Analisa Kualitas Keamanan Pada Aplikasi Slims Akasia Dengan Metode NIST SP 800-115 DAN OWASP

Hena Sulaeman[1], Ahsani Takwim[2]
Departemen Teknik Informatika
Universitas Teknologi Bandung
Bandung, Indonesia
e-mail: [1]henasulaeman@utb-univ.ac.id, [2]ahsanitakwim10@gmail.com

Correspondence : *e-mail*: henasulaeman@utb-univ.ac.id

Diajukan: 24 Agustus 2024; Direvisi: 24 Agustus 2024; Diterima: 28 Agustus 2024

Abstrak

Penelitian ini bertujuan untuk menganalisis kualitas keamanan pada aplikasi SLiMS Akasia versi 8 yang digunakan oleh perpustakaan Universitas XYZ. Seiring dengan perkembangan teknologi, serangan siber seperti ransomware, phishing, dan deface website semakin sering terjadi dan semakin kompleks. Aplikasi SLiMS Akasia, yang merupakan perangkat lunak open source berbasis web, telah mengalami serangan semacam itu, sehingga diperlukan penilaian kerentanan secara menyeluruh. Penelitian ini menggunakan metode NIST SP 800-115 untuk mengidentifikasi potensi kerentanan pada sistem dan mengusulkan langkah-langkah mitigasi. Pendekatan kualitatif digunakan dalam penelitian ini, melibatkan tinjauan literatur, wawancara dengan staf perpustakaan, dan observasi lapangan untuk mengevaluasi praktik keamanan yang sudah berjalan. Penilaian dilakukan dengan mengikuti tahapan penetration testing, perencanaan, penemuan, serangan, dan pelaporan untuk mengukur pertahanan keamanan sistem. Selain itu, panduan uji kerentanan OWASP digunakan untuk menguji berbagai aspek keamanan seperti pengumpulan informasi, autentikasi, dan penanganan kesalahan. Tujuan utama dari penelitian ini adalah meningkatkan ketahanan sistem terhadap ancaman siber dan melindungi sumber daya digital perpustakaan. Melalui analisis ini, diharapkan implementasi langkah-langkah keamanan yang efektif dapat secara signifikan mengurangi risiko serangan siber di masa mendatang.

Kata kunci: *NIST SP 800-115, OWASP, Vulnerability Assesment*

Abstract

This study aims to analyze the security quality of the SLiMS Akasia application, version 8, used by Universitas XYZ library. As technology advances, cyber-attacks such as ransomware, phishing, and website defacement are becoming more frequent and complex. SLiMS Akasia, an open-source web-based software, has experienced such attacks, prompting the need for a thorough vulnerability assessment. Using the NIST SP 800-115 method, this research identifies potential vulnerabilities in the system and proposes mitigation measures. The study employs a qualitative approach, involving literature reviews, interviews with library staff, and field observations to evaluate the security practices in place. The assessment focuses on penetration testing stages planning, discovery, attack, and reporting to evaluate the system's defenses. OWASP's vulnerability testing guide is used to assess various security aspects, such as information gathering, authentication, and error handling. The ultimate goal is to improve the system's resilience against cyber threats and ensure the library's digital resources remain secure. Through this analysis, it is expected that the implementation of effective security measures will significantly reduce the risk of future attacks.
Keywords*: NIST SP 800-115, OWASP, Vulnerability Assessment*

1. Pendahuluan

Perkembangan teknologi membawa dampak yang signifikan terhadap layanan perpustakaan. Implementasi berbagai aplikasi berbasis web dan mobile memungkinkan pengguna mengakses koleksi perpustakaan secara online dari mana saja dan kapan saja. Namun , semakin kompleksnya system informasi, semakin pula besar potensi terjadinya serangan siber. Serangan *ransomware*, *phising*, sql *injection*, *deface* website, *XSS* adalah beberapa contoh yang sering terjadi.

Senayan *Library* Management *System* (SLiMS) adalah sebuah perangkat lunak (software) *open source* (gratis) dan berbasis web yang dapat digunakan untuk mengelola perpustakaan secara digital. SLiMS dapat digunakan untuk mengelola koleksi tercetak dan terekam di perpustakaan, dan memiliki fitur-fitur seperti pengolahan data transaksi anggota, peminjaman dan pengembalian koleksi, dan pengingat jadwal pengembalian buku [1].

Saat ini, bagian perpustakaan Universitas XYZ telah menggunakan aplikasi SliMS Akasia versi 8 untuk pengelolaan perpustakaan secara digital. Namun, berdasarkan informasi yang diterima dari admin perpustakaan, aplikasi SliMS Akasia ini pernah mengalami serangan siber, seperti *deface*, di mana halaman utama aplikasi SliMS Akasia diubah dengan tampilan yang tidak seharusnya ditampilkan kepada *public*. *Vulnerability assessment* adalah proses untuk mengidentifikasi, evaluasi dan klasifikasi kerentanan dengan tujuan untuk memberitahukan kepada entitas bahwa ada celah yang bisa disalahgunakan oleh orang yang tidak bertanggung jawab dan dapat merugikan entitas atau organisasi yang menggunakan system tersebut [2].

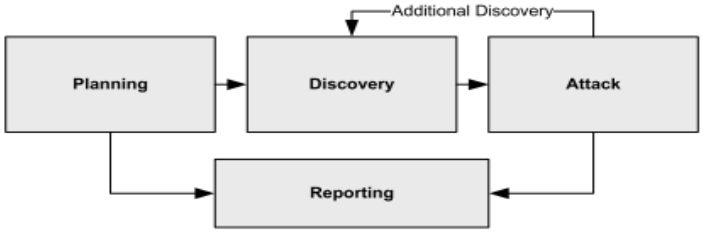
Penelitian ini bertujuan untuk mengidentifikasi dan analisis kerentanan yang terjadi pada perangkat lunak *open source* senayan *library management system* versi 8 (AKASIA) yang digunakan oleh bagian perpustakaan Universitas XYZ. Dengan melakukan *vulnerability* assessment, diharapkan dapat ditemukan langkah-langkah mitigasi yang efektif untuk melindungi system informasi dan berbagai jenis serangan siber.

2. Metode Penelitian

Metode penelitian yang digunakan adalah metode kualitatif. Metode kualitatif di pilih dengan tujuan untuk membantu dalam menganalisis dan observasi praktik keamanan yang sedang berjalan dan peneliti bisa melakukan penyelidikan kejadian, fenomena kehidupan dan meminta individu-individu menceritakan kembali tentang suatu kejadian yang sebelumnya sudah pernah terjadi pada objek yang sedang di teliti [3].

Pengumpulan data dilakukan dengan studi literatur dan bagian terkait yang mengelola sistem di Universitas XYZ, serta observasi di lapangan. Studi literatur di lakukan untuk mengidentifikasi, mencari landasan teori yang berkaitan dengan aplikasi SLiMS Akasia untuk menjadikan sebagai kerangka berfikir dalam penelitian ini dan Observasi dilakukan untuk mengamati langsung ke lapangan, mencoba fungsionalitas dari fitur-fitur yang ada di aplikasi SLiMS Akasia versi 8 serta proses bisnis yang berjalan di bagian perpustakaan.

Metode penelitian untuk pengujian dan penilaian keamanan informasi mengacu pada dokumen yang dikeluarkan oleh National *Institute of Standar and Technology* (NIST) dengan kode dokumen NIST SP 800-115. NIST-SP 800 115 merupakan panduan teknis untuk pengujian dan penilaian keamanan informasi, metodologi yang dikhususkan untuk membantu organisasi dalam melakukan perencanaan test keamanan informasi [4]. Tahapan penetration testing yang digunakan adalah 4 tahap yaitu sebagai berikut:



Gambar 1. *Four-Stage Penetration Testing Methodelogy*

2.1. *Planning*

Pada tahap ini akan mengumpulkan informasi yang di perlukan untuk pelaksanaan penilaian aplikasi target yang akan di nilai, ancaman yang terkait pada aplikasi, kontrol keamanan yang akan digunakan dan *scope penetration testing*.

2.2. *Discovery*

Tahap ini berfungsi untuk mengumpulkan data tentang sistem yang akan diuji dan menganalisisnya untuk mengidentifikasi kerentanan potensial.

2.3. *Attack*

Tahap ini berfungsi untuk mengidentifikasi dan menguji kerentanan dalam sistem. Ini mencakup penggunaan berbagai alat dan teknik untuk mengeksploitasi kelemahan yang ditemukan selama pengumpulan informasi.

2.4. *Reporting*

Pada tahap ini dilakukan reporting kerentanan, hasil dokumentasi kerentanan dan tingkat keparahan kerentanan serta rekomendasi perbaikan yang harus dilakukan. Selanjutnya parameter *Open Web Application Security Project* (OWASP) yang digunakan adalah daftar testing kerentanan yang merujuk pada dokumen testing guide 4.0 [5*] testing guide* tersebut berfungsi untuk menjelaskan bagaimana cara menguji dan membuktikan bukti-bukti kerentanan dalam aplikasi akibat kekurangan control keamanan yang telah diidentifikasi [6] . Sebagai acuan standar pengujian kerentanan aplikasi SLiMS Akasia 8 merujuk pada standar kontrol berikut :

Tabel 1. Standar Kontrol OWASP Versi 4

ID Kontrol	Standar Kontrol
Information Gathering	
OTG-INFO-001	*Conduct Search Engine Discovery and Reconnaissance for Information Leakage*
OTG-INFO-002	*Fingerprint Web Server*
OTG-INFO-003	*Review Webserver Metafiles for Information Leakage*
OTG-INFO-004	*Enumerate Applications on Webserver*
OTG-INFO-005	*Review Webpage Comments and Metadata for Information Leakage*
OTG-INFO-006	*Identify application entry points*
OTG-INFO-007	*Map execution paths through application*
OTG-INFO-008	*Fingerprint Web Application Framework*
OTG-INFO-009	*Fingerprint Web Application*
OTG-INFO-010	*Map Application Architecture*
Configuration and Deploy Management Testing	
OTG-CONFIG-001	*Test Network/Infrastructure Configuration*
OTG-CONFIG-002	*Test Application Platform Configuration*
OTG-CONFIG-003	*Test File Extensions Handling for Sensitive Information*
OTG-CONFIG-004	*Backup and Unreferenced Files for Sensitive Information*
OTG-CONFIG-005	*Enumerate Infrastructure and Application Admin Interfaces*
OTG-CONFIG-006	*Test HTTP Methods*
OTG-CONFIG-007	*Test HTTP Strict Transport Security*
OTG-CONFIG-008	*Test RIA cross domain policy*
Identity Management Testing	
OTG-IDENT-001	*Test Role Definitions*
OTG-IDENT-002	*Test User Registration Process*
OTG-IDENT-003	*Test Account Provisioning Process*
OTG-IDENT-004	*Testing for Account Enumeration and Guessable User Account*
OTG-IDENT-005	*Testing for Weak or unenforced username policy*
OTG-IDENT-006	*Test Permissions of Guest/Training Accounts*
OTG-IDENT-007	*Test Account Suspension/Resumption Process*
Authentication Testing	
OTG-AUTHN-001	*Testing for Credentials Transported over an Encrypted Channel*
OTG-AUTHN-002	*Testing for default credentials*
OTG-AUTHN-003	*Testing for Weak lock out mechanism*
OTG-AUTHN-004	*Testing for bypassing authentication schema*
OTG-AUTHN-005	*Test remember password functionality*
OTG-AUTHN-006	*Testing for Browser cache weakness*
OTG-AUTHN-007	*Testing for Weak password policy*
OTG-AUTHN-008	*Testing for Weak security question/answer*
OTG-AUTHN-009	*Testing for weak password change or reset functionalities*
OTG-AUTHN-010	*Testing for Weaker authentication in alternative channel*
Authorization Testing	
OTG-AUTHZ-001	*Testing Directory traversal/file include*
OTG-AUTHZ-002	*Testing for bypassing authorization schema*
OTG-AUTHZ-003	*Testing for Privilege Escalation*
OTG-AUTHZ-004	*Testing for Insecure Direct Object References*
Session Management Testing	
OTG-SESS-001	*Testing for Bypassing Session Management Schema*
OTG-SESS-002	*Testing for Cookies attributes*
OTG-SESS-003	*Testing for Session Fixation*
OTG-SESS-004	*Testing for Exposed Session Variables*

OTG-SESS-005	*Testing for Cross Site Request Forgery*
OTG-SESS-006	*Testing for logout functionality*
OTG-SESS-007	*Test Session Timeout*
OTG-SESS-008	*Testing for Session puzzling*
Input Validation Testing	
OTG-INPVAL-001	*Testing for Reflected Cross Site Scripting*
OTG-INPVAL-002	*Testing for Stored Cross Site Scripting*
OTG-INPVAL-003	*Testing for HTTP Verb Tampering*
OTG-INPVAL-004	*Testing for HTTP Parameter pollution*
OTG-INPVAL-006	*Testing for SQL Injection*
	Oracle Testing
	SQL Server Testing
	Testing PostgreSQL
	MS Access Testing
OTG-INPVAL-007	*Testing for LDAP Injection*
OTG-INPVAL-008	*Testing for ORM Injection*
OTG-INPVAL-009	*Testing for XML Injection*
OTG-INPVAL-010	*Testing for SSI Injection*
OTG-INPVAL-011	*Testing for XPath Injection*
OTG-INPVAL-012	*IMAP/SMTP Injection*
OTG-INPVAL-013	*Testing for Code Injection*
	Testing for Local File Inclusion
	Testing for Remote File Inclusion
OTG-INPVAL-014	*Testing for Command Injection*
OTG-INPVAL-015	*Testing for Buffer overflow*
	Testing for Heap overflow
	Testing for Stack overflow
	Testing for Format string
OTG-INPVAL-016	*Testing for incubated vulnerabilities*
OTG-INPVAL-017	*Testing for HTTP Splitting/Smuggling*
Error Handling	
OTG-ERR-001	*Analysis of Error Codes*
OTG-ERR-002	*Analysis of Stack Traces*
Cryptography	
OTG-CRYPST-001	*Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection*
OTG-CRYPST-002	*Testing for Padding Oracle*
OTG-CRYPST-003	*Testing for Sensitive information sent via unencrypted channels*
Business Logic Testing	
OTG-BUSLOGIC-001	*Test Business Logic Data Validation*
OTG-BUSLOGIC-002	*Test Ability to Forge Requests*
OTG-BUSLOGIC-003	*Test Integrity Checks*
OTG-BUSLOGIC-004	*Test for Process Timing*
OTG-BUSLOGIC-005	*Test Number of Times a Function Can be Used Limits*
OTG-BUSLOGIC-006	*Testing for the Circumvention of Work Flows*
OTG-BUSLOGIC-007	*Test Defenses Against Application Mis-use*
OTG-BUSLOGIC-008	*Test Upload of Unexpected File Types*
OTG-BUSLOGIC-009	*Test Upload of Malicious Files*
Client Side Testing	
OTG-CLIENT-001	*Testing for DOM based Cross Site Scripting*
OTG-CLIENT-002	*Testing for JavaScript Execution*
OTG-CLIENT-003	*Testing for HTML Injection*
OTG-CLIENT-004	*Testing for Client Side URL Redirect*
OTG-CLIENT-005	*Testing for CSS Injection*
OTG-CLIENT-006	*Testing for Client Side Resource Manipulation*
OTG-CLIENT-007	*Test Cross Origin Resource Sharing*
OTG-CLIENT-008	*Testing for Cross Site Flashing*
OTG-CLIENT-009	*Testing for Clickjacking*
OTG-CLIENT-010	*Testing WebSockets*
OTG-CLIENT-011	*Test Web Messaging*
OTG-CLIENT-012	*Test Local Storage*

Manfaat yang akan diperoleh pada penilaian risiko ini diantaranya adalah bisa mengurangi terjadinya risiko yang lebih serius yang akan berdampak pada aplikasi atau system yang digunakan [7]. Untuk perhitungan tingkat keparahan menggunakan kalkulator yang sudah di sediakan CVSS (*Common Vulnerability Scoring System*) *Version* 3 [8].

Tabel 2 *Findings Severity Ratings*

Severity	***CVSS V3 Score Range***	***Definition***
Critical	9.0-10.0	Eksploitasi bersifat langsung dan biasanya menghasilkan kompromi di tingkat sistem. Disarankan untuk segera membuat rencana tindakan dan melakukan penambalan.
High	7.0-8.9	Eksploitasi lebih sulit namun dapat menyebabkan peningkatan hak istimewa dan berpotensi hilangnya data atau downtime. Disarankan untuk membuat rencana tindakan dan penambalan sesegera mungkin.
Moderate	4.0-6.9	Kerentanan ada tetapi tidak dapat dieksploitasi atau memerlukan langkah tambahan seperti rekayasa sosial. Disarankan untuk membuat rencana tindakan dan perbaikan setelah masalah prioritas tinggi diselesaikan.
Low	0.1-3.9	Kerentanan tidak dapat dieksploitasi namun akan mengurangi permukaan serangan organisasi. Disarankan untuk membuat rencana tindakan dan patch selama masa pemeliharaan berikutnya.
Informational	N/A	Tidak ada kerentanan. Informasi tambahan diberikan mengenai item yang diperhatikan selama pengujian, kontrol yang kuat, dan dokumentasi tambahan.

3. Hasil Pembahasan Penelitian

Bagian ini membahas hasil dari vulnerability assessment terhadap aplikasi SLiMS Akasia versi 8 dengan merujuk pada tahapan pentesting yang ada di dokumen NIST-SP 800-115 yaitu *planning, discovery, attack* dan *reporting* [9]. Tujuan utama pada penelitan ini adalah untuk mengevaluasi kualitas keamanan aplikasi SliMS Akasia versi 8 dan mempersiapkannya agar lebih tahan terhadap ancaman siber.

3.1. *Planning*

Pada tahap ini, domain yang akan diuji dibatasi ruang lingkupnya sesuai dengan persetujuan dari tim terkait di Universitas XYZ. Pembatasan ini hanya mencakup domain pada aplikasi SliMS Akasia, dan pengujian seperti *scanning* dan pentesting akan dilakukan pada aplikasi kloning yang telah disiapkan, sehingga tidak mengganggu aplikasi yang sudah berjalan di lingkungan *production*.

3.2. *Discovery*

Pada tahap ini, dilakukan *information gathering* mengumpulkan data-data yang di perlukan pada system target dengan cara scanning *system* pada target untuk mendapatkan data seperti alamat IP, spesifikasi server, teknologi server yang digunakan, hostname, domain name, *registry*, TLD dan lain lain untuk dijadikan landasan atau dasar pada tahap *attack* atau penyerangan [10].

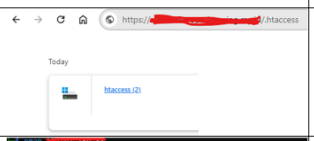
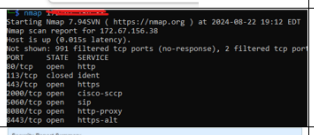
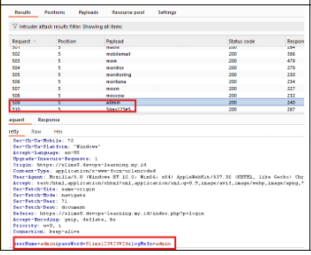
3.3. *Attack*

Pada tahap ini dilakukan analisis dari hasil pemindaian dengan menggunakan alat atau tools burpsuite, kalilinux dan sqlmap. Burp suite digunakan untuk pengujian penetrasi pada aplikasi SLiMS Akasia 8 dengan proses analisis *intercepting proxy*, *scanner*, *intruder* dan *repeater*. Kalilinux digunakan untuk mengeksploitasi kerentanan dengan menggunakan Metasploit framework. *Sqlmap* digunakan untuk mengeksploitasi kerentanan *SQL Injection*.

3.4. *Reporting*

Pada tahap ini dilakukan pelaporan kerentanan keamanan yang ditemukan dalam sistem, aplikasi SLiMS Akasia versi 8 dengan tujuan dari *reporting* ini adalah untuk mengidentifikasi masalah keamanan, memberikan informasi yang jelas dan rinci tentang kerentanan yang ditemukan, dan merekomendasikan tindakan perbaikan atau mitigasi.

Tabel 3. *Reporting*

Vulnerability Testing	*Description*	*Risk Level*	*Evidence*	*CVSS Score*
Webserver Metafiles for Information Leakage	kerentanan yang terjadi ketika file metafile atau file konfigurasi yang digunakan oleh server web secara tidak sengaja diekspos kepada publik.	Informational		CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Enumerate Applications on Webserver	Pengintaian pada saat penetration testing untuk mengumpukan informasi port yang terbuka (open port) karena kesalahan konfigurasi	Informational		CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Lack or Misconfiguration Security Header(s)	Heeader respons HTTP yang dapat digunakan aplikasi Anda untuk meningkatkan keamanan aplikasi Anda. Setelah ditetapkan, header respons HTTP ini dapat membatasi browser modern agar tidak mengalami kerentanan yang dapat dicegah dengan mudah.	Informational		CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:N
Stored Cross Site Scripting	muncul ketika suatu aplikasi menerima data dari sumber yang tidak tepercaya, dan menyertakan data tersebut dalam respons HTTP selanjutnya dengan cara yang tidak aman.	Moderate		CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:N/I:L/A:N (4.3)
Possible Brute Force	"brute force" berasal dari penyerang yang menggunakan upaya yang terlalu kuat untuk mendapatkan akses ke akun pengguna	Moderate		CVSS:3.1/AV:N/AC:H/PR:N/UI:N/S:U/C:L/I:N/A:L

Tabel 4. Data Mitigation

Vulnerability	***Mitigation***
Webserver Metafiles for Information Leakage	Konfigurasi Server Web untuk Menolak Akses ke *Metafile*, Hapus *Metafile* yang Tidak Diperlukan, Gunakan File .gitignore (untuk Repositori Git) dan Nonaktifkan *Listing* Direktori
Enumerate Applications on Webserver	Konfigurasi Server Web untuk Membatasi Informasi yang di publish, Menggunakan *Firewall* Aplikasi Web (WAF) dan Membatasi Akses ke Direktori dan File Sensitif
Lack or Misconfiguration Security Header(s)	Konfigurasikan CSP dengan menetapkan aturan ketat tentang sumber daya yang boleh dimuat, ambahkan header ini dengan nilai nosniff untuk mencegah MIME *type sniffing* dan Tambahkan header ini untuk memastikan semua komunikasi dilakukan melalui HTTPS
Stored Cross Site Scripting	Validasi dan Sanitasi Input, *Escape Output* ke HTML dan Gunakan *Framework* yang Aman
Possible Brute Force	*Rate Limiting* : Batasi jumlah percobaan login yang dapat dilakukan dalam waktu tertentu. Misalnya, batasi hingga 5 percobaan login dalam 5 menit.

4. Kesimpulan

Berdasarkan hasil pengujian yang sudah dilakukan terdapat 5 kerentanan yang sudah teridentifikasi dengan tingkat risiko *informational* dan moderate, walaupun belum menemukan tingkat risiko dengan level *critical* atau *high*, serangan bisa saja terjadi dari berbagai pihak, baik dari dalam ataupun dari luar. Oleh karena itu, sangat penting untuk tetap waspada dan proaktif dalam memperbaiki kerentanan yang ada, serta terus meningkatkan langkah-langkah keamanan yang diterapkan pada aplikasi SLiMS Akasia. Mengabaikan kerentanan dengan tingkat risiko lebih rendah dapat membuka celah bagi penyerang untuk mengeksploitasi sistem di masa depan, sehingga tindakan pencegahan yang tepat harus segera diambil untuk melindungi integritas dan keamanan infrastruktur.

Daftar Pustaka

[1] Komunitas Pengembang Senayan, "Dokumentasi Slims 8 Akasia." Accessed: Aug. 18, 2024. [Online]. Available: https://slims.gitbooks.io/slims8_doc/content/preface.html

[2] H. Jurnal, R. Farismana, and D. Pramadhana, "VULNERABILITY ASSESSMENT UNTUK ANALISIS TINGKAT KEAMANAN PADA SISTEM INFORMASI REPOSITORI KARYA ILMIAH POLITEKNIK XYZ," Online, 2023.

[3] A. Rustamana, N. Rohmah, P. F. Natasya, and R. Raihan, "CENDIKIA PENDIDIKAN," vol. 5, 2024, doi: 10.9644/sindoro.v4i5.3317.

[4] E. Z. Darojat, E. Sediyono, and I. Sembiring, "Vulnerability Assessment Website E-Government dengan NIST SP 800-115 dan OWASP Menggunakan Web Vulnerability Scanner," *JURNAL SISTEM INFORMASI BISNIS*, vol. 12, no. 1, pp. 36–44, Sep. 2022, doi: 10.21456/vol12iss1pp36-44.

[5] M. Meucci and A. Muller, "4.0 Testing Guide." [Online]. Available: http://www.owasp.org

[6] L. Costaner and dan Musfawati, "ANALISIS KEAMANAN WEB SERVER OPEN JOURNAL SYSTEM (OJS) MENGGUNAKAN METODE ISSAF DAN OWASP (STUDI KASUS OJS UNIVERSITAS LANCANG KUNING)."

[7] B. Ghozali, K. Kusrini, and S. Sudarmawan, "Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) Untuk Penilaian Risk Rating," *Creative Information Technology Journal*, vol. 4, no. 4, p. 264, Jan. 2019, doi: 10.24076/citec.2017v4i4.119.

[8] Inc. A. R. Reserved. Forum of Incident Response and Security Teams, "Common Vulnerability Scoring System Version 3." Accessed: Aug. 23, 2024. [Online]. Available: https://www.first.org/cvss/calculator/3.0

[9] K. A. Scarfone, M. P. Souppaya, A. Cody, and A. D. Orebaugh, "Technical guide to information security testing and assessment.," Gaithersburg, MD, 2008. doi: 10.6028/NIST.SP.800-115.

[10] Y. Thurfah Afifa Rosaliah and B. Hananto, *Pengujian Celah Keamanan Website Menggunakan Teknik Penetration Testing dan Metode OWASP TOP 10 pada Website SIM xxx*. 2021.