and active probing, Arachni contributes valuable depth and accuracy, making their combined use a robust strategy for comprehensive vulnerability assessment.

**TABLE 2.** Summary of DAST scanning finding based on severity risk.

|  |  |  |
| --- | --- | --- |
| **Risk** | **ZAP** | Arachni |
| High | 4 | 1 |
| Medium | 6 | 3 |
| Low | 9 | 2 |

**TABLE 3.** Mapping ZAP and Arachni finding based on OWASP category.

|  |  |  |
| --- | --- | --- |
| **OWASP category** | **ZAP** | Arachni |
| A01: Broken Access Control | 1 |  |
| A02: Cryptographic Failures | 2 |  |
| A05: Security Misconfiguration | 13 | 4 |
| A06: Vulnerable Components | 1 |  |
| A07: Identification Failures | 1 | 1 |
| A08: Software and Data Integrity Failures | 1 |  |

Table 3 consolidates the findings from both tools by mapping them against OWASP Top 10 categories, providing a comparative overview of vulnerability coverage. Notably, Security Misconfiguration (A05) emerges as the most frequently detected category, with OWASP ZAP identifying 13 instances and Arachni 4. This prevalence underscores a critical insight: a significant portion of the vulnerabilities discovered in the tested application stem from misconfigurations that, in many cases, could be mitigated through relatively straightforward server and application configuration adjustments. Issues such as missing security headers, improper cookie settings, exposed sensitive files, and inadequate access controls often do not require complex code changes but rather careful tuning of server settings, web server directives, and security policies.

The dominance of security misconfiguration as a vulnerability category highlights a persistent challenge in web application security, ensuring that default or legacy configurations are hardened to meet modern security standards. Many of these weaknesses can be addressed by implementing best practices such as enabling HTTP Strict Transport Security (HSTS), configuring Content Security Policy (CSP), setting secure cookie attributes (HttpOnly, SameSite), and restricting directory access. The fact that such issues remain widespread suggests gaps in deployment processes, insufficient security awareness, or lack of automated configuration management in the development and operational lifecycle.

## DAST Testing Based on WSTG Standard

Three main types of testing were conducted in this research, guided by the OWASP Web Security Testing Guide (WSTG) to ensure a focused and systematic assessment of critical vulnerabilities. The tests performed include Reflected Cross-Site Scripting (XSS), SQL Injection (SQLi), and manual error handling analysis, corresponding to WSTG-INPV-01, WSTG-INPV-05, and WSTG-ERRH-01 respectively, as summarized in Table 6. The primary focus of this research is the login page of the application, as it represents a critical attack surface and a common entry point for attackers seeking unauthorized access.

**TABLE 4.** Mapping proposed testing with WSTG category.

|  |  |  |
| --- | --- | --- |
| **WSTG category** | **Testing** | **Objective** |
| WSTG-INPV-01 | Reflected Cross Site Scripting | CSP header not set, Cross-Domain JavaScript Source File Inclution, dan X-Content-Type-Header Missing. |
| WSTG-INPV-05 | SQL Injection | PII disclosure dan CSP header not set. |
| WSTG-ERRH-01 | Improper Error Handling | Information Disclosure - Debug Error Messages |

The Reflected XSS test aims to identify vulnerabilities where untrusted input is immediately reflected in the user’s browser without proper sanitization, enabling attackers to execute malicious scripts that can lead to session hijacking, defacement, or redirection. The SQL Injection (SQLi) test detects flaws in handling user input within SQL queries, allowing attackers to manipulate databases or extract sensitive data. SQLi testing involved injecting payloads with special characters and SQL synta, such as tautologies and union queries into form inputs and URL parameters, with systematic fuzzing performed using OWASP ZAP. Manual error handling tests evaluated the application’s response to unexpected inputs or system errors, focusing on exposure of sensitive information like stack traces or database errors, aligning with WSTG-ERRH-01.

Analysis of the findings, as summarized in Table 5 Based on (a) ZAP and (b) Arachni, indicates a high Likelihood Factor for vulnerability discovery. This is primarily due to the ease with which automated tools like OWASP ZAP can identify sensitive user information—such as full names and email addresses—that ideally should remain protected. The accessibility of these vulnerabilities does not require advanced cybersecurity expertise, as the automated nature of the scanning tools simplifies detection. As a result, attackers with minimal skills could exploit these weaknesses, increasing the risk of successful attacks despite the presence of basic security controls. Conversely, the vulnerability assessment conducted with the Arachni tool, following the established methodology, yielded an overall average vulnerability rating classified as LOW. This suggests that while vulnerabilities are present, their potential impact or exploitability within the tested environment is relatively limited. A detailed summary of the scoring and specific results is provided in Table 6, offering further insights into the quantitative evaluation underpinning this conclusion.

**TABLE 5.** Likehood Factor and Impact Factor based on (a) ZAP and (b) Arachni.

1. ZAP result (b) Arachni result

A screenshot of a survey

Description automatically generated A screenshot of a survey

Description automatically generated

# conclusion

This research aims to assess the security of the University of Muhammadiyah Malang (UMM) public digital library platform, with a particular focus on its web login services. Vulnerability testing was conducted using Dynamic Application Security Testing (DAST) tools, specifically OWASP ZAP and Arachni. A detailed analysis was performed based on the OWASP Top 10 2021 framework, recognized as a leading security best practice. To reduce complexity, the study concentrated on Data Validation and Error Handling failures, narrowing the scope from ten to four key observations.

Initial scans identified 19 risks using ZAP and 5 risks using Arachni. Further analysis leveraged the OWASP Web Security Testing Guide (WSTG), followed by in-depth testing targeting Cross-Site Scripting (XSS), SQL Injection (SQLi), and manual error handling. Based on calculations of Likelihood and Impact Factors, the assessed website presents a moderate risk level according to ZAP, while Arachni’s evaluation indicates a low risk level.

# References

[1] Michelle Moore, “No TiTop Cybersecurity Threats to Watch in 2025tle.” [Online]. Available: https://onlinedegrees.sandiego.edu/faculty/michelle-moore/

[2] T. M. R. Centre, “The Trend 2025 Cyber Risk Index,” 2025, [Online]. Available: https://www.trendmicro.com/vinfo/us/security/news/threat-landscape/trend-2025-cyber-risk-report

[3] “20 Recent Cyber Attacks & What They Tell Us About the Future of Cybersecurity.” Accessed: Aug. 08, 2025. [Online]. Available: https://secureframe.com/blog/recent-cyber-attacks

[4] “OWASP Top Ten | OWASP Foundation.” Accessed: Aug. 08, 2025. [Online]. Available: https://owasp.org/www-project-top-ten/

[5] A. Dann, H. Plate, B. Hermann, S. E. Ponta, and E. Bodden, “Identifying Challenges for OSS Vulnerability Scanners-A Study & Test Suite,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3613–3625, 2022, doi: 10.1109/TSE.2021.3101739.

[6] F. O. Sonmez and B. G. Kilic, “Holistic Web Application Security Visualization for Multi-Project and Multi-Phase Dynamic Application Security Test Results,” *IEEE Access*, vol. 9, pp. 25858–25884, 2021, doi: 10.1109/ACCESS.2021.3057044.

[7] A. Jakobsson and I. Häggström, “Study of the techniques used by OWASP ZAP for analysis of vulnerabilities in web applications,” p. 69, 2022, [Online]. Available: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1675227&dswid=-4307

[8] A. M. Irzan and E. Sulistiyani, “Owasp Zap vs Arachni: Which One is Better in Vulnerability Assesment?,” *2024 9th Int. Conf. Informatics Comput. ICIC 2024*, pp. 1–6, 2024, doi: 10.1109/ICIC64337.2024.10956935.

[9] A. Wijayanto, E. Utami, and A. B. Prasetio, “Analysis of Vulnerability Webserver Office Management of Information and Documentation Diskominfo using OWASP Scanner,” *2020 2nd Int. Conf. Cybern. Intell. Syst. ICORIS 2020*, pp. 8–12, 2020, doi: 10.1109/ICORIS50180.2020.9320833.

[10] N. Anantharaman and B. Wukkadada, “Identifying the Usage of Known Vulnerabilities Components Based on OWASP A9,” *2020 Int. Conf. Emerg. Smart Comput. Informatics, ESCI 2020*, pp. 88–91, 2020, doi: 10.1109/ESCI48226.2020.9167645.

[11] D. F. Priambodo, G. S. Ajie, H. A. Rahman, A. C. F. Nugraha, A. Rachmawati, and M. R. Avianti, “Mobile Health Application Security Assesment Based on OWASP Top 10 Mobile Vulnerabilities,” *2022 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2022 - Proc.*, pp. 25–29, 2022, doi: 10.1109/ICITSI56531.2022.9970949.

[12] V. N. Nanisura Damanik and S. U. Sunaringtyas, “Secure code recommendation based on code review result using owasp code review guide,” *2020 Int. Work. Big Data Inf. Secur. IWBIS 2020*, pp. 153–157, 2020, doi: 10.1109/IWBIS50925.2020.9255559.

[13] S. Alazmi and D. C. De Leon, “Customizing OWASP ZAP: A Proven Method for Detecting SQL Injection Vulnerabilities,” *Proc. - 2023 IEEE 9th Int. Conf. Big Data Secur. Cloud, IEEE Int. Conf. High Perform. Smart Comput. IEEE Int. Conf. Intell. Data Secur. BigDataSecurity-HPSC-IDS 2023*, pp. 102–106, 2023, doi: 10.1109/BigDataSecurity-HPSC-IDS58521.2023.00028.

[14] A. Choiriyah and N. Qomariasih, “Security Analysis on Websites Belonging to the Health Service Districts in Indonesia Based on the Open Web Application Security Project (OWASP) Top 10 2021,” *Proceeding - Int. Conf. Inf. Technol. Comput. 2023, ICITCOM 2023*, pp. 267–272, 2023, doi: 10.1109/ICITCOM60176.2023.10442816.

[15] J. Li and H. Li, “Evolution of Application Security based on OWASP Top 10 and CWE/SANS Top 25 with Predictions for the 2025 OWASP Top 10,” *Proc. 8th Int. Conf. Inven. Comput. Technol. ICICT 2025*, pp. 1178–1183, 2025, doi: 10.1109/ICICT64420.2025.11004742.

[16] S. and S. Pascoe, C. , Quinn, “The NIST Cybersecurity,” p. 32, 2024, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.29.pdf

[17] B. Al-Sada, A. Sadighian, and G. Oligeri, “MITRE ATT&CK: State of the Art and Way Forward,” *ACM Comput. Surv.*, vol. 57, no. 1, 2024, doi: 10.1145/3687300.