Jurnal Informasi dan Teknologi

http://www.jidt.org

2022 Vol. 4 No. 3 Hal: 160-165 e-ISSN: 2714-9730

Analisis Metode Open Web Application Security Project (OWASP) Menggunakan Penetration Testing pada Keamanan Website Absensi

Ichsan Octama Riandhanu✉

Universitas Gunadarma

ichsano.r@gmail.com

Abstract

The use of technology in various fields increases mobility, one of which is the creation of websites to share and manage information. Information system security that cannot interfere with the infrastructure of an organization or company. Many system vulnerabilities or system problems occur on the internet. These problems can be in the form of Malware attacks, Exploits and Injection databases. This problem can be minimized by implementing security from hackers' interference or attacks by means of penetration testing (Pentest), which is testing carried out on the web legally by imitating the form of hackers. To detect web security, an analysis of the vulnerabilities of a web is required in accordance with the security standardization of the Open Web Application Security Project (OWASP) using security tools. Web-based vulnerability analysis with the OWASP method using security tools is able to determine the security level of an application, based on the results of tests that have been carried out where the results of the research provide some suggestions or recommendations about website vulnerabilities, which can be used by the website development team to improve website security.

Keywords: Vulnerability, OWASP, Website, Penetration Testing, Security.

Abstrak

Penggunaan teknologi dalam berbagai bidang meningkatkan mobilitas, salah satunya dengan pembuatan website untuk berbagi dan mengelola informasi. Keamanan sistem informasi yang tidak baik dapat mengganggu infrastruktur suatu organisasi atau perusahaan. Masalah kerentanan atau gangguan keamanan sistem banyak terjadi di internet. Masalah tersebut dapat berjenis serangan Malware, Eksploitasi dan Injeksi database. Masalah tersebut dapat diminimalisir dengan menerapkan pengamanan web dari gangguan atau serangan hacker dengan cara *penetration testing* (Pentest) yaitu pengujian yang dilakukan terhadap web secara legal dengan aktifitas menyerupai *hacker*. Untuk mendeteksi keamanan web dibutuhkan sebuah analisis terhadap kerentanan sebuah web yang sesuai dengan standarisasi kemanan Open Web Application Security Project (OWASP) dengan menggunakan tools security. Analisis kerentanan aplikasi berbasis web dengan metode OWASP dengan menggunakan tools security mampu mengetahui tingkat keamanan suatu aplikasi, berdasarkan hasil pengujian yang telah dilakukan dimana hasil dari penelitian memberikan beberapa saran atau rekomendasi tentang kerentanan situs web, yang dapat digunakan oleh tim developer situs web untuk meningkatkan keamanan situs web tersebut.

Kata kunci: Kerentanan, OWASP, Website, Penetration Testing, Keamanan.

1. Pendahuluan

Penerapan teknologi di berbagai bidang semakin membuat mobilitas semakin tinggi, salah satunya dengan membuat website untuk bertukar dan mengelola informasi. Pandemi Covid-19 mengakibatkan beberapa perusahaan menerapkan kerja secara *work from home* (WFH) sehingga membuat mobilitas dalam bekerja menjadi semakin tinggi dan dibutuhkan sistem untuk menunjang pekerjaan tersebut. Perusahaan X adalah perusahaan yang bergerak dibidang impor yang pada saat ini sudah menerapkan sistem bekerja dari rumah atau *work from home*, dengan adanya sistem bekerja tersebut terdapat tuntutan dibukanya akses website absensi yang sebelumnya dilakukan secara manual menggunakan sidik jari menjadi menggunakan media website yang dipublish melalui internet. Aplikasi sistem absensi saat ini sudah berjalan dan sudah dipakai oleh pengguna internal perusahaan. Seluruh fungsi yang ada dalam aplikasi sebelumnya sudah melalui tahap uji coba dan sudah sesuai dengan kebutuhan karyawan perusahaan. Seiring meningkatnya kebutuhan penggunaan website tersebut maka perlu diiringi dengan tingkat keamanan yang baik, sehingga diperlukan upaya untuk mengamankan website tersebut baik dari sisi infrastrukur dan aplikasi.

Salah satu cara paling akurat untuk mengevaluasi sikap keamanan informasi organisasi adalah dengan mengamati bagaimana organisasi tersebut berdiri melawan serangan, cara terbaik untuk memastikan bahwa sistem aman adalah dengan mencoba pengujian penetrasi, pengujian penetrasi sering kali memungkinkan analisis keamanan menemukan kerentanan baru [1].

Salah satu metode untuk menguji sistem berbasis web adalah metode OWASP (*Open Web Application Security Project*) Top 10 sebuah metode yang dirilis oleh komunitas OWASP yang berisikan 10 daftar

Diterima: 18-09-2022 | Revisi: 25-09-2022 | Diterbitkan: 30-09-2022 | doi: 10.37034/jidt.v4i3.236

teratas celah keamanan yang dapat mengancam keamanan suatu website [2]. OWASP memprioritaskan 10 besar berdasarkan daya eksploitasi, prevalensi umum, kemudahan deteksi dan dampak parahnya [3].

Untuk memahami kerentanan umum dalam aplikasi web membantu pelaku bisnis lebih siap dalam melindungi data mereka dari serangan dan OWASP Top 10 dapat membantu dalam pemilihan tindakan yang yang harus diambil untuk mengurangi kerentanan [4].

Analisis kerentanan aplikasi berbasis web dengan teknik OWASP versi 4 mampu mengetahui keamanan suatu aplikasi. Berdasarkan hasil pengujian kerentanan pada website menggunakan beberapa tahapan kategori yaitu tahap *Authentication Testing Authorization*, Session *Management Testing*, *Input Validation Testing*, dan *Error Handling* pada metode OWASP versi 4 dapat dijadikan sebagai standar penilaian keamanan aplikasi berbasis web [5].

Pengujian dengan menggunakan alat pemindai kerentanan dan menggunakan OWASP sebagai standart untuk langkah pengujian dapat ditemukan kerentanan informasi berupa versi webserver, permintaan GET dan POST, sistematika URL, *framework* website, *website builder component*, dan arsitektur web [6]. Pengujian dengan menggunakan metode OWASP dapat menemukan kerentanan kelemahan yang masuk pada kategori *authentication testing* [7].

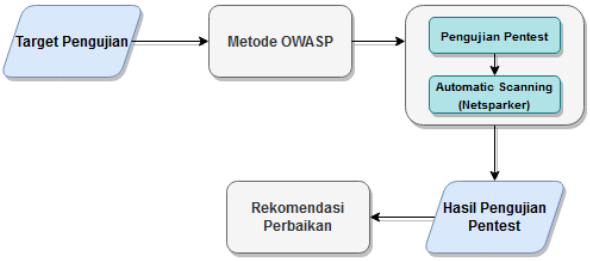
Penggunaan tools *Web Application Vulnerability Scanners* (WAVS) dapat membantu pengembang untuk mengidentifikasi kerentanan. Menggunakan tools *vulnerability scanner* seperti Acunetix WVS dapat membantu assesment dan perbaikan untuk mengurangi vulnerability yang ditemukan [8]. Pengujian menggunakan alat OWASP-ZAP memiliki deteksi kerentanan yang lebih tinggi beri peringkat dalam kategori *open source* kemudian Acunetix dan NetSparker memiliki tingkat positif palsu yang lebih sedikit [9]. Hasil pemindaian menggunakan alat *vulnerability scanner* dapat memberikan informasi kerentanan berdasarkan tingkat risiko [10].

Pengujian menggunakan metode OWASP dengan menggunakan alat pemindai kerentanan perlu dilakukan konfigurasi untuk memaksimalkan tingkat akurasi [11].

Pada penelitian ini dilakukan analisis *open web application security project* (OWASP) *top* 10 2017 menggunakan metode penetration testing dalam menentukan vulnerability website absensi perusahan X.

2. Metodologi Penelitian

Penelitian ini dimulai dengan menentukan target pengujian yaitu webite absensi, kemudian dilanjutkan dengan penerapan metode OWASP sebagai standart pengujian aplikasi web. Langkah berikutnya adalah pengujian *pentest* dengan menggunakan alat pemindai kerentanan netsparker. Tapan terakhir adalah analisa hasil pengujian pentest, hasil analisa pengujian pentest akan dijadikan rekomendasi perbaikan kerentanan kepada tim pengembang aplikasi. Alur penelitian dari setiap langkah yang dilakukan dalam penelitian ini dapat disajikan pada Gambar 1.



Gambar 1. Alur Penelitian

2.1. Target Pengujian

Judul Penelitian ini dilaksanakan pada website absensi pada perusahaan X yang terletak di wilayah Jakarta Selatan. Perusahaan ini penulis pilih karena sebagai tempat penulis bekerja dan juga memanfaatkan teknologi website untuk absensi yang dapat diakses dari dalam dan dari luar kantor. Pada Tabel 1 berisi informasi terkait infrastruktur web absensi.

Tabel 1. Target Informasi

Informasi Infrastruktur	
Sistem Operasi	CentOS 7.9
Hostname	https://sub.domain.com
Versi PHP	7.0.33
Webserver	Apache 2.4.6
Jenis Aplikasi	Web
Jenis Akses	Intranet dan Internet

2.2. Metode OWASP

Strategi pengujian kerentanan terhadap aplikasi web menggunakan metode OWASP top 10 dilakukan dengan pendekatan box testing. Terdapat 10 Parameter yang digunakan pada OWASP TOP 10 sebagai berikut [12]:

a) Injection

Injeksi merupakan cara untuk memasukkan data ke dalam aplikasi yang kemudian diinterpretasikan atau dieksekusi oleh aplikasi. Tahap ini memastikan penyerang tidak dapat mengirim data yang tidak sah ke dalam aplikasi. Contoh dari kasus injeksi yaitu penyerang memodifikasi nilai parameter ‘id’ pada browser untuk mengirim ‘or’1’=’1.

b) Broken Authentication

Otentikasi dan fungsi manajemen sesi dalam aplikasi web digunakan untuk memverifikasi identitas pengguna. Implementasi yang salah dari fungsi-fungsi ini memungkinkan penyerang untuk mengkompromikan kata sandi, kunci, atau token sesi. Contoh skenario serangan yaitu melakukan *brute force login* pada web untuk mencari penggunaan sandi yang lemah dan tidak kompleks.

c) Sensitive Data Exposure

Paparan data sensitif terjadi ketika informasi tidak dilindungi secara memadai. Contoh data sensitif adalah info pembayaran, kredensial, nomor telepon, alamat email, data pribadi, dan catatan kesehatan. Tahap ini untuk memastikan tidak terdapat data sensitif yang tidak dilindungi. Contoh skenario serangan yaitu attacker memonitor traffic dan mencuri sesi cookie pada website yang tidak menerapkan https.

d) XML External Entities

Penyerang dapat mengeksploitasi prosesor XML yang rentan jika mereka dapat mengunggah XML atau memasukkan konten yang tidak bersahabat dalam dokumen XML, mengeksploitasi kode, dependensi, atau integrasi yang rentan.

e) *Broken Access Control*

Kelemahan dalam kontrol akses umum terjadi karena kurangnya deteksi otomatis, dan kurangnya pengujian fungsional yang efisien oleh pengembang aplikasi. Penyerang yang dapat mengeksploitasi kerentanan kontrol akses dapat memodifikasi atau menghapus semua data, melakukan fungsi yang berada di luar batas pengguna, atau mendapatkan akses tidak sah ke informasi.

f) Security Misconfiguration

Kelemahan ini dapat dimanfaatkan oleh penyerang untuk mendapatkan akses yang tidak sah atau bahkan sepenuhnya membahayakan sistem. Kesalahan konfigurasi keamanan sering kali dapat dideteksi oleh pemindai otomatis. Contoh skenario serangan yaitu penyerang menemukan list directory yang terbuka.

g) Cross-Site Scripting

Masalah XSS sangat umum dan dapat ditemukan di sekitar dua pertiga dari semua aplikasi. Ada tiga bentuk Cross-Site Scripting: Reflected XSS, Stored XSS, dan DOM XSS, yang semuanya dapat dideteksi dan dieksploitasi dengan alat otomatis. Contoh skenario serangan yaitu penyerang mencuri session ID untuk dikirim ke website penyerang sehingga, penyerang dapat mengambil alih session pengguna.

h) Insecure Deserialization

Aplikasi atau API (*Application programming interface*) yang melakukan deserialize objek yang tidak bersahabat atau rusak dari sumber yang tidak tepercaya akan rentan terhadap serangan. Deserialisasi yang tidak aman dapat menyebabkan serangan replay, serangan injeksi, serangan eskalasi hak istimewa atau dalam kasus terburuk, eksekusi kode jarak jauh.

i) Using Components with Known Vulnerabilities

Pada tahap ini memastikan penggunaan aplikasi pendukung tidak menggunakan versi kadaluarsa yang dapat menyebabkan aplikasi menjadi rentan.

j) Insufficient Logging and Monitoring

Pada tahap ini memastikan monitoring dan respon dengan tepat sehingga dapat mendeteksi penyerang dan mencegah terjadinya insiden yang lebih besar.

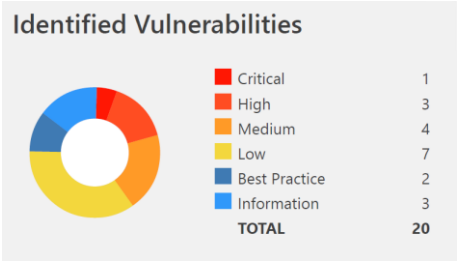
2.3. Pengujian Pentest

Pengujian penetrasi (*Penetration Testing*) adalah pendekatan proaktif yang mapan untuk mengevaluasi keamanan aset digital dengan secara aktif mengidentifikasi dan mengeksploitasi kerentanan yang ada [13]. Strategi pengujian kerentanan terhadap aplikasi web absensi menggunakan metode OWASP menggunakan bentuk analisis dinamis (*Dynamic Analysis)* yang dilakukan pada domain dimana aplikasi web target beroperasi [14], [15], [16]. Hasil data kerentanan didapatkan melalui proses pemindaian dan penilaian kerentanan dengan alat pemindaian kerentanan dalam melakukan pengujian kerentanan aplikasi web menggunakan Netsparker [17], [18], [19], [20]. Proses pemindaian dilakukan dengan cara memasukkan alamat website target pada aplikasi netsparker dan dilakukan optimalisasi konfigurasi *policy.*

3. Hasil dan Pembahasan

3.1. Hasil Pengujian

Setelah proses pemindaian pada aplikasi sub.domain.com selesai dilakukan menggunakan Netsparker, hasil analisis dinamis (*Dynamic Analysis*) disajikan pada Gambar 2.



Gambar 2. Laporan Pemindaian Netsparker

Rangkuman hasil pemindaian kerentanan yang ditemukan berdasarkan tingkat resiko disajikan pada Tabel 2.

Tabel 2. Rangkuman Total Jumlah Kerentanan

Tingkat Resiko	Jumlah Kerentanan
Critical (Kritis)	1
High (Tinggi)	3
Medium (Menengah)	4
Low (Rendah)	7
Best Practice (Praktik terbaik)	2
Information (Informasi)	3

Rangkuman hasil pemindaian kerentanan yang ditemukan berdasarkan kategori OWASP *top* 10 disajikan pada Tabel 3.

Tabel 3. Rangkuman kerentanan berdasarkan kategori OWASP

OWASP top 10	Kerentanan
A1 - Injection	Tidak ditemukan
A2 – Broken authentication	Tidak ditemukan
A3 – Sensitive data exposure	Ditemukan
A4 – XML external entities (XXE)	Tidak ditemukan
A5 – Broken access control	Tidak ditemukan
A6 – Security misconfiguration	Ditemukan
A7 – Cross-site scripting	Tidak ditemukan
A8 – Insecure deserlization	Tidak ditemukan
A9 – Using component with known vulnerabilities	Ditemukan
A10 – Insufficient logging & monitoring	Tidak ditemukan

Perincian hasil pemindaian kerentanan berdasarkan OWASP top 10 yang berhasil ditemukan dapat dilihat sebagai berikut:

a. A3 – Sensitive Data Exposure

Hasil pengujian menggunakan netsparker ditemukan kerentanan pada kategori *sensitive data exposure* dengan perincian disajikan pada Tabel 4.

Tabel 4. Sensitive Data Exposure

Kerentanan	Method	Url	Resiko
Weak Cipher Enabled	GET	https://sub.domain.com/	Medium
HTTP Strict Transport Security (HSTS) Policy Not Enabled	GET	https://sub.domain.com/	Medium
Cookie Not Marked as Secure	GET	https://sub.domain.com/	Low
Insecure Transportation Security Protocol Supported (TLS 1.0)	GET	https://sub.domain.com/	Low

b. A6 - Security Misconfiguration

Temuan berikutnya terdapat kerentanan pada kategori *security misconfiguration* dengan perincian seperti pada Tabel 5.

Tabel 5. Kategori *security misconfiguration*

Kerentanan	Method	Url	Resiko
Autocomplete is Enabled	GET	https://sub.domain.com/	Rendah
Missing Content-Type Header	GET	https://sub.domain.com/__assets/template/bower_components/bootstrap/dist/fonts/glyphicons-halflings-regular.woff2	Rendah
Missing X-Frame-Options Header	GET	https://sub.domain.com/	Rendah
TRACE/TRACK Method Detected	TRACE	https://sub.domain.com/	Rendah
Version Disclosure (PHP)	GET	https://sub.domain.com/	Rendah

c. A9 - *Using Components With Known Vulnerabilities*

Hasil pengujian menggunakan netsparker ditemukan juga kerentanan pada kategori using *components with known vulnerabilities* dengan perincian disajikan pada Tabel 6.

Tabel 6. Kategori *Using Components with Known Vulnerability*

Kerentanan	Method	Url	Resiko
Out-of-date Version (PHP)	GET	https://sub.domain.com/	Kritis
Out-of-date Version (Chart.js)	GET	https://sub.domain.com/	Tinggi
Out-of-date Version (jQuery Validation)	GET	https://sub.domain.com/ __assets/template/bower_components/Chart.js/Chart.js	Tinggi
Out-of-date Version (Moment.js)	GET	https://sub.domain.com/ __assets/pluginsall/jquery-validation/dist/jquery.validate.js	Tinggi
Out-of-date Version (Bootstrap)	GET	https://sub.domain.com/ __assets/template/bower_components/bootstrap/dist/js/bootstrap.min.js?v=1629341241	Sedang
Out-of-date Version (jQuery)	GET	https://sub.domain.com/	Sedang

3.2. Rekomendasi Perbaikan

Pada tahap ini dilakukan analisa terhadap pengujian dengan memberikan rekomendasi tindak lanjut sebagai pedoman untuk perbaikan selanjutnya. Rekomendasi yang diberikan berdasarkan hasil pengujian dan analisa dari OWASP. Hasil rekomendasi tindakan perbaikan dikelompokan berdasarkan tingkat resiko dimulai dari resiko kritis hingga resiko rendah, untuk hasilnya adalah sebagai berikut:

a) A3 – *Sensitive Data Exposure*

Rekomendasi tindakan perbaikan kerentanan pada kategori *sensitive data exposure* disajikan pada Tabel 7.

Tabel 7. Sensitive data exposure

Kerentanan	Url	Tindakan Perbaikan
Weak Cipher Enabled	https://sub.domain.com/	Disable penggunaan enkripsi yang lemah dengan cara melakukan konfigurasi SSLCipherSuite pada file httpd.conf
HTTP Strict Transport Security (HSTS) Policy Not Enabled	https://sub.domain.com/	Konfigurasi fitur *redirect* HTTP *request* ke HTTPS
Cookie Not Marked as Secure	https://sub.domain.com/	Menandai semua cookie dengan aman
Insecure Transportation Security Protocol Supported (TLS 1.0)	https://sub.domain.com/	Mengnonaktifkan TLS 1.0 dan mengganti menggunakan TLS 1.2 pada modul *mod_ssl*

b) A6 - Security Misconfiguration

Rekomendasi tindakan perbaikan kerentanan pada kategori *security misconfiguration* dengan perincian disajikan pada Tabel 8.

Tabel 8. Kategori *security misconfiguration*

Kerentanan	Url	Tindakan Perbaikan
Autocomplete is Enabled	https://sub.domain.com/	Menambahkan atribut *autocomplate=" off"* pada form tag
Missing Content-Type Header	https://sub.domain.com/__assets/template/bower_components/bootstrap/dist/fonts/glyphicons-halflings-regular.woff2	Memastikan content-type header sesuai dengan tipe pada sumber
Missing X-Frame-Options Header	https://sub.domain.com/	Menambahkan konfigurasi Header set X-Frame-Options: "SAMEORIGIN" pada file httpd.conf
TRACE/TRACK Method Detected	https://sub.domain.com/	Disable *method* TRACK/TRACE pada file httpd.conf
Version Disclosure (PHP)	https://sub.domain.com/	Konfigurasi *expose_php* menjadi *off* pada file php.ini

c) A9 - *Using Components With Known Vulnerabilities*

Rekomendasi tindakan perbaikan kerentanan pada kategori using *components with known vulnerabilities* dengan perincian disajikan pada Tabel 9.

Tabel 9. Kategori *Using Components with Known Vulnerability*

Kerentanan	Url	Tindakan Perbaikan
Out-of-date Version (PHP)	https://sub.domain.com/	melakukan *upgrade* instalasi PHP ke versi terbaru
Out-of-date Version (Chart.js)	https://sub.domain.com/	Melakukan *update* versi chart.js ke versi 3.9.1
Out-of-date Version (jQuery Validation)	https://sub.domain.com/__assets/template/bower_components/Chart.js/Chart.js	Melakukan *upgrade* ke versi 1.19.5
Out-of-date Version (Moment.js)	https://sub.domain.com/ __assets/pluginsall/jquery-validation/dist/jquery.validate.js	Melakukan update versi moment.js ke versi 2.29.4
Out-of-date Version (Bootstrap)	https://sub.domain.com/__assets/template/bower_components/bootstrap/dist/js/bootstrap.min.js?v=1629341241	Melakukan upgrade ke versi 3.4.1
Out-of-date Version (jQuery)	https://sub.domain.com/	Melakukan upgrade jquery ke versi 3.6.0

4. Kesimpulan

Hasil dari penelitian ini dapat diambil kesimpulan bahwa analisis keamanan aplikasi berbasis web menggunakan metode OWASP telah terbukti mampu mengetahui kerentanan keamanan yang berada pada aplikasi web absensi sub.domain.com. Berdasarkan hasil pengujian ditemukan kerentanan pada tingkat ancaman kritis (*critical*) sebanyak 1 temuan, tingkat tinggi (*high*) sebanyak 3 temuan, tingkat menengah (*medium*) sebanyak 4 temuan dan tingkat rendah (*low*) sebanyak 7 temuan. Kerentan yang ditemukan berdasarkan kategori OWASP *top* 10 adalah *sensitive data exposure*, *security misconfiguration*, dan *using components with known vulnerabilities*, rekomendasi dari hasil pengujian diharapkan dapat membantu tim developer aplikasi memperbaiki dan menutup celah kerentanan.

Daftar Rujukan

[1] Zeebaree, S. R. M., Jacksi, K., & Zebari, R. R. (2020). Impact analysis of SYN flood DDoS attack on HAProxy and NLB cluster-based web servers. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1), 505–512. DOI: https://doi.org/10.11591/ijeecs.v19.i1.pp505-512 .

[2] Yudiana, Y., Elanda, A., & Buana, R. L. (2021). Analisis Kualitas Keamanan Sistem Informasi E-Office Berbasis Website Pada STMIKRosma Dengan Menggunakan OWASP Top 10. *CESS (Journal of Computer Engineering, System and Science)*, *6*(2), 37-43. DOI: https://doi.org/10.24114/cess.v6i2.24777 .

[3] Idris, I., Majigi, M. U., Abdulhamid, S., Olalere, M., & Rambo, S. I. (2017). Vulnerability assessment of some key Nigeria government websites. *International Journal of Digital Information and Wireless Communications*, *7*(3), 143-153. DOI: http://dx.doi.org/10.17781/P002309 .

[4] Bach-Nutman, M. (2020). Understanding the top 10 owasp vulnerabilities. *arXiv preprint arXiv:2012.09960*. DOI: https://doi.org/10.48550/arXiv.2012.09960 .

[5] Yunus, M. (2019). Analisis Kerentanan Aplikasi Berbasis Web Menggunakan Kombinasi Security Tools Project Berdasarkan Framework Owasp Versi 4. *Jurnal Ilmiah Informatika Komputer*, *24*(1), 37-48. DOI: http://dx.doi.org/10.35760/ik.2019.v24i1.1988 .

[6] Pratama, I. P. A. E., & Wiradarma, A. A. B. A. (2019). Open source intelligence testing using the owasp version 4 framework at the information gathering stage (case study: X company). *International Journal of Computer Network and*

Information Security, *11*(7), 8-12. DOI: http://dx.doi.org/10.5815/ijcnis.2019.07.02 .

[7] Guntoro, G., Costaner, L., & Musfawati, M. (2020). Analisis Keamanan Web Server Open Journal System (Ojs) Menggunakan Metode Issaf Dan Owasp (Studi Kasus Ojs Universitas Lancang Kuning). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *5*(1), 45-55. DOI: http://dx.doi.org/10.29100/jipi.v5i1.1565 .

[8] Zirwan, A. (2022). Pengujian dan Analisis Keamanan Website Menggunakan Acunetix Vulnerability Scanner. *Jurnal Informasi dan Teknologi*, 70-75. DOI: https://doi.org/10.37034/jidt.v4i1.190 .

[9] Shahid, J., Hameed, M. K., Javed, I. T., Qureshi, K. N., Ali, M., & Crespi, N. (2022). A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions. *Applied Sciences*, *12*(8), 4077. DOI: http://dx.doi.org/10.3390/app12084077 .

[10] Riadi, I., & Raharja, P. A. (2019). Vulnerability analysis of E-voting application using open web application security project (OWASP) framework. *International Journal of Advanced Computer Science and Applications*, *10*(11). DOI: https://dx.doi.org/10.14569/IJACSA.2019.0101118 .

[11] Fathurrahmad, F., & Ester, E. (2020). Automatic Scanner Tools Analysis As A Website Penetration Testing: Automatic Scanner Tools Analysis As A Website Penetration Testing. *Jurnal Mantik*, *4*(2), 1138-1144. DOI: https://doi.org/10.35335/mantik.Vol4.2020.886.pp1138-1144 .

[12] Nedeljković, N., Vugdelija, N., & Kojić, N. (2020, October). Use of "OWASP Top 10" in web application security. *In Fourth International Scientific Conference on Recent Advances in Information Technology, Tourism, Economics, Management and Agriculture* (p. 25). DOI: https://doi.org/10.31410/ITEMA.2020.25 .

[13] Filiol, E., Mercaldo, F., & Santone, A. (2021). A method for automatic penetration testing and mitigation: A red hat approach. *Procedia Computer Science*, 192, 2039-2046. DOI: http://dx.doi.org/10.1016/j.procs.2021.08.210 .

[14] Viriya, A., & Muliono, Y. (2021). Peeking and Testing Broken Object Level Authorization Vulnerability onto E-Commerce and E-Banking Mobile Applications. *Procedia Computer Science*, *179*, 962-965. DOI: https://doi.org/10.1016/j.procs.2021.01.101 .

[15] Aryanti, D., & Utamajaya, J. N. (2021). Analisis Kerentanan Keamanan Website Menggunakan Metode Owasp (Open Web Application Security Project) Pada Dinas Tenaga Kerja. *Jurnal Syntax Fusion*, *1*(03), 15-25. DOI: https://doi.org/10.54543/fusion.v1i03.53 .

[16] Mateus, E., & Serrão, C. (2021). Vulnerability assessment of Angolan university web applications. *Vulnerability assessment of Angolan university web applications*, 518-525. DOI: http://dx.doi.org/10.5220/0010716800003058 .

[17] Gultom, L. M., & Harahap, M. (2018). Analisis Celah Keamanan Website Instansi Pemerintahan di Sumatera Utara. *Jurnal Teknovasi: Jurnal Teknik dan Inovasi Mesin Otomotif, Komputer, Industri dan Elektronika*, *2*(2), 1-7. DOI: http://dx.doi.org/10.55445/teknovasi.v2i2.54 .

[18] Mateo Tudela, F., Bermejo Higuera, J. R., Bermejo Higuera, J., Sicilia Montalvo, J. A., & Argyros, M. I. (2020). On Combining Static, Dynamic and Interactive Analysis Security Testing Tools to Improve OWASP Top Ten Security Vulnerability Detection in Web Applications. *Applied Sciences*, *10*(24), 9119. DOI: https://doi.org/10.3390/app10249119 .

[19] Fachri, F., Fadlil, A., & Riadi, I. (2021). Analisis Keamanan Webserver Menggunakan Penetration Test. *Jurnal Informatika*, *8*(2), 183-190. DOI: https://doi.org/10.31294/ji.v8i2.10854 .

[20] Ula, M. (2019). Evaluasi Kinerja Software Web Penetration Testing. *TECHSI-Jurnal Teknik Informatika*, *11*(3), 336-352. DOI: https://doi.org/10.29103/techsi.v11i3.1996