Analisis Kerentanan Situs Web KopKar Syariah PT BSIN menggunakan OWASP Zed Attack Proxy

M. Wahidin[1*], Dhian Nur Rahayu[2], R. Mega Yulianto[3]
[1]Program Studi Sistem Informasi, STMIK ROSMA, Karawang, Indonesia
[2,3]Program Studi Teknik Informatika, STMIK ROSMA, Karawang, Indonesia
Email: m.wahidin@dosen.rosma.ac.id, dhian@dosen.rosma.ac.id, r.mega@dosen.rosma.ac.id

Abstract
Koperasi Karyawan Syariah PT BSIN has a website to support interaction with members and make it a promotional media.Important member data stored and processed in the web application, it is necessary to test the vulnerability of the site. Vulnerability testing is done using OWASP Zed Attack Proxy. This application was developed by OWASP (Open Web Application Security Project), a non-profit organization that focuses on web application security. This test shows a vulnerability with a risk level of Medium, Low, Informational and no high risk level is found. From these vulnerabilities, it is necessary to anticipate and improved by web developers so that they can ensure the security.

Keywords: *Vulnerability Testing, OWASP, Zed Attack Proxy*

Abstrak
Koperasi Karyawan Syariah PT BSIN memiliki situs web untuk mendukung interaksi dengan anggota serta menjadikannya sebagai media promosi. Mengingat adanya data-data penting anggota yang tersimpan dan diproses di aplikasi web tersebut maka perlu dilakukan pengujian terhadap kerentanan situs. Pengujian kerentanan dilakukan dengan menggunakan OWASP Zed Attack Proxy. Aplikasi ini dikembangkan oleh OWASP (Open Web Application Security Project), organisasi nonprofit yang berfokus pada keamanan aplikasi web. Pengujian ini menunjukkan kerentanan dengan tingkat risiko Medium, Low , Informational dan tidak ditemukan adanya tingkat risiko yang High. Dari kerentanan tersebut perlu dilakukan antisipasi serta perbaikan oleh pengembang web sehingga dapat menjamin keamanannya.

Kata Kunci: Pengujian Kerentanan, OWASP, Zed Attact Proxy

Article History :

Received 08, Januari, 2024
Revised 15, Januari, 2024
Accepted 22, Januari, 2024

Corresponding Author:

Nama Penulis, M. Wahidin
Departemen, Sistem Informasi
Instansi, STMIK Rosma
Alamat. Jl. Parahyangan, Adiarsa Barat
Email Penulis. m.wahidin@dosen.rosma.ac.id

P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

1. Pendahuluan

Koperasi adalah sebuah badan usaha yang memiliki anggota perorangan atau badan hukum koperasi dengan berlandaskan pada prinsip koperasi sebagai gerakan ekonomi kerakyatan berasaskan kekeluargaan [1]. Koperasi Karyawan Syariah PT BSIN Adalah koperasi yang memiliki unit usaha simpan pinjam pembiayaan syariah, toko, dan Armada. Koperasi ini beranggotakan karyawan yang bekerja di perusahaan. Tujuan didirikannya Koperasi ini adalah untuk meningkatkan kesejahteraan anggota ta dan turut membangun tatanan perekonomian Sesuai dengan prinsip syariah. Seiring dengan berkembangnya teknologi, Koperasi Karyawan Syariah PT BSIN berusaha memberikan layanan terbaik dengan meluncurkan website Yang dapat diakses oleh anggota. Selain untuk memberikan informasi bagi anggota, website tersebut dapat dimanfaatkan untuk melihat aktivitas anggota seperti besaran simpanan pokok, simpanan wajib, simpanan sukarela, serta pembiayaan.

Aplikasi web yang dipublish ke internet seringkali mendapatkan serangan dari hacker atau peretas. Perlu dilakukan langkah-langkah pencegahan agar data anggota koperasi karyawan Syariah PT BSIN aman dari tindakan tersebut. oleh karenanya perlu dilakukan Analisis kerentanan

OWASP (Open Web Application Security Project), organisasi nonprofit yang berfokus pada keamanan aplikasi web, merekomendasikan sepuluh ancaman keamanan aplikasi web yaitu Injection, Broken Authentication, Sensitive Data Exposure, XML External Entities (XXE), Broken Access Control, Security Misconfiguration, Cross-Site Scripting (XSS), Insecure Deserialization, Using Components with Known Vulnerabilities, dan Insufficient Component with Known Vulnearbilities [2]. OWASP mengembangkan aplikasi yang dapat digunakan untuk menganalisis web, yaitu Zed Attack Proxy (ZAP). Analisis yang dilakukan oleh ZAP meliputi Vulnerability Assessment, Penetration Testing, Runtime Testing dan Code Review [3]

Penelitian ini bertujuan untuk menganalisis kerentanan website Koperasi Karyawan Syariah PT BSIN, hasil penelitian kerentanan ini dapat menjadi rekomendasi untuk melakukan tindakan pengamanan. Penelitian serupa telah dilakukan oleh peneliti sebelumnya antara lain analisis kerentanan serangan XSS menggunakan ZAP yang berhasil menemukan kerentanan di aplikasi smart payment [4]. Penelitian lainnya melakukan analisis kerentanan menggunakan metode OWASP Mantra. Penelitian ini menghasilkan analisis kerentanan terhadap website penerimaan mahasiswa baru [5]. Penggunaan OWASP sebagai metode analisis kerentanan dilakukan penelitian menggunakan Sistematic Literature Review menghasilkan analisis website terbaik adalah yang lolos tahapan pengujian OWASP terbanyak [6].

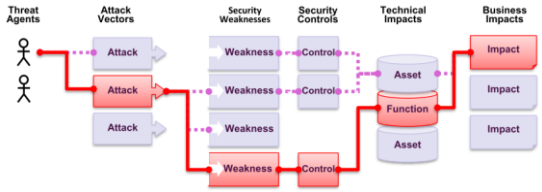
2. Tinjauan Pustaka

a. Risiko Keamanan Aplikasi

Hacker atau peretas berpotensi menggunakan banyak jalur berbeda melalui aplikasi web untuk membahayakan bisnis atau organisasi. Masing-masing jalur ini mewakili risiko yang mungkin, atau mungkin tidak, cukup serius untuk mendapat perhatian. Terkadang jalur ini sepele untuk ditemukan dan dieksploitasi, dan terkadang sangat sulit. Demikian pula, kerugian yang ditimbulkan mungkin tidak berdampak apa-apa, atau dapat membuat kerugian yang lebih besar. Untuk menentukan risiko bagi organisasi, dapat dilakukan dengan mengevaluasi kemungkinan yang terkait dengan setiap ancaman, serangan, dan kelemahan keamanan kemudian menggabungkannya

P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

dengan perkiraan dampak teknis dan bisnis bagi organisasi. Bersama-sama, faktor-faktor ini menentukan risiko secara keseluruhan.



Gambar 1. Ancaman keamanan dan dampaknya (OWASP, 2017)

b. OWASP TOP 10

Merupakan 10 daftar celah keamanan yang dapat mengancam keamanan sebuah *web application* [7]. Daftar ini diperbaharui sesuai dengan perkembangan kerentanan, versi terakhir yang di release adalah versi 2017. Pada saat penelitian ini dilakukan, OWASP sedang menyusun versi 2021 untuk menggantikan versi sebelumnya.

OWASP Top 10 - 2013		OWASP Top 10 - 2017
A1 – Injection	→	A1:2017-Injection
A2 – Broken Authentication and Session Management	→	A2:2017-Broken Authentication
A3 – Cross-Site Scripting (XSS)	↘	A3:2017-Sensitive Data Exposure
A4 – Insecure Direct Object References [Merged+A7]	∪	A4:2017-XML External Entities (XXE) [NEW]
A5 – Security Misconfiguration	↘	A5:2017-Broken Access Control [Merged]
A6 – Sensitive Data Exposure	↗	A6:2017-Security Misconfiguration
A7 – Missing Function Level Access Contr [Merged+A4]	∪	A7:2017-Cross-Site Scripting (XSS)
A8 – Cross-Site Request Forgery (CSRF)	☒	A8:2017-Insecure Deserialization [NEW, Community]
A9 – Using Components with Known Vulnerabilities	→	A9:2017-Using Components with Known Vulnerabilities
A10 – Unvalidated Redirects and Forwards	☒	A10:2017-Insufficient Logging&Monitoring [NEW,Comm.]

Gambar 2. Perbandingan kerentanan OWASP Top 10 versi 2013 dengan 2017 (OWASP, 2017)

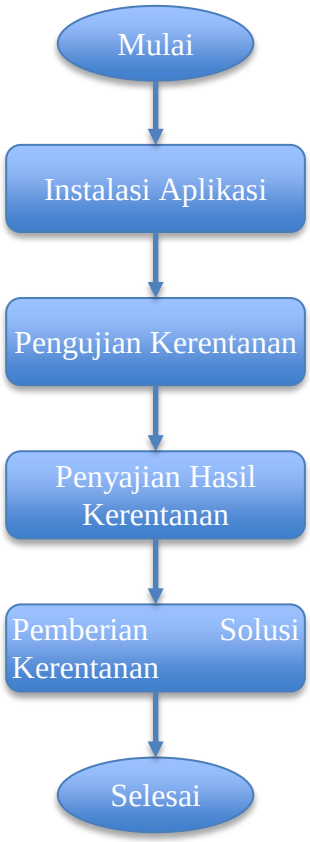
c. OWASP Zed Attack Proxy (ZAP)

OWASP ZAP adalah aplikasi *Open Source* yang dikembangakan oleh OWASP untuk melakukan analisis kerentanan dengan cara [8] :

1. *Vulnerability Assessment*, Sistem dipindai dan dianalisis untuk masalah keamanan.
2. *Penetration Testing*, Sistem menjalani analisis dan serangan dengan simulasi.
3. *Runtime Testing*, Sistem menjalani analisis dan pengujian keamanan dari pengguna akhir.
4. *Code Review*, Kode sistem menjalani tinjauan dan analisis mendetail yang secara khusus mencari kerentanan keamanan.

3. Metode

Penelitian dilakukan melalui 4 tahap, tahap pertama yang dilakukan ialah instalasi aplikasi, kemudian dilanjut dengan pengujian kerentanan, kemudian setelahnya dilakukan penyajian hasil dari pengujian kerentanan, dan terakhir adalah memberikan solusi dari kerentanan web. Skema metode penelitian dapat dilihat pada gambar berikut :



Gambar 3. Metode Penelitian

4. Hasil dan Pembahasan

Dengan menggunakan apikasi OWASP Zed Attcak Proxy, didapati bahwa Website Koperasi Karyawan Syariah PT BSIN memiliki 11 kerentanan. Langkah langkah

Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi
P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

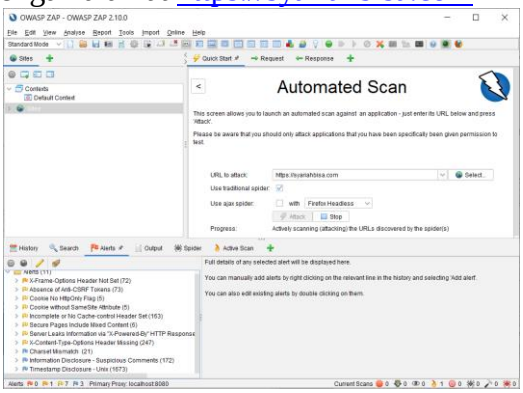
pengujian kerentanan tersebut adalah sebagai berikut

a. Instalasi Aplikasi

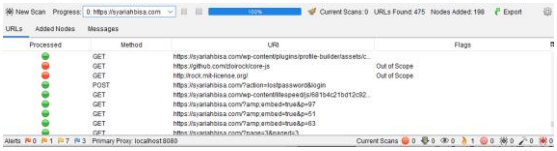
Dalam melakukan pengujian kerentanan website pada penelitian ini menggunakan Owap ZAP versi 2.10.0 yang digunakan sebagai aplikasi penguji kerentanan.

b. Prose Pengujian Kerentanan

Proses pengujian dan deteksi kerentanan pada Website Koperasi Karyawan Syariah PT BSIN dilakukan dengan menggunakan aplikasi OWASP ZAP seperti yang telah dijelaskan sebelumnya. Gambar 4 dan 5 merupakan proses memindai kerentanan pada website dengan alamat https://syariahbisa.com



Gambar 4. Kerentanan Website Koperasi Karyawan Syariah PT BSIN



Gambar 5. Proses Pemindaian Website

Setelah melakukan proses pemindaian, hasilnya dapat dilihat pada Tabel 1 yang menjelaskan jumlah jenis peringatan dan level resikonya. Secara total terdapat 11 jenis peringatan pada website Koperasi Karyawan Syariah PT BSIN. Detail mengenai jenis peringatan dan jumlah halaman web yang beresiko dapat dilihat pada Tabel 2.

Tabel 1. Resiko Kerentanan

Level Resiko	Jumlah Jenis Peringatan
High	0
Medium	1
Low	7
Informational	3

Tabel 2. Jenis Peringatan Kerentanan

Jenis	Level Resiko	Jumlah Contoh
X-Frame-Options Header Not Set	Medium	72
Absence of Anti-CSRF Tokens	Low	73
Cookie No HttpOnly Flag	Low	5
Cookie without SameSite Attribute	Low	5
Incomplete or No Cache-control Header Set	Low	163
Secure Pages Include Mixed Content	Low	6
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	185
X-Content-Type-Options Header Missing	Low	247
Charset Mismatch	Informational	21
Information Disclosure - Suspicious Comments	Informational	172
Timestamp Disclosure - Unix	Informational	1673

c. Rekomendasi Penanganan Kerentanan

Dari hasil proses pengujian kerentanan menggunakan aplikasi OWASP ZAP, perlu dilakukan perbaikan sesuai dengan kerentanan yang didapatkan. Berikut adalah rekomendasi penanganan kerentanan :

1). X-Frame-Options Header Not Set

Header X-Frame-Options tidak disertakan dalam respons HTTP untuk melindungi dari serangan '*ClickJacking*'. *Clickjacking* merupakan jenis serangan pada aplikasi web yang membuat korbannya secara tidak sengaja mengklik elemen halaman web yang sebenarnya. Sebagian besar browser web modern mendukung header HTTP *X-Frame-Options*. Pastikan hal tersebut di setting pada semua halaman web yang ada di situs web. Referensi :

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

2). Absence of Anti-CSRF Tokens

P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

Tidak ada token Anti CSRF yang ditemukan dalam formulir pengiriman HTML. *Cross-Site Request Forgery* (CSRF) adalah serangan yang melibatkan pemaksaan korban untuk mengirim permintaan HTTP ke tujuan target tanpa sepengetahuan atau niat mereka untuk melakukan tindakan sebagai korban. Penyebab dasarnya adalah fungsionalitas aplikasi yang menggunakan tindakan URL/formulir yang dapat diprediksi dengan cara yang berulang. Sifat serangannya adalah CSRF mengeksploitasi kepercayaan yang dimiliki situs web untuk pengguna. Sebaliknya, *Cross-Site Scripting* (XSS) mengeksploitasi kepercayaan yang dimiliki pengguna untuk situs web. Seperti XSS, serangan CSRF tidak harus lintas situs, tetapi bisa pemalsuan permintaan lintas situs juga dikenal sebagai CSRF, XSRF, *one-click attack, session riding, confused deputy*, dan *sea surf*. Referensi: http://projects.webappsec.org/Cross-Site-Request-Forgery, dan http://cwe.mitre.org/data/definitions/352.html

3). Cookie No HttpOnly Flag

Cookie telah disetel tanpa *flag HttpOnly*, yang berarti bahwa *cookie* dapat diakses oleh JavaScript. Jika skrip berbahaya dapat dijalankan di halaman ini, maka *cookie* akan dapat diakses dan dapat dikirim ke situs lain. Jika ini adalah *cookie* sesi, maka pembajakan sesi mungkin terjadi. Untuk pencegahannya pastikan bahwa *flag HttpOnly* disetel untuk semua *cookie*. Referensi: https://owasp.org/www-community/HttpOnly

4). Cookie without SameSite Attribute

Cookie telah disetel tanpa atribut *SameSite*, yang berarti *cookie* dapat dikirim sebagai hasil dari permintaan '*cross-site'*. Atribut *SameSite* adalah tindakan yang efektif untuk pemalsuan permintaan *cross-site*, penyertaan skrip *cross-site*, dan *timing attacks*. Solusinya dengan memastikan atribut *SameSite* diatur ke *'lax'* atau idealnya *'strict'* untuk semua *cookie*. Referensi: https://tools.ietf.org/html/draft-ietf-httpbis-cookie-same-site

5). Incomplete or No Cache-control Header Set

Header cache-control belum disetel dengan benar atau tidak ada, memungkinkan *browser* dan *proxy* untuk menyimpan konten dalam *cache*. Penanganannya dengan memastikan header HTTP kontrol-cache disetel dengan *no-cache, no-store, must-revalidate*. Referensi:https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching dan https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Cache-Control.

6). Secure Pages Include Mixed Content

Halaman tersebut mencakup konten campuran, yaitu konten yang diakses melalui HTTP, bukan HTTPS. Penanganannya, halaman yang tersedia melalui SSL/TLS harus sepenuhnya terdiri dari konten yang dikirimkan melalui SSL/TLS. Halaman tidak boleh berisi konten apa pun yang dikirimkan melalui HTTP yang tidak terenkripsi. Ini termasuk konten dari situs pihak ketiga. Referensi: https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

7). Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)

Server web/aplikasi membocorkan informasi melalui satu atau lebih header respons HTTP "*X-Powered-By*". Akses ke informasi tersebut dapat memfasilitasi penyerang untuk mengidentifikasi kerangka kerja/komponen lain yang diandalkan oleh aplikasi web dan kerentanan yang mungkin

P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

dialami oleh komponen tersebut. Penanganannya dengan memastikan server web, server aplikasi, *load balancer*, dan lain lain dikonfigurasi untuk menekan header "*X-Powered-By*". Referensi : http://blogs.msdn.com/b/varunm/archive/2013/04/23/remove-unwanted-http-response-headers.aspx dan http://www.troyhunt.com/2012/02/shhh-dont-let-your-response-headers.html

8). X-Content-Type-Options Header Missing

Header *Anti-MIME-Sniffing X-Content-Type-Options* tidak disetel ke *'nosniff'*. Ini memungkinkan versi *Internet Explorer* dan *Chrome* yang lebih lama untuk melakukan *sniffing* MIME (*Multipurpose Internet Mail Extensions*) pada *response body*, yang berpotensi menyebabkan *response body* ditafsirkan dan ditampilkan sebagai tipe konten selain tipe konten yang dideklarasikan. *Firefox* versi saat ini (awal 2014) dan lawas akan menggunakan tipe konten yang dideklarasikan (jika ada yang disetel), daripada melakukan *sniffing* MIME. Solusinya dengan memastikan aplikasi/server web menyetel header *Content-Type* dengan tepat, dan menyetel *header X-Content-Type-Options* ke *'nosniff'* untuk semua halaman web. Jika memungkinkan, pastikan bahwa pengguna akhir menggunakan browser web yang sesuai standar dan modern yang tidak melakukan *sniffing* MIME sama sekali, atau yang dapat diarahkan oleh aplikasi web/server web untuk tidak melakukan *sniffing* MIME. Referensi: http://msdn.microsoft.com/en-us/library/ie/gg622941%28v=vs.85%29.aspx dan https://owasp.org/www-community/Security_Headers

9). Charset Mismatch

Pemeriksaan ini mengidentifikasi respons di mana header HTTP *Content-Type* mendeklarasikan rangkaian karakter yang berbeda dari rangkaian karakter yang ditentukan oleh badan HTML atau XML. Ketika ada ketidakcocokan charset antara header HTTP dan isi konten, browser Web dapat dipaksa ke mode sniffing konten yang tidak diinginkan untuk menentukan rangkaian karakter konten yang benar. Penyerang dapat memanipulasi konten pada halaman untuk ditafsirkan dalam pengkodean pilihan mereka. Misalnya, jika penyerang dapat mengontrol konten di awal halaman, mereka dapat menyuntikkan skrip menggunakan teks yang disandikan UTF-7 dan memanipulasi beberapa browser untuk menafsirkan teks tersebut. Solusinya adalah dengan memaksa UTF-8 untuk semua konten teks di header HTTP dan tag meta dalam HTML atau deklarasi penyandian dalam XML. Referensi: http://code.google.com/p/browsersec/wiki/Part2#Character_set_handling_and_detection

10). Information Disclosure - Suspicious Comments

Tanggapan tersebut tampaknya berisi komentar mencurigakan yang dapat membantu penyerang. Catatan: Kecocokan yang dibuat dalam blok skrip atau file bertentangan dengan seluruh konten, bukan hanya komentar. Penanganannya dengan menghapus semua komentar yang mengembalikan informasi yang dapat membantu penyerang dan perbaiki masalah mendasar yang mereka rujuk.

11). Timestamp Disclosure – Unix

Stempel waktu diungkapkan oleh aplikasi/server web – Unix. Solusinya dengan mengkonfirmasikan secara manual bahwa data stempel waktu tidak sensitif, dan bahwa data tidak dapat digabungkan untuk mengungkapkan pola yang dapat dieksploitasi. Referensi: http://projects.webappsec.org/w/page/13246936/Information%20Leakage

P-ISSN : 1907-8420
E-ISSN : 2621-1106
DOI : 10.35969/interkom.v18i4.321

5. Penutup

Penelitian ini menghasilkan analisis kerentanan pada website Koperasi Karyawan Syariah PT BSIN. Kerentanan yang ditemukan adalah X-Frame-Options Header Not Set, Absence of Anti-CSRF Tokens, Cookie No HttpOnly Flag, Cookie without SameSite Attribute, Incomplete or No Cache-control Header Set, Secure Pages Include Mixed Content, Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s), X-Content-Type-Options Header Missing, Charset Mismatch, Information Disclosure - Suspicious Comments, Timestamp Disclosure - Unix. Setelah ditemukan kerentanan kemudian diberikan pencegahan untuk mengatasinya.

Daftar Pustaka

[1] A. Elanda, R. Buana, and R. Lintang. 2020. Analisis Keamanan Sistem Informasi Berbasis Website Dengan Metode Open Web Application Security Project (OWASP) Versi 4: Systematic Review. CESS Vol 5 No 2: 185-191.

[2] B. Subana, A. Fadlil, and Sunardi. 2020. Web Server Security Analysis Using The OWASP Mantra Method. Jurnal Mantik Vol 4 No 1: 107-116.

[3] D. Sukrianto, M. Alhafizh. 2019. Pemanfaatan Teknologi Berbasis Web Sistem Informasi Koperasi Syariah Pada Pengadilan Agama Pekanbaru. Jurnal Intra Tech Vol 3 No 2: 42-53.

[4] Guntoro, L. Costaner, and Musfawati. 2020. Analisis Keamanan Web Server Open Journal System (OJS) Menggunakan Metode ISSAF Dan OWASP (Studi Kasus OJS Universitas Lancang Kuning). JIPI Vol 5 No 1: 45-55.

[5] H. Mumtahana, S. Nita, A. Tito. 2017. Pemanfaatan Web E-Commerce untuk Meningkatkan Strategi Pemasaran. Khazanah Informatika Vol 3 No 1: 6-15.

[6] I. Riadi, R. Umar, T. Lestari. 2020. Analisis Kerentanan Serangan Cross Site Scripting (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP. JISKA Vol 5 No 3: 146-152.

[7] M. Yunus. 2021. Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework Owasp Versi 4. Jurnal Ilmiah Informatika Komputer Vol 24 No 1: 37-48.

[8] OWASP. 2018. OWASP ZAP 2.9 Getting Started Guide, The OWASP Foundation

[9] OWASP. 2017. OWASP Top 10 - 2017 The Ten Most Critical Web Application Security Risks, The OWASP Foundation

[10] R. Wibowo, A. Sulaksono. 2021. Web Vulnerability Through Cross Site Scripting (XSS) Detection with OWASP Security Shepherd. IJIS Vol 3 No 2: 149-159.

[11] Yudiana, A. Elanda, R. Buana, and R. Lintang. 2021. Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIK Rosma Dengan Menggunakan OWASP Top 10. CESS Vol 6 No 2: 37-43