Multidisciplinary Output Research For Actual and International Issue (MORFAI Journal) ISSN (e): 2808-6635
Volumes 5 No. 2 (2025)

SECURITY ANALYSIS OF A WEB-BASED ACADEMIC INFORMATION SYSTEM AT XYZ UNIVERSITY USING VULNERABILITY ASSESSMENT TECHNIQUES

Imun Faizal[1], Khairunnisak Nur Isnaini[2*], Mohammad Imron[3]
[1,2,3] Informatics Departement , Amikom Purwokerto University, Indonesia
E-mail: imunfaizal99@gmail.com[1], nisak@amikompurwokerto.ac.id[2*], imron@amikompurwokerto.ac.id[3]

Received	: 29 June 2025	Published	: 09 August 2025
Revised	: 10 July 2025	DOI	: https://doi.org/10.54443/morfai.v5i2.3791
Accepted	: 25 July 2025	Link Publish	: https://radjapublika.com/index.php/MORFAI/article/view/3791

Abstract

This study aims to evaluate the security of a web-based academic information system at XYZ University using vulnerability assessment techniques. The system plays a vital role in supporting academic and administrative processes but stores sensitive data that makes it vulnerable to cyber threats. The research method consists of four main stages: defining the assessment scope, conducting vulnerability scanning using OWASP ZAP, analyzing the identified vulnerabilities based on type and severity using the OWASP Top Ten standard, and reporting the findings along with mitigation recommendations. The scanning results revealed 14 types of vulnerabilities, including the absence of anti-CSRF tokens, misconfigured security headers, and the use of outdated or vulnerable JavaScript libraries. Although no critical vulnerabilities were found, the identified issues still pose significant risks if left unaddressed. This study highlights the importance of regular security audits and the implementation of standardized web security practices. The proposed mitigation strategies are expected to enhance the overall cybersecurity posture of academic information systems and serve as a reference for developing more robust information security policies in higher education institutions.

Keywords: ***cybersecurity, vulnerability assessment, academic information system, OWASP ZAP, web application security***

INTRODUCTION

The advancement of information technology has had a significant impact across various sectors, including higher education. Web-based applications have become the primary choice for digital transformation due to their flexibility, efficiency, and cross-platform accessibility (Hamidy & Yasin, 2024). One of the most critical implementations in this context is the use of web-based academic information systems, which many universities have adopted to manage academic and administrative processes digitally. Academic information systems play a vital role in providing integrated services to students, lecturers, and academic staff, including course registration, grade management, class schedules, and institutional communication (Zulfa et al., 2025)(Saadah et al., 2022). The widespread use of such systems offers numerous benefits, such as improving efficiency, reducing manual bureaucracy, and facilitating data-driven decision-making (Dellia et al., 2025). However, this high dependency on web systems also brings increasing security risks, especially concerning the protection of sensitive institutional data.

The education sector ranks among the most vulnerable to cyberattacks in the country. Academic systems store a wide range of sensitive information, such as personal student data, academic transcripts, course records, and administrative archives, which can be valuable targets for malicious exploitation. Unfortunately, many higher education institutions still lack regular security audits, proper risk assessments, and adequate defense mechanisms (Budiyanto & Mabruri, 2025). To address these risks, regular security evaluation is essential. One widely accepted and applied approach is vulnerability assessment, which is the process of identifying, analyzing, and prioritizing potential security weaknesses before they can be exploited (Syaifudin et al., 2025). Commonly used tools in this method include OWASP ZAP for detecting web application vulnerabilities such as *Cross-Site Scripting (XSS)* and *SQL Injection*, Nmap for network and open port scanning, and Nikto for web server configuration analysis (Ending Narhudin et al., 2024; Rahman & Fatkhur Razak, 2024).

Numerous studies have demonstrated the effectiveness of vulnerability assessment techniques in uncovering weaknesses in academic systems at various universities. For instance, (Pramuja Inngam Fanani et al., 2025) revealed that academic portals lacking proper input filtering and regular system updates were highly susceptible to malicious script injection. However, only a limited number of studies have specifically analyzed the security of web-based academic information systems in Indonesian universities with a systematic and technical approach (Ariyadi et al., 2023).Given this context, the objective of this study is to conduct a security analysis of the web-based academic information system at XYZ University using vulnerability assessment techniques. This research employs tools such as OWASP ZAP to identify potential security flaws and provide mitigation recommendations based on the findings. The outcomes of this study are expected to contribute to the development of stronger information security policies in higher education institutions and serve as a reference for future research in the field of academic web application security.

LITERATURE REVIEW

The use of web-based academic information systems has become a primary necessity in higher education institutions to support both academic and administrative activities. These systems enable the integration of services such as course registration, grade management, class scheduling, and communication among members of the academic community(Melani, 2023). However, this digital transformation also brings significant challenges, particularly in terms of data security. Research conducted by (Supartini & Parenreng, 2023) highlights that web-based systems are highly vulnerable to attacks such as Cross-Site Scripting (XSS), SQL Injection, and insecure session management. Therefore, vulnerability assessment is employed as a standard method to thoroughly evaluate the security of web applications. Furthermore, a study by (Marpaung, 2025) reveals that many universities have not yet implemented adequate protections, such as input filtering and regular system updates. This is concerning, as the education sector is one of the primary targets of cyberattacks in Indonesia due to its storage of sensitive data such as student biodata, academic transcripts, and administrative records. (Mustofa et al., 2025) emphasize the importance of a systematic technical approach in evaluating academic systems. However, there is still a lack of research that specifically focuses on the security evaluation of academic information systems within Indonesian universities using a structured penetration testing method. Based on these various studies, it can be concluded that there remains a lack of comprehensive technical protection in the implementation of academic information systems across many universities in Indonesia. Therefore, this research aims to address that gap by conducting a direct security analysis of the web-based academic system at XYZ University and providing relevant technical mitigation recommendations based on the results of the vulnerability assessment.

METHOD

This study employs the Vulnerability Assessment method, which is a systematic process used to identify, analyze, and evaluate vulnerabilities within a web-based system (Hasibuan & Handoko, 2023). The primary objective of this approach is to detect potential security weaknesses before they can be exploited by unauthorized parties, while also providing a foundation for proactive security improvements. The object of this research is the web-based Academic Information System used at XYZ University, which serves as the core platform for managing student and faculty academic data. The assessment process consists of four structured stages as illustrated in Image 1.

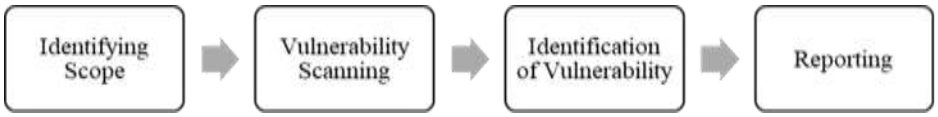

Image 1: Research Stages

1. Identifying Scope
   The initial stage of this research is to determine the scope of the study to be conducted. In this context, the researcher uses the web-based Academic Information System implemented at XYZ University as the object of research.
2. Vulnerability Scanning
   Vulnerability scanning is the process carried out to identify and discover weaknesses or vulnerabilities within a system [13]. In this stage, the researcher uses the OWASP ZAP tool to perform the scanning.
3. Identification of Vulnerabilities
   After the scanning process is completed, the researcher analyzes the identified vulnerabilities based on their type, severity level, and potential impact on the system. The analysis is conducted with reference to the OWASP Top Ten standards and the documentation results from OWASP ZAP.
4. Reporting
   In the final stage, the researcher compiles a report that includes a description of the discovered vulnerabilities, the risk level of each finding, and mitigation recommendations to improve system security. The report is prepared as a consideration for system administrators in planning improvements and enhancing security.

As a form of compliance with ethical aspects in cybersecurity research, the researcher has obtained official permission from the information system administrators at XYZ University before conducting the testing. The testing was carried out in a limited manner without damaging the system or accessing/manipulating sensitive data, in accordance with ethical hacking principles.

RESULTS AND DISCUSSION

Dalam pengujian kerentanan yang dilakukan untuk menemukan celah keamanan pada sistem informasi akademik berbasis web di Universitas XYZ, digunakan pendekatan sistematis yang meliputi identifikasi ruang lingkup, pemindaian kerentanan, analisis hasil, dan pelaporan. Pemindaian dilakukan menggunakan OWASP ZAP, alat open-source yang dirancang untuk mendeteksi berbagai jenis kerentanan aplikasi web secara otomatis. Hasil pemindaian diklasifikasikan berdasarkan jenis dan tingkat keparahan (severity) sesuai standar OWASP, kemudian dianalisis dampaknya terhadap sistem dan diberikan rekomendasi mitigasi sebagai langkah perbaikan.

1. Identifying Scope
   In this study, the testing scope is focused on the web-based Academic Information System at XYZ University. Defining this scope aims to ensure that the vulnerability assessment process is conducted in a targeted and comprehensive manner on the selected system.
2. Vulnerability Scanning
   The vulnerability scanning process was carried out using the OWASP ZAP tool, which functions to identify potential security loopholes within the web-based Academic Information System. This tool is used to evaluate the application's security level by automatically detecting various types of vulnerabilities. The scanning is specifically focused on the university's web application as an effort to assess the reliability and resilience of the system against security threats.
   Below are the scanning results obtained using OWASP ZAP.

		Confidence				
		User Confirmed	High	Medium	Low	Total
Risk	High	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)	0 (0.0%)
	Medium	0 (0.0%)	2 (14.3%)	2 (14.3%)	1 (7.1%)	5 (35.7%)
	Low	0 (0.0%)	2 (14.3%)	3 (21.4%)	0 (0.0%)	5 (35.7%)
	Informational	0 (0.0%)	0 (0.0%)	2 (14.3%)	2 (14.3%)	4 (28.6%)
	Total	0 (0.0%)	4 (28.6%)	7 (50.0%)	3 (21.4%)	14 (100%)

Image 2: OWASP ZAP Scanning Results Based on Risk and Confidence Levels

3. Identification of Vulnerabilities

After the vulnerability scanning process was completed, OWASP ZAP successfully identified several vulnerabilities present in the system. The types of vulnerabilities detected are summarized in **Table 1** below:

Table 1. OWASP ZAP Alert Scanning Results

No	Type of Vulnerability (Alert Type)
1	Absence of Anti-CSRF Tokens
2	Content Security Policy (CSP) Header Not Set
3	Hidden File Found
4	Missing Anti-clickjacking Header
5	Vulnerable JavaScript Library
6	Cookie Without Secure Flag
7	Cookie Without SameSite Attribute
8	Server Leaks Version Information via "Server" HTTP Response Header
9	Strict-Transport-Security Header Not Set
10	X-Content-Type-Options Header Missing
11	Information Disclosure – Suspicious Comments
12	Modern Web Application
13	Re-examine Cache-Control Directives
14	Session Management Response Identified

Based on the table above, a total of 14 types of vulnerabilities were successfully identified. The detected vulnerabilities span various aspects, including session management, security header configurations, the use of external libraries, and the unintentional exposure of information to the public. These findings indicate that, although no critical vulnerabilities were detected, there are weaknesses that—if left unaddressed—could potentially be exploited by unauthorized parties.

The next stage of this process is reporting, in which each identified vulnerability is further analyzed based on its risk level. Mitigation recommendations are then provided to enhance the overall security posture of the system.

4. Reporting

The reporting stage is the final documentation process of all scanning results and vulnerability analyses found in the web-based Academic Information System at XYZ University. This documentation not only includes the name and impact of each type of vulnerability but also provides technical recommendations as mitigation solutions to reduce or eliminate the associated risks.

The following is a summary table of the reporting results:

Table 2. Reporting

Vulnerability Name	Vulnerability Impact	Mitigation Solution
Absence of Anti-CSRF Tokens	The application is vulnerable to forged user requests, potentially leading to unauthorized data changes without user consent.	Add CSRF tokens to all forms and data-modifying requests to ensure request authenticity.
Content Security Policy (CSP) Header Not Set	The application is more susceptible to malicious scripts injected by third parties.	Set the CSP header to only allow content from trusted sources.
Hidden File Found	Hidden files such as.*git* or.*env* are publicly accessible and may expose system configuration information.	Ensure sensitive files are not accessible via the web server; configure *.htaccess* or server rules to block access.
Missing Anti-clickjacking Header	Without headers like X-Frame-Options, the application can be loaded in an iframe by malicious sites to hijack user clicks.	Add the *X-Frame-Options* header with the value *DANY* or *SAMEORIGIN* to prevent framing.
Vulnerable JavaScript Library	Insecure JavaScript libraries can be exploited by attackers.	Update libraries to the latest version and perform regular audits of external dependencies.
Cookie Without Secure Flag	Cookies without the Secure flag may be transmitted over unencrypted connections, increasing the risk of theft.	Apply the Secure flag to all sensitive cookies so they are only sent over HTTPS.
Cookie Without SameSite Attribute	Without the SameSite attribute, cookies can be sent in cross-site requests, increasing CSRF risks.	Add the SameSite=Lax or SameSite=Strict attribute to cookies as appropriate.
Server Leaks Version Information via "Server" HTTP Response Header Field	Server version info in HTTP headers may aid attackers in identifying underlying software.	Remove or obscure server version information from HTTP headers.
Strict-Transport-Security Header Not Set	Without HSTS, users may access the site over insecure HTTP.	Add the Strict-Transport-Security header to enforce HTTPS usage.
X-Content-Type-Options Header Missing	Without this header, browsers may misinterpret content types, which can be exploited for script execution.	Add the X-Content-Type-Options header with the value nosniff to prevent content type guessing.
Information Disclosure - Suspicious Comments	Code comments may reveal sensitive information or application logic that attackers can exploit.	Remove or obfuscate sensitive comments before deploying the application.
Modern Web Application	Modern technologies may introduce new attack vectors requiring enhanced protection.	Apply up-to-date security practices such as using HTTPS, input validation, and a Content Security Policy.
Re-examine Cache-control Directives	Improper caching can result in sensitive data being stored and served from cache.	Use appropriate Cache-Control and Pragma headers to prevent sensitive data from being cached.
Session Management Response Identified	Improper session configuration can lead to session hijacking.	Secure session management by using secure and HttpOnly cookies, setting session expiration, and implementing robust invalidation mechanisms.

Based on the summary of the scanning results in Table 2, the majority of identified vulnerabilities are categorized as weaknesses in web application security configurations—particularly related to HTTP header settings, session management, and the use of outdated third-party libraries. Each finding has been analyzed based on its potential impact on the confidentiality, integrity, and availability (CIA triad) of the system. Although the overall risk level ranges from low to medium, the accumulation of these vulnerabilities can significantly increase the system's attack surface and open the door to chained exploitation techniques. Therefore, implementing the recommended mitigation measures is crucial to reducing the risk exposure and strengthening the overall security posture of the academic information system.

CONCLUSION

Based on the security analysis of the web-based Academic Information System at XYZ University, a total of 14 types of vulnerabilities were identified across various components of the system, including security header configurations, session management, and the use of external libraries. Although no critical vulnerabilities were found, these issues still pose potential risks if not properly addressed. This study demonstrates that vulnerability assessment methods, particularly using OWASP ZAP, are effective in uncovering commonly overlooked security weaknesses. Immediate mitigation measures such as configuring appropriate security headers, updating external libraries, and implementing best web security practices are strongly recommended. Furthermore, this research suggests that higher education institutions should perform regular security audits and adopt ethical hacking practices to maintain the integrity and security of their information systems.

REFERENCES

Ariyadi, T., Widodo, T. L., Apriyanti, N., & Kirana, F. S. (2023). Analisis Kerentanan Keamanan Sistem Informasi Akademik Universitas Bina Darma Menggunakan OWASP. *Techno.Com*, *22*(2), 418–429. https://doi.org/10.33633/tc.v22i2.7562

Budiyanto, D., & Mabruri, M. (2025). Pentingnya Keamanan Siber dalam Era Digital: Tinjauan Global dan Kondisi di Indonesia. *Prosiding Seminar Nasional Sains Dan Teknologi Seri III Fakultas Sains Dan Teknologi, Universitas Terbuka*, *2*(1), 981–994.

Dellia, P., Hasan, M. A., Buana, D. S., Sari, D., & Savitri, C. (2025). Analisis Kepuasan Pengguna Siakad Menggunakan Metode Sus. *Jurnal Ilmiah Teknik Dan Ilmu Komputer*, *4*(2), 92–101.

Ending Narhudin, D., Irawan, B., & Bahtiar, A. (2024). Evaluasi Keamanan Website Menggunakan Metode Owasp: Penilaian Terhadap Serangan Injeksi Sql Dan Cross-Site Scripting (Xss). *JATI (Jurnal Mahasiswa Teknik Informatika)*, *8*(1), 675–680. https://doi.org/10.36040/jati.v8i1.8700

Hamidy, F., & Yasin, I. (2024). Penerapan Metode Moving Average Dalam Penentuan Harga Pokok Penjualan Barang Berbasis Web. *CHAIN: Journal of Computer Technology, Computer Engineering, and Informatics*, *2*(2), 67–76. https://doi.org/10.58602/chain.v2i2.115

Hasibuan, A. F., & Handoko, D. (2023). Analisis Keretanan Website Dengan Aplikasi Owasp Zap. *Jurnal Ilmu Komputer Dan Sistem Informasi*, *2*(2), 257–270.

Marpaung, J. N. (2025). Cyber Security in Indonesian Higher Education Institutions: Lessons Learned from Recent Cyber Attacks. *Jurasik (Jurnal Riset Sistem Informasi Dan Teknik Informatika)*, *10*(1), 343. https://doi.org/10.30645/jurasik.v10i1.876

Melani, C. (2023). Pengembangan Sistem Informasi Akademik Berbasis Web untuk Manajemen Data Mahasiswa, Dosen, dan Jadwal Kuliah di Perguruan Tinggi. *Teknologi Informasi*, *3*(4), 1–19. http://www.portaldata.org/index.php/cyberarea/article/view/396%0Ahttp://www.portaldata.org/index.php/cyberarea/article/download/396/384

Mustofa, P. Z., Sumaryana, Y., & Ruuhwan, R. (2025). Penetration Testing Pada Domain xyz.ac.id Menggunakan OWASP 10. *E-Jurnal JUSITI (Jurnal Sistem Informasi Dan Teknologi Informasi)*, *13*(2), 175–182. https://doi.org/10.36774/jusiti.v13i2.1637

Pramuja Inngam Fanani, G., Muhammad Amirul Mu'min, & Tristanti, N. (2025). Analisis dan Pengujian Kerentanan Website Menggunakan OWASP ZAP. *Jurnal Riset Sistem Dan Teknologi Informasi*, *3*(1), 36–50. https://doi.org/10.30787/restia.v3i1.1886

Rahman, R., & Fatkhur Razak, D. (2024). Pengujian Penetrasi Jaringan Menggunakan Owasp Zap Dan Sqlmap Untuk Mengidentifikasi Kerentanan Keamanan Website. *Jurnal Riset Sistem Informasi (JISSI)*, *1*(4), 11.

Saadah, ana wasilatu, Azizah, wafiq azizah, Permadani, H. indah, & Saputri, L. (2022). Implementasi Sistem Informasi Akademik (Siakad) Di Universitas Tulungagung Dalam Meningkatkan Efisiensi Dan Kualitas Pelayanan Pendidikan Ana. *Implementasi Sistem Informasi Akademik (Siakad) Di Universitas Tulungagung Dalam Meningkatkan Efisiensi Dan Kualitas Pelayanan Pendidikan Ana*.

Supartini, R., & Parenreng, J. M. (2023). Deteksi Serangan SQL Injection pada Website dengan Menggunakan Metode Reguler Expression. *Progressive Information, Security, Computer, and Embedded System*, *1*(2), 107–114. https://doi.org/10.61255/pisces.v1i2.101

Syaifudin, M. R., Murtadho, M. A., Wafa, M. S., & Masrur, M. (2025). *KOMPUTA : Jurnal Ilmiah Komputer dan Informatika Analisis Keamanan Website Kampus UNIPDU Melalui Metode Vulnerability Assessment (VA)*

dengan Menggunakan Tools Acunetix UNIPDU Campus Website Security Analysis Through Vulnerability Assessment (VA) Metho. *14*(1), 7–12. https://doi.org/10.34010/komputa.v14i1.

Zulfa, A. A., Ibrahim, T., & Arifudin, O. (2025). Peran Sistem Informasi Akademik Berbasis Web Dalam Upaya Meningkatkan Efektivitas Dan Efisiensi Pengelolaan Akademik Di Perguruan Tinggi. *Jurnal Tahsinia*, *6*(1), 115–134.