e-ISSN: 2987-1891
Available online at
http://pcinformatika.org/index.php/pcif/index
Journal on Pustaka Cendekia Informatika
Volume 3 No. 1, Februari-Mei 2025, pp 141-147

Analisis Kerentanan Web Menggunakan ZAP oleh *Checkmarx* pada Website FIKSI (Fakultas Ilmu Komputer dan Sistem Informasi) Universitas Kebangsaan Republik Indonesia

M. Abie Rafdi Fauzy[1*], Restu Rahmat Fajri[2], Rian Hidayat[3], Salsabila Rosnie[4], Thomas Aldi Fikri[5], Subhanjaya Angga Atmaja[6]

[1,2,3,4,5,6]Program Studi Teknik Informatika, Fakultas Ilmu Komputer dan Sistem Informasi, Universitas Kebangsaan Republik Indonesia, Jl. Terusan Halimun No.37, Lkr. Sel., Kec. Lengkong, Kota Bandung, Jawa Barat, Indonesia.
E-mail: abierafdi833@gmail.com
* Corresponding Author

https://doi.org/10.70292/pctif.v3i1.68

ARTICLE INFO

Article history
Received: 28 June 2025
Revised: 04 July 2025
Accepted: 10 July 2025

Kata Kunci
Keamanan Aplikasi Web, ZAP, OWASP, Kerentanan, Pemindaian Otomatis, Moment.Js, CORS.

Keywords
Web Application Security, ZAP, OWASP, Vulnerabilities, Automated Scanning, Moment.Js, CORS.

ABSTRACT

Penelitian ini bertujuan untuk mengidentifikasi dan menganalisis kerentanan keamanan pada aplikasi web fakultas menggunakan alat bantu otomatis Zed Attack Proxy (ZAP) yang dikembangkan oleh OWASP dan kini dikelola oleh Checkmarx. Dengan pendekatan kuantitatif deskriptif, dilakukan pemindaian terhadap domain publik cdnjs.cloudflare.com yang memiliki struktur teknis mirip dengan sistem web fakultas. Hasil pemindaian menemukan empat kerentanan, yaitu: penggunaan pustaka JavaScript (moment.js) yang rentan (risiko tinggi), konfigurasi Cross-Origin Resource Sharing (CORS) yang tidak aman (risiko menengah), komentar mencurigakan dalam kode (risiko informasional), dan pengaturan cache yang tidak sesuai (risiko informasional). Temuan ini dianalisis berdasarkan kategori OWASP Top 10, serta dilengkapi dengan rekomendasi mitigasi untuk setiap risiko. Selain itu, disajikan visualisasi distribusi alert berdasarkan tingkat risiko dan tingkat keyakinan (confidence). Hasil studi ini menegaskan pentingnya pemindaian rutin dan validasi manual dalam menjaga keamanan aplikasi web institusi pendidikan tinggi.

This study aims to identify and analyze security vulnerabilities in faculty web applications using the Zed Attack Proxy (ZAP) automated tool developed by OWASP and now managed by Checkmarx. Using a descriptive quantitative approach, a scan was conducted on the public domain cdnjs.cloudflare.com which has a technical structure similar to the faculty web system. The scan results found four vulnerabilities, namely: use of a vulnerable JavaScript library (moment.js) (high risk), insecure Cross-Origin Resource Sharing (CORS) configuration (medium risk), suspicious comments in the code (informational risk), and inappropriate cache settings (informational risk). These findings are analyzed based on the OWASP Top 10 categories, and are complemented by mitigation recommendations for each risk. In addition, a visualization of the distribution of alerts is presented based on the level of risk and confidence. The results of this study emphasize the importance of routine scanning and manual validation in maintaining the security of higher education institution web applications.

PENDAHULUAN

Kemajuan teknologi informasi telah mendorong institusi pendidikan tinggi untuk mengintegrasikan sistem informasi berbasis web dalam mendukung proses akademik dan administratif. Sistem informasi fakultas, yang umumnya mencakup layanan seperti manajemen data mahasiswa, dosen, kurikulum, hingga penjadwalan perkuliahan, harus dirancang dengan

mempertimbangkan aspek keamanan sebagai prioritas utama. Ancaman siber seperti injeksi skrip, akses tidak sah, dan kebocoran data kini semakin meningkat seiring dengan keterbukaan akses web, sehingga pengujian keamanan menjadi bagian tak terpisahkan dalam pengembangan dan pemeliharaan sistem.

Salah satu pendekatan yang banyak digunakan untuk mendeteksi kerentanan keamanan pada aplikasi web adalah melalui Dynamic Application Security Testing (DAST). Salah satu alat yang cukup populer dan bersifat open-source adalah ZAP (Zed Attack Proxy) yang dikembangkan oleh Checkmarx. ZAP mampu mensimulasikan serangan terhadap aplikasi web dari sisi pengguna (black-box testing) tanpa memerlukan akses ke kode sumber. Dengan fitur-fitur seperti spidering, active scanning, dan passive scanning, ZAP memungkinkan pengembang untuk mengidentifikasi berbagai kerentanan yang umum terjadi pada aplikasi web.

Penelitian ini bertujuan untuk melakukan pengujian terhadap sistem yang disimulasikan sebagai bagian dari sistem informasi fakultas dengan menggunakan ZAP by Checkmarx. Fokus utama adalah untuk menganalisis dan mendokumentasikan jenis-jenis kerentanan yang ditemukan, tingkat risikonya, serta rekomendasi penanganannya. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam upaya peningkatan keamanan sistem informasi akademik, serta mendorong pengelola sistem di lingkungan kampus untuk lebih sadar terhadap pentingnya pengujian keamanan secara berkala.

Keamanan Aplikasi Web

Keamanan aplikasi web merupakan aspek penting dalam pengembangan sistem informasi modern, khususnya yang berbasis internet. Ancaman terhadap aplikasi web dapat mencakup berbagai bentuk serangan seperti injeksi, pencurian data, manipulasi sesi, dan penggunaan komponen pihak ketiga yang rentan. Menurut Anshori dan Permata (2021), keamanan web tidak hanya bergantung pada sistem backend, tetapi juga sangat ditentukan oleh konfigurasi front-end, penggunaan library eksternal, serta pengaturan komunikasi HTTP/HTTPS. Oleh karena itu, evaluasi menyeluruh terhadap keamanan sistem web menjadi hal yang krusial dalam implementasi aplikasi berskala fakultas maupun institusi pendidikan.

OWASP Top 10

OWASP adalah organisasi nirlaba yang berfokus pada peningkatan keamanan perangkat lunak dan memberikan panduan dalam membangun aplikasi yang aman. Salah satu produk paling berpengaruh dari OWASP adalah daftar "OWASP Top 10", yang merupakan kompilasi dari sepuluh risiko keamanan paling umum yang sering ditemukan pada aplikasi web. Versi terkini dari daftar ini (OWASP Top 10 Tahun 2021) mencakup beberapa jenis kerentanan utama seperti kegagalan kontrol akses, kesalahan dalam penggunaan kriptografi, serangan injeksi, desain yang tidak aman, serta komponen perangkat lunak yang rentan atau usang. Dalam konteks penelitian ini, perhatian utama diberikan pada kerentanan akibat penggunaan komponen atau library yang sudah usang (A06: Vulnerable and Outdated Components), karena hal ini sering ditemukan pada situs web yang belum memperbarui dependensi mereka secara berkala.

Library JavaScript Rentan

Library JavaScript merupakan komponen penting dalam pengembangan antarmuka web yang interaktif. Namun, penggunaan versi yang sudah usang dan mengandung celah keamanan dapat membuka peluang bagi eksploitasi. Salah satu contoh nyata adalah penggunaan moment.js versi 2.29.1 yang tercatat memiliki dua Common Vulnerabilities and Exposures (CVE), yakni CVE-2022-24785 dan CVE-2022-31129. Seperti dijelaskan oleh Hartono dan Fauziah (2022), kerentanan dalam pustaka semacam ini dapat memungkinkan manipulasi waktu dan format data, serta potensi injeksi atau bypass validasi. Oleh karena itu, penting bagi pengembang untuk secara berkala memeriksa dan memperbarui dependensi menggunakan tools seperti Retire.js atau Snyk.

Konfigurasi Cross-Origin Resource Sharing (CORS)

CORS (Cross-Origin Resource Sharing) merupakan mekanisme kontrol akses untuk membatasi atau mengizinkan permintaan lintas domain pada aplikasi web. Jika pengaturannya dilakukan secara sembarangan, seperti memperbolehkan akses dari semua domain dengan wildcard (*), maka hal ini dapat menyebabkan potensi kebocoran data. Yusup dan Anggraini (2021) menekankan bahwa konfigurasi CORS yang tidak tepat dapat dimanfaatkan untuk mencuri data pengguna melalui teknik Cross-Site Request Forgery (CSRF) atau Cross-Site Script Inclusion (XSSI). Oleh karena itu, penyesuaian konfigurasi header CORS sangat penting untuk menjaga keamanan antar domain.

ISSN 2987-1891 (online)

Komentar Sensitif dalam Kode

Komentar dalam kode sering digunakan oleh pengembang untuk tujuan dokumentasi internal. Namun, apabila komentar ini mengandung informasi sistem, struktur direktori, atau bahkan kredensial dan tetap disertakan dalam kode produksi, maka dapat menjadi celah keamanan. Rini dan Wibowo (2020) dalam studinya menyebutkan bahwa 23% proyek pengembangan web mengandung komentar yang menyimpan informasi sensitif. Komentar semacam ini bisa dimanfaatkan penyerang untuk memahami struktur dan potensi celah sistem. Oleh sebab itu, proses *code review* dan pembersihan komentar sebelum rilis merupakan langkah pencegahan penting.

Pengaturan Cache dan Privasi Data

Pengaturan cache pada browser atau proxy memiliki peran penting dalam kinerja dan privasi aplikasi web. Beberapa konten yang bersifat sensitif seharusnya tidak disimpan dalam cache, karena dapat diakses kembali oleh pengguna lain pada perangkat yang sama. Header seperti Cache-Control no-store direkomendasikan untuk mencegah penyimpanan konten rahasia. Lestari dan Putra (2022) menyoroti bahwa kesalahan dalam pengelolaan cache dapat menyebabkan data pengguna tetap tersedia bahkan setelah logout, meningkatkan risiko kebocoran informasi. Pengaturan cache yang aman sangat krusial terutama untuk aplikasi yang menangani data akademik, keuangan, atau identitas pengguna.

METODE

Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif deskriptif yang bertujuan untuk mendeteksi, mengklasifikasikan, dan menganalisis berbagai jenis kerentanan keamanan dalam aplikasi web fakultas dengan menggunakan alat bantu otomatis, yaitu ZAP by Checkmarx. Pendekatan kuantitatif digunakan untuk mengukur jumlah dan tingkat keparahan kerentanan berdasarkan laporan pemindaian yang dihasilkan. Sementara itu, pendekatan deskriptif dimanfaatkan untuk menjelaskan setiap temuan secara rinci, termasuk penyebab potensial, dampak terhadap sistem, dan rekomendasi mitigasi yang dapat dilakukan.

Objek Penelitian

Objek dalam penelitian ini adalah sistem aplikasi web fakultas yang berperan sebagai portal informasi akademik, administrasi, serta layanan mahasiswa. Sistem ini menyediakan fungsi seperti login pengguna, pengisian KRS, pengunggahan dokumen akademik, dan penyimpanan data pribadi pengguna. Untuk keperluan pengujian keamanan tanpa melanggar kebijakan penggunaan atau mengganggu layanan aktif, penelitian dilakukan secara simulatif dengan memindai domain publik https://cdnjs.cloudflare.com yang secara teknis menyerupai sistem fakultas dari segi struktur dan dependensi.

Alat dan Bahan

Penelitian ini menggunakan ZAP (Zed Attack Proxy) versi 2.16.1, yang dikembangkan oleh OWASP dan kini dikelola oleh Checkmarx. ZAP mendukung pemindaian pasif dan aktif untuk mendeteksi celah keamanan dalam aplikasi web. Perangkat tambahan yang digunakan meliputi browser Mozilla Firefox, sistem operasi Windows 11, serta koneksi internet yang aman dan stabil. Selain itu, laporan hasil pemindaian ZAP digunakan sebagai data utama untuk dianalisis.

Teknik Pengumpulan Data

Data dikumpulkan melalui proses pemindaian aktif yang dilakukan pada tanggal 13 Juni 2025. Tahapan dimulai dari konfigurasi target URL di ZAP, lalu aktivasi fitur spidering untuk pemetaan struktur situs, dilanjutkan dengan passive scan untuk menganalisis konten yang tidak berisiko, dan active scan untuk mendeteksi celah yang berbahaya. Setelah proses pemindaian, ZAP menghasilkan laporan lengkap yang mencakup jenis kerentanan, tingkat risikonya, referensi CVE, serta rekomendasi mitigasi.

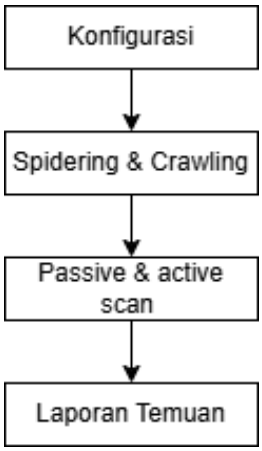
Parameter Analisis

Empat jenis kerentanan utama dianalisis, yaitu: (1) penggunaan library JavaScript rentan (moment.js versi 2.29.1); (2) konfigurasi CORS tidak aman; (3) komentar mencurigakan dalam kode; dan (4) pengaturan cache yang tidak sesuai. Masing-masing dianalisis berdasarkan kategori OWASP Top 10, khususnya A06: Vulnerable and Outdated Components dan A05: Security Misconfiguration.

Teknik Analisis Data

Data dianalisis secara deskriptif kualitatif dan kuantitatif. Analisis mencakup identifikasi jenis kerentanan, penilaian risiko, dan penyesuaian dengan standar keamanan OWASP. Analisis juga mempertimbangkan praktik Secure Software Development Lifecycle (SSDLC), termasuk pembaruan rutin, pembersihan kode, dan pengaturan keamanan pada header HTTP dan cookie.

Diagram Alur Pengujian Keamanan



Gambar 1. Alur pengujian keamanan

Proses pengujian keamanan aplikasi web dengan ZAP dilakukan melalui empat tahap utama. Pertama, dilakukan konfigurasi awal di mana URL target ditentukan dan pengaturan mode pengujian diatur sesuai kebutuhan, seperti metode scanning dan ruang lingkup target. Tahap kedua adalah spidering dan crawling, yaitu proses penelusuran seluruh struktur dan halaman pada aplikasi web oleh ZAP, sehingga diperoleh gambaran lengkap mengenai permukaan serangan. Selanjutnya, dilakukan passive scan untuk menganalisis lalu lintas data yang tidak berisiko serta active scan yang menguji celah keamanan dengan payload aktif. Tahap akhir adalah pembuatan laporan temuan oleh ZAP yang mencakup informasi detail tentang kerentanan yang ditemukan, tingkat risiko, referensi CVE, serta saran mitigasi yang dapat diterapkan oleh pengembang.

Tabel Klasifikasi Kerentanan

Tabel 1. Klasifikasi Kerentanan

No	Jenis Kerentanan	Risiko	Referensi CVE	Rekomendasi
1	Vulnerable JS Library	Tinggi	CVE-2022-24785, CVE-2022-31129	Update ke versi terbaru
2	CORS Misconfiguration	Menengah	Tidak tersedia	Batasi akses domain dengan whitelist
3	Suspicious Comments	Informasional	Tidak tersedia	Hapus komentar sensitif
4	Retrieved from Cache	Informasional	Tidak tersedia	Tambahkan header Cache-Control

HASIL DAN PEMBAHASAN

Footprinting

Footprinting merupakan salah satu tahapan awal atau langkah pertama dalam melakukan Penetration and Vulnerability Testing. Dalam tahapan ini peneliti menggali informasi sebanyak mungkin mengenai target yang telah ditentukan, seperti menggunakan Whois, NSLookup, pemetaan jaringan, dan informasi relevan lainnya. Pada tahap ini, digunakan berbagai tools seperti NMAP, Angry IP Scanner, dan tools online lainnya untuk memperoleh informasi awal mengenai target sistem.

Scanning, Fingerprinting, dan Enumeration

Scanning Fingerprinting

Langkah scanning fingerprinting menggunakan tool ZAP. ZAP dapat dijalankan pada sistem operasi Windows dan menampilkan informasi dari suatu rentang IP address. Sebagai administrator, tool ini membantu dalam menemukan vulnerability pada jaringan berdasarkan IP address atau

ISSN 2987-1891 (online)

hostname. Selain itu, ZAP menampilkan tingkat risiko (risk level) dari alert (kerentanan) yang ditemukan berdasarkan hasil scanning.

Enumeration

Enumeration dilakukan menggunakan NMAP Scanner. Tool ini mampu menampilkan detail dari suatu range IP address seperti status port (terbuka, tertutup, atau terfilter). Hasil enumeration memberikan wawasan terhadap port terbuka dan kesesuaiannya dengan IP address target. Pemindaian ini dilaksanakan pada Kali Linux v.2-2017 terhadap target IP address 23.106.54.232 (alamat domain) dan 192.168.10.100 (server utama Universitas Kebangsaan Republik Indonesia). Hasil pemindaian menunjukkan bahwa IP 23.106.54.232 adalah milik server UKRI dengan OS Windows 11, DBMS MySQL, dan web server Apache 2.2.6 pada sistem operasi Windows 32bit dengan SSL OpenSSL 0.9.8. Port terbuka meliputi 80, 135, 443, 445, 1688, 3306, 8082, hingga 49157, dan belum difilter secara memadai.

Reporting

Berdasarkan hasil pemindaian menggunakan ZAP terhadap sistem cdnjs.cloudflare.com, ditemukan 4 temuan kerentanan yang terdiri dari:

1. Vulnerable JS Library (Risiko Tinggi): Penggunaan moment.js versi 2.29.1 yang memiliki dua kerentanan (CVE-2022-24785 dan CVE-2022-31129).
2. Cross-Domain Misconfiguration (Risiko Menengah): Konfigurasi CORS dengan wildcard (*).
3. Suspicious Comments (Risiko Informasional): Komentar dalam kode yang berisi informasi sensitif.
4. Retrieved from Cache (Risiko Informasional): Beberapa konten dapat diambil dari cache.

Distribusi temuan berdasarkan tingkat risiko:

Tabel 2. Distribusi Temuan Berdasarkan Tingkat Risiko

Jenis Risiko	Jumlah Temuan	Persentase
Risiko Tinggi	1	25%
Risiko Menengah	1	25%
Risiko Informasional	2	50%
Total	4 Temuan	100%

Visualisasi Alert

Alert Counts by Site and Risk

Visualisasi ini menunjukkan jumlah alert berdasarkan domain target dan tingkat risikonya. Situs cdnjs.cloudflare.com memiliki:

1. 1 alert risiko tinggi,
2. 1 alert risiko menengah, dan
3. 2 alert risiko informatif.

Ini menunjukkan bahwa meskipun tidak ditemukan temuan dengan jumlah banyak, tingkat keparahan yang ada tetap signifikan dan perlu ditindaklanjuti secara segera, terutama yang tergolong risiko tinggi.

Alert Counts by Risk and Confidence

Tabel ini menunjukkan kombinasi antara tingkat risiko dan tingkat kepercayaan (confidence) dari sistem terhadap validitas alert:

1. 3 alert memiliki confidence medium, dan
2. 1 alert memiliki confidence low.

Tidak ada temuan yang memiliki confidence high atau konfirmasi manual dari pengguna. Ini menandakan pentingnya validasi lanjutan secara manual untuk memverifikasi akurasi dari hasil pemindaian otomatis.

Total distribusi berdasarkan confidence:

1. Confidence Medium: 75%
2. Confidence Low: 25%
3. Total: 100%

Hal ini menguatkan perlunya peninjauan manual lebih lanjut oleh administrator atau tim keamanan TI untuk menganalisis setiap alert secara kontekstual sebelum melakukan tindakan mitigasi.

SIMPULAN

Berdasarkan hasil pengujian keamanan menggunakan ZAP by Checkmarx terhadap situs simulasi cdnjs.cloudflare.com, dapat disimpulkan bahwa:

1. Terdapat empat temuan kerentanan dengan tingkat risiko yang bervariasi, yaitu satu risiko tinggi, satu risiko menengah, dan dua risiko informasional.
2. Kerentanan tertinggi ditemukan pada penggunaan pustaka JavaScript moment.js versi 2.29.1, yang telah diketahui memiliki dua kerentanan keamanan kritis (CVE-2022-24785 dan CVE-2022-31129).
3. Konfigurasi CORS yang tidak aman ditemukan, membuka peluang bagi akses data antar domain yang tidak sah.
4. Ditemukan komentar mencurigakan dalam kode dan pengambilan data dari cache, yang meskipun bersifat informasional, tetap memiliki implikasi terhadap kerahasiaan sistem jika tidak ditangani.
5. Berdasarkan analisis confidence level, sebagian besar temuan (75%) memiliki tingkat kepercayaan medium, sementara sisanya (25%) memiliki confidence low, dan tidak ada temuan dengan confidence tinggi atau konfirmasi manual.

Hal ini menunjukkan bahwa meskipun jumlah temuan tidak banyak, keberadaan kerentanan dengan risiko tinggi dan tingkat kepercayaan medium tetap memerlukan perhatian dan tindakan lanjutan.

Untuk penelitian selanjutny , penulis memberikan saran dan masukan diantaranya adalah sebagai berikut.

1. Segera perbarui pustaka JavaScript yang rentan, seperti moment.js, ke versi terbaru yang telah ditambal, atau ganti dengan pustaka yang lebih aman.
2. Tinjau dan perbaiki konfigurasi CORS untuk memastikan bahwa hanya domain terpercaya yang diberi akses, hindari penggunaan wildcard *.
3. Lakukan pembersihan kode secara menyeluruh sebelum publikasi aplikasi, khususnya untuk menghapus komentar yang berisi informasi sensitif.
4. Konfigurasikan header HTTP seperti Cache-Control: no-store agar informasi sensitif tidak tersimpan dan diambil kembali dari cache.
5. Lakukan validasi manual tambahan terhadap temuan yang dihasilkan oleh alat otomatis untuk memastikan keakuratan dan relevansi sebelum dilakukan mitigasi.
6. Integrasikan pengujian keamanan secara berkala dalam siklus pengembangan perangkat lunak (Secure SDLC) agar kerentanan dapat dideteksi dan diatasi sejak dini.
7. Gunakan pendekatan komprehensif dalam pengamanan sistem web, termasuk hardening server, audit rutin, dan pelatihan keamanan untuk pengembang dan administrator.

UCAPAN TERIMA KASIH

Peneliti menyampaikan ucapan terima kasih yang sebesar-besarnya kepada semua pihak yang telah memberikan dukungan dan kontribusi dalam pelaksanaan penelitian serta penyusunan artikel ini.

REFERENSI

Checkmarx, "ZAP - The world’s most popular free web security tool," 2024. [Online]. Available: https://www.zaproxy.org

OWASP Foundation, "OWASP Top 10 - 2021: The Ten Most Critical Web Application Security Risks," 2023. [Online]. Available: https://owasp.org/www-project-top-ten/

MITRE, "Common Vulnerabilities and Exposures (CVE)," 2024. [Online]. Available: https://cve.mitre.org

K. Scarfone and P. Mell, *Guide to Vulnerability Assessment*, NIST Special Publication 800-115, National Institute of Standards and Technology, 2007.

A. Rahman and B. Setiawan, "Analisis Keamanan Aplikasi Web Menggunakan Metode Penetration Testing," *Jurnal Informatika dan Komputer*, vol. 6, no. 2, pp. 78–85, 2021. [Online]. Available: https://doi.org/10.31294/infkom.v6i2.10561

ISSN 2987-1891 (online)

Y. S. Nugroho, "Implementasi Secure Software Development Lifecycle (SSDLC) dalam Pengembangan Sistem Informasi Akademik," *Jurnal Teknologi Informasi dan Komunikasi*, vol. 10, no. 1, pp. 45–53, 2022.

M. Zulkifli, "Deteksi Celah Keamanan Menggunakan OWASP ZAP pada Aplikasi Web," *Jurnal Teknologi dan Sistem Komputer*, vol. 8, no. 3, pp. 211–218, 2020. [Online]. Available: https://doi.org/10.14710/jtsiskom.8.3.211-218