Journix: Journal of Informatics and Computing
Vol. 1 No. 1, 2025 : 10-20
ISSN : 3090-6784 | DOI : 10.63866/journix.v1i1.1

Security Analysis of XYZ Website Using OWASP Zap Tools

Muhammad Amirul Mu'min [1*], Yana Safitri [2], Galih Pramuja Inngam Fanani [3], Setiawan Ardi Wijaya [4], Novi Tristanti [5]

[1] Ilmu Komputer, Universitas Muhammadiyah Bima, Bima, Indonesia
[2] Ilmu Komputer, Universitas Qamarul Huda Badaruddin Bagu, Lombok, Indonesia
[3] Sistem dan Teknologi Informasi, Universitas 'Aisyiyah Surakarta, Surakarta, Indonesia
[4] Ilmu Komputer, Universitas Muhammadiyah Riau, Riau, Indonesia
[5] Informatika, Universitas Muhammadiyah Karanganyar, Indonesia

Email: muhamirul98@gmail.com
(* : corresponding author)

ABSTRACT - In the growing digital era, website security is a critical aspect that must be considered. Vulnerabilities such as Cross-Site Scripting (XSS), Clickjacking, and Man-in-the-Middle can pose serious risks to data integrity and security. Therefore, effective tools are needed to identify and evaluate such vulnerabilities to prevent costly exploitation. This research aims to analyze security vulnerabilities on the website using OWASP ZAP (Zed Attack Proxy) as a penetration testing tool, and provide mitigation recommendations to improve system security. The method used is penetration testing by utilizing OWASP ZAP to identify security vulnerabilities on the website. The research stages include testing, analyzing the results, and preparing mitigation recommendations based on the findings of vulnerabilities such as A01, A03, and A04. The results showed that OWASP ZAP successfully identified various vulnerabilities, including XSS, Clickjacking, and Man-in-the-Middle. Recommended mitigation measures include configuring security headers and protecting sensitive data to prevent exploitation. OWASP ZAP proved to be effective in detecting and evaluating security vulnerabilities on websites. In addition, the tool also raises awareness of the importance of strong security policies. With the implementation of mitigation recommendations, website owners can better protect sensitive data, maintain user trust, and stay safe in an increasingly complex digital environment.

KEYWORDS: Information Security, Website, OWASP Zap, XSS

Analisis Keamanan Website XYZ Menggunakan Tools OWASP Zap

ABSTRAK - Dalam era digital yang semakin berkembang, keamanan website menjadi aspek kritis yang harus diperhatikan. Kerentanan seperti Cross-Site Scripting (XSS), Clickjacking, dan Man-in-the-Middle dapat menimbulkan risiko serius bagi integritas dan keamanan data. Oleh karena itu, diperlukan alat yang efektif untuk mengidentifikasi dan mengevaluasi kerentanan tersebut guna mencegah eksploitasi yang merugikan. Penelitian ini bertujuan untuk menganalisis kerentanan keamanan pada website menggunakan OWASP ZAP (Zed Attack Proxy) sebagai alat penetration testing, serta memberikan rekomendasi mitigasi untuk meningkatkan keamanan sistem. Metode yang digunakan adalah penetration testing dengan memanfaatkan OWASP ZAP untuk mengidentifikasi kerentanan keamanan pada website. Tahapan penelitian meliputi pengujian, analisis hasil, dan penyusunan rekomendasi mitigasi berdasarkan temuan kerentanan seperti A01, A03, dan A04. Hasil penelitian menunjukkan bahwa OWASP ZAP berhasil mengidentifikasi berbagai kerentanan, termasuk XSS, Clickjacking, dan Man-in-the-Middle. Langkah mitigasi yang direkomendasikan meliputi konfigurasi header keamanan dan perlindungan data sensitif untuk mencegah eksploitasi. OWASP

ZAP terbukti efektif dalam mendeteksi dan mengevaluasi kerentanan keamanan pada website. Selain itu, alat ini juga meningkatkan kesadaran akan pentingnya kebijakan keamanan yang kuat. Dengan implementasi rekomendasi mitigasi, pemilik website dapat lebih melindungi data sensitif, mempertahankan kepercayaan pengguna, dan menjaga keamanan di tengah lingkungan digital yang semakin kompleks.

KATA KUNCI: Keamanan Informasi, Website, OWASP Zap, XSS

Received : 10-02-2025	**Revised :** 03-03-2025	**Published :** 13-03-2025

1. PENDAHULUAN

Dalam era digital yang terus berkembang, keamanan informasi menjadi aspek yang sangat krusial [1]. Website, sebagai media komunikasi dan interaksi utama di dunia maya, sering kali menjadi target serangan siber [2]. Kerentanan pada website dapat mengakibatkan kerugian yang signifikan, baik bagi pemilik website maupun pengguna [3]. Dengan meningkatnya jumlah pengguna *internet* dan kompleksitas serangan siber yang semakin canggih, penting bagi pemilik website untuk menerapkan langkah-langkah keamanan yang efektif [4]. Serangan seperti *malware*, *phishing*, dan DDoS (*Distributed Denial of Service*) tidak hanya merusak reputasi bisnis, tetapi juga dapat mengakibatkan kehilangan data sensitif dan menurunkan kepercayaan pengguna [5]. Selain itu, regulasi yang ketat terkait perlindungan data pribadi, seperti GDPR dan UU PDP, semakin menuntut pemilik website untuk menjaga keamanan informasi. Oleh karena itu, edukasi tentang keamanan siber dan penerapan praktik terbaik menjadi kunci untuk melindungi aset digital [6].

Meskipun kesadaran akan pentingnya keamanan website semakin meningkat, banyak pemilik website masih belum sepenuhnya memahami kerentanan yang mungkin ada pada sistem mereka. Beberapa masalah umum yang sering ditemukan meliputi kerentanan terhadap serangan SQL *Injection*, *Cross-Site Scripting* (XSS), dan serangan DdoS [7]. Kurangnya pemahaman tentang cara mengidentifikasi dan memitigasi kerentanan ini dapat membuat website rentan terhadap eksploitasi oleh pihak yang tidak bertanggung jawab [8]. Selain itu, kurangnya alat yang tepat untuk melakukan analisis keamanan secara mendalam juga menjadi hambatan dalam menjaga keamanan website [9]. Untuk mengatasi masalah kerentanan keamanan pada website, solusi yang diusulkan adalah dengan menggunakan OWASP ZAP sebagai alat untuk melakukan pemindaian dan analisis keamanan [10]. OWASP ZAP memungkinkan pengguna untuk mengidentifikasi berbagai jenis kerentanan dengan cara yang sistematis dan terstruktur [11]. Setelah kerentanan teridentifikasi, langkah selanjutnya adalah memberikan rekomendasi perbaikan yang sesuai untuk meningkatkan keamanan website tersebut [12]. Dengan demikian, pemilik website dapat mengambil langkah proaktif untuk melindungi aset digital mereka dari serangan siber [13].

Penelitian ini akan fokus pada analisis keamanan website menggunakan alat OWASP ZAP [14]. Tools ini dipilih karena merupakan tools *open-source* yang telah terbukti efektif dalam mengidentifikasi berbagai jenis kerentanan keamanan pada aplikasi web. Namun, penelitian ini hanya akan membatasi pada pemindaian dan analisis kerentanan yang dapat diidentifikasi oleh OWASP ZAP, seperti SQL *Injection*, XSS, dan kerentanan umum lainnya. Penelitian ini tidak akan mencakup aspek keamanan lain seperti keamanan fisik server atau keamanan jaringan [15].

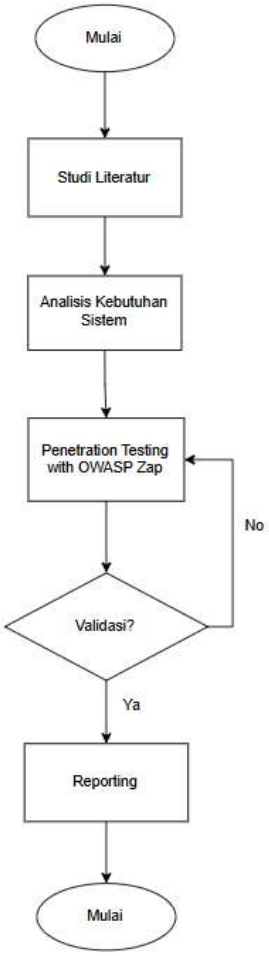
Tujuan dari penelitian ini adalah untuk melakukan pemindaian keamanan pada sebuah website menggunakan OWASP ZAP, menganalisis hasil pemindaian tersebut, dan memberikan rekomendasi perbaikan yang dapat meningkatkan keamanan website. Melalui penelitian ini, diharapkan pembaca dapat memahami pentingnya keamanan web dan bagaimana alat seperti

OWASP ZAP dapat digunakan untuk mengidentifikasi dan mengatasi kerentanan yang ada. Selain itu, penelitian ini juga bertujuan untuk meningkatkan kesadaran akan pentingnya penerapan praktik keamanan siber yang baik dalam pengembangan dan pemeliharaan website.

2. METODE PENELITIAN

2.1 Tahapan Penelitian

Penelitian ini menerapkan kerangka kerja untuk menganalisis kerentanan website menggunakan aplikasi OWASP ZAP, dengan menguji website XYZ sebagai objek penelitian guna menilai tingkat kerentanannya terhadap berbagai serangan, sebagaimana ditampilkan pada Gambar 1. Tahapan penelitian diawali dengan studi literatur, yaitu mencari referensi terkait penelitian dalam bentuk jurnal, buku, atau sumber lainnya untuk memperdalam pemahaman tentang topik yang dibahas. Selanjutnya, dilakukan analisis kebutuhan sistem untuk menyiapkan segala keperluan penelitian, termasuk alat dan bahan yang diperlukan. Setelah persiapan selesai, tahap berikutnya adalah melakukan penetration testing menggunakan alat OWASP ZAP guna mengidentifikasi celah keamanan pada website yang diuji. Terakhir adalah reporting yaitu membuat laporan dari hasil analisis dan identifikasi yang ditemukan selama penelitian.



Gambar 1. Alur Penelitian

2.2 Alat dan Bahan

Penelitian melibatkan penggunaan alat yang terdiri dari perangkat lunak dan perangkat keras untuk mendukung analisis serta pengujian. Sementara itu, bahan mengacu pada data atau sumber daya yang digunakan dalam eksperimen. Pemilihan alat dan bahan yang sesuai sangat penting untuk memastikan hasil yang akurat dan sesuai dengan tujuan penelitian. Tahap ini merupakan bagian dari persiapan sebelum melakukan penetration testing pada aplikasi situs web. Alat dan bahan seperti yang ditampilkan dalam Tabel 1.

Tabel 1. Alat bahan dan kegunaan

Tools	Spesifikasi
Laptop	OS: Windows 10 64 bit Processor: Intel Core i5-8565U quad-core 2,8GHz RAM: 8GB DDR4 VGA: NVidia GeForce MX150 SSD: 128GB
OWASP Zap	Versi: 2.11.1
Web Browser	Google Chrome

3. HASIL DAN PEMBAHASAN

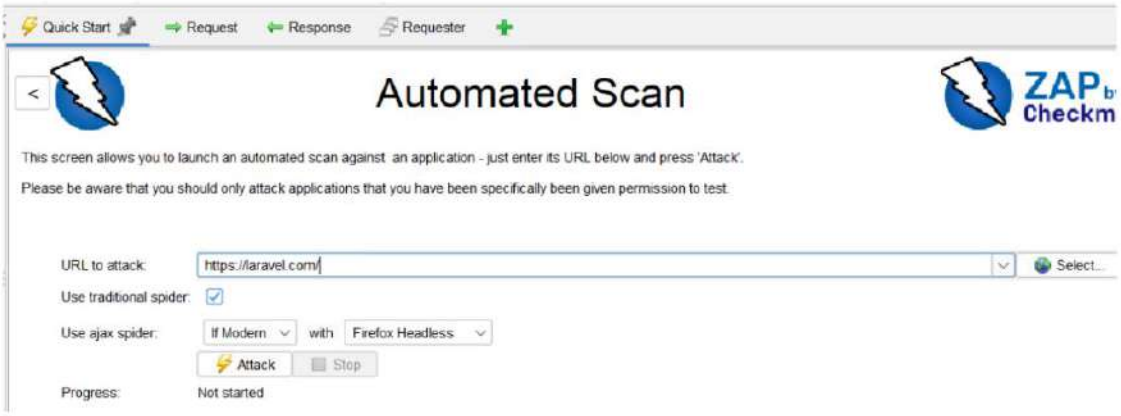
Penelitian ini menggunakan OWASP ZAP untuk membantu mengidentifikasi serta mengatasi potensi kerentanan keamanan pada web.

3.1 *Penetration Testing*

Tahap ini merupakan proses pengujian keamanan sistem atau website dengan cara mensimulasikan serangan siber untuk mengidentifikasi kerentanan atau celah keamanan yang mungkin dieksploitasi oleh pihak yang tidak bertanggung jawab. Tujuannya adalah untuk mengevaluasi tingkat keamanan sistem dan memberikan rekomendasi perbaikan guna memperkuat pertahanan terhadap ancaman yang mungkin terjadi.

a. *Input* URL Website XYZ

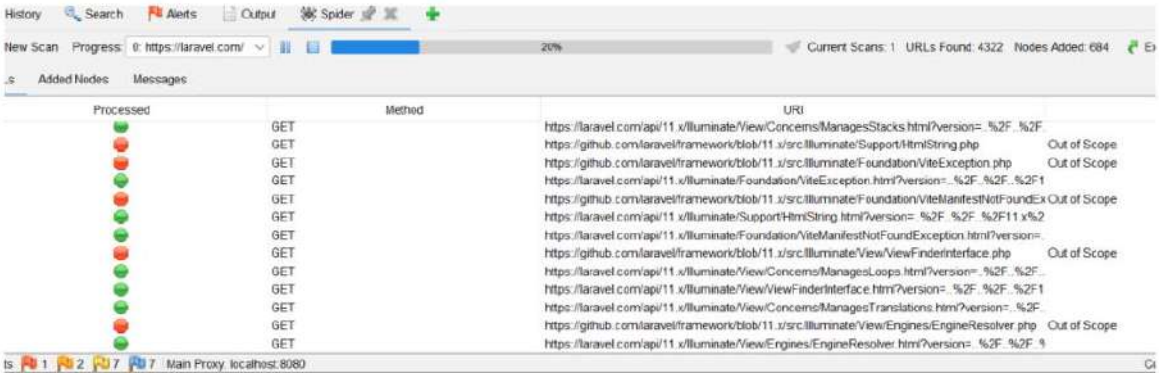
Pada tahap ini, pengguna memasukkan URL web ke kolom URL untuk diuji, lalu memilih opsi *Use traditional spider* dan *Use AJAX* spider guna membantu pemindaian. Setelah itu, tombol *attack* ditekan untuk memulai pemindaian otomatis. OWASP ZAP kemudian menganalisis situs web yang telah dimasukkan, seperti yang ditampilkan pada Gambar 2.



Gambar 2. Tampilan *Automated Scan*

b. Proses *Penetration Testing*

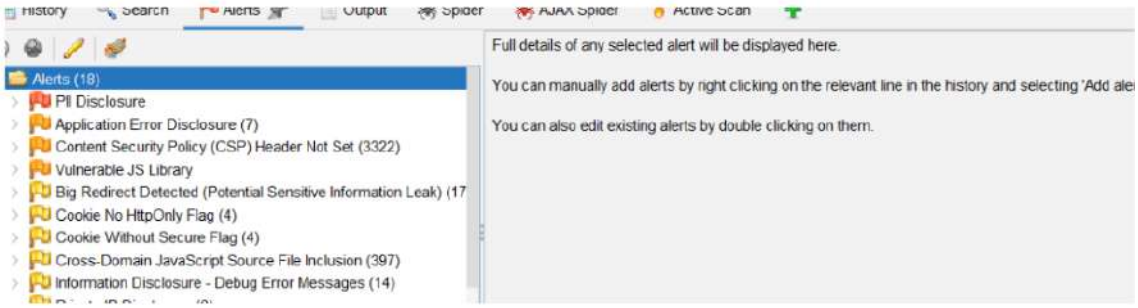
Setelah itu, proses *Penetration Testing (scanning attcak*) berlangsung dengan beberapa hal yang perlu diperhatikan: pemindaian telah mencapai 20%, daftar URL yang diproses mencakup metode HTTP serta URL yang dianalisis, ikon warna menunjukkan status (hijau untuk risiko rendah, merah untuk risiko tinggi), URL yang ditandai sebagai *Out of Scope* tidak ikut dipindai, dan bagian *Messages* menampilkan *log* atau notifikasi tambahan yang membantu dalam *debugging*. Hasil dapat dilihat pada Gambar 3.



Gambar 3. Proses *Penetration Testing*

c. Hasil *Penetration Testing*

Setelah uji penetrasi dilakukan pada situs web, akan muncul alert yang menampilkan berbagai kerentanan yang terdeteksi. Gambar 4. menunjukkan hasil kerentanan yang ditemukan selama proses penetration testing.



Gambar 4. Hasil *Penetration Testing*

Gamabr 4. Hasil pengujian menunjukkan bahwa website yang diuji memiliki 18 kerentanan. Dari jumlah tersebut, satu kerentanan berada pada tingkat tinggi, sehingga pemilik website perlu memberikan perhatian penuh dan segera melakukan perbaikan untuk mengatasi masalah keamanan tersebut.

3.2 Reporting dan Mitigasi

Pada tahapan ini, seluruh kerentanan yang berhasil diidentifikasi selama proses pengujian akan didokumentasikan dan dilaporkan secara rinci. Laporan tersebut mencakup deskripsi setiap kerentanan, tingkat keparahannya, serta potensi dampak yang dapat ditimbulkan jika kerentanan tersebut dieksploitasi. Selain itu, tahapan ini juga memberikan rekomendasi mitigasi atau solusi perbaikan untuk setiap alert yang ditemukan, termasuk langkah-langkah teknis yang dapat diimplementasikan untuk memperbaiki celah keamanan tersebut. Tujuannya adalah memastikan bahwa pemilik website dapat mengambil tindakan yang tepat untuk meningkatkan keamanan sistem dan mengurangi risiko serangan siber di masa mendatang. Hasil dapat dilihat pada tabel 2.

Tabel 2. Reporting dan mitigasi kerentanan

Kerentanan	***Alerts***	**Deskripsi**	**Dampak**	**Mitigasi**
A01: *Broken Access Control*	Information Disclosure - Debug Error Messages	Responsnya tampaknya berisi pesan kesalahan umum yang dikembalikan oleh platform seperti ASP.NET, dan server Web seperti IIS dan Apache. Anda dapat mengonfigurasi daftar pesan debug umum.	• Akses Data Sensitif Pengguna dapat mengakses informasi rahasia, seperti data pribadi, informasi keuangan, atau data perusahaan. • Perubahan Data Tanpa Izin Pengguna yang tidak sah dapat memodifikasi atau menghapus data penting, seperti catatan pelanggan, inventaris, atau laporan. • Eskalasi Privilege Pengguna biasa dapat meningkatkan hak aksesnya ke tingkat	• Least Privilege Berikan hak akses hanya sesuai kebutuhan pengguna, hindari akses berlebih. • Validasi dan Verifikasi Akses Gunakan autentikasi kuat (OAuth, SAML) dan cek izin pengguna sebelum memberikan akses. • Role-Based Access Control (RBAC) Tentukan peran dengan hak akses spesifik untuk kemudahan pengelolaan. • Validasi Server-Side Lakukan kontrol akses di sisi server, bukan

			administratif, yang dapat membahayakan seluruh sistem. • Kehilangan Kepercayaan Kerentanan ini dapat menyebabkan	hanya pada frontend. • Uji Keamanan Gunakan alat otomatis (OWASP ZAP, BurpSuite) dan lakukan pengujian manual
A03: *Injection*	Big Redirect Detected (Potential Sensitive Information Leak)	Server telah merespons dengan pengalihan yang sepertinya memberikan respons besar. Hal ini mungkin menunjukkan bahwa meskipun server mengirimkan pengalihan, server juga merespons dengan konten isi (yang mungkin mencakup detail sensitif, PII, dll.).	1. Pencurian Data Penyerang dapat mengakses data sensitif dalam database. 2. Modifikasi Data Data penting dapat diubah, dihapus, atau disisipkan secara tidak sah. 3. Eskalasi Privilege Penyerang bisa mendapatkan akses administratif ke sistem. 4. Kerusakan Sistem Perintah berbahaya dapat menyebabkan sistem tidak stabil atau rusak. 5. Kehilangan Kepercayaan Insiden dapat merusak reputasi organisasi dan menyebabkan kerugian finansial.	• Gunakan Parameterized Queries atau ORM Hindari concatenation query, gunakan prepared statements atau Object-Relational Mapping (ORM). • Validasi Input Validasi dan sanitasi semua data masukan sesuai tipe yang diharapkan. • Gunakan Stored Procedures Batasi interaksi langsung dengan database menggunakan prosedur tersimpan. • Least Privilege Berikan hak akses

		minimal pada akun database untuk membatasi kerusakan. • Escaping Input Gunakan mekanisme escaping pada input jika tidak dapat menggunakan parameterized queries. • Audit dan Pengujian Lakukan pengujian aplikasi (seperti SQLMap atau OWASP ZAP) untuk mendeteksi kerentanan injection. • Pembaruan Rutin Selalu perbarui sistem dan pustaka untuk mengatasi kelemahan yang diketahui.

A04: *Insecure Design*	PII Disclosure	Responsnya berisi Informasi Identifikasi Pribadi, seperti nomor CC, SSN, dan data sensitif serupa	1. Eksploitasi Fitur yang Tidak Aman Penyerang dapat memanfaatkan logika bisnis atau desain yang cacat. 2. Kehilangan Data Data sensitif dapat diakses, dimodifikasi, atau dihapus oleh pihak tidak sah. 3. Kerentanan Keamanan Tambahan Desain yang lemah membuka jalan bagi serangan lain seperti injection atau broken access control. 4. Kegagalan Sistem Sistem dapat menjadi tidak stabil atau berhenti beroperasi karena desain yang buruk. 5. Kerugian Reputasi dan Finansial Kehilangan kepercayaan pelanggan dan biaya tinggi untuk perbaikan pasca-insiden.	• Integrasi Secure Development Lifecycle (SDLC) Terapkan keamanan sejak tahap desain hingga implementasi. • Analisis Risiko Identifikasi potensi ancaman dan desain sistem untuk memitigasi risiko. • Prinsip Least Privilege Minimalisasi hak akses dan kontrol terhadap semua fungsi sistem. • Validasi Input dan Output Pastikan semua data masukan dan keluaran memenuhi aturan keamanan. • Uji Desain secara Berkala Lakukan penilaian keamanan desain oleh tim internal atau eksternal. • Penerapan Best Practices Ikuti panduan seperti OWASP

		Secure Design Principles. • Pembaruan dan Evaluasi Rutin Tinjau dan tingkatkan desain sesuai Perkembangan ancaman keamanan.

4. KESIMPULAN

Penetration Testing dengan OWASP ZAP adalah alat yang efektif untuk mengidentifikasi dan mengevaluasi kerentanan keamanan pada website. Kerentanan yang terdeteksi, seperti A01, A03, dan A04, meliputi ancaman seperti Cross-Site Scripting (XSS), Clickjacking, dan Man-in-the-Middle, sehingga langkah mitigasi seperti konfigurasi header keamanan dan perlindungan data sensitif sangat diperlukan untuk mencegah eksploitasi. Selain mendeteksi dan memperbaiki ancaman, OWASP ZAP juga meningkatkan kesadaran akan pentingnya kebijakan keamanan yang kuat dalam aplikasi web. Dengan demikian, pemilik website dapat lebih melindungi data sensitif, mempertahankan kepercayaan pengguna, dan menjaga keamanan di lingkungan digital yang semakin kompleks.

5. DAFTAR PUSTAKA

[1] G. Kusuma, "Implementasi OWASP Zap Untuk Pengujian Keamanan Sistem Informasi Akademik," *J. Teknol. Inf. J. Keilmuan dan Apl. Bid. Tek. Inform.*, vol. 16, no. 2, hal. 178–186, 2022, doi: 10.47111/jti.v16i2.3995.

[2] B. Subana, A. Fadlil, dan Sunardi, "Web Server Security Analysis Using The OWASP Mantra Method," *J. Mantik*, vol. 4, no. 36, hal. 107–116, 2020, [Daring]. Tersedia pada: https://iocscience.org/ejournal/index.php/mantik/index.

[3] I. G. A. S. P. Wijaya, G. M. A. Sasmita, dan I. P. A. E. Pratama, "Web Application Penetration Testing on Udayana University's OASE E-learning Platform Using Information System Security Assessment Framework (ISSAF) and Open Source Security Testing Methodology Manual (OSSTMM)," *Int. J. Inf. Technol. Comput. Sci.*, vol. 16, no. 2, hal. 45–56, 2024, doi: 10.5815/ijitcs.2024.02.04.

[4] M. Agreindra Helmiawan, E. Firmansyah, I. Fadil, Y. Sofivan, F. Mahardika, dan A. Guntara, "Analysis of Web Security Using Open Web Application Security Project 10," in *2020 8th International Conference on Cyber and IT Service Management, CITSM 2020*, 2020, hal. 1–5, doi: 10.1109/CITSM50537.2020.9268856.

[5] S. Thakre dan S. Bojewar, "Studying the Effectiveness of Various Tools in Detecting the Protecting Mechanisms Implemented in Web-Applications," *Proc. Int. Conf. Inven. Res. Comput. Appl. ICIRCA 2018*, no. Icirca, hal. 1316–1321, 2018, doi: 10.1109/ICIRCA.2018.8597363.

[6] U. Ravindran dan R. V. Potukuchi, "A Review on Web Application Vulnerability Assessment and

Penetration Testing," *Rev. Comput. Eng. Stud.*, vol. 9, no. 1, hal. 1–22, 2022, doi: 10.18280/rces.090101.

[7] F. M. M. Mokbal, W. Dan, A. Imran, L. Jiuchuan, F. Akhtar, dan W. Xiaoxi, "MLPXSS: An Integrated XSS-Based Attack Detection Scheme in Web Applications using Multilayer Perceptron Technique," *IEEE Access*, vol. 7, hal. 1–14, 2019, doi: 10.1109/ACCESS.2019.2927417.

[8] H. Saputra, A. Z. Abidin, F. Faldi, dan M. T. Sumadi, "Penetration Testing on Mail Server Website using the OWASP Method," *J. Mandiri IT*, vol. 12, no. 2, hal. 58–65, 2023, [Daring]. Tersedia pada: https://ejournal.isha.or.id/index.php/Mandiri/article/view/232.

[9] A. Wijayanto, E. Utami, dan A. B. Prasetio, "Analysis of Vulnerability Webserver Office Management of Information and Documentation Diskominfo using OWASP Scanner," in *2020 2nd International Conference on Cybernetics and Intelligent System, ICORIS 2020*, 2020, hal. 1–5, doi: 10.1109/ICORIS50180.2020.9320833.

[10] A. Fadlil, I. Riadi, dan M. A. Mu'Min, "Mitigation from SQL Injection Attacks on Web Server using Open Web Application Security Project Framework," *Int. J. Eng. Trans. A Basics*, vol. 37, no. 4, hal. 635–645, 2024, doi: 10.5829/ije.2024.37.04a.06.

[11] Carlos P. Flores Jr., "Evaluation of Common Security Vulnerabilities of State Universities and Colleges Websites Based on OWASP," *J. Electr. Syst.*, vol. 20, no. 5s, hal. 1396–1404, 2024, doi: 10.52783/jes.2471.

[12] I. Riadi, A. Fadlil, dan M. Amirul, "OWASP Framework-Based Network Forensics to Analyze the SQLi Attacks on Web Servers," *Matrik J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 22, no. 3, hal. 481–493, 2023, doi: 10.30812/matrik.v22i3.3018.

[13] Sunardi, I. Riadi, dan P. A. Raharja, "Vulnerability Analysis of E-Voting Application using Open Web Application Security Project (OWASP) Framework," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 11, hal. 135–143, 2019, doi: 10.14569/IJACSA.2019.0101118.

[14] Nurbojatmiko, A. Lathifah, F. Bil Amri, dan A. Rosidah, "Security Vulnerability Analysis of the Sharia Crowdfunding Website Using OWASP-ZAP," *2022 10th Int. Conf. Cyber IT Serv. Manag. CITSM 2022*, no. September, 2022, doi: 10.1109/CITSM56380.2022.9935837.

[15] M. Alenezi, M. Nadeem, dan R. Asif, "SQL Injection Attacks Countermeasures Assessments," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 21, no. 2, hal. 1121–1131, 2020, doi: 10.11591/ijeecs.v21.i2.pp1121-1131.