Jurnal Informasi dan Teknologi

http://www.jidt.org

2020 Vol. 2 No. 4 Hal: 133-138 ISSN: 2714-9730 (*electronic*)

Meningkatkan Keamanan Sistem Informasi Puskesmas Terpadu dengan Metode Grey-Box Penetration Test Menggunakan Computer Assisted Audit Techniques

Nopan Pirsa[1]✉, Sumijan[2]

[1,2]Universitas Putra Indonesia YPTK Padang

nopan.pirsa@gmail.com

Abstract

Information system security is something the owner must pay attention to in order to avoid cyber crime. Information systems that have security vulnerabilities can threaten an organization's critical infrastructure. Security vulnerabilities are any kind of vulnerability that allows attackers to enter into the system illegally and perform unwanted acts. Penetration tests on information systems need to be carried out to ensure its security. This study aims to improve the security of the integrated health center information system in Payakumbuh City so that data and information is guaranteed its security. The method used is the gray box penetration test using computer assisted audit techniques. The results of this study found 97 high vulnerability categories, 1 medium vulnerability category and 26 low vulnerability categories. The Payakumbuh City Information and Communication Office can take advantage of the penetration test results as a reference to improve the security of the Payakumbuh integrated health center information system.

Keywords: Information System Security, Penetration Testing, Vulnerability Assessment, Computer Assisted Audit Techniques, OWASP ZAP.

Abstrak

Keamanan sistem informasi merupakan hal yang wajib diperhatikan oleh pemiliknya agar terhindar dari kejahatan siber. Sistem informasi yang memiliki kerentanan keamanan dapat mengancam infrastruktur penting suatu organisasi. Kerentanan keamanan adalah segala jenis celah yang memungkinkan penyerang untuk dapat masuk kedalam sistem secara illegal dan melakukan tidakan yang tidak diinginkan. *Penetration test* pada sistem informasi perlu dilakukan untuk memastikan keamanannya. Penelitian ini bertujuan untuk meningkatkan keamanan sistem informasi puskesmas terpadu Kota Payakumbuh sehingga data dan informasi yang ada terjamin keamanannya. Metode yang digunakan adalah *grey box penetration test* menggunakan *computer assisted audit techniques*. Hasil penelitian ini ditemukan 97 kerentanan kategori tinggi, 1 kerentanan kategori sedang dan 26 kerentanan kategori rendah. Dinas Komunikasi dan Informatika Kota Payakumbuh dapat memanfaatkan hasil *penetration test* sebagai referensi untuk meningkatkan keamanan sistem informasi puskesmas terpadu Kota Payakumbuh.

Kata kunci: Keamanan Sistem Informasi, Pengujian Penetrasi, Penilaian Kerentanan, Teknik Audit Berbantuan Komputer, OWASP ZAP.

1. Pendahuluan

Pemanfaatan sistem informasi saat ini sangat banyak digunakan untuk menunjang kinerja suatu organisasi. Sistem informasi berbasis web menjadi pilihan utama dikarenakan kemudahan dalam mengakses dan mendistribusikan. Semua sistem informasi berbasis web rentan terhadap peretasan [1]. Dibutuhkan suatu keamanan dalam sebuah sistem informasi berbasis web [2].

Tahun 2019, terdata sebanyak 4241 (empat ribu dua ratus empat puluh satu) aduan yang terdiri dari 699 aduan tidak terverifikasi dan 3542 (tiga ribu lima ratus empat puluh dua) aduan terverifikasi. Proses verifikasi meliputi proof of concept dari bukti-bukti laporan (screenshoot, link, database, correlated file) maupun kelengkapan identitas pelapor. Kerentanan merupakan jenis aduan siber yang paling banyak diajukan [3].

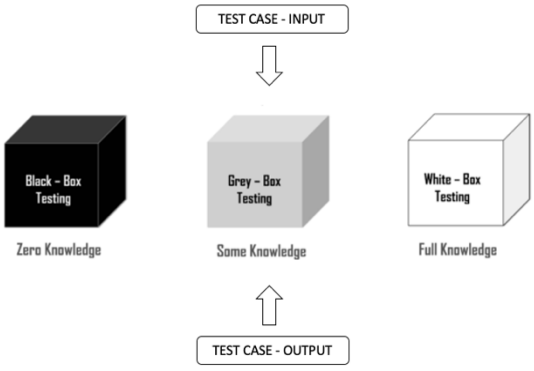
Sistem informasi puskesmas terpadu Kota Payakumbuh (SIPADUKO) adalah aplikasi berbasis web yang dikembangkan Pemerintah Kota Payakumbuh melalui Dinas Komunikasi dan Informatika. Aplikasi ini dibangun untuk menunjang kinerja puskesmas. Mulai dari pendaftaran pasien hingga pengambilan obat di apotik dikelola oleh aplikasi ini. Aplikasi dapat diakses melalui jaringan publik. Dengan dibukanya akses melalui jaringan publik, aplikasi berpotensi diserang oleh peretas. Berdasarkan laporan Dinas Komunikasi dan Informatika Kota Payakumbuh, percobaan-percobaan peretasan banyak terjadi pada aplikasi tersebut.

Diterima: 04-09-2020 | Revisi: 15-09-2020 | Diterbitkan: 31-12-2020 | DOI: 10.37034/jidt.v2i4.79

Aktifitas peretasan sistem bukan berasal dari orang yang berada di luar organisasi saja, tetapi orang yang berada di dalam organisasi juga berpotensi melakukannya. Peretas biasanya menargetkan sistem informasi yang sudah mereka kenal dengan baik akses sebagai user [4]. User yang telah diberikan otorisasi untuk mengakses sistem, memiliki peluang yang besar untuk dapat meretas sistem tersebut.

Penetration test digunakan untuk mengidentifikasi risiko yang mungkin terjadi ketika penyerang mendapatkan akses ke sebuah sistem. Dengan melakukan penetration test, celah keamanan pada sistem dapat ditutup sebelum serangan yang sebenarnya terjadi. Salah satu tujuan utama penetration test adalah untuk menciptakan keamanan suatu sistem [5].

Penetration test pada sistem perlu dilakukan untuk memastikan data yang disimpan pada server tetap aman [6]. Penetration test adalah proses yang dilakukan untuk mengungkap dan menemukan kerentanan pada suatu sistem [7]. Ada tiga strategi penetration test berdasarkan lingkup dan jenis audit yaitu *Black Box Testing*, *White Box Testing* dan *Grey Box Testing* [8].



Gambar 1. Strategi Penetration Test

Computer assisted audit technique (CAAT) adalah penggunaan teknologi untuk membantu penyelesaian proses audit [9]. CAAT memudahkan untuk mengakses berbagai file yang bersifat elektronik dan melakukan proses audit dengan baik sehingga penipuan dapat dicegah diawal [10]. Penggunaan CAAT memiliki dampak yang positif terhadap kinerja auditor dalam melakukan proses audit [11].

Zed Attack Proxy (ZAP) adalah perangkat lunak proyek unggulan dari Open Web Application Security Project (OWASP) yang membantu proses penteration test [12]. Menggunakan Web Application Firewall (WAF) untuk melindungi aplikasi berbasis web memang dapat mencegah peretasan, akan tetapi belum menyelesaikan masalah dikarenakan kerentanan berada pada sisi aplikasi. Salah satu cara untuk mengatasi hal ini adalah dengan melakukan pengujian keamanan pada aplikasi.

2. Metodologi Penelitian

Untuk menyelesaikan suatu masalah, diperlukan langkah-langkah yang tepat. Langkah-langkah tersebut dituangkan dalam bentuk kerangka kerja yang dapat dilihat pada Gambar 2.

Gambar 2. Kerangka Kerja Penelitian

Uraian langkah dari Gambar 2 adalah sebagai berikut :

2.1. Mengidentifikasi Masalah

Pada tahapan ini ditentukan ruang lingkup permasalahan. Berdasarkan informasi dari Dinas Komunikasi dan Informatika Kota Payakumbuh, aplikasi Sistem Informasi Puskesmas Terpadu Kota Payakumbuh (SIPADUKO) pernah berhasil ditembus oleh hacker. Hal ini menandakan masih ada terdapat kerentanan pada aplikasi SIPADUKO.

2.2. Melakukan Studi Literatur

Pada tahapan ini dilakukan pengumpulan informasi lebih lanjut dari permasalahan yang ada. Informasi didapatkan dari berbagai sumber tertulis dengan cara mempelajari datanya. Sumber tertulis yang dipelajari berupa tesis, disertasi, buku, paper dan artikel yang terkait dengan penelitian ini.

2.3. Persiapan Penetration Test

Pada tahapan ini dilakukan persiapan sebelum melakukan penetration test pada aplikasi SIPADUKO. Persiapan yang dilakukan adalah sebagai berikut:

a. Meminta izin kepada Dinas Komunikasi dan Informatika Kota Payakumbuh untuk melakukan *penetration test* pada aplikasi SIPADUKO yang sudah di duplikasi untuk keperluan penetration test.

b. Meminta data *username* dan *password* seluruh level *user* mulai dari tata usaha, loket, perawat, dokter, laboratorium dan apoteker untuk digunakan dalam proses *penetration test*.

c. Memasang OWASP ZAP versi 2.9.0.

2.4. Mengumpulkan Informasi Aplikasi

Pada tahapan ini dilakukan pengumpulan seluruh informasi aplikasi SIPADUKO melalui ZAP *proxy server*. Informasi didapatkan dengan cara mengakses semua menu, proses dan laporan dari aplikasi SIPADUKO untuk semua user yang telah diberikan. Informasi yang didapatkan berupa *URL*, nama *class*, nama *function*, *POST* atau *GET* parameter, *http response* dan lainnya akan disimpan pada *session* OWASP ZAP.

2.5. Mengidentifikasi Kerentanan Aplikasi

Pada tahapan ini dilakukan pemindaian kerentanan aplikasi SIPADUKO dengan bantuan OWASP ZAP *active scan*. Semua informasi yang sudah disimpan pada *session* OWASP ZAP akan di pindai kerentanannya. Hasil dari pemindaian ini juga akan disimpan pada *session* OWASP ZAP.

5. Exploitation

Pada tahapan ini dilakukan exploitasi pada aplikasi SIPADUKO dengan cara manual dan menggunakan tool yang spesifik untuk kerentanan berdasarkan hasil pemindaian. Eksploitasi dilakukan bertujuan untuk memverifikasi hasil pemindaian kerentanan. Kerentanan dengan hasil verifikasi *false positive* tidak akan di laporkan.

2.6. Hasil Penetration Test

Pada tahapan ini akan dilakukan analisis dari hasil *exploitation*. Semua kerentanan yang ditemukan pada aplikasi SIPADUKO akan dianalisis penyebabnya beserta cara menutup kerentanannya. Laporan dan saran akan diberikan kepada Dinas Komunikasi dan Informatika Kota Payakumbuh untuk meningkatkan keamanan aplikasi SIPDUKO agar tujuan dan manfaat penelitian ini dapat dicapai untuk meningkatkan keamanan aplikasi SIPADUKO.

3. Hasil dan Pembahasan

Untuk melakukan pengujian keamanan dengan metode *grey box penetration test* peneliti membutuhkan data *user* dan *password* aplikasi SIPADUKO. Data user aplikasi SIPADUKO dapat dilihat pada Tabel 1.

Tabel 1. Data Username Aplikasi SIPADUKO

Username	Password	Keterangan
super_admin	xxxxxxxx	Username administrator aplikasi
user_tata_usaha	xxxxxxxx	Username Kepala Tata Usaha Puskesmas
user_loket	xxxxxxxx	Username petugas loket
user_perawat_u	xxxxxxxx	Username perawat poli umum
user_dokter_u	xxxxxxxx	Username dokter poli umum
user_perawat_g	xxxxxxxx	Username perawat poli gigi
user_dokter_g	xxxxxxxx	Username dokter poli gigi
user_bidan	xxxxxxxx	Username bidan poli KIA
user_labor	xxxxxxxx	Username petugas laboratorium Puskesmas
user_apotek	xxxxxxxx	Username petugas apotek Puskesmas

Dengan menggunakan semua *username* dari aplikasi SIPADUKO pada Tabel 1, dilakukan pengujian keamanan pada aplikasi SIPADUKO dengan tahapan:

3.1. Planning

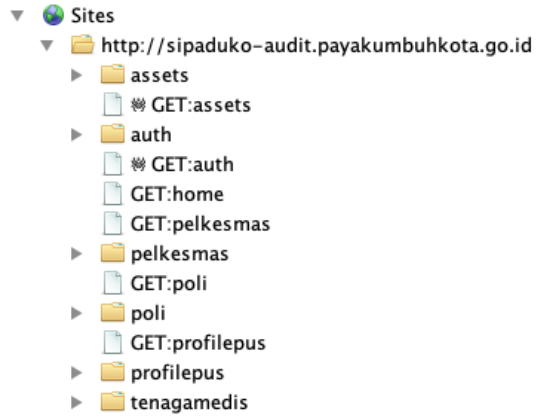
Pada tahapan ini peneliti mempersiapkan semua kebutuhan untuk penetration test. Untuk pengujian keamanan peneliti menggunakan computer assisted audit technique (CAAT) tool OWASP ZAP versi 2.9.0 dan tools exploit. Spesifikasi peralatan untuk penetration test ini dapat diihat pada Tabel 2.

Tabel 2. Spesifikasi Peralatan Untuk Penetration Test

Nama Peralatan	Spesifikasi
Laptop	OS: macOS Catalina Versi 10.15.3 64 bit Processor: 1,6 GHz Dual-Core Intel Core i5 RAM: 4 GB 1600 MHz DDR3 VGA: Intel HD Graphics 6000 1536 MB HDD: SSD 128 GB
Cisco AnyConnect	Versi 4.8.03036
OWASP ZAP	Versi 2.9.0
Koneksi Internet	Up to 10 Mbps
Web Browser	Firefox Mozilla versi 68.0.1

3.2. Information Gathering

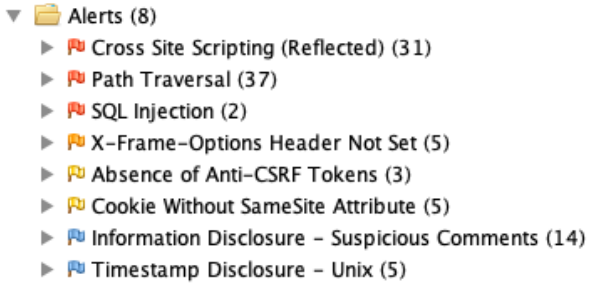
Pada tahap ini peneliti mengumpulkan semua informasi aplikasi SIPADUKO berdasarkan data user yang sudah didapatkan dari Dinas Kominfo. Informasi dikumpulkan dengan cara mengakses semua menu melalui *browser* yang sudah di pasang *proxy* OWASP ZAP. Informasi dikumpulkan untuk setiap data user yang ada pada aplikasi dan menyimpan hasilnya pada session OWASP ZAP. Hasil pengumpulan informasi menggunakan *username* user_tata_usaha didapat sebanyak 60 *URL* seperti pada Gambar 3.



Gambar 3. Hasil Pengumpulan Informasi User user_tata_usaha

3.3. Vulnerability Scanning

Pada tahapan ini dilakukan pemindaian kerentanan aplikasi SIPADUKO menggunakan OWASP ZAP *passive scan* dan *active scan*. Pemindaian dilakukan untuk setiap user berdasarkan informasi aplikasi SIPADUKO yang sudah dikumpulkan pada tahap sebelumnya. Hasil *vulnerability scanning* menggunakan *username* user_tata_usaha ditemukan 3 *high alert*, 1 *medium alert*, 2 *low alert* dan 2 *informational alert* seperti pada Gambar 4.



Gambar 3. Hasil Pemindaian User user_tata_usaha

3.4. Exploitation

Pada tahapan ini dilakukan ekploitasi terhadap kerentanan yang ditemukan pada tahap *vulnerability scanning*. Eksploitasi bertujuan untuk memverifikasi hasil *vulnerability scanning*. Eksploitasi dilakukan dengan cara manual dan menggunakan *tool* yang spesifik untuk kerentanannya. Hasil ekspoitasi yang didapat adalah sebagai berikut:

a. Kerentanan *Cross Site Scripting (Reflected)*

Cross Site Scripting (Reflected) adalah kerentanan yang disebabkan oleh parameter *input* yang tidak di validasi. Penyerang akan mengirimkan *payload* berupa *client side script* pada parameter *input*. Serangan yang berhasil akan menyebabkan *payload* yang dikirimkan akan dieksekusi di dalam *browser*. Verifikasi hasil pemindaian dilakukan dengan cara memeriksa secara manual *response* dari aplikasi. Dari enam puluh *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil 53 URL terkonfirmasi dapat dieksploit dan 7 *URL false positive* atau tidak memiliki kerentanan ini.

b. Kerentanan *Path Traversal*

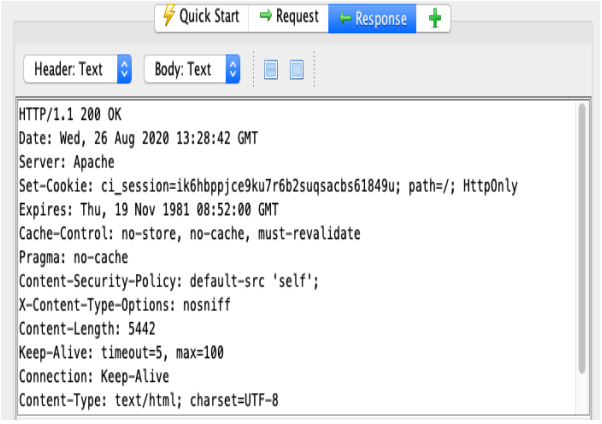
Path Traversal adalah kerentanan yang disebabkan oleh *permission* akses ke direktori yang tidak dimanage dengan baik serta parameter *input* yang tidak di validasi. Penyerang akan mengirimkan *payload* berupa *path* direktori pada parameter *input*. Serangan yang berhasil akan menyebabkan isi dari *file* pada *payload* yang dikirimkan akan tampil pada *browser*. Verifikasi hasil pemindaian dilakukan dengan cara memeriksa secara manual *response* dari aplikasi. Dari lima belas *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil semuanya *false positive* atau tidak memiliki kerentanan ini.

c. Kerentanan *SQL Injection*

SQL Injection adalah kerentanan yang disebabkan oleh parameter *input* yang tidak di validasi dan dieksekusi oleh *query DBMS*. Penyerang akan mengirimkan *payload* berupa *script SQL* pada parameter input. Serangan yang berhasil akan menyebabkan *script SQL* yang dikirimkan akan dieksekusi oleh *query DBMS*. Verifikasi hasil pemindaian dilakukan dengan menggunakan *tool* SQLmap versi 1.4.8.10. Dari empat puluh enam *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil 44 URL terkonfirmasi dapat dieksploit dan 2 URL *false positive* atau tidak memiliki kerentanan ini.

d. Kerentanan *X-Frame-Options Header Not Set*

X-Frame-Options Header Not Set adalah kerentanan yang disebabkan karena tidak disertakannya X-Frame-Options dalam *http response*. Kerentanan ini berada pada sisi *web server* yang dapat dimanfaatkan oleh penyerang untuk serangan *click jacking*. Verifikasi dilakukan dengan cara melihat *http response* dari *web server* SIPADUKO. Hasil verifikasi kerentanan ini ditemukan *http response* tidak menyertakan X-Frame-Options seperti Gambar 4.



Gambar 4. Http Response Web Server Aplikasi SIPADUKO

e. Kerentanan *Application Error Disclosure*

Application Error Disclosure adalah kerentanan yang disebabkan oleh tampilnya pesan *error* atau *warning* pada halaman *web*. Pesan tersebut dapat dimanfaatkan oleh penyerang sebagai sarana pengumpulan informasi untuk melakukan serangan. Verifikasi dilakukan dengan cara memeriksa *response* dari aplikasi secara manual. Dari tujuh *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil 5 *URL* terkonfirmasi memiliki kerentanan ini dan 2 *URL false positive* atau tidak memiliki kerentanan ini.

f. Kerentanan Cookie Without SameSite Attribute

Cookie Without SameSite Attribute adalah kerentanan yang disebabkan oleh tidak disertakannya same site attribute untuk cookie pada http response. Kerentanan ini berada pada sisi web server yang dapat dimanfaatkan oleh penyerang untuk melakukan serangan cross-site request forgery, cross-site script inclusion, dan timing attacks. Verifikasi dilakukan dengan cara memeriksa http response dari web server. Hasil verifikasi kerentanan ini ditemukan http response tidak menyertakan same site attribute seperti Gambar 4.

g. Kerentanan *Absence of Anti-CSRF Tokens*

Absence of Anti-CSRF Tokens adalah kerentanan yang disebabkan oleh tidak digunakannya *anti-CSRF tokens* pada *form*. Kerentanan ini dapat digunakan penyerang untuk melakukan serangan *cross-site request forgery* pada aplikasi. Verifikasi dilakukan dengan cara memeriksa *response* dari aplikasi secara manual. Dari dua puluh lima *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil 20 *URL* terkonfirmasi memiliki kerentanan ini dan 5 *URL false positive* atau tidak memiliki kerentanan ini.

h. Kerentanan *Information Disclosure - Debug Error Messages*

Information Disclosure - Debug Error Messages adalah kerentanan yang disebabkan oleh tampilnya pesan *debug error* pada *browser*. Kerentanan ini dapat dimanfaatkan oleh penyerang sebagai sarana pengumpulan informasi untuk melakukan serangan. Verifikasi dilakukan dengan cara memeriksa *response* dari aplikasi secara manual. *URL* yang terindikasi memiliki kerentanan ini adalah 1 *URL* dengan hasil verifikasi *false positive* atau tidak memiliki kerentanan ini.

i. Kerentanan *Information Disclosure - Suspicious Comments*

Information Disclosure - Suspicious Comments adalah kerentanan yang disebabkan oleh *comment out* pada *coding* yang dianggap mencurigakan atau mengandung data sensitif. Kerentanan ini dapat dimanfaatkan oleh penyerang sebagai sarana pengumpulan informasi untuk melakukan serangan. Verifikasi dilakukan dengan cara memeriksa *comment out* dari aplikasi secara manual. Dari empat puluh enam *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil semua *URL false positive* atau tidak memiliki kerentanan ini.

j. Kerentanan *Timestamp Disclosure - Unix*

Timestamp Disclosure - Unix adalah kerentanan yang disebabkan oleh tampilnya informasi *timestamp unix* pada *browser*. Kerentanan ini dapat dimanfaatkan oleh penyerang sebagai sarana pengumpulan informasi untuk melakukan serangan. Verifikasi dilakukan dengan cara memeriksa *timestamp* yang tampil mengandung informasi penting atau tidak secara manual. Dari delapan *URL* yang terindikasi memiliki kerentanan ini, didapatkan hasil semua *URL false positive* atau tidak memiliki kerentanan ini.

4. Kesimpulan

Pengujian keamanan pada aplikasi sitem informasi puskesmas terpadu Kota Payakumbuh dengan metode *grey box penetration test* menggunakan teknik audit berbantuan komputer *tool* OWASP ZAP, dapat menemukan 97 kerentanan kategori tinggi, 1 kerentanan kategori sedang dan 26 kerentanan kategori rendah. Hasil pengujian dapat dimanfaatkan oleh Dinas Komunikasi dan Informatika Kota Payakumbuh sebagai referensi untuk meningkatkan keamanan sistem informasi puskesmas terpadu Kota Payakumbuh.

Daftar Rujukan

[1] Nagendran, K., Adithyan, A., Chethana, R., Camillus, P., & Bala, S. V. K. B. (2019). Web Application Penetration Testing. *International Journal of Innovative Technology and Exploring Engineering (IJITEE), 8*(10), 1029-1035. DOI: https://dx.doi.org/10.35940/ijitee.J9173.0881019 .

[2] Nur, R. M., Na'am, J., Nurcahyo, G. W., & Arlis, S. (2019). Peningkatan Keamanan Website Menggunakan Metode XML dengan Framework Codeigniter. *Indonesian Journal of Computer Science, 8*(2), 156-163. DOI: https://dx.doi.org/10.33022/ijcs.v8i2.188 .

[3] Votipka, D., Stevens, R., Redmiles, E., Hu, J., & Mazurek, M. (2018). Hackers Vs. Testers: A Comparison of Software Vulnerability Discovery Processes. *IEEE Symposium on Security and Privacy (SP)*, 374-391. DOI: https://dx.doi.org/10.1109/SP.2018.00003 .

[4] Shebli, H. M. Z. A., & Beheshti, B. D. (2018). A Study on Penetration Testing Process And Tools. *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*. DOI: https://dx.doi.org/10.1109/lisat.2018.8378035 .

[5] Setiawan, E. B., & Setiyadi, A. (2018). Web Vulnerability Analysis and Implementation. *In IOP Conference Series: Materials Science and Engineering, 407*(1). DOI: https://dx.doi.org/10.1088/1757-899X/407/1/012081 .

[6] Simran, T. G., & Sasikala, D. (2019). Vulnerability Assessment of Web Applications using Penetration Testing. *International Journal of Recent Technology and Engineering (IJRTE), 8*(4), 1552-1556. DOI: https://dx.doi.org/10.35940/ijrte.B2133.118419

[7] 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: https://dx.doi.org/10.35760/ik.2019.v24i1.1988 .

[8] Jaber, R. J., & Wadi, R. M. A. (2018). Auditors' Usage of Computer-Assisted Audit Techniques (CAATs): Challenges and Opportunities. *In Conference on e-Business, e-Services and e-Society*, 365-375. DOI: https://dx.doi.org/10.1007/978-3-030-02131-3_33 .

[9] Asniarti, A., & Muda, I. (2019). The Effect of Computer Assisted Audit Tools on Operational Review of Information Technology Audits. *In 1st International Conference on Social Sciences and Interdisciplinary Studies*. DOI: https://dx.doi.org/10.2991/icssis-18.2019.5 .

[10] Wicaksono, A., Laurens, S., & Novianti, E. (2018). Impact Analysis of Computer Assisted Audit Techniques Utilization on Internal Auditor Performance. *In 2018 International Conference on Information Management and Technology (ICIMTech)*, 267-271. DOI: https://dx.doi.org/10.1109/ICIMTech.2018.8528198 .

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

[12] Clincy, V., & Shahriar, H. (2018). Web Application Firewall: Network Security Models and Configuration. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 1, 835-836. DOI: https://dx.doi.org/10.1109/COMPSAC.2018.00144 .