WEB SECURITY ANALYSIS AND TESTING USING OWASP ZAP PENETRATION TESTING CASE STUDY OF WIDYA DHARMA UNIVERSITY WEBSITE PONTIANAK

Theresia Elsa Deshinta[1], Thommy Willay[2]
[1,2]Universitas Widya Dharma Pontianak, Indonesia
Email: 22412844_theresia_deshinta@widyadharma.ac.id

Abstract

As web digital technology advances rapidly, threats to web applications such as hacking, data leaks, and exploitation of web application vulnerabilities are becoming increasingly complex and diverse. Unaddressed security vulnerabilities may result in critical information disclosure, unauthorized data manipulation, and disruption to system operations. This study employs penetration testing with the latest OWASP ZAP version 2.17.0 as the primary security testing tool to perform a systematic vulnerability assessment on the official website of Universitas Widya Dharma Pontianak, with the OWASP Top 10 2025 framework utilized as the benchmark for vulnerability classification and categorization. This study is aim to identify security vulnerabilities that's undetected on the website, which can open up opportunities for attackers to attack the university's official website, as well to improve the security of the university website. The results show that there are 17 security vulnerabilities, comprising one high category vulnerability, six medium category vulnerabilities, six low category vulnerabilities, and four informational category vulnerabilities. The most frequently found vulnerability category identified was A02 Security Misconfiguration. This study concludes that the OWASP ZAP tool is effective in identifying previously undetected security vulnerabilities, while the OWASP Top 10 2025 framework makes it easier to categorize the security vulnerabilities found on the website.

Keywords: *Website, Security, Penetration Testing, OWASP, Vulnerability.*

A. INTRODUCTION

Digital technology and web-based information continue to develop rapidly. Web-based information systems are becoming a vital link, facilitating access and use of information. Many organizations use web applications, including higher education institutions in Indonesia, such as Widya Dharma University in Pontianak (Putri et al., 2025). This web-based information system functions not only as an important source of information for students, lecturers, and administrative staff, but also as an important platform for managing student data, class schedules, course registration, and other information related to academic activities (Ramadhan & Ilmananda, 2024). With an academic information system, students can easily and quickly access the information they need, making the academic process more effective (Eko setiawan & Fachri, 2025).

However, as digital technology develops very rapidly, threats to web applications are also increasing with various types of threats, such as hacking, data leaks, and exploitation of web application vulnerabilities (Imtias et al., 2025; Edy Listartha et al., 2022). If those security threats are not handled properly, they can result in leaks of important information, data changes, and even disruptions to system operations. (Ni'am & Tulodo, 2025). There are several factors that can cause these threats to occur, such as errors in writing program code and misconfiguration (Aryanti et al., 2021).

Therefore, based on the existing problems, this study analyzes the security of the web-based information system of Widya Dharma University Pontianak using penetration testing, as

well as the OWASP Top 10 2025 and OWASP ZAP frameworks as tools for conducting security scanning and analysis (Muhammad Amirul Mu'min et al., 2025; Elfatiha et al., 2024). OWASP is renowned for its contribution in identifying the most critical vulnerabilities in the field of web application security through the OWASP Top 10 list and also providing solutions for each vulnerability found (Li & Li, 2025; Darwis et al., 2022).

A website is an alternative for presenting information on the internet in the form of text, images, video, sound, or interactively, either statically or dynamically, and has the advantage of connecting links between one document and another (Hidayatulloh & Saptadiaji, 2021; Alfarizi1 et al., 2022).

Penetration testing is an approach that focuses on identifying security vulnerabilities and providing systematic remediation solutions before the vulnerabilities can be exploited (Pahlawansah et al., 2025). A security gap in a website is called a vulnerability (Fandier Saragih et al., 2023).

OWASP (Open Web Application Security Project) Top 10 framework is a method for testing web-based systems that provides information about security vulnerabilities and suggested improvements that can be implemented in web-based services. This method covers ten key vulnerabilities that can compromise a website's security. These vulnerabilities are constantly evolving as technology advances (Nurelasari et al., 2024; Hermanto & Haeruddin, 2022; Izumi & Widiasari, 2022).

OWASP Zed Attack Proxy (ZAP), developed by the OWASP organization, is one of the most popular and well-maintained open-source web security testing tools. The OWASP method can be used as a standard in assessing the effectiveness of web application security (Charly et al., 2022). OWASP ZAP offers several features that allow users to perform automated scans for security vulnerabilities in web applications and provide detailed reports that help in understanding and fixing those vulnerabilities (Putra et al., 2024).

Several previous studies have proven that the OWASP Top 10 framework and the OWASP Zap tool successfully detect undetected security vulnerabilities on a website. Research by Rofiq et al., 2025 used the OWASP ZAP tool to assist in an in-depth evaluation of the SIAKAD website of the Faculty of Medicine, Surabaya State University. This study concluded that OWASP ZAP is an effective tool for auditing website security while providing a clear roadmap for improving and enhancing an institution's digital security posture. Research Yudiana et al., 2021 also succeeded in finding vulnerabilities in the website-based e-office information system at STMIK ROSMA using OWASP ZAP, there were 13 vulnerabilities successfully found in the web-based e-office information system.

Then there is also research conducted by Arief et al., 2025 which used the OWASP Top 10 framework to analyze subdomain vulnerabilities in the SIMPELMAS web-based information system. The study successfully identified vulnerabilities in the SIMPELMAS website and compared them to ten OWASP vulnerability points. Two vulnerabilities were discovered: Security Misconfiguration and Identification and Authentication Failures. Furthermore, research by Muttaqin et al., 2025 used the OWASP Top 10 framework to classify security vulnerabilities in the school quality management system's login feature. The classifications were divided into several categories: Identification and Authentication Failures, Security Misconfiguration, and Vulnerable Components.

This study uses the OWASP Top 10 2025 framework because it is the latest version of the OWASP Top 10 framework and uses the OWASP ZAP tool to find security gaps or vulnerabilities in the web-based information system of Widya Dharma University Pontianak. The security vulnerabilities found will be grouped based on the OWASP Top 10 category. It is hoped that this study can help in improving the quality of security on the university website by finding undetected security vulnerabilities.

B. METHOD

This study analyzes the vulnerability of web-based information systems using penetration testing with OWASP ZAP on the research object, namely the website of Widya Dharma University Pontianak and uses the OWASP Top 10 framework as a standard in analyzing and grouping vulnerabilities from the official campus website (Dwi Cahyani et al., 2022). The research flow consists of four steps, that is determination research object, data collection, website vulnerability testing, and lastly test result and conclusion, as shown in Figure 1.

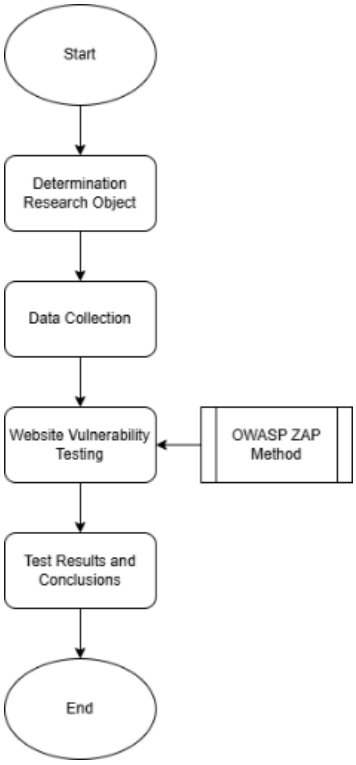

Figure 1. Research Flow

The following is an explanation of each research flow in Figure 1:

1. Determination Research Object

The first stage of the research was to determine the research object. The object used in the research was the website of Widya Dharma University Pontianak at https://widyadharma.ac.id/. This website is the official website of Widya Dharma University Pontianak and can be accessed by anyone to view various academic information presented by the university.

2. Data Collection

The next step is to collect data by searching for various journal sources on the same topic as the research to strengthen our basic understanding of the research topic. The journal sources used were those published no more than five years ago, between 2021 and 2025.

3. Website Vulnerability Testing

The security level of the Widya Dharma University Pontianak website will be analyzed using the latest version of the OWASP ZAP tool, namely 2.17.0, which was updated on December 15, 2025. The main page of the OWASP ZAP tool is displayed as in Figure 2.

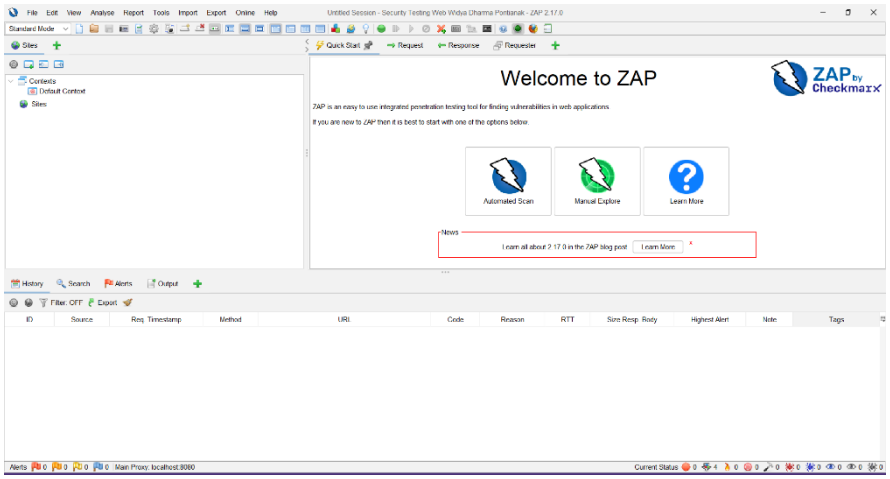

Figure 2. Display of the OWASP ZAP User Interface Tools

The OWASP ZAP tool includes an Automated Scan feature used to test website security and identify previously undetected vulnerabilities. This feature ensures a thorough vulnerability identification process without the need for potentially harmful exploits, thus maintaining web security. The display when entering the target link to the website to be tested is shown in Figure 3.

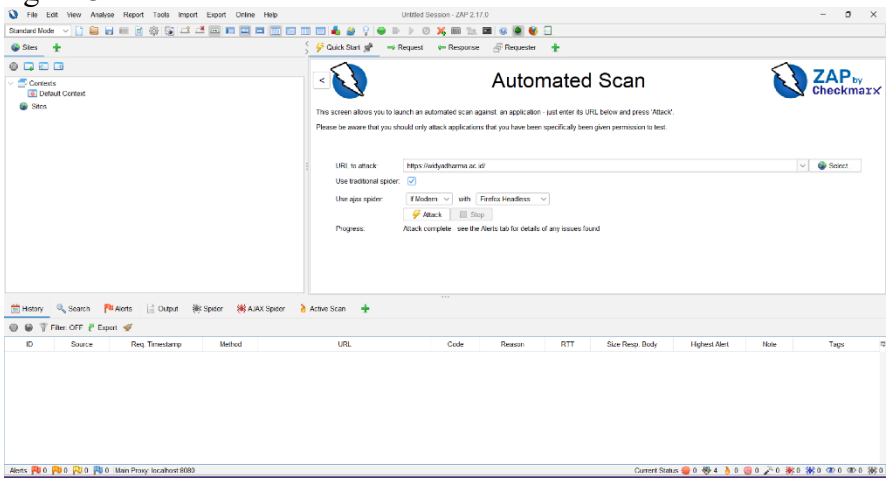

Figure 3. Target Link

After entering the target link, the next step is to press the attack button to scan the targeted website. The scanning process in OWASP ZAP begins with SpiderScan to collect URLs, then continues with Active Scan to test for vulnerabilities, and finally generates alerts as a report of security findings.

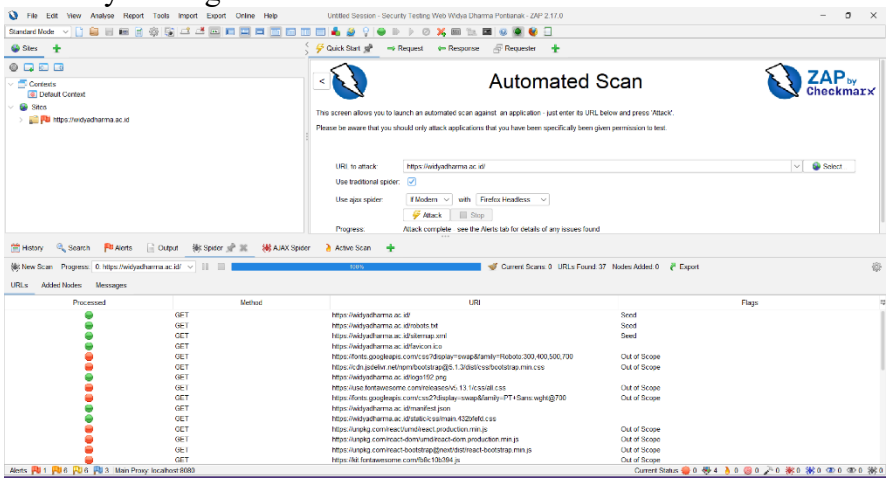

Figure 4. Spider Scan

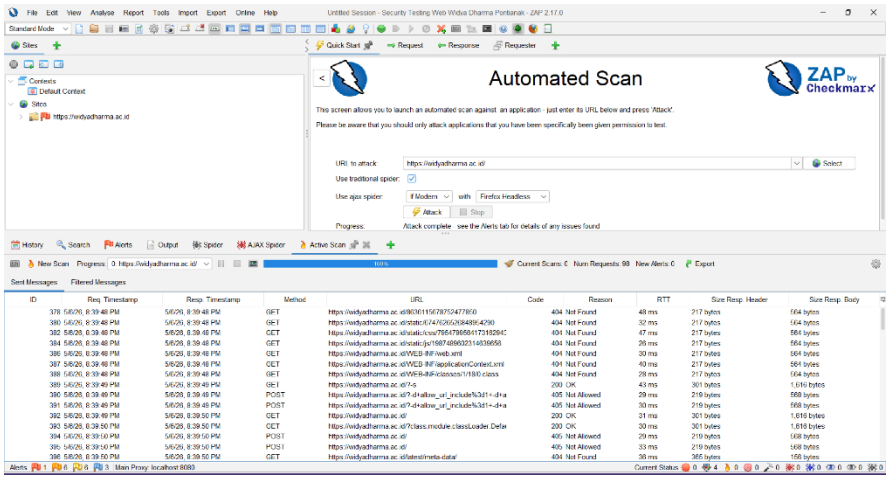

Figure 5. Active Scan

4. Test Results and Conclusions

The discovered security vulnerabilities will be classified based on their root cause and impact, and then grouped based on the OWASP Top 10 2025 framework. Conclusions from the testing of the Widya Dharma University Pontianak website will also be presented. The following are ten website security risks that serve as benchmarks for categorizing security vulnerabilities and are continuously updated to the latest version, the OWASP Top 10 2025:

a. A01 – Broken Access Control: Users have access control for functions or data they should not have access to.
b. A02 – Security Misconfiguration: If an application's security configuration is not set up correctly, it can create a risk of security vulnerabilities.
c. A03 – Software Supply Chain Failure: Vulnerabilities or changes to third-party components that are unsafe or unverified, which can open security gaps in the system.
d. A04 - Cryptographic Failures: Vulnerabilities that occur due to errors in the use or implementation of encryption, resulting in sensitive information being intercepted or stolen by irresponsible parties.
e. A05 – Injection: A vulnerability in an application that allows malicious input, such as code or commands, from an untrusted user to be sent to the application and causes the application to execute part of that input as a command.
f. A06 – Insecure Design: Security issues arise because a system was designed without security in mind from the outset. As a result, the system has security vulnerabilities even when implemented correctly.
g. A07 - Authentication Failures: When an attacker manages to trick the system into recognizing an invalid user or mistaking it for a valid user.
h. A08 - Software/Data Integrity Failures: Software and data integrity failures occur when code and infrastructure fail to protect themselves from invalid or untrustworthy code or data. This means the code or data is perceived as trusted and valid when it is not.
i. A09 - Logging and Alerting Failures: Lack of security logging and monitoring makes attacks and breaches difficult to detect and makes it difficult to respond quickly to ongoing security incidents.
j. A10 - Mishandling of Exceptional Conditions: External condition mishandling is common in software. This occurs when a program fails to prevent, detect, and respond to unusual and unexpected situations. As a result, crashes, unexpected behavior, and vulnerabilities can occur.

C. RESULTS AND DISCUSSION

This study used penetration testing with OWASP ZAP to conduct security tests on the Widya Dharma University Pontianak website. The homepage of the website is shown in Figures 6 and 7.

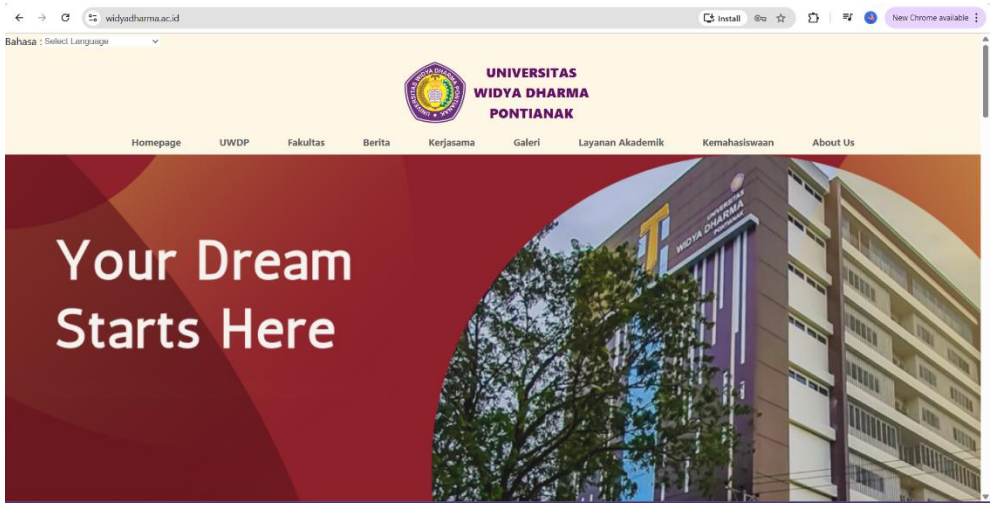

Figure 6. Website Homepage 1

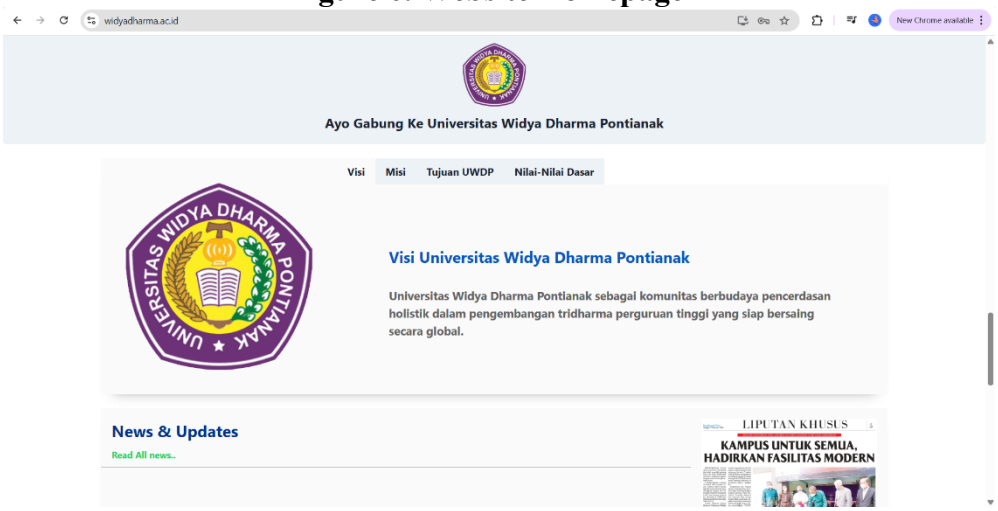

Figure 7. Website Homepage 2

The feature used to perform security testing is Automated Scan, which can detect security vulnerabilities in a website and provide alerts for any identified flaws. The resulting alerts can be seen in Figure 8.

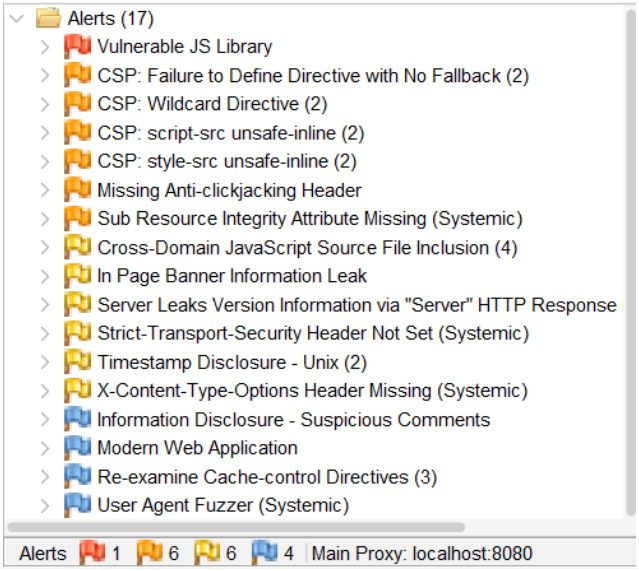

Figure 8. Alert Results

Based on Figure 8, 17 alerts were found on the Widya Dharma University Pontianak website after conducting security testing. The alerts are divided into four risk levels, as shown in Table 1, along with the number of risks.

Tabel 1. Risk Level

Color	Level	Number of Risk
Red	High Risk	1
Orange	Medium Risk	6
Yellow	Low Risk	6
Blue	Informational	4

Table 1 shows one high-severity vulnerability, six medium-severity vulnerabilities, six low-severity vulnerabilities, and four informational-level vulnerabilities. Each discovered vulnerability will be categorized based on the OWASP Top 10 2025 framework, as shown in Table 2.

Tabel 2. Security Vulnerability Classification

Code	Security Risk	Alerts	Level
A01	Broken Access Control	None	None
A02	Security Misconfiguration	CSP: Failure to Define Directive with No Fallback CSP: Wildcard Directive CSP: script-src unsafe-inline CSP: style-src unsafe-inline Missing Anti-clickjacking Header Server Leaks Version Information via "Server" HTTP Response Header Field Strict-Transport-Security Header Not Set	Low to Medium

		X-Content-Type-Options Header Missing Re-examine Cache-control Directives	
A03	Software Supply Chain Failure	Vulnerable JS Library Cross-Domain JavaScript Source File Inclusion	Medium to High
A04	Cryptographic Failures	None	None
A05	Injection	None	None
A06	Insecure Design	Modern Web Application	Informational
A07	Authentication Failures	None	None
A08	Software/Data Integrity Failures	Sub Resource Integrity Attribute Missing	Medium
A09	Logging and Alerting Failures	In Page Banner Information Leak Information Disclosure - Suspicious Comments Timestamp Disclosure - Unix	Low
A10	Mishandling of Exceptional Conditions	User Agent Fuzzer	Informational

Below is a brief explanation of why some alerts are placed in certain categories and their risk impact:

1. CSP (Content Security Policy) and Missing Headers Alerts: This security vulnerability is classified in category A02 because if the CSP and headers are not configured correctly, the website does not have basic rules/guidelines regarding content or behavior on the website, making it easier for attackers to carry out attacks.
2. Vulnerable JS Library: This vulnerability is classified as A03 because it is a third-party component used on a website. If the library is outdated or untrusted, the vulnerability could be exploited by attackers, as all identified vulnerabilities in previous versions are publicly recorded through the Common Vulnerabilities and Exposures (CVE) database.
3. Sub Resource Integrity Attribute Missing: SRI is a security feature that ensures that external files loaded on a website have not been manipulated by an attacker. Without resource attribute integrity, a web system cannot detect whether an external file has been modified by an attacker.
4. In Page Banner Information Leak dan Timestamp Disclosure – Unix: This security vulnerability is classified in category A09 because the web system cannot control what information is displayed, so internal web information, such as framework versions and other technical details of the system are also accidentally displayed, which can potentially be exploited by attackers.

The security vulnerabilities that were successfully identified have several recommended solutions that can be implemented, as shown in Table 3.

Tabel 3. Recommended Solutions

Code	Risk	Solution
A02	Security Misconfiguration	A safe installation process must be implemented, including: - Task to review and update configurations to comply with all security notes, updates and patches.

		- Automated process to verify the effectiveness of configurations and settings across all environments. - Proactively add central configuration to prevent excessive error messages.
A03	Software Supply Chain Failure	- Update old library to the latest version. - Ensure JavaScript source files are only loaded from trusted sources, and that those sources cannot be controlled by the end user of the application.
A08	Software/Data Integrity Failures	- Ensure there is a review process for code and configuration changes to minimize the possibility of malicious code or configuration entering the software workflow. - Ensure software workflows have proper separation, configuration, and access control to ensure the integrity of the code flowing through the build and deployment process.
A09	Logging and Alerting Failures	- Ensure every part of the application that contains security controls is logged, whether successful or unsuccessful.

D. CONCLUSION

The results of security testing on the Widya Dharma University Pontianak website showed that 17 vulnerabilities were successfully identified using penetration testing with OWASP ZAP and categorized into the OWASP Top 10 2025 framework. Overall, the vulnerabilities found were one high-level vulnerability, six medium-level vulnerabilities, six low-level vulnerabilities, and four informational-level vulnerabilities. The most frequently found vulnerability was in category A02 Security Misconfiguration.

The study shows that the official website of Widya Dharma University Pontianak still needs to be updated and its security improved, to prevent potential attacks from irresponsible parties. It can be concluded that security testing using penetration testing with OWASP ZAP successfully identified previously undetected security vulnerabilities, which could potentially open up gaps for attackers to carry out attacks. The study also shows that the OWASP Top 10 2025 framework makes it easy to group all found vulnerabilities into one specific category, thus facilitating the search for needed solutions. Further website security research is expected to utilize another feature in OWASP ZAP, namely Ajax Spider Scan, which can produce a wider testing scope, so that security holes that were not detected in this study can be found in future research.

REFERENCES

Alfarizi1, M., K, M. N., H, M. A., & Ashari4, I. F. (2022). Vulnerability Analysis and Proven on the neonime.co Website Using OWASP ZAP 4 and XSpear. *JTKSI (Jurnal Teknologi Komputer dan Sistem Informasi)*, *5*(2), 75–81. https://doi.org/doi:10.56327/jtksi.v5i2.1130

Arief, M. I., Anwar, D. S., & Supriatman, A. (2025). Analisis Kerentanan Website Melalui Pendekatan Penetration Testing Berdasarkan Standar Owasp Top 10 Studi Kasus

Simpelmas Universitas XYZ. *JEIS: Jurnal Elektro dan Informatika Swadharma*, *5*(2), 93–104. https://doi.org/10.56486/jeis.vol5no2.798

Aryanti, D., Nurholis, & Nashar Utamajaya, J. (2021). Analisis Kerentanan Keamanan Website Menggunakan Metode Owasp (Open Web Application Security Project) Pada Dinas Tenaga Kerja. *Jurnal Syntax Fusion*, *1*(03), 15–25. https://doi.org/10.54543/fusion.v1i03.53

Charly, P., Diatmika, K. E., Prayoga, I. M. P., & Listartha, I. M. E. (2022). Pendeteksian Keamanan Website SMA Greenschool Menggunakan Metode Owasp dengan Pengujian XSS. *Format: Jurnal Ilmiah Teknik Informatika*, *11*(1), 77. https://doi.org/10.22441/10.22441/format.2022.v11.i1.008

Darwis, E., Junaedy, & Musdar, I. A. (2022). Analisis Kerentanan Website Renovaction Menggunakan Rangkaian Security Tools Project Berdasarkan Framework Owasp. *KHARISMA Tech*, *17*(1), 1–15. https://doi.org/10.55645/kharismatech.v17i1.170

Dwi Cahyani, D., Windy Puspita Dewi, L. P., Rama Suryadi, K. D., & Edy Listartha, I. M. (2022). Analisis Kerentanan Website SMP Negeri 3 Semarapura Menggunakan Metode Pengujian Rate Limiting dan OWASP. *INSERT: Information System and Emerging Technology Journal*, *2*(2), 106–112. https://doi.org/10.23887/insert.v2i2.42936

Edy Listartha, I. M., Premana Mitha, I. M. A., Aditya Arta, M. W., & Yuda Arimika, I. Km. W. (2022). Analisis Kerentanan Website SMA Negeri 2 Amlapura Menggunakan Metode OWASP (Open Web Application Security Project). *SIMKOM*, *7*(1), 23–27. https://doi.org/10.51717/simkom.v7i1.63

Eko setiawan, & Fachri, F. (2025). Pengujian dan Mitigasi Kerentanan Website Sistem Informasi Akademik Universitas Ma'arif Nahdlatul Ulama Kebumen dengan OWASP ZAP. *Cyber Security dan Forensik Digital*, *8*(1), 25–33. https://doi.org/10.14421/csecurity.2025.8.1.5190

Fandier Saragih, N., Reinhard Tamalawe, & Indra M Sarkis. (2023). Analisis dan Implementasi Secure Code Pada Pengembangan Sistem Keamanan Website Fikom-Methodist.Com Menggunakan Penetration Testing dan Owasp Zap. *Jurnal TIMES*, *12*(1), 28–39. https://doi.org/10.51351/jtm.12.1.2023690

Hermanto, H., & Haeruddin, H. (2022). Peningkatan Sistem Keamanan Website Menggunakan Metode OWASP. *Jurnal Ilmu Komputer dan Bisnis*, *13*(1), 94–104. https://doi.org/10.47927/jikb.v13i1.277

Hidayatulloh, S., & Saptadiaji, D. (2021). Penetration Testing pada Website Universitas ARS Menggunakan Open Web Application Security Project (OWASP). *Jurnal Algoritma*, *18*(1), 77–86. https://doi.org/10.33364/algoritma/v.18-1.827

Imtias, M. B., Umam, K., Mustofa, H., & Subowo, M. H. (2025). Comparative Analysis of Penetration Testing Frameworks: OWASP, PTES, and NIST SP 800-115 for Detecting Web Application Vulnerabilities. *Journal of Applied Informatics and Computing*, *9*(6), 3689–3696. https://doi.org/10.30871/jaic.v9i6.9846

Izumi, A. C., & Widiasari, I. R. (2022). "Siasat" UKSW (Universitas Kristen Satya Wacana) Website Security Analysis Using Owasp (Open Web Application Security Project). *Jurnal Teknik Informatika (JUTIF)*, *3*(3), 763–770. https://doi.org/https://doi.org/10.20884/1.jutif.2022.3.3.X

Li, J., & Li, H. (2025). Evolution of Application Security based on OWASP Top 10 and CWE/SANS Top 25 with Predictions for the 2025 OWASP Top 10. *2025 International Conference on Inventive Computation Technologies (ICICT)*, 1178–1183. https://doi.org/10.1109/ICICT64420.2025.11004742

Muhammad Amirul Mu'min, Yana Safitri, Galih Pramuja Inngam Fanani, Setiawan Ardi Wijaya, & Novi Tristanti. (2025). Security Analysis of XYZ Website Using OWASP

Zap Tools. *Journix: Journal of Informatics and Computing*, *1*(1), 10–20. https://doi.org/10.63866/journix.v1i1.1

Muttaqin, M. F., Ferdiansyah, D., Majapahit, S. A., & Rijayanti, R. (2025). Analisis Keamanan Fitur Login Aplikasi: Studi Kasus Sistem Manajemen Mutu Sekolah OWASP Top 10 dengan OWASP ZAP. *Pasinformatik*, *4*(2).

Ni'am, M. F. K., & Tulodo, R. P. (2025). Analisis Kerentanan Website Menggunakan Metode Penetration Testing Dengan Standar Keamanan OWASP Top 10:2021 Studi Kasus Website Sistem Informasi Manajemen Laboratorium Dinas Kesehatan Kabupaten Tegal. *Jurnal Rekayasa Teknik dan Ilmu Komputer (Jurektik)*, *6*(2), 762–773.

Nurelasari, E., Gumilang, D., & Farabi, A. (2024). Analisis Keamanan Sistem Website Menggunakan Metode Open Web Application Security Project (Owasp) pada Simantep.Id. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(3), 3049–3054.

Pahlawansah, H., Basmar, Muh. F., & Yusuf, M. (2025). Analisis Kerentanan Website SMK Muhammadiyah 2 Bontoala Makassar Menggunakan Metode OWASP (Open Web Application Security Project). *BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer*, *6*(2), 92–100. https://doi.org/10.37148/bios.v6i2.180

Putra, F. P. E., Ubaidi, U., Hamzah, A., Pramadi, W. A., & Nuraini, A. (2024). Systematic Literature Review: Security Gap Detection on Websites Using Owasp Zap. *Brilliance: Research of Artificial Intelligence*, *4*(1), 348–355. https://doi.org/10.47709/brilliance.v4i1.4227

Putri, V. R., Sobandi, A., & Santoso, B. (2025). Analysis of Information System Security Using OWASP ZAP on a Web-Based Electronic Archiving System. *Telematika*, *22*(3), 28–42. https://doi.org/10.31315/telematika.v22i3.14241

Ramadhan, M. F. A., & Ilmananda, A. S. (2024). Analisis Ancaman Keamanan Pada Sistem Informasi Akademik Kampus Menggunakan Metode Owasp Zap. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(4), 7985–7991.

Rofiq, F. A., Mahbubi, M., Kalokajaya, D. A., Yunus, A. F., Permana, R. S., Taufiq, A. M., Prismana, I. G. L. P. E., & Habibi, M. W. (2025). Pengujian Kerentanan dan Mitigasi Website SIAKAD Fakultas Kedokteran UNESA dengan OWASP ZAP. *RIGGS: Journal of Artificial Intelligence and Digital Business*, *4*(4), 2671–2679. https://doi.org/10.31004/riggs.v4i4.3686

Umar, R., Riadi, I., & Elfatiha, M. I. A. (2024). Security analysis of web-based academic information system using owasp framework. *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 4(3), 277-288. https://doi.org/10.22219/kinetik.v9i4.2015

Yudiana, Y., Elanda, A., & Buana, R. L. (2021). Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10. *CESS (Journal of Computer Engineering, System and Science)*, *6*(2), 185. https://doi.org/10.24114/cess.v6i2.24777