**Jurnal Ilmiah Universitas Batanghari Jambi** e-ISSN: 2549-4236, p-ISSN: 1411-8939
Lembaga Penellitian dan Pengabdian kepada Masyarakat
Vol 24, No 3 (2024): Oktober, 2451-2457
DOI: 10.33087/jiubj.v24i3.5488

# Perbandingan Evaluasi Kerentanan Menggunakan *Tenable Nessus Scanner* dan *Owasp Zed Attack Proxy* untuk Meningkatkan Keamanan Sistem Informasi Kepegawaian di Universitas Merdeka Malang

**Rizca Wenny, Fandi Yulian Pamuji**
Sistem Informasi Universitas Merdeka Malang
Correspondence: 20083000005@student.unmer.ac.id; fandi.pamuji@unmer.ac.id

**Abstrak.** Penelitian ini bertujuan untuk membandingkan analisis kerentanan antara Tenable Nessus Scanner dan OWASP Zed Attack Proxy (ZAP) untuk meningkatkan keamanan situs web Sistem Informasi Kepegawaian (SIMPEG) di Universitas Merdeka Malang. Metodologi penelitian mencakup penggunaan alat Nessus dan OWASP ZAP untuk memindai situs web SIMPEG terhadap potensi kerentanan. Temuan penelitian ini menunjukkan bahwa OWASP ZAP mengidentifikasi beberapa kerentanan aplikasi web yang kritis seperti ketiadaan token Anti-CSRF, tidak adanya header Content Security Policy (CSP), dan header Anti-Clickjacking yang hilang, yang sangat penting untuk menjaga keamanan dan integritas data pengguna. Di sisi lain, Nessus Scanner lebih fokus pada kerentanan infrastruktur jaringan dan server. Hasilnya menunjukkan bahwa OWASP ZAP lebih efektif untuk keamanan aplikasi web dalam konteks ini. Rekomendasi diberikan untuk mengatasi kerentanan yang diidentifikasi dan meningkatkan keamanan keseluruhan situs web SIMPEG.

**Kata Kunci:** analisis kerentanan, keamanan web, Tenable Nessus, OWASP ZAP, SIMPEG.

***Abstract.*** *This study aims to compare the vulnerability analysis between Tenable Nessus Scanner and OWASP Zed Attack Proxy (ZAP) for improving the security of the Human Resource Information System (HRIS) website at Universitas Merdeka Malang. The research methodology includes the use of both Nessus and OWASP ZAP tools to scan the HRIS website for potential vulnerabilities. The findings of this research indicate that OWASP ZAP identified several critical web application vulnerabilities such as the absence of Anti-CSRF tokens, lack of Content Security Policy (CSP) headers, and missing Anti-Clickjacking headers, which are essential for maintaining the security and integrity of user data. On the other hand, Nessus Scanner focused more on network and server infrastructure vulnerabilities. The results suggest that OWASP ZAP is more effective for web application security in this context. Recommendations are provided to address the identified vulnerabilities and enhance the overall security of the HRIS website.*

***Keywords:*** *vulnerability analysis, web security, Tenable Nessus, OWASP ZAP, HRIS.*

## PENDAHULUAN

Sistem Informasi Manajemen Kepegawaian adalah sistem informasi terpadu, yang meliputi pendataan pegawai, pengolahan data, prosedur, tata kerja, sumber daya manusia dan teknologi informasi untuk menghasilkan informasi yang cepat, lengkap dan akurat dalam rangka mendukung administrasi kepegawaian (Agung & Arifin, 2020). Sistem Informasi Manajemen Kepegawaian (SIMPEG) adalah sistem informasi yang terpadu yang memungkinkan pengelolaan data kepegawaian yang efektif dan efisien. SIMPEG memudahkan pegawai dalam mengelola informasi kepegawaian dan memahami syarat-syarat untuk peningkatan jabatan, golongan, atau jenis pekerjaan. Selain itu, sistem ini memungkinkan pengolahan data, prosedur, tata kelola kerja, sumber daya manusia, dan teknologi informasi untuk menghasilkan informasi yang cepat dan akurat (Amri & Adi, 2023).

Pembuatan Sistem Informasi Kepegawaian (SIMPEG) di universitas merdeka malang bertujuan untuk mengelola data pegawai secara efisien dan efektif. SIMPEG membantu universitas dalam mencatat informasi lengkap tentang setiap pegawai, termasuk riwayat pendidikan, pengalaman kerja, dan prestasi. Selain itu, SIMPEG mempermudah pencatatan kehadiran, absensi, dan pengelolaan cuti, memungkinkan universitas untuk menjadwalkan kegiatan akademik dan administratif dengan lebih baik. Melalui SIMPEG, proses penggajian dan pengelolaan tunjangan juga dapat dilakukan dengan cepat dan akurat, sesuai dengan kebijakan dan peraturan yang berlaku

(Widyawan & Idris, 2021). Data yang terdapat dalam SIMPEG juga digunakan untuk mengevaluasi kinerja pegawai, serta merencanakan pengembangan sumber daya manusia yang sesuai. Dengan demikian, SIMPEG membantu universitas dalam memenuhi kebutuhan administrasi dan manajemen SDM, serta menjaga kepatuhan terhadap peraturan yang berlaku.

Seringkali, aplikasi web yang telah dipublikasikan di internet rentan terhadap serangan dari pihak yang tidak diinginkan, seperti hacker atau peretas (Taufan, 2022). Aplikasi web, termasuk Sistem Informasi Kepegawaian Universitas Merdeka Malang, rentan terhadap berbagai jenis serangan siber seperti SQL injection, cross-site scripting (XSS), dan denial of service (DoS), yang dapat mengancam keamanan data dari Dosen, Mahasiswa, dan petugas universitas. Ancaman dari pihak yang tidak diinginkan, seperti hacker atau peretas, memiliki potensi untuk mengeksploitasi kerentanan dalam aplikasi web SIMPEG untuk mengakses, memanipulasi, atau bahkan mencuri data sensitif. Oleh karena itu, penting untuk mengambil tindakan pencegahan yang efektif guna melindungi data tersebut dari akses yang tidak sah. Analisis kerentanan menjadi langkah awal yang krusial untuk mengevaluasi keamanan aplikasi web secara menyeluruh, sehingga potensi risiko keamanan yang mungkin terjadi dapat diidentifikasi dengan tepat. Perlindungan data dari Dosen, Mahasiswa, dan petugas universitas menjadi fokus utama, mengingat informasi pribadi, akademik, dan administratif yang sensitif harus dijaga kerahasiaannya.

Nessus Scanner merupakan Pemindai kerentanan yang memiliki tingkat akurasi tinggi, dengan insiden false positive yang rendah. Tenable telah menguji dan membuktikan bahwa ia hanya menghasilkan 32 false positives dalam satu juta pemindaian (Kamilah & Hendrawan, 2019). Nessus menyediakan cakupan kerentanan yang luas dengan harga yang terjangkau. Sebagai platform open-source, Nessus menggunakan Arsitektur Common Vulnerabilities and Exposure. Nessus dapat diintegrasikan dengan produk Tenable lainnya seperti Tenable.io dan Tenable.sc. Nessus tersedia dalam berbagai versi, termasuk Nessus Expert, Nessus Professional, Nessus Manager, dan Nessus Agent (Nessus Vulnerability Scanner: Network Security Solution, n.d.). OWASP ZAP adalah scanner vulnerabilities yang dibuat oleh organisasi OWASP. Ini adalah proyek OWASP yang paling aktif karena terus dikembangkan dan bersifat open source (Yudiana et al., 2021).

Penelitian Susanti (2024) melakukan analisis uji kualitas keamanan website PPDB SMK X menggunakan metode isaff menemukan bahwa website PPDB SMK X dapat diakses dengan baik tanpa masalah besar meskipun ditemukan beberapa kerentanan pada server-side software namun secara keseluruhan tidak ada masalah signifikan yang dapat membahayakan situs web ini. Penggunaan Nessus Scanner sebagai metode analisis kerentanan dilakukan penelitian menggunakan Sistematic Literature Review menghasilkan analisis website terbaik adalah yang lolos tahapan pengujian tahapan pengujian OWASP terbanyak. Penelitian Wahidin dkk (2024) mengaanalisis kerentanan situs web KopKar Ayariah PT BSIN menggunakan OWASP Zed Attack Proxy yang berhasil menunjukan kerentanan dengan tingkat risiko Medium, Low, Informational dan tidak ditemukan adanya risiko yang High.

Peningkatan keamanan situs Web SIAKAD Universitas Merdeka Malang melalui analisis kerentanan dengan tenable nessus scanner dan owasp zed attack proxy ini sangat penting dalam menghadapi potensi ancaman keamanan siber yang mungkin dihadapi oleh Sistem Informasi Kepegawaian (SIMPEG) di Universitas Merdeka Malang. Pusat pengelolaan data kepegawaian, keamanan SIMPEG menjadi fokus utama, terutama di masa di mana aplikasi web rentan terhadap serangan tidak diinginkan. Dengan penekanan pada analisis kerentanan menggunakan Nessus Scanner & Owasp Zed Attack Proxy, penelitian ini bertujuan untuk memberikan pemahaman mendalam mengenai risiko keamanan yang mungkin terjadi, sehingga dapat menyusun rekomendasi tindakan pengamanan yang sesuai dan efektif untuk menjaga keamanan data dosen, mahasiswa, dan staf universitas.

## METODE

Penelitian ini menggunakan pendekatan eksperimental untuk menganalisis kerentanan website SIMPEG Universitas Merdeka Malang, dengan tujuan mengidentifikasi potensi risiko keamanan dan merancang rekomendasi tindakan pengamanan yang tepat. Metode analisis deskriptif digunakan untuk menggambarkan

kondisi keamanan situs web dan mengidentifikasi kerentanan yang mungkin diekspos oleh serangan siber. Lokasi penelitian di Universitas Merdeka Malang dipilih karena universitas ini memiliki Sistem Informasi Kepegawaian (SIMPEG) yang menjadi objek penelitian, memungkinkan peneliti melakukan analisis langsung serta berinteraksi dengan pengelola SIMPEG, dosen, dan staf administrasi akademik untuk memperoleh pemahaman lebih mendalam. Pengumpulan data dilakukan melalui observasi langsung, wawancara dengan tim IT universitas, dan pengumpulan serta analisis dokumen resmi terkait SIMPEG. Penelitian ini juga melibatkan uji kerentanan menggunakan alat Nessus dan OWASP Zed Attack Proxy, serta teknik analisis data dari hasil pemindaian untuk memberikan rekomendasi pengamanan yang tepat.

## HASIL

**Tabel 1**
**Hasil scan kerentanan menggunakan Owasp Zap**

| No | Alerts | Jumlah Contoh | Level |
|---|---|---|---|
| 1 | Absence Of Anti-Csrf Tokens | 3 | Medium |
| 2 | Content Security Policy (Csp) Header Not Set | 7 | Medium |
| 3 | Missing Anti-Clickjacking Header | 3 | Medium |
| 4 | Cookie No Httponly Flag | 2 | Low |
| 5 | Cookie Without Secure Flag | 1 | Low |
| 6 | Cookie Without Samesite Attribute | 2 | Low |
| 7 | X-Content-Type-Options Header Missing | 7 | Low |
| 8 | Authentication Request Identified | 1 | Info |
| 9 | Charset Mismatch (Header Versus Meta Content-Type Charset) | 3 | info |
| 10 | Modern Web Application | 3 | Info |
| 11 | Session Management Response Identified | 3 | info |
| 12 | User Controllable Html Element Attribute (Potential Xss) | 1 | Info |

Sumber: data olahan

Berdasarkan Tabel 1 hasil scan menggunakan Owasp Zap ditemukan beberapa kerentanan, diantaranya:.

1. Kerentanan pada level menengah mencakup :
   a. Absence Of Anti-Csrf Tokens. Tidak adanya token Anti-CSRF (Cross-Site Request Forgery) adalah kerentanan di mana aplikasi web tidak menggunakan token untuk melindungi dari serangan CSRF. Tanpa token ini, penyerang bisa membuat pengguna melakukan tindakan yang tidak diinginkan di aplikasi web tersebut.
   b. Content Security Policy (CSP) Header Not Set. Content Security Policy (CSP) adalah mekanisme keamanan yang membantu mendeteksi dan mengurangi serangan XSS (Cross-Site Scripting) dan data injection. Jika header CSP tidak diatur, aplikasi web lebih rentan terhadap berbagai serangan yang dapat menginjeksi skrip berbahaya.
   c. Missing Anti-Clickjacking Header. Anti-clickjacking header, seperti X-Frame-Options, mencegah halaman web ditampilkan dalam frame atau iframe dari situs lain. Tanpa header ini, aplikasi web rentan terhadap serangan clickjacking, di mana pengguna dapat ditipu untuk mengklik sesuatu yang berbeda dari yang mereka maksudkan.
2. Kerentanan pada level rendah mencakup :
   a. Cookie No HttpOnly Flag. Flag HttpOnly pada cookie mencegah cookie diakses melalui JavaScript. Tanpa flag ini, cookie bisa diakses oleh skrip berbahaya, meningkatkan risiko pencurian cookie dan serangan XSS.
   b. Cookie Without Secure Flag. Flag Secure pada cookie memastikan bahwa cookie hanya dikirim melalui koneksi HTTPS. Tanpa flag ini, cookie bisa dikirim melalui koneksi HTTP yang tidak aman, meningkatkan risiko intersepsi dan pencurian data.
   c. Cookie Without SameSite Attribute. Atribut SameSite pada cookie membantu mencegah pengiriman cookie dalam permintaan lintas situs, yang bisa digunakan dalam serangan CSRF. Tanpa atribut ini, cookie lebih rentan terhadap serangan tersebut.

d. X-Content-Type-Options Header Missing. Header X-Content-Type-Options: nosniff mencegah browser menebak-nebak jenis konten dari respons server. Tanpa header ini, browser bisa salah mengidentifikasi jenis konten, yang dapat dimanfaatkan oleh penyerang untuk menyisipkan skrip berbahaya.

3. Kerentanan pada level informasi mencakup :
   a. Authentication Request Identified. Kerentanan ini menunjukkan bahwa permintaan otentikasi terdeteksi. Jika tidak dikelola dengan baik, ini bisa menunjukkan titik lemah di mana penyerang dapat mencoba mendapatkan akses tidak sah.
   b. Charset Mismatch (Header Versus Meta Content-Type Charset). Ketidakcocokan charset antara header HTTP dan meta tag HTML bisa menyebabkan browser salah menampilkan atau memproses karakter. Ini bisa dieksploitasi oleh penyerang untuk melakukan serangan XSS atau injeksi kode.
   c. Modern Web Application. Sebagai aplikasi web modern, mungkin menggunakan teknologi terbaru yang juga memiliki kerentanan keamanan terbaru. Pengembang harus tetap waspada dan terus memperbarui pengetahuan dan praktik keamanan.
   d. Session Management Response Identified. Identifikasi respons manajemen sesi menunjukkan adanya titik di mana sesi pengguna diatur atau dikelola. Jika tidak aman, ini bisa menjadi titik lemah di mana penyerang dapat membajak sesi pengguna.
   e. User Controllable Html Element Attribute (Potential XSS). Atribut elemen HTML yang dapat dikontrol pengguna bisa dieksploitasi untuk melakukan serangan XSS. Jika pengguna dapat memasukkan data yang dimasukkan langsung ke dalam atribut HTML tanpa validasi yang memadai, ini membuka peluang bagi penyerang untuk menyisipkan skrip berbahaya.

**Tabel 2**
**Hasil scan kerentanan menggunakan Tenable Nessus Scanner**

| No | Level | Alerts | Jumlah Contoh |
|---|---|---|---|
| 1 | Low | ICMP Timestamp Request Remote Date Disclosure | 1 |
| 2 | Low | Web Server Allows Password Auto-Completion | 2 |
| 3 | Medium | HTTP TRACE / TRACK Methods Allowed | 3 |
| 6 | Medium | Web Application Potentially Vulnerable to Clickjacking | 1 |
| 7 | None | HTTP Server Type and Version | 2 |
| 9 | None | Traceroute Information | 1 |
| 10 | None | Web Server No 404 Error Code Check | 1 |
| 11 | None | Web mirroring | 1 |
| 12 | None | SSL Certificate Information | 1 |
| 13 | None | Web Server Directory Enumeration | 1 |
| 14 | None | Unknown Service Detection: Banner Retrieval | 1 |

Sumber: data olahan

Berdasarkan Tabel 2 hasil scan menggunakan Owasp Zap ditemukan beberapa kerentanan, diantaranya:

1. Kerentanan pada level rendah mencakup:
   a. ICMP Timestamp Request Remote Date Disclosure. Kerentanan ini memungkinkan penyerang untuk mendapatkan informasi waktu dari server target melalui permintaan ICMP timestamp. Meskipun dampaknya rendah, informasi waktu ini dapat digunakan untuk sinkronisasi serangan yang lebih kompleks.
   b. Web Server Allows Password Auto-Completion. Kerentanan ini terjadi ketika server web mengizinkan fitur auto-complete untuk bidang kata sandi. Ini dapat meningkatkan risiko pengungkapan kata sandi jika perangkat pengguna dicuri atau diakses oleh pihak yang tidak berwenang.
2. Kerentanan pada level menengah mencakup:
   a. HTTP TRACE / TRACK Methods Allowed. Server web yang mengizinkan metode HTTP TRACE atau TRACK dapat dieksploitasi untuk menyerang aplikasi web melalui teknik cross-site tracing (XST). Ini dapat menyebabkan

pengungkapan informasi sensitif seperti cookie dan header HTTP.

b. Web Application Potentially Vulnerable to Clickjacking. Kerentanan ini menunjukkan bahwa aplikasi web mungkin rentan terhadap clickjacking, di mana penyerang dapat membajak klik pengguna untuk melakukan tindakan yang tidak diinginkan. Ini biasanya terjadi karena kurangnya header keamanan seperti X-Frame-Options.

3. Kerentanan pada level informasi mencakup:
   a. HTTP Server Type and Version. Informasi tentang jenis dan versi server HTTP dapat diakses, yang dapat memberikan petunjuk kepada penyerang tentang potensi kerentanan spesifik pada versi tersebut.
   b. Traceroute Information. Informasi yang diperoleh dari traceroute dapat digunakan oleh penyerang untuk memahami struktur jaringan dan mengidentifikasi titik lemah dalam jalur jaringan.
   c. Web Server No 404 Error Code Check. Server web tidak mengembalikan kode kesalahan 404 saat halaman tidak ditemukan. Ini dapat mengindikasikan konfigurasi yang salah atau masalah lain yang mungkin tidak langsung jelas.
   d. Web Mirroring. Kerentanan ini menunjukkan bahwa situs web dapat dengan mudah disalin atau dicerminkan oleh penyerang, yang dapat digunakan untuk serangan phishing atau pencurian konten.
   e. SSL Certificate Information. Informasi tentang sertifikat SSL dapat diakses, yang dapat memberikan wawasan tentang pengaturan keamanan komunikasi server. Meski informasi ini umum diakses, dapat membantu penyerang dalam merencanakan serangan.
   f. Web Server Directory Enumeration. Server web memungkinkan enumerasi direktori, yang dapat digunakan oleh penyerang untuk menemukan file dan direktori yang tidak dimaksudkan untuk diakses publik.
   g. Unknown Service Detection. Banner Retrieval: Layanan yang tidak diketahui terdeteksi melalui retrieval banner, yang dapat memberikan informasi tentang layanan yang berjalan pada server dan membantu penyerang dalam merencanakan serangan.

*Solusi penanganan kerentanan*

Pada OWASP ZAP, terdapat beberapa kerentanan yang terdeteksi. Untuk kerentanan tingkat menengah, di antaranya adalah ketiadaan token Anti-CSRF, yang dapat diatasi dengan mengimplementasikan token Anti-CSRF pada semua formulir dan permintaan yang memodifikasi data, serta menggunakan framework atau library yang mendukung Anti-CSRF secara native seperti Django, Laravel, atau Spring. Selain itu, header Content Security Policy (CSP) yang tidak diset juga merupakan masalah, yang dapat diatasi dengan menambahkan header CSP pada konfigurasi server web untuk membatasi sumber daya yang dapat diambil oleh browser. Konfigurasi CSP yang baik bisa seperti ini: Content-Security-Policy: default-src 'self'; script-src 'self' 'nonce-...'; style-src 'self' 'nonce-...';. Juga, ketiadaan header Anti-Clickjacking dapat diatasi dengan menambahkan header X-Frame-Options atau Content-Security-Policy dengan frame-ancestors pada konfigurasi server, misalnya: X-Frame-Options: DENY atau Content-Security-Policy: frame-ancestors 'none'.

Untuk kerentanan tingkat rendah, seperti Cookie tanpa flag HttpOnly, solusinya adalah dengan menyetel flag HttpOnly pada semua cookie untuk mencegah akses melalui JavaScript, contohnya: Set-Cookie: sessionId=abc123; HttpOnly. Selain itu, untuk Cookie tanpa flag Secure, dapat diatasi dengan menyetel flag Secure pada semua cookie untuk memastikan cookie hanya dikirim melalui koneksi HTTPS, misalnya: Set-Cookie: sessionId=abc123; Secure. Kerentanan lainnya seperti Cookie tanpa atribut SameSite, dapat diatasi dengan menyetel atribut SameSite pada semua cookie untuk mencegah pengiriman dalam permintaan lintas situs, contohnya: Set-Cookie: sessionId=abc123; SameSite=Strict. Ketiadaan header X-Content-Type-Options juga bisa diatasi dengan menambahkan header X-Content-Type-Options: nosniff pada konfigurasi server untuk mencegah penebakan jenis konten oleh browser.

Kerentanan tingkat informasi yang terdeteksi termasuk permintaan autentikasi yang teridentifikasi, yang bisa diatasi dengan memastikan semua titik autentikasi dilindungi dengan enkripsi seperti HTTPS dan mengimplementasikan otentikasi dua faktor (2FA) untuk meningkatkan keamanan. Selain itu, ketidakcocokan charset (header versus meta

content-type charset) dapat diatasi dengan memastikan charset di header HTTP dan meta tag HTML konsisten, contohnya: Content-Type: text/html; charset=UTF-8 di header HTTP dan <meta charset="UTF-8"> di HTML. Aplikasi web modern harus terus diperbarui dan komponen terkaitnya harus menggunakan teknologi terbaru yang aman, serta melakukan penilaian keamanan secara berkala dan mengikuti praktik terbaik dalam pengembangan aplikasi web. Untuk respons manajemen sesi yang teridentifikasi, solusinya adalah dengan mengimplementasikan mekanisme manajemen sesi yang aman seperti regenerasi ID sesi setelah login, serta memastikan sesi pengguna dienkripsi dan berumur pendek. Kerentanan potensi XSS akibat atribut HTML yang dapat dikontrol pengguna dapat diatasi dengan memvalidasi dan mensanitasi semua input pengguna untuk mencegah injeksi skrip berbahaya, serta menggunakan library sanitasi seperti DOMPurify untuk membersihkan input pengguna.

Solusi untuk kerentanan yang ditemukan oleh Nessus Scanner juga beragam. Untuk kerentanan tingkat rendah seperti ICMP Timestamp Request Remote Date Disclosure, dapat diatasi dengan menonaktifkan permintaan ICMP timestamp pada server dan mengonfigurasi firewall untuk memblokir permintaan ICMP timestamp. Sedangkan, server web yang mengizinkan auto-completion untuk bidang kata sandi bisa diatasi dengan menambahkan atribut autocomplete="off" pada semua bidang formulir kata sandi di HTML. Kerentanan tingkat menengah seperti metode HTTP TRACE atau TRACK yang diizinkan bisa diatasi dengan menonaktifkan metode TRACE dan TRACK pada konfigurasi server web, misalnya konfigurasi Apache: TraceEnable off. Aplikasi web yang rentan terhadap clickjacking dapat diatasi dengan mengimplementasikan header X-Frame-Options atau Content-Security-Policy dengan frame-ancestors untuk mencegah clickjacking, misalnya: X-Frame-Options: DENY atau Content-Security-Policy: frame-ancestors 'none';.

Untuk kerentanan tingkat informasi, seperti tipe dan versi server HTTP yang teridentifikasi, solusinya adalah dengan menyembunyikan informasi jenis dan versi server dengan mengonfigurasi server untuk tidak mengirim informasi ini dalam header HTTP, misalnya konfigurasi Apache: ServerTokens Prod. Informasi traceroute dapat diatasi dengan mengonfigurasi firewall untuk memblokir permintaan traceroute dari luar jaringan yang dipercayai. Selain itu, server web yang tidak mengembalikan kode kesalahan 404 yang tepat ketika halaman tidak ditemukan harus dikonfigurasi dengan benar.

Kerentanan web mirroring dapat diatasi dengan menggunakan mekanisme anti-crawling dan pemblokiran IP untuk mencegah web mirroring, serta menerapkan CAPTCHA untuk memvalidasi permintaan pengguna yang mencurigakan. Informasi sertifikat SSL dapat diatasi dengan memastikan sertifikat SSL diperbarui dan dikelola dengan baik, menggunakan sertifikat SSL dari penyedia tepercaya dan mengaktifkan fitur seperti OCSP stapling untuk meningkatkan keamanan. Enumerasi direktori pada server web dapat diatasi dengan menonaktifkan enumerasi direktori pada konfigurasi server web, misalnya konfigurasi Apache: Options -Indexes. Sedangkan, deteksi layanan yang tidak diketahui melalui retrieval banner dapat diatasi dengan menyembunyikan informasi layanan dengan mengonfigurasi server untuk tidak mengirim banner layanan, misalnya konfigurasi untuk berbagai layanan seperti SSH dan FTP untuk menonaktifkan banner.

## SIMPULAN

Berdasarkan hasil analisis kerentanan terhadap situs web sdm.unmer.ac.id, dapat disimpulkan bahwa OWASP ZAP lebih baik dibandingkan Nessus dalam konteks ini. OWASP ZAP mampu mendeteksi berbagai kerentanan yang sangat spesifik terhadap aplikasi web, seperti Absence of Anti-CSRF Tokens, Content Security Policy (CSP) Header Not Set, dan Missing Anti-Clickjacking Header, yang langsung mempengaruhi integritas, keamanan, dan privasi data pengguna. Selain itu, OWASP ZAP mengklasifikasikan kerentanan berdasarkan level risiko (medium, low, informational), yang membantu tim keamanan memprioritaskan perbaikan berdasarkan tingkat keparahan ancaman. OWASP ZAP juga memberikan perhatian khusus pada kerentanan yang dapat dieksploitasi untuk serangan Cross-Site Scripting (XSS) dan Cross-Site Request Forgery (CSRF), yang merupakan ancaman umum dan berbahaya pada aplikasi web. Sebaliknya, Nessus Scanner lebih fokus pada kerentanan yang berhubungan dengan

infrastruktur jaringan dan server, seperti metode TRACE/TRACK, informasi traceroute, dan detail sertifikat SSL, yang meskipun penting, kurang relevan dibandingkan dengan kerentanan aplikasi web yang ditemukan oleh OWASP ZAP dalam konteks situs web sdm.unmer.ac.id. Oleh karena itu, untuk keamanan aplikasi web sdm.unmer.ac.id, OWASP ZAP adalah alat yang lebih efektif dan sesuai karena mampu mengidentifikasi kerentanan yang lebih spesifik dan kritis terhadap aplikasi web.

## DAFTAR PUSTAKA

Agung, B., & Arifin, M. 2020. *Sistem Informasi Manajemen Kepegawaian pada Administrasi dan Pelayanan Kepegawaian dalam Kerangka Merit System di Lingkungan Kementerian Hukum dan HAM*: *Teknis subtantif sistem informasi kepegawaian*, Depok: BPSDM KUMHAM Press.

Amri, M., Waidah, D. F., & Adi, F. T. 2023. Analisis Sistem Informasi Manajemen Kepegawaian di Badan Kepegawaian dan Pengembangan Sumber Daya Manusia (BKPSDM) Kabupaten Karimun. *JURNAL TIKAR*, *4*(1), 41-50.

Kamilah, I., & Hendri Hendrawan, A. 2019. Analisis Keamanan Vulnerability pada Server Absensi Kehadiran Laboratorium di Program Studi Teknik Informatika. *Jurnal UMJ*, *16*, 1–9.

Prasetyo Taufan. 2022. Pengamanan Jaringan Komputer Dengan Intrusion PreventionSystem (IPS) Berbasis Sms Gateway. *Teknologipintar.Org*, *2*(6), 1–13.

Susanti, D. 2024. Analisis Uji Kualitas Keamanan Website PPDB SMK X Menggunakan Metode Isaaf. *Jurnal IndraTech*, *5*.

Wahidin, M., Rahayu, D. N., & Yulianto, R. M. 2024. Analisis Kerentanan Situs Web KopKar Syariah PT BSIN menggunakan OWASP Zed Attack Proxy. *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi*, *18*(4), 25–31.

Widyawan, D. C., & Idris, A. 2021. Implementasi Sistem Informasi Manajemen Kepegawaian (Simpeg) Di Badan Kepegawaian Pendidikan dan Pelatihan Daerah Kota Samarinda. *Jurnal Administrative Reform*, *8*(2), 125.

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.