Vulnerability Assessment Method for Website Security

Rifaldi Dwi Anugrah[a,1], Erick Irawadi Alwi[a,2]

[a] *Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muslim Indonesia, Jl. Urip Sumoharjo km.05, Makassar dan 90231, Indonesia*
[1] dandibongkardandi@gmail.com; [2] erick.alwi@umi.ac.id
**corresponding author*

ARTICLE INFORMATION

Received : 18 – 11 – 2024
Revised : 12 – 09 – 2025
Published : 29 – 10 – 2025

Keywords:
Website Security
OWASP Zap
Vulnerability Assessment Cross-site Scripting Account Take Over

ABSTRACT

Website security is crucial in today's digital era as a medium for information and communication. The same applies to the Studio FIKOM UMI website used by the Faculty of Computer Science at Universitas Muslim Indonesia. This study aims to evaluate the security posture of the Studio FIKOM UMI website against cyberattacks and to identify the most likely attack vectors targeting the site. The research adopts a Vulnerability Assessment methodology to analyze, identify, and categorize the risk levels of discovered vulnerabilities within the existing networked system. Information was gathered from multiple sources—websites, journals, scholarly works, books, and online resources. The method is applied to uncover vulnerabilities present on the Studio FIKOM UMI website. The assessment revealed vulnerabilities based on alerts from OWASP ZAP scanning, including: Vulnerable JavaScript Library, X-Frame-Options header not set, absence of anti-CSRF tokens, cookies without the HttpOnly flag, cookies without the SameSite attribute, cross-domain JavaScript source inclusion, incomplete or missing Cache-Control/Pragma headers, X-Powered-By response header exposed, and missing X-Content-Type-Options header. The overall risk ratings comprised medium risk (4 findings) and low risk (7 findings). In terms of confidence, there were medium confidence (8 findings) and high confidence (3 findings) alerts. Based on validation of the OWASP ZAP findings, two items map to the OWASP Top 10: Broken Access Control (risk: medium, confidence: high) and Cross-Site Scripting (XSS) (risk: medium, confidence: medium). Consequently, the most plausible attack scenarios include cross-site scripting and account takeover.

I. Introduction

In today's digital era, the presence of websites as an information medium is crucial—especially for educational institutions such as Universitas Muslim Indonesia. Websites are used to convey information to both students and lecturers; therefore, it is essential that they operate optimally and securely.

However, alongside the rapid development of technology and information, the rate of cybercrime has increased, particularly in the realm of website security. Cyberattacks such as cross-site scripting (XSS) and SQL injection pose real threats to educational institutions. These attacks not only risk damaging an institution's reputation but can also lead to leaks of sensitive data and a loss of user trust.

The Faculty of Computer Science at Universitas Muslim Indonesia operates a dedicated certification website for students and lecturers as a professional standardization platform, namely Studio FIKOM UMI. The Studio FIKOM UMI website has previously experienced a cyberattack using brute-force techniques. This attack exploited broken authentication due to weak passwords, enabling attackers to obtain administrator access and even steal critical data from the site.

A study by Imam Riadi, Agus Yudhana, and Yunanri W., titled "Security Analysis of the Open Journal System Website Using the Vulnerability Assessment Method," found that OJS version 2.4.7 contains numerous vulnerabilities and is not recommended for use [1]. Another study by Yulia Taryana and Nono Heryana, "Security Analysis of the BPJS Kesehatan Website Using the Vulnerability Assessment Method," identified eleven vulnerabilities on the BPJS Kesehatan website and proposed remediation measures [2]. Further, Erick Irawadi Alwi and Lutfi Budi Ilmawan, in "Security Analysis of the Academic Information System (SIAKAD) at University XYZ Using the Vulnerability Assessment Method," reported findings that included 1 critical threat,

6 high-severity, 14 medium-severity, and several low-severity issues [3]. In another study, Alif Muhammad Akmal, Nono Heryana, and Arip Solehudin, "Security Analysis of the Singaperbangsa Karawang Website Using the Vulnerability Assessment Method," identified 2 high-risk vulnerabilities, 5 medium, 2 low, and several informational items [4]. Additional research by Mira Orisa and Michael Ardita, "Vulnerability Assessment to Improve Web Security Quality," concluded that Nmap is capable of conducting vulnerability assessments to check for malware/phishing issues, assess exposure to denial-of-service attacks, and detect vulnerabilities to SQL injection [5].

Building on prior studies and the incident that occurred on the Studio FIKOM UMI website, this research is titled: "Evaluating the Security of the Studio FIKOM UMI Website Against Cyberattacks Using the Vulnerability Assessment Method.".

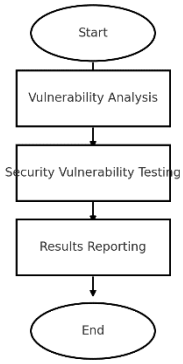

Figure 1. Research Stages

The initial stage involves problem identification based on information obtained from several sources by visiting the Studio FIKOM UMI website and consulting with the site administrators. This identification also draws on other elements such as dissertations related to this research topic and additional supporting information.

The second stage is Vulnerability Analysis. After identifying the problems and gathering information, an automated scan is performed using suitable tools. The researcher analyzes potential vulnerabilities in the target and evaluates their severity, which then informs the exploitation phase.

The testing/exploitation phase of website vulnerabilities uses Burp Suite, guided by the collected information, to conduct cross-site scripting (XSS) and account takeover attack scenarios. The results of this testing are then analyzed further to determine appropriate remediation measures for the previously identified issues.

After testing, a reporting phase is conducted to document the discovered vulnerabilities within an information and network security framework, covering the entire research flow—from identification, vulnerability analysis, and exploitation to test-result reporting and recommended fixes to address the vulnerabilities..

II. Method

Vulnerability assessment is the process used to analyze, identify, classify, and test points that could serve as entry vectors for attacks so they can be prevented and remediated. Another function of vulnerability assessment is to detect weaknesses within a system on a computer network. As a first step in testing, information must be gathered about the physical server, the type of network in use, and various other details related to the server network. The stages of a vulnerability assessment include:

A. Vulnerability Identification

The goal of this step is to analyze the security of an application, server, or other system by scanning it with automated tools or by testing and evaluating it manually. The analysis also relies on vulnerability databases to identify security weaknesses.

B. Analysis

This step aims to determine the origin and cause of the vulnerabilities identified in the first step. For example, a root cause may be that the components in use are outdated versions.

C. Risk Assessment

This step prioritizes the vulnerabilities. It involves a security analysis that evaluates and elevates the severity level of each vulnerability discovered.

D. Remediation

The goal of this step is to close security gaps or vulnerabilities. This is usually carried out by developers to determine the most effective mitigation path, such as updating backend code or upgrading software to a newer version. The stages of the vulnerability assessment are shown in Figure 2.

Figure 2. Vulnerability Assessment

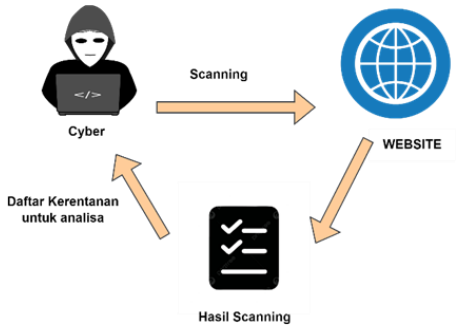

Figure 3. Testing Topology

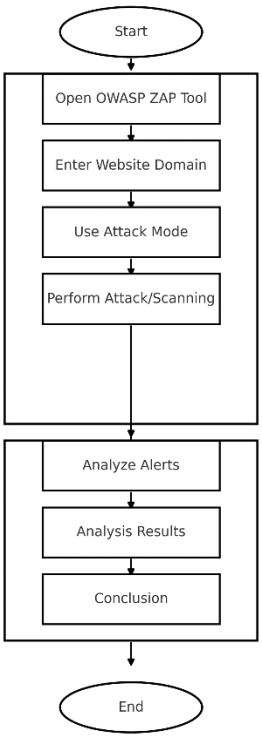

Figure 4. Testing Scenario

Scanning: In this stage, the OWASP ZAP tool is used to collect information such as the domain, IP address, application type, application version, and existing vulnerabilities.

After scanning, the researcher analyzes the alerts or the list of vulnerabilities displayed by OWASP ZAP. These alerts present potential vulnerabilities found during the scanning process.

Based on the analyzed alerts, the researcher compiles the results indicating the types of vulnerabilities discovered, their severity levels, and the potential impact on the website.

III. Results and Discussion

A. Scanning

This stage performs a scanning process using the OWASP ZAP tool to help identify the types of vulnerabilities, their risk levels (low, medium, or high), and the confidence level for accessing the Studio FIKOM UMI website. The OWASP ZAP scan provides information on vulnerabilities, as shown in Figure 5.

Figure 5. OWASP ZAP Scanning

Table 1. Studio FIKOM UMI Website Vulnerabilities

Vulnerability (Studio FIKOM UMI)		
Type	Risk	Confidence
Vulnerable JS Library	Medium	Medium
X-Frame-Options Header Not Set	Medium	Medium
Absence of Anti-CSRF Tokens	Low	Medium
Cookie Without HttpOnly Flag	Low	High
Cookie Without SameSite Attribute	Low	High
Cross-Domain JavaScript Source File Inclusion	Low	Medium
Incomplete or No Cache-Control and Pragma HTTP Header Set	Low	Medium
X-Powered-By HTTP Response Header Fields	Low	Medium
X-Content-Type-Options Header Missing	Low	Medium

After scanning, a vulnerability analysis is carried out in which the attacker analyzes XSS entry points based on the scan results. The objective is to identify XSS vulnerabilities in the target. The OWASP ZAP scan revealed several issues, including: Vulnerable JS Library, X-Frame-Options Header Not Set, Absence of Anti-CSRF Tokens, Cookie Without HttpOnly Flag, Cookie Without SameSite Attribute, Cross-Domain JavaScript Source File Inclusion, Incomplete or No Cache-Control and Pragma HTTP Header Set, Server Leaks Information via “X-Powered-By” HTTP Response Header Fields, and X-Content-Type-Options Header Missing.

The next step analyzes the vulnerabilities identified by OWASP ZAP. The following explains each finding on the Studio FIKOM UMI website.

1) Vulnerable JS Library

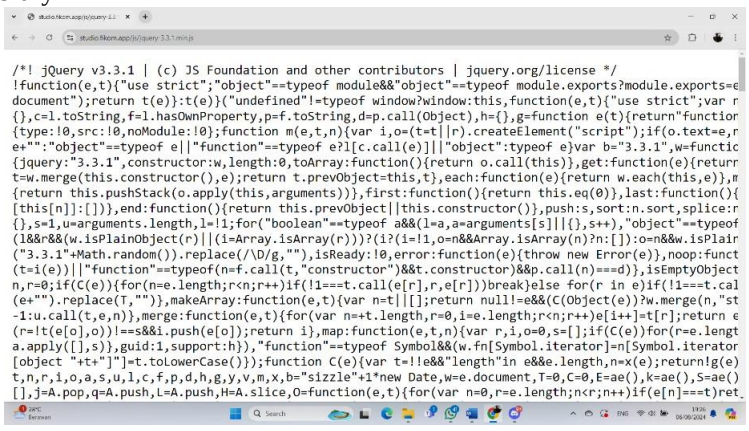

Figure 6. Vulnerable JS Library

In Figure 6, an attacker can inject malicious JavaScript into a site that relies on a vulnerable JS library. The injected code may execute in users' browsers, allowing the attacker to steal data, control the browser, or perform other harmful actions.

2) X-Frame-Options Header Not Set

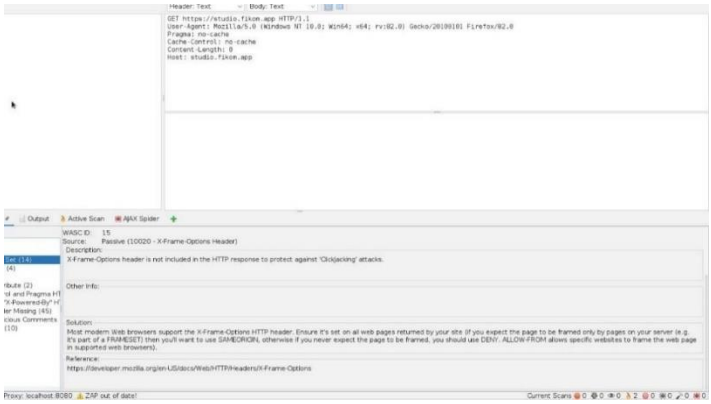

Figure 7. X-Frame-Options Header Not Set

As shown in Figure 7, this header allows developers to control whether a web page may be displayed in a frame or iframe. If it is not set, an attacker can trap your page within a frame and present it alongside malicious content. This enables clickjacking attacks, tricking users into clicking dangerous buttons or links.

3) Absence of Anti-CSRF Tokens

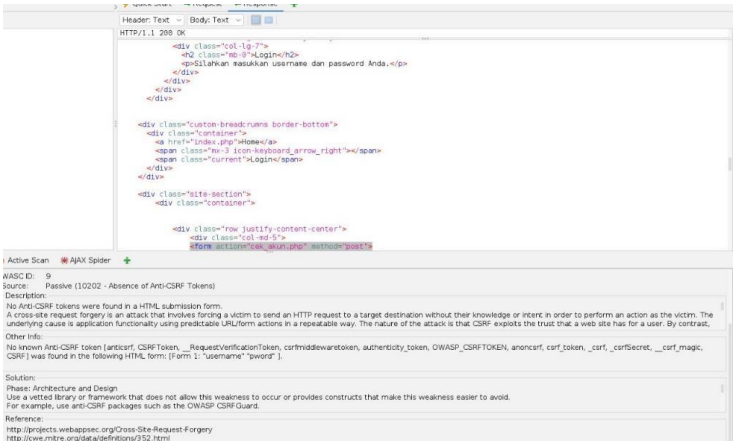

Figure 8. Absence of Anti-CSRF Tokens

In Figure 8, random tokens that protect users from CSRF are absent. CSRF occurs when attackers trick users into sending unintended HTTP requests to a trusted site. Without anti-CSRF tokens, attackers can perform various actions on behalf of users without their knowledge or consent.

4) Cookie Without HttpOnly Flag

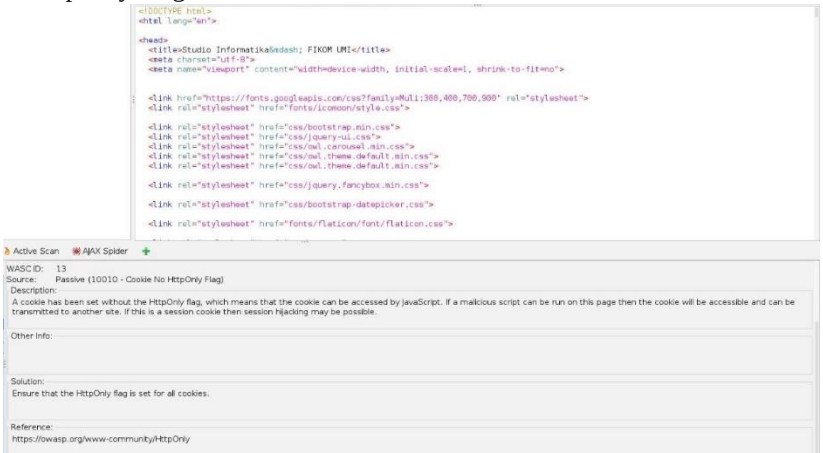

Figure 9. Cookie Without HttpOnly Flag

In Figure 9, the cookie is accessible to JavaScript. If malicious scripts run on the page, the cookie can be read and exfiltrated to another site. If it is a session cookie, session hijacking becomes possible.

5) Cookie Without SameSite Attribute

Figure 10. Cookie Without SameSite Attribute

In Figure 10, the cookie lacks a "SameSite" value in its header. The SameSite attribute controls how cookies are sent in cross-site requests. Without it, cookies may be transmitted in all types of requests, including cross-site requests.

6) Cross-Domain JavaScript Source File Inclusion

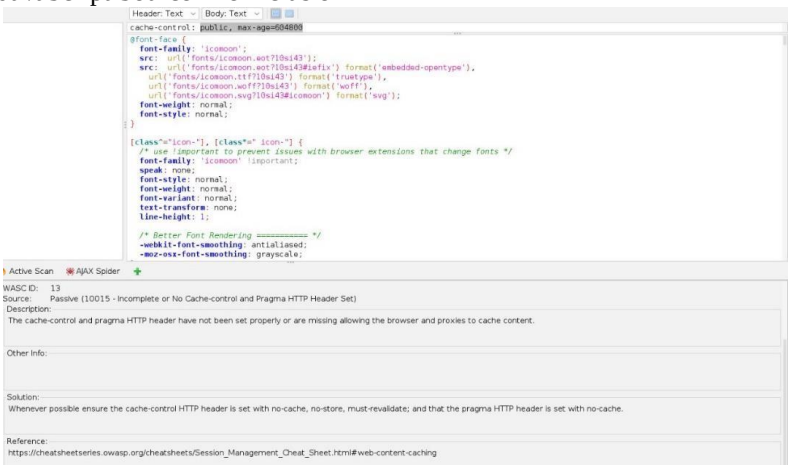

Figure 11. Cross-Domain JavaScript Source File Inclusion

In Figure 11, this vulnerability allows attackers to include malicious JavaScript files from other domains into a web page. This can enable data theft, browser control, or other unwanted actions.

7) Incomplete or No Cache-Control and Pragma HTTP Header Set

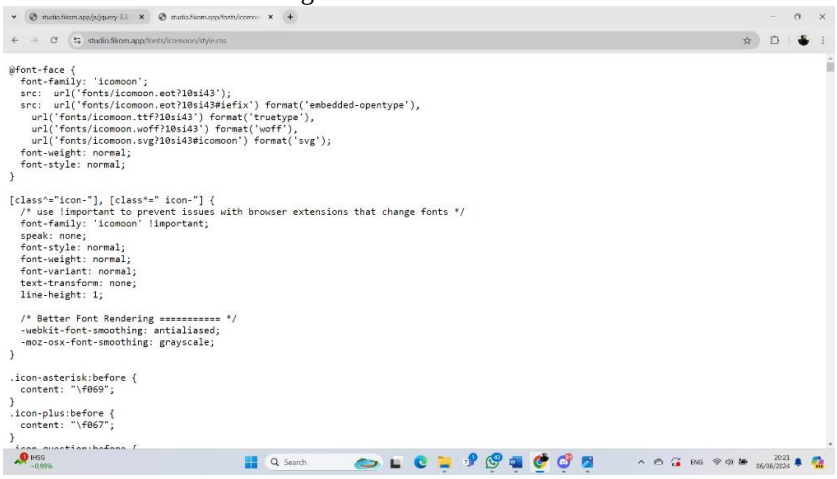

Figure 12. No Cache-Control and Pragma HTTP Header Set

As shown in Figure 12, these headers let developers control how the browser stores and accesses web content. If misconfigured or absent, attackers may intercept and modify content before it is displayed, facilitating phishing or malware attacks.

8) X-Powered-By HTTP Response Header Fields

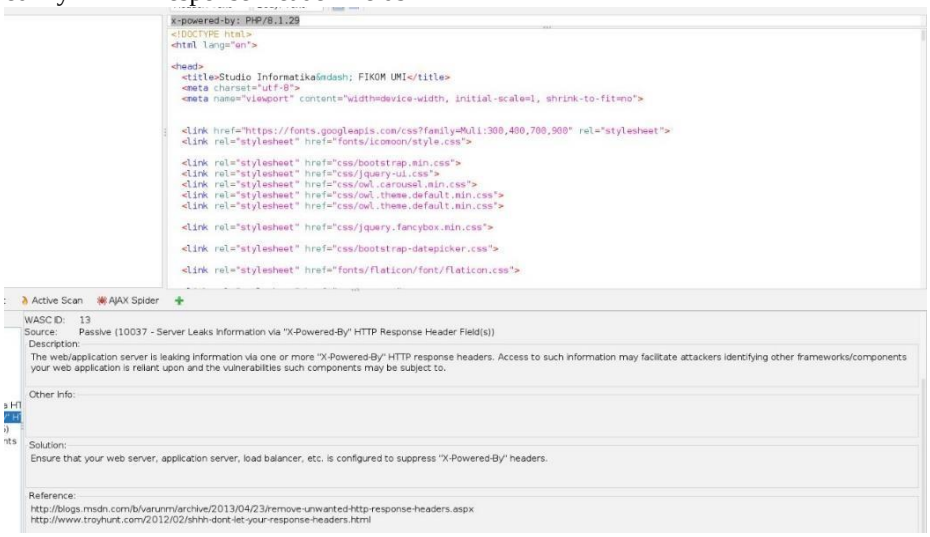

Figure 13. X-Powered-By HTTP Response Header Fields

In Figure 13, this response header reveals the software technology used by the application. Because many servers include it by default, exposing it can pose a security risk; attackers may leverage known, technology-specific vulnerabilities.

9) X-Content-Type-Options Header Missing

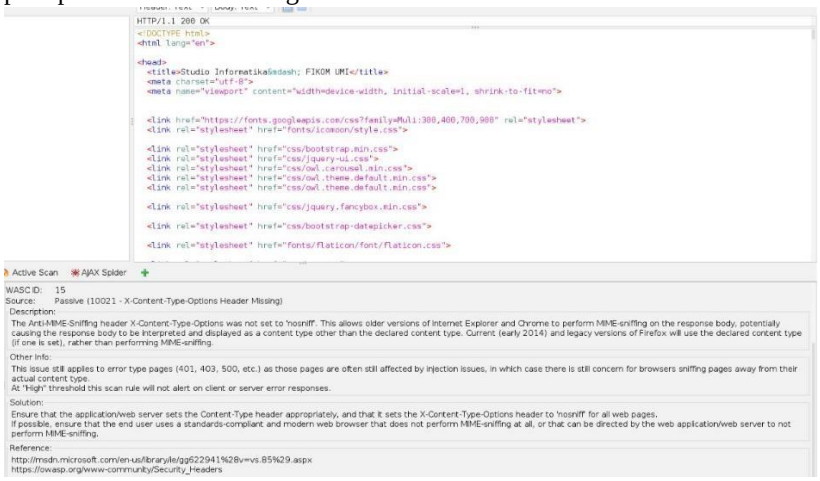

Figure 14. X-Content-Type-Options Header Missing

In Figure 14, this header prevents the browser from MIME-sniffing the content type, helping to mitigate XSS. If it is not set, attackers may exploit content-type guessing to inject and execute malicious code.

B. Discussion

The results of testing conducted on the Studio FIKOM UMI website (https://studio.fikom.app) identified two vulnerabilities: Cross-Site Scripting (XSS) and Broken Access Control, as shown in Table 2.

Table 2. Website Vulnerabilities

Vulnerability (Studio FIKOM UMI)	Location	Impact	Risk	Confidence
Cross-Site Scripting	Search bar	Redirection to malicious websites	Medium	Medium
Broken Access Control	URL	Files can be browsed; potential access to database	Low	High

Table 2 shows two vulnerabilities with different risk and confidence levels: Cross-Site Scripting and Broken Access Control. For XSS, the risk is medium because the effect does not directly disrupt the Studio FIKOM

UMI website's operation but impacts its users, thereby undermining the site's credibility. The likelihood of success is medium, as attackers must trick users into clicking a crafted link. For Broken Access Control, the risk is medium because files (including system code) can be browsed; the likelihood of success is high since special techniques are not required to exploit this weakness.

Below is a brief explanation of the risks and recommended mitigations for the identified vulnerabilities (see Table 3).

Table 3. Types of Vulnerabilities

No	Vulnerability Type	Risk	Recommendation
1	Cross-Site Scripting (XSS)	This vulnerability arises when the application does not properly validate or filter user input, allowing the injection of malicious script code. XSS attacks can enable attackers to steal user data or run malicious scripts in the user's browser.	Always validate and sanitize user input in the search input field. Ensure only expected characters are accepted and avoid accepting or executing input that contains script code.
2	Broken Access Control	This occurs when access controls are not correctly enforced, allowing users to gain unauthorized access to resources they should not be able to reach. This may expose sensitive data and restricted functionality.	Always perform server-side authorization checks before granting access to particular resources or features. Do not rely solely on client-side access controls, as they can be bypassed, enabling access to sensitive files via URLs.

In the Risk column, the table describes the causes or faults in the website system that allow the vulnerability to occur and the possible impacts. The Recommendation column contains remediation steps that the IT team for the Studio FIKOM UMI website can implement.

IV. Conclusion

This study found vulnerabilities on the Studio FIKOM UMI website based on alerts from OWASP ZAP scanning, including: Vulnerable JS Library, X-Frame-Options Header Not Set, Absence of Anti-CSRF Tokens, Cookie Without HttpOnly Flag, Cookie Without SameSite Attribute, Cross-Domain JavaScript Source File Inclusion, Incomplete or No Cache-Control and Pragma HTTP Header Set, X-Powered-By HTTP Response Header Fields, and X-Content-Type-Options Header Missing—with medium risk (4 findings) and low risk (7 findings). For confidence levels, there were medium confidence (8 findings) and high confidence (3 findings) alerts.

Based on the verification of OWASP ZAP results, two OWASP Top 10 findings were confirmed: Broken Access Control (level 5, medium risk, high confidence) and Cross-Site Scripting (XSS) (medium risk, medium confidence). Given these vulnerabilities, the overall security level remains low and requires immediate remediation. The identified weaknesses can adversely affect the relationship with clients and the reputation of the Faculty of Computer Science, UMI, as a service provider. Consequently, the most likely attacks against the Studio FIKOM UMI website are Cross-Site Scripting and Account Takeover.

References

[1] I. Riadi, A. Yudhana, and Y. W, "Analisis Keamanan Website Open Journal System Menggunakan Metode Vurnerability Assessment," J. Teknol. Inf. dan Ilmu Komput., vol. 7, no. 4, pp. 853–860, 2019, doi: 10.25126/jtiik.202071928.

[2] Y. Taryana and N. Heryana, "ANALISIS KEAMANAN WEBSITE BPJS KESEHATAN MENGGUNAKAN METODE VULNERABILITY ASESEMENT."

[3] E. Irawadi Alwi and L. Budi Ilmawan, "Analisis Keamanan Sistem Informasi Akademik (SIAKAD) Universitas XYZ Menggunakan Metode Vulnerability Assessment," 2021.

[4] J. Pendidikan and D. Konseling, "Analisis Keamanan Website Universitas Singaperbangsa Karawang Menggunakan Metode Vulnerability Assessment."

[5] M. Orisa and M. Ardita, "VULNERABILITY ASSESMENT UNTUK MENINGKATKAN KUALITAS KEMANAN WEB," 2021.

5] M. Orisa and M. Ardita, “VULNERABILITY ASSESMENT UNTUK MENINGKATKAN KUALITAS KEMANAN WEB,” 2021.

[6] W. Wahyudin, H. Kuswara, R. Resti, and S. Dalis, “Metode Vulnerability Assesment Dalam Pengujian Kinerja Sistem Keamanan Website Points of Sales,” Comput. Sci., vol. 4, no. 1, pp. 44–52, 2024, doi: 10.31294/coscience.v4i1.2978.

[7] Y. Mulyanto, E. Haryanti, and J. Jumirah, “Analisis Keamanan Website Sman 1 Sumbawa Menggunakan Metode Vulnerability Asesement,” J. Inform. Teknol. dan Sains, vol. 3, no. 3,

pp. 394–400, 2021, doi: 10.51401/jinteks.v3i3.1260.

[8] M. I. Fadillah, U. Yunan, K. S. Yanto, and M. Fathinuddin, “Analisis Security Mitigation dengan Metode Vulnerability Assesment and Penetration Testing (VAPT) (Kasus Website Kerja Praktek dan Pengabdian Masyarakat),” J. Sains Komput. Inform. (J-SAKTI, vol. 7, no. 2, pp. 753–764, 2023.

[9] M. A. Aziz, “Vulnerability Assesment Untuk Mencari Celah Keamanan Web Aplikasi E- Learning Pada Universitas Xyz,” J. Eng. Comput. Sci. Inf. Technol., vol. 2, no. 1, 2023, doi: 10.33365/jecsit.v1i1.13.

[10] Rifky Lana Rahardian, “Analisis Keamanan Web New Kuta Golf Menggunakan Metode Vulnerability Assessments Dan Perhitungan Security Metriks,” J. Inform. Dan Tekonologi Komput., vol. 2, no. 3, pp. 256–265, 2022, doi: 10.55606/jitek.v2i3.582.

[11] S. A. Putra, A. Budiono, and U. Y. K. Septo, “Vulnerability Assesment Web ProposalTugas Akhir Mahasiswa MenggunakanAcunetix dan NMAP,” eProceedings …, vol. 10, no. 2, pp. 1615–1622, 2023, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/19972%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/download/19972/19337

[12] A. M. Marpaung, F. Husnah, S. D. Efita, and A. B. Nasution, “Perancangan Sistem Keamanan Website Dengan Metode Hill Chiper,” J. Sains dan Teknol., vol. 3, no. 1, pp. 120–129, 2023, doi: 10.47233/jsit.v3i1.494.

[13] I. A. Hakim, F. A. Pratama, R. A. Sitorus, A. Firdaus, and S. Fadilah, “Meningkatkan Kewaspadaan Terhadap Kejahatan Cyber Di Era Digital Di SMK Negeri 8 Kabupaten Tangerang,” vol. 1, no. 4, pp. 188–194, 2023.

[14] R. Armando, I. G. A. K. A. Melyantara, R. Elfariani, D. F. A. Latuconsina, and M. Nasrullah, “IT Support Website Security Evaluation Using Vulnerability Assessment Tools,” J. Inf. Syst. Informatics, vol. 4, no. 4, pp. 949–957, 2022, doi: 10.51519/journalisi.v4i4.330.

[15] B. A, “10 Tools untuk Vulnerability Assessment,” MENGGUNAKAN.ID. Accessed: May 14, 2024. [Online]. Available: https://www.menggunakan.id/10-tools-untuk-vulnerability- assessment/