CESS (Journal of Computer Engineering System and Science) **p-ISSN :2502-7131**
Vol. 6 No. 2 Juli 2021 **e-ISSN :2502-714x**

ANALISIS KUALITAS KEAMANAN SISTEM INFORMASI *E-OFFICE* BERBASIS WEBSITE PADA STMIK ROSMA DENGAN MENGGUNAKAN OWASP TOP 10

Yudiana[1], Anggi Elanda[2], Robby Lintang Buana[3]

[123]STMIK ROSMA
Jl. Kertabumi No. 62 Karawang 41311
yudiana@rosma.ac.id[1], anggi@rosma.ac.id[2], robby.buana@mhs.rosma.ac.id [3]

Abstrak- **STMIK ROSMA Karawang telah menerapkan pengarsipan dokumen surat di kampus, yang mana kesemuanya diatur secara daring (online) menggunakan sistem informasi e-office berbasis web. Sejak tahun 2020, Sistem Informasi e-office beberapa kali mengalami pengembangan dari sisi fitur yaitu fitur booking nomor surat dengan metode multiple insert query maupun data yang disimpan. Data tersebut menyimpan data dokumen surat baik itu surat masuk, surat keputusan, surat keuangan, surat keluar ketua, surat keluar wakil ketua dan surat keluar akademik STMIK ROSMA. Mengingat pentingnya data yang tersimpan maka perlu diterapkan pengujian keamanan dari sistem informasi e-office. Pengujian keamanan tersebut dilakukan untuk mengetahui tingkat kerentanan agar terhindar dari pihak yang tidak bertanggung jawab. Salah satu metode untuk menguji sistem informasi berbasis web adalah metode OWASP (Open Web Application Security Project) TOP 10 yang dikeluarkan oleh owasp.org sebuah organisasi nonprofit yang berdedikasi pada keamanan aplikasi berbasis web. Hasil pengujian menggunakan OWASP TOP 10 menunjukkan 10 kerentanan yang sering terjadi terhadap sistem informasi berbasis website sehingga perlu dilakukan perbaikan lebih lanjut oleh pihak pengembang sistem informasi e-office STMIK ROSMA.**

Kata Kunci - **keamanan sistem informasi, pentest, owasp, framework**

Abstract- **STMIK ROSMA Karawang has implemented the filing of mail documents on campus, all of which are arranged online using a web-based e-office information system. Since 2020, the e-office Information System has undergone several development in terms of features, namely the mail number booking feature with multiple insert query methods and stored data. The data stores data of letter documents be it incoming letters, decrees, financial letters, outgoing letters of the chairman, outgoing letters of vice chairmen and academic exit letters of STMIK ROSMA. Given the importance of stored data, security testing of e-office information systems is necessary. These security tests are conducted to determine the level of vulnerability to avoid irresponsible parties. One method to test web-based information systems is the OWASP (Open Web Application Security Project) TOP 10 method issued by owasp.org a non-profit organization dedicated to web-based application security. The test results using OWASP TOP 10 show 10 vulnerabilities that often occur to website-based information systems so that further improvements are needed by the developer of STMIK ROSMA e-office information system.**

Keywords - **information system security, pentest, owasp, framework.**

I. PENDAHULUAN

Tahun 2020 STMIK ROSMA untuk memenuhi pengarsipan dokumen surat baik itu surat masuk, surat keputusan, surat keuangan, surat keluar ketua, surat keluar wakil ketua dan surat keluar akademik menggunakan Sistem Informasi e-office berbasis website. Sistem Informasi e-office hanya bisa digunakan oleh lembaga internal STMIK ROSMA yang diakses melalui laman e-office.rosma.ac.id. Sejak tahun 2020, Sistem Informasi e-office beberapa kali mengalami pengembangan dari sisi fitur yaitu fitur booking nomor surat dengan metode multiple insert query. Berdasarkan hasil wawancara dengan pengelola Sistem Informasi e-office STMIK ROSMA hingga saat ini belum melakukan penetration test pada Sistem Informasi e-office STMIK ROSMA.

Data yang tersimpan pada database e-office berdasarkan identifikasi website telah mencapai lebih dari 2 GB. Database tersebut menyimpan pengarsipan dokumen surat. Aktivitas surat menyurat dalam suatu instansi memiliki urgensitas sendiri yang harus mendapatkan perhatian juga dari manajemen perusahaan atau instansi. Dalam jalannya kegiatan surat menyurat, tentu setiap surat menyimpan informasi tertentu yang memiliki tingkat kepentingannya sendiri, sehingga memang hanya orang-orang yang memiliki keperluan saja yang boleh melihat dan mengakses surat tersebut. oleh karena itu tingkat keamanan dari surat menyurat dalam suatu organisasi perlu mendapatkan perhatian khusus agar tidak terjadi hal-hal yang

merugikan organisasi, yang tentunya disebabkan oleh orang-orang yang tidak bertanggung jawab. Oleh karena itu, maka perlu diterapkan pengujian keamanan dari Sistem Informasi e-office STMIK ROSMA. Pengujian keamanan tersebut dilakukan untuk mengetahui tingkat kerentanan agar terhindar dari serangan dari pihak yang tidak bertanggung jawab.

Salah satu metode untuk menguji sistem informasi e-office berbasis web adalah metode OWASP (*Open Web Application Security Project*) Top 10 sebuah metode yang dirilis oleh komunitas *OWASP* yang berisikan 10 daftar teratas celah keamanan yang dapat mengancam keamanan suatu website daftar ini terus berkembang dan berubah-ubah mengikuti perkembangan teknologi website yang terus berkembang. Metode ini bebas digunakan oleh siapa saja yang ingin mengetahui kerentanan dari sebuah aplikasi web.

Dari penjelasan pada latar belakang diatas maka dilakukan penelitian untuk menerapkan Pengujian Sistem Informasi e-office STMIK ROSMA menggunakan OWASP ZAP dan Acunetix Web Vulnerability, karena dengan menggunakan aplikasi OWASP ZAP sebuah website dapat diuji sesuai kerentanan yang sering terjadi menurut OWASP Top 10 dan Acunetix Web Vulnerability Scanner adalah salah satu aplikasi scanner web terkemuka yang sangat baik sebagai solusi untuk memecahkan masalah keamanan situs web.

II. TINJAUAN PUSTAKA

Penetration Test

Penetration Testing adalah sebuah metode pengujian terhadap sebuah sistem atau jaringan komputer yang bertujuan untuk mengevaluasi keamanan sistem atau jaringan komputer tersebut.[1]

OWASP ZAP

OWASP ZAP adalah sebuah *tools vulnerabilities scanner* yang dibuat oleh organisasi *OWASP tools* ini adalah suatu proyek dari *OWASP* yang paling aktif karena terus dikembangkan *tools* ini bersifat *opensource* sehingga siapa saja juga bisa mengembangkan *tools* ini [1].

OWASP TOP 10

OWASP Top 10 atau yang biasa disebut *OWASP 10* adalah sebuah metode yang dirilis oleh komunitas *OWASP* yang berisikan 10 daftar teratas celah keamanan yang dapat mengancam keamanan suatu website daftar ini terus berkembang dan berubah-ubah mengikuti perkembangan teknologi website yang terus berkembang [2].

Parameter yang digunakan pada OWASP TOP 10 adalah sebagai berikut :

1. A1 – *Injection*
   Celah injeksi, seperti SQL,OS dan LDAP injection terjadi ketika data yang berbahaya dikirim ke interpreter sebagai bagian dari perintah atau query. Data berbahaya milik penyerang dapat mengelabui interpreter untuk mengeksekusi perintah yang tidak diinginkan atau mengakses data secara ilegal .
2. A2 *Broken Authentication*
   Fungsi aplikasi yang berhubungan dengan otentikasi dan manajemen sesi sering tidak diterapkan dengan benar, sehingga memungkinkan penyerang untuk mengambil password-password, kunci-kunci, token-token sesi, atau untuk mengeksploitasi celah implementasi lainnya untuk mengambil identitas pengguna.
3. A3 *Sensitive Data Exposure*
   Banyak aplikasi web yang tidak melindungi data rahasia dengan baik. Seperti kartu kredit, nomor PIN, dan kewenangan-kewenangan otentikasi. Penyerang akan mencuri atau merubah data yang diproteksi dengan lemah, untuk melakukan tindak pencurian identitas, kejahatan lewat kartu kredit, pencurian identitas, atau kejahatan lainnya. Data rahasia pantas terlindungi dengan baik menggunakan sandi enkripsi ketika tinggal atau dalam transit, sebagai pencegahan khusus atas tidak kejahatan ketika ketika tampil di browser.
4. A4 *XML External Entities* (XXE)
   Banyak bahasa XML versi lebih tua atau yang tidak dikonfigurasi dengan baik mengevaluasi referensi entitas eksternal dalam dokumen XML. Entitas eksternal dapat digunakan untuk mengungkapkan file internal menggunakan URL file, pembagian file internal, pemindaian port internal, eksekusi kode jarak jauh, dan penolakan serangan layanan
5. A5 *Broken Access Control*
   Pembatasan pada apa yang diizinkan oleh pengguna yang diautentikasi sering kali tidak dilakukan dengan benar. Penyerang dapat mengeksploitasi kelemahan ini untuk mengakses fungsionalitas dan / atau data yang tidak sah, seperti mengakses akun pengguna lain, melihat file sensitif, memodifikasi data pengguna lain, mengubah hak akses, dll.
6. A6 *Security Misconfiguration*
   Keamanan yang baik memerlukan konfigurasi keamanan yang terperinci dan telah menyeluruh pada framework aplikasi, aplikasi pada server, web server, database dan sistem operasi. Semua pengaturan ini harus didefinisikan, diimplementasikan dan dipelihara, karena terdapat banyak aplikasi yang dirilis tanpa konfigurasi default yang aman. Termasuk menjaga semua software untuk tetap terbarukan (up to date).
7. A7 *Cross-Site Scripting* (*XSS*)
   Celah XSS terjadi ketika sebuah aplikasi mengambil data berbahaya dan mengirimkannya ke web browser dengan tanpa memvalidasi atau melepaskan konten tersebut

dengan benar. XSS mengijinkan penyerang mengeksekusi script di browser target sehingga dapat leluasa mengambil alih data pengguna, merubah tampilan website target, atau mengarahkan korban ke laman yang berbahaya.

8. A8 *Insecure Deserialization*
   Eksploitasi deserialization adalah agak sulit, seperti di luar rak eksploitasi jarang bekerja tanpa perubahan atau menyesuaikan eksploitasi kode yang mendasarinya.
9. A9 *Using Components with Known Vulnerabilities*
   Komponen, seperti libraries, frameworks, dan modul perangkat lunak lainnya, dijalankan dengan hak yang sama seperti aplikasi. Jika komponen rentan dieksploitasi, serangan semacam itu dapat memfasilitasi hilangnya data serius atau pengambilalihan server. Aplikasi dan API yang menggunakan komponen dengan kerentanan yang diketahui dapat merusak pertahanan aplikasi dan memungkinkan berbagai serangan dan berdampak.
10. A10 *Insufficient Logging* & *Monitoring*
    Logging / Pencatatan dan pemantauan yang tidak memadai, ditambah dengan integrasi yang hilang atau tidak efektif dengan respons insiden, memungkinkan penyerang untuk menyerang sistem lebih lanjut, mempertahankan exploitasi, beralih ke lebih banyak sistem, dan mengutak-atik, mengekstrak, atau menghancurkan data. Sebagian besar studi pelanggaran menunjukkan waktu untuk mendeteksi pelanggaran lebih dari 200 hari, biasanya terdeteksi oleh pihak eksternal daripada proses internal atau pemantauan [1].

ISSAF

Framework ISSAF adalah standar pengujian penetrasi yang digunakan untuk menguji ketahanan situs web, yang memiliki beberapa keunggulan dibandingkan kontrol keamanan lainnya, dan berfungsi sebagai jembatan antara pandangan teknis dan manajerial [3].

DAST

DAST (Dynamic Application Security Testing) adalah teknik pengetesan pada interface yang terbuka pada sebuah aplikasi. Pengujian dilakukan dari luar ke dalam, teknologi ini sudah cukup lama dikenal, dan sebagian besar orang aplikasi sudah akrab dengan teknologi ini. DAST sangat baik dalam menemukan kerentanan eksternal. Kelemahan dari DAST adalah ketergantungan pada security experts yang mengimplementasi, sehingga sulit untuk diukur [4].

SAST

SAST adalah sudut pandang developer yang dapat mendeteksi security vulnerabilities lebih awal melalui review source code tanpa perlu aplikasi yang terdeploy sehingga menggunakan SAST pada saat system development [5].

CVE

Common Vulnerabilities and Exposures (CVE) adalah sistem yang menyediakan menyediakan metode referensi terkait kerentanan (vulnerability) dan paparan (exposure) keamanan informasi yang diketahui publik [6].

NVD

National Vulnerability Database (NVD) adalah gudang pemerintah AS untuk data manajemen kerentanan berbasis standar yang direpresentasikan menggunakan Security Content Automation Protocol (SCAP) [7].

WAF

Web Application Firewall (WAF) adalah bentuk khusus dari firewall aplikasi yang memfilter, memantau, dan memblokir lalu lintas HTTP ke dan dari layanan web. Dengan memeriksa lalu lintas HTTP, ini dapat mencegah serangan yang mengeksploitasi kerentanan aplikasi web yang diketahui, seperti injeksi SQL, skrip lintas situs (XSS), penyertaan file, dan konfigurasi sistem yang tidak tepat [8].

NIST

Framework ini menyediakan mekanisme penilaian yang memungkinkan organisasi/perusahaan menentukan kemampuan cybersecurity saat ini, menetapkan sasaran individual, dan membuat rencana untuk memperbaiki dan memelihara program cybersecurity [9].

TLS

Transport Layer Security (TLS) adalah protokol yang dirancang untuk menyediakan komunikasi yang aman melalui web. Oleh karena itu, situs web yang menggunakan TLS menyediakan jalur komunikasi yang aman antara web mereka server dan browser web mencegah penyadapan, pembajakan, dan serangan aktif lainnya [10].

Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

TABEL I
PENELITIAN TERDAHULU TENTANG PENGUJIAN *WEBSERVER* MENGGUNAKAN *OWASP*

Literatur	Judul	Penulis	Target	Tools
[11]	Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online)	Mohammad Muhsin, Adi Fajaryanto	Aplikasi Ujian Online	*WebScarab , Brutus , Browser Mozilla Firefox , Wfuzz , Dirb , Zed Attack Proxy , OWASP CSRF Tester*
[12]	Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi *Security Tools Project* Berdasarkan *Framework OWASP* Versi 4	Moh Yunus	*Website www.xyz.com*	*Browser Mozilla Firefox , Netsparker , OWASP ZAP Attack Proxy , Google Chrome (Plugin)*
[13]	Penerapan Metode *ISSAF* dan *OWASP* versi 4 Untuk Uji Kerentanan Web Server	Dr. Raden Teduh Dirgahayu, S.T., M.Sc., Yudi Prayudi, S.Si., M.Kom., Adi Fajaryanto	*WebServer*	*WebScarab , Brutus , Browser Mozilla Firefox , Wfuzz , Dirb , Zed Attack Proxy , OWASP CSRF Tester*
[14]	Analisis Deteksi Vulnerability Pada Webserver Open Jurnal System Menggunakan OWASP Scanner	Yunanri. W, Imam Riadi, Anton Yudhana	*Open Journal System*	*Open Web Application Security Project (OWASP)*
[15]	Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode Owasp (Open Web Application Security Project) untuk Penilaian Risk Rating	Bahrun Ghozali, Kusrini, Sudarmawan	Sistem Informasi Harga Komoditas Utama	*OWASP Risk Rating, Acunetix Web Vulnerability Scanner.*

III. METODE PENELITIAN

Metode penelitian ini diselesaikan dengan tahap-tahap kegiatan dalam Gbr 1.

Gbr. 1 Tahapan Penelitian

1. *Studi Literatur*

Tahap ini bertujuan untuk menjelaskan kajian pustaka dari teori-teori penunjang yang mendukung konstruksi penelitian. Kegiatan ini dilakukan dengan membaca buku, jurnal, artikel laporan penelitian, dan situs-situs di internet.

2. *Identifikasi Website*

Sistem informasi e-office STMIK ROSMA adalah suatu sistem yang dibuat untuk memenuhi pengarsipan dokumen surat di kampus, yang mana kesemuanya diatur secara daring (online). Dalam sistem informasi ini Lembaga internal rosma dapat mengarsipkan dokumen surat baik itu surat masuk, surat keputusan, surat keuangan, surat keluar ketua, surat keluar waket dan surat keluar akademik. Sistem Informasi ini dapat diakses melalui laman e-office.rosma.ac.id.

3. *Metode OWASP TOP 10*

OWASP merupakan organisasi non-profit amal di Amerika Serikat yang didirikan pada tanggal 21 April 2004 yang berdedikasi untuk membuat framework pengujian keamanan yang bebas digunakan oleh siapa saja. Pengujian kerentanan Sistem Informasi e-office STMIK Rosma yang digunakan pada OWASP TOP 10 adalah sebagai berikut.

a. A1 Injection
   Pengujian A1 Injection menggunakan OWASP ZAP secara otomatis menggunakan Tools Active Scan Rules dengan versi rilis.
b. A2 Broken Authentication
   Pengujian A2 Broken Authentication menggunakan OWASP ZAP secara otomatis menggunakan tools Access Control Testing.
c. A3 Sensitive Data Exposure
   Pengujian A3 Sensitive Data Exposure menggunakan OWASP ZAP secara otomatis menggunakan Tools Active Scan Rules dengan versi rilis.
d. A4 XML External Entities (XXE)
   Pengujian A4 XML External Entities (XXE) menggunakan OWASP ZAP secara otomatis menggunakan Tools Active Scan Rules dengan versi rilis.
e. A5 Broken Access Control
   Pengujian A5 Broken Access Control menggunakan OWASP ZAP secara otomatis menggunakan Tools Active Scan Rules dengan versi rilis
f. A6 Security Misconfiguration
   Pengujian A6 Security Misconfiguration menggunakan OWASP ZAP secara manual menggunakan tools Access Control Testing.
g. A7 Cross-Site Scripting (XSS)
   Pengujian A7 Cross-Site Scripting (XSS) menggunakan OWASP ZAP secara otomatis menggunakan Tools Active Scan Rules dengan versi rilis
h. A8 Insecure Deserialization

Pengujian A8 Insecure Deserialization menggunakan OWASP ZAP secara otomatis menggunakan Tools Insecure deserialization active scanner & Java Serialization Handling.

i. A9 Using Components with Known Vulnerabilities
Pengujian A9 Using Components with Known Vulnerabilities menggunakan OWASP ZAP secara otomatis menggunakan Tools Passive Scan Rules versi Alpha.

j. A10 Insufficient Logging & Monitoring
pengujian A10 Insufficient Logging & Monitoring menggunakan OWASP ZAP secara menggunakan Tools Access Control.

4. Pengujian Penetrasi

Untuk melihat lebih detail mengenai pengujian kerentanan berdasarkan *OWASP Top 10* terhadap Sistem Informasi e-office STMIK Rosma menggunakan *OWASP ZAP* dapat melihat tabel berikut ini.

TABEL II
PENGUJIAN PENETRASI *OWASP ZAP*

OWASP TOP 10	Cara	Tools
A1-Injection	Otomatis	*Active Scan Rules*
A2-Broken Authentication	Otomatis	*Access Control Testing*
A3-Sensitive Data Exposure	Otomatis	*Active Scan Rules*
A4-XML External Entities	Otomatis	*Active Scan Rules*
A5-Broken Access Control	Otomatis	*Active Scan Rules*
A6-Security Misconfiguration	Manual	*Access Control Testing*
A7-Cross Site Scripting	Otomatis	*Active Scan Rules*
A8-Insecure Deserialization	Otomatis	*Insecure Deserialization Active Scanner & Java Serialization Handling*
A9- Using Components with Known Vulnerabilities	Otomatis	*Passive Scan Rules*
A10-Using Components with Known Vulnerabilities	Otomatis	*Access Control*

5. Hasil Pengujian

Setelah melakukan pengujian penetrasi menggunakan *OWASP ZAP* maka akan mendapatkan hasil pengujian terhadap sistem informasi e-office yang akan menjadi tolak ukur dari kegiatan analisis.

6. Analisis

Kegiatan Analisis dilakukan berdasarkan hasil dari melakukan pengujian penetrasi menggunakan *OWASP ZAP* yang akan menentukan kualitas keamanan sistem informasi e-office STMIK ROSMA.

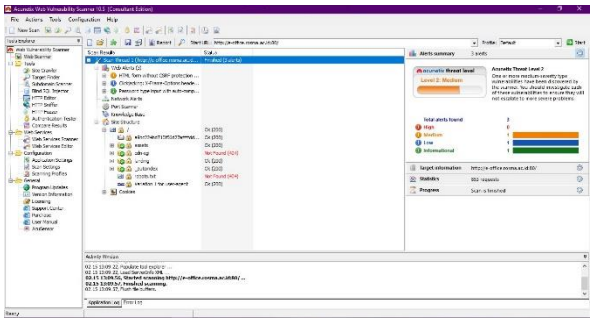
7. Kesimpulan

Kesimpulan berisi uraian hasil pengujian dan analisis dari sistem informasi e-office STMIK ROSMA dan dari hasil .

IV. HASIL DAN PEMBAHASAN

A. Identifikasi Kerentanan

Identifikasi kerentanan dalam penelitian ini menggunakan aplikasi Acunetix untuk mengetahui tingkat kerentanan yang ada. Berikut adalah hasil scan Acunetix :



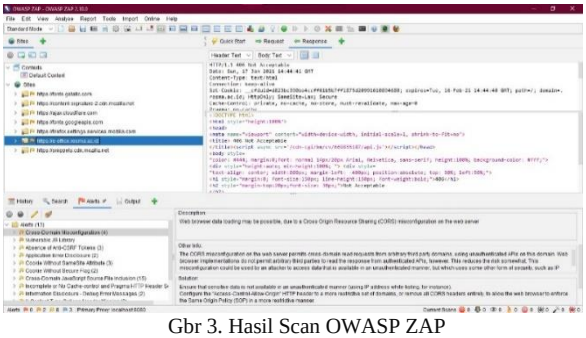
Gbr 2.Hasil Scan Acunetix

Berikut ini merupakan hasil pengujian sistem informasi e-office STMIK ROSMA menggunakan Acunetix Web Vulnerability, dalam melakukan pengujian sistem informasi e-office STMIK Rosma terdapat 3 kerentanan yaitu :

1. HTML form without CSRF protection
   Kerentanan : Sedang
2. Clickjacking: X-Frame-Options header missing
   Kerentanan : Rendah
3. Password type input with auto-complete enabled
   Kerentanan : Informatif

Oleh karena itu, Acunetix Web Vulnerability menilai sistem informasi e-office STMIK ROSMA berada di Level 2 : Sedang dengan memiliki 1 kerentanan Sedang, 1 kerentanan Rendah dan 1 kerentanan Informatif.

B. Penetrasi OWASP ZAP



Gbr 3. Hasil Scan OWASP ZAP

Berdasarkan Hasil Pengujian OWASP ZAP sistem informasi e-office STMIK ROSMA memiliki 13 kerentanan yaitu, :

1. Cross-Domain Misconfiguration (4)
2. Vulnerable JS library
3. Absence of Anti-CSRF Tokens (3)
4. Application Error Disclosure (2)
5. Cookie Without SameSite Attribute (3)
6. Cookie Without Secure Flag (2)
7. Cross-Domain JavaScript Source File Inclusion (15)
8. Incomplete or No Cache-control and Pragma HTTP Header Set (35)
9. Information Disclosure – Debug Error Messages (2)
10. X-Content-Type-Options Header Missing (3)
11. Information Disclosure - Suspicious Comments (6)
12. Loosely Scoped Cookies (14)
13. Timestamp Disclosure – Unix (75)

C. Penetrasi OWASP ZAP Berdasarkan OWASP TOP 10

Berdasarkan hasil pengujian OWASP ZAP Sistem Informasi e-office STMIK ROSMA memiliki 13 kerentanan. 4 dari 13 kerentanan tersebut termasuk ke dalam daftar OWASP TOP 10. Berikut hasil pengujian OWASP ZAP berdasarkan OWASP TOP 10 yang disajikan ke dalam Tabel III.

TABEL III
HASIL PENGUJIAN OWASP ZAP TOP 10

OWASP TOP 10	Tools	Kerentanan
A1 Injection	*Active Scan Rules*	Tidak Rentan
A2 Broken Authentication	*Access Control Testing*	Tidak Rentan
A3 Sensitive Data Exposure	*Active Scan Rules*	Rendah
A4 XML External Entities (XXE)	*Active Scan Rules*	Tidak Rentan
A5 Broken Access Control	*Passive Scan Rules*	Tidak Rentan
A6 Security Misconfiguration	*Access Control Testing*	Sedang
A7 Cross-site Scripting (XSS)	*Active Scan Rules*	Rendah
A8 Insecure Deseralization	*Active Scan Rules*	Rendah
A9 Using Components with Known Vulnerabilities	*Passive Scan Rules*	Tidak Rentan
A10 Insufficient Logging & Monitoring	*Access Control*	Tidak Rentan

Berdasarkan Tabel III Sistem informasi e-office terdapat kerentanan di A3 Sensitive Data Exposure dengan menggunakan tools Active Scan Rules memiliki tingkat kerentanan Low, A6 Security Misconfiguration menggunakan tools Access Control Testing dengan tingkat kerentanan Medium, A7 Cross Site Scripting menggunakan tools Active Scan Rules dengan tingkat kerentanan Low, A8 Insecure Deseralization menggunakan tools Active Scan Rules dengan tingkat kerentanan Low.

D. Rekomendasi

Untuk meminimalisir kerentanan yang dihasilkan dalam pengujian sistem informasi e-office maka pengembang sistem dapat mengikuti saran berikut ini.

1. Konfigurasikan header HTTP "Access-Control-Allow-Origin" ke kumpulan domain yang lebih ketat, atau hapus semua header CORS seluruhnya, untuk memungkinkan browser web menerapkan Kebijakan Asal yang Sama (SOP) dengan cara yang lebih ketat.
2. Harap tingkatkan ke versi terbaru jquery.
3. Gunakan pustaka atau kerangka kerja terverifikasi yang tidak memungkinkan kelemahan terjadi atau menyediakan konstruksi yang membuat kelemahan ini lebih mudah dihindari.
4. Pertimbangkan untuk menerapkan mekanisme untuk memberikan referensi / pengenal
5. kesalahan unik ke klien (browser) saat mencatat detail di sisi server dan tidak memaparkannya kepada pengguna.
6. Pastikan atribut SameSite disetel ke 'lax' atau idealnya 'tight' untuk semua cookie.
7. Setiap kali cookie berisi informasi sensitif atau merupakan token sesi, cookie harus selalu diteruskan menggunakan saluran terenkripsi.

8. Pastikan file sumber JavaScript dimuat hanya dari sumber terpercaya, dan sumber tidak dapat dikontrol oleh pengguna akhir aplikasi.
9. Kapan pun memungkinkan, pastikan header HTTP kontrol-cache disetel dengan no-cache, no-store, must-revalidate; dan bahwa header HTTP pragma disetel no-cache.
10. Nonaktifkan pesan debug sebelum melanjutkan ke produksi.
11. Pastikan bahwa aplikasi / server web menyetel tajuk Jenis Konten dengan tepat, dan bahwa itu menetapkan header X-Content-Type-Option ke 'nosniff' untuk semua halaman web.
12. Hapus semua komentar yang mengembalikan informasi yang dapat membantu penyerang dan memperbaiki masalah mendasar yang dirujuknya.
13. Selalu lingkup cookie ke FQDN (Fully Qualified Domain Name).
14. Konfirmasikan secara manual bahwa data stempel waktu tidak sensitif, dan bahwa data tidak dapat digabungkan untuk mengungkap pola yang dapat dieksploitasi.
15. Periksa apakah formulir ini memerlukan perlindungan CSRF dan terapkan tindakan pencegahan CSRF jika perlu.
16. Konfigurasikan server web untuk menyertakan header X-Frame-Options.
17. Perlengkapan otomatis kata sandi harus dinonaktifkan dalam aplikasi sensitif.

V. PENUTUP

A. Kesimpulan

Berdasarkan pengujian menggunakan OWASP ZAP menunjukkan bahwa sistem informasi e-office memiliki 13 kerentanan dan berdasarkan OWASP TOP 10 sistem informasi e-office STMIK ROSMA terdeteksi memiliki 4 kerentanan yaitu *Sensitive Data Exposure*, *Security Misconfiguration*, *Cross Site Scripting*, dan *Insecure Deseralization* maka dengan dilakukannya pengujian penetration test kualitas keamanan sistem informasi e-office berada di tingkat sedang sehingga perlu dilakukan perbaikan lebih lanjut oleh pihak pengembang sistem informasi e-office STMIK ROSMA dengan mengikuti rekomendasi dari penulis.

B. Saran

Berdasarkan kesimpulan diatas maka perlu dilakukan penelitian dengan metode ISSAF (Information System Security Assessment Framework) agar dapat diketahui kerentanan dari sisi web server
.

REFERENSI

[1] OWASP, “The ten Most Critical Web Application Security Risk,” http://www.owasp.org, 2017.
[2] O. ZAP, “ZAPping the OWASP Top 10,” https://www.zaproxy.org/docs/guides/zapping-the-top-10/, 2020. .
[3] I. G. A. S. Sanjaya, G. M. A Sasmita and D. M. S. Arsa. “Evaluasi Keamanan Website Lembaga X Melalui Penetration Testing Menggunakan Framework ISSAF,” *J. Ilm. Merpati*, vol. Vol. 8, No, 2020.
[4] WASC, “Kriteria Evaluasi Pemindai Keamanan Aplikasi Web versi 1.0,” 2013.
[5] A. Santoso, “Static Application Security Testing SAST VS Dynamic Application Security Testing DAST,” 2020.
[6] Mitre Corporation, “CVE is sponsored by the National Cyber Security Division of the U.S. Department of Homeland Security.,” 2013.
[7] nvd.nist.gov, “NVD - CVSS v2 Equations,” 2013.
[8] T. Target, “Web Application Firewall,” 2018.
[9] V. I. Sugara, H. Syahrial and M. Syafrullah. “Sistem Pemeriksa Keamanan Informasi Menggunakan National Institute Of Standards And Technology (Nist) Cybersecurity Framework,” *Komputasi J. Ilm. Ilmu Komput. dan Mat*., vol. Vol 16, No, 2019.
[10] R. T. and T. S. Nicolas-Rocca, “Application Level Security in a Public Library : A Case Study,” *Inf. Technol. Libr*., 2018.
[11] M. Muhsin, “Penerapan Pengujian Keamanan Web Server Menggunakan Metode OWASP versi 4 (Studi Kasus Web Server Ujian Online),” *Multitek Indones*., vol. 9, No. 1, pp. 31–42.
[12] M. Yunus, “Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework OWASP Versi 4,” *J. Ilm. Inform. Komput*., vol. 24, No. 1, pp. 38–50, 2019.
[13] R. T. Dirgahayu, Y. Prayudi and A. Fajaryanto. “Penerapan Metode ISSAF dan OWASP versi 4 Untuk Uji Kerentanan Web Server,” *J. Ilm. NERO*, vol. 1 No. 3, pp. 190–197, 2015.
[14] Y. W, I. Riadi and A. Yudhana. “Analisis Deteksi Vulnerability Pada Webserver Open Jurnal System Menggunakan OWASP Scanner,” *JURTI*, vol. Vol. 2 No., 2018.
[15] B. Ghozali, Kusrini and Sudarman. “Mendeteksi Kerentanan Keamanan Aplikasi Website Menggunakan Metode OWASP (Open Web Application Security Project) untuk Penilaian Risk Rating,” *Citec J.*, vol. Vol. 4 No., 2017.