Journal of Information Technology System
https://journal.ummat.ac.id/index.php/jintens/index

Feb 2023
Vol. 1 No. 1

ANALISIS KERENTANAN WEB SERVER PADA APLIKASI ELEARNING (STUDI KASUS UNIVERSITAS MUHAMMADIYAH MATARAM)

[1]Muhammad Rizkillah, [2]Fitri Astutik
[1]Prodi Sistem Teknologi Informasi, Fakultas Teknik, Universitas Muhammadiyah Mataram, Jl. KH. Ahmad Dahlan No. 1 Pagesangan, Mataram, Indonesia 83127
[2]Prodi Pendidikan Teknologi Informasi, FSTT, Universitas Pendidikan Mandalika, Jl. Pemuda No. 59 A, Mataram, Indonesia 83125
ryzkillah@ummat.ac.id, fitriastutik@undikma.ac.id

ABSTRAK

UMMAT telah memiliki *web server* yang menjalankan beberapa aplikasi berbasis website yang bersifat publik. Salah satunya adalah sistem pembelajaran online (*elearning*) yang mulai diimplementasikan sejak tahun 2018 dan penggunaannya semakin meningkat secara signifikan sejak terjadinya pandemi. Walaupun *web server* yang dimiliki UMMAT belum pernah mengalami *cracking* yang mengakibatkan kerusakan tetapi tetap memerlukan pengujian keamanan informasi yang bertujuan untuk menguji tingkat kerentanan *web server*. Pengujian kerentanan *web server* dalam penelitian ini menggunakan *tools Zad Attack Proxy (ZAP)* sebagai perangkat lunak yang merupakan salah satu aplikasi yang dikembangkan oleh OWASP. ZAP merupakan aplikasi untuk menemukan kerentanan dalam suatu aplikasi web dengan cara menyediakan *scanner* otomatis. Kelebihan dari ZAP ini di antaranya bersifat mudah diinstal, *community based*, *open source*, *intercepting proxy*, *traditional & ajax spider*, *active scanner*, *growing add ons*, *forced browsing*, *fuzzer*, *dynamic*, *smart card support*, SSL cerificates, *integrared*, dan *web socket support.* Aplikasi elearning UMMAT memiliki 13 jenis kerentanan, yaitu *SQL Injection, .htaccess Information Leak, Directory Browsing, Vulnerable JS Library, Absence of Anti-CSRF Tokens, Cookie No HttpOnly Flag, Cookie Without SameSite Attribute, Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s), X-Content-Type-Options Header Missing, Content-Type Header Missing, Information Disclosure - Suspicious Comments, Loosely Scoped Cookie, Timestamp Disclosure – Unix*. Level resiko yang ada pada kerentanan aplikasi elearning UMMAT terdiri dari 1 kasus dalam resiko tinggi, 3 kasus dalam resiko menengah, 5 kasus dalam resiko rendah dan 4 kasus yang bersifat informasi.

ABSTRACT

UMMAT already has a web server that runs several public website-based applications. One of them is the online learning system (e-learning) which has been implemented since 2018 and its use has increased significantly since the pandemic. Although the web server owned by UMMAT has never experienced cracking which causes damage, it still requires information security testing which aims to test the level of web server vulnerability. The web server vulnerability testing in this study uses the Zad Attack Proxy (ZAP) tool as software which is one of the applications developed by OWASP. ZAP is an application to find vulnerabilities in a web application by providing an automatic scanner. The advantages of this ZAP include being easy to install, community based, open source, intercepting proxy, traditional & ajax spider, active scanner, growing add ons, forced browsing, fuzzer, dynamic, smart card support, SSL cerificates, integrared, and web socket. support. The UMMAT e-learning application has 13 types of vulnerabilities, namely SQL Injection, .htaccess Information Leak, Directory Browsing, Vulnerable JS Library, Absence of Anti-CSRF Tokens, Cookie No HttpOnly Flag, Cookie Without SameSite Attribute, Server Leaks Information via "X-Powered- By "HTTP Response Header Field (s), X-Content-Type-Options Header Missing, Content-Type Header Missing, Information Disclosure - Suspicious Comments, Loosely Scoped Cookie, Timestamp

Disclosure - Unix. The level of risk in the UMMAT e-learning application vulnerability consists of 1 case in high risk, 3 cases in medium risk, 5 cases in low risk and 4 cases of information in nature.

A. LATAR BELAKANG

Di masa pandemi COVID 19 saat ini, *internet* memegang peranan sangat penting dalam segala bentuk proses bisnis yang ada di dunia. Hal ini juga terjadi pada dunia pendidikan bahkan di beberapa negara, sekolah-sekolah dan perguruan tinggi sempat tutup dikarenakan terjadinya pandemi berdampak pada tuntutan perguruan tinggi di Indonesia berusaha melakukan peningkatan dalam layanan TIK untuk mendukung proses pendidikan, termasuk Universitas Muhammadiyah Mataram (UMMAT).

Meningkatnya penggunaan internet selama pandemi juga berdampak pada meningkatnya kejahatan siber di Indonesia. Menurut Badan Siber dan Sandi Negara (BSSN) sepanjang bulan Januari hingga Agustus 2020 terdapat hampir 190 juta upaya serangan siber yang terjadi di Indonesia.

Gambar 1. Data Jumlah Serangan Siber Januari - Agustus 2019/2020(Pusat Operasi Keamanan Siber Nasional)

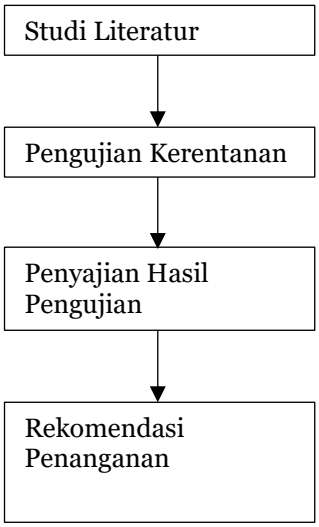
UMMAT telah memiliki *web server* yang menjalankan beberapa aplikasi berbasis website yang bersifat publik. Salah satunya adalah sistem pembelajaran online (*elearning*) yang mulai diimplementasikan sejak tahun 2018 dan penggunaannya semakin meningkat secara signifikan sejak terjadinya pandemi. Walaupun *web server* yang dimiliki UMMAT belum pernah mengalami *cracking* yang mengakibatkan kerusakan tetapi tetap memerlukan pengujian keamanan informasi yang bertujuan untuk menguji tingkat kerentanan *web server*.

Dalam penelitian ini akan dilakukan pengujian *web server elearning* yang dimiliki UMMAT dengan menggunakan pengujian kerentanan dengan menggunakan metode *Open Web Application Security Project* (OWASP).

Pengujian kerentanan *web server* dalam penelitian ini menggunakan *tools Zad Attack Proxy (ZAP)* sebagai perangkat lunak yang merupakan salah satu aplikasi yang dikembangkan oleh OWASP. ZAP merupakan aplikasi untuk menemukan kerentanan dalam suatu aplikasi web dengan cara menyediakan *scanner* otomatis. Kelebihan dari ZAP ini di antaranya bersifat mudah diinstal, *community based*, *open source*, *intercepting proxy*, *traditional & ajax spider*, *active scanner*, *growing add ons*, *forced browsing*, *fuzzer*, *dynamic*, *smart card support*, SSL cerificates, *integrared*, dan *web socket support.*

B. METODE PENELITIAN

Adapun tahap dalam penelitian ini terbagi atas 4 (empat) tahap, yaitu studi literatur, pengujian kerentanan, penyajian hasil pengujian, rekomendasi penanganan. Skema penelitian dapat dilihat pada Gambar 2.



Gambar 2. Metode Penelitian

1. Studi Literatur

Pada tahapan ini akan dilakukan studi literatur. Studi literatur bertujuan untuk menjelaskan kajian pustaka berdasarkan teori-teori penunjang yang digunakan sebagai bahan penelitian. Adapun studi literatur ini didapatkan dari membaca buku, jurnal, artikel maupun dari internet.

2. Pengujian Kerentanan

Pada tahapan ini dilakukan pengujian kerentanan dengan cara melakukan *scanning* pada target. Adapun alat kebutuhan yang digunakan untuk pengujian kerentanan adalah:

a. Analisa kebutuhan sistem

Adapun analisis kebutuhan sistem meliputi:

1. Perangkat Keras

Untuk pengujian kerentanan dibutuhkan perangkat keras, berikut perangkat keras yang digunakan:

- Laptop Macbook Pro 2,8 GHz Intel Core i7
- Ram 8 GB
- Hardisk 128 SSD

2. Perangkat Lunak

Untuk pengujian kerentanan dibutuhkan perangkat lunak, berikut perangkat lunak yang digunakan:

- macOS High Sierra
- OWASP ZAP

3. Penyajian Hasil Pengujian

Pada tahapan ini akan dilakukan penyajian hasil dari pengujian dengan membahas apa saja kerentanan yang ada pada elearning UMMAT.

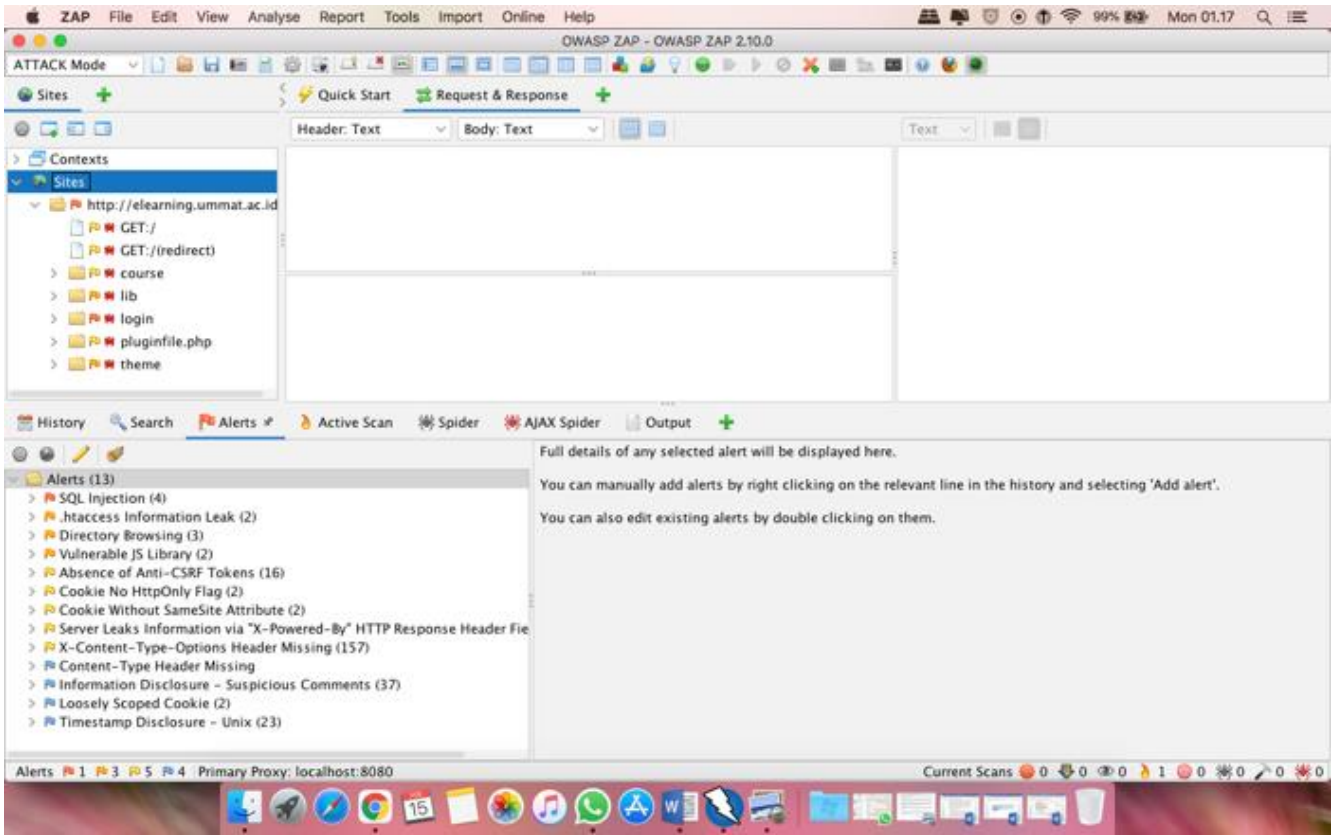
4. Rekomendasi Penanganan

Pada tahapan ini akan dibahas mengenai rekomendasi untuk menangani kerentanan yang ada pada elearning UMMAT.

C. HASIL DAN PEMBAHASAN

1. Pengujian Kerentanan

Proses pengujian kerentanan menggunakan OWASP ZAP versi 2.10.0 yang membutuhkan waktu scanning hampir 120 menit. Setelah proses scanning selesai maka akan didapatkan laporan kerentanan pada elearning,ummat.ac.id.



Gambar 3. Hasil *scanning* pengujian

Dari hasil pengujian terdapat 13 jenis kerentanan yang terbagi atas 4 tingkatan resiko kemungkinan dan dampak, yaitu 1 kasus dengan tingkat resiko tinggi, 3 kasus dengan tingkat resiko menengah, 5 kasus dengan resiko rendah dan 4 kasus hanya bersifat informasi.

2. Penyajian Hasil Pengujian

Kerentanan Elearning UMMAT			
No	**Jenis**	**Risk**	**Confidence**
1	SQL Injection	High	Medium
2	.htaccess Information Leak	Medium	Medium
3	Directory Browsing	Medium	Medium
4	Vulnerable JS Library	Medium	Medium
5	Absence of Anti-CSRF Tokens	Low	Medium
6	Cookie No HTTPOnly Flag	Low	Medium
7	Cookie Without SameSite Attribute	Low	Medium
8	Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	Medium

9	X-Content-Type-Options Header Missing	Low	Medium
10	Content-Type Header Missing	Informational	Medium
11	Information Disclosure - Suspicious Comments	Informational	Low
12	Loosely Scoped Cookie	Informational	Low
13	Timestamp Disclosure - Unix	Informational	Low

Tabel 1. Hasil Pengujian

Identifikasi untuk menentukan tingkat resiko pada penelitian ini menggunakan metode pemodelan ancaman (*threat modeling*). Metode ini dapat digunakan untuk memperkirakan tingkat keparahan semua risiko terhadap aplikasi web dan membuat keputusan berdasarkan informasi tentang apa yang harus dilakukan terhadap risiko tersebut. Risiko diukur dengan skala 0 hingga 9 dan dibagi menjadi tiga bagian seperti pada Tabel 2.

Tingkat Kemungkinan dan Dampak	
0 sampai <3	Low (Rendah)
3 sampai <6	Medium (Menengah)
6 sampai 9	High (Tinggi)

Tabel 2. Level Resiko

3. Rekomendasi Penanganan

Berikut merupakan penjelasan dari kerentanan yang ada pada *elearning* UMMAT dengan beberapa skenario yang dapat dilakukan oleh peretas dan penanganan dalam mengatasi kerentanan :

1. SQL Injection.
   Merupakan salah satu dari 10 resiko keamanan aplikasi pada OWASP. Dengan menggunakan SQL Injection, memungkinkan peretas dapat menganalisa struktur basis data dalam aplikasi sehingga mampu mengambil alih aplikasi dengan sangat mudah. Untuk solusianya sebaiknya melakukan beberapa kebijakan antara lain :
   a. Batasi panjang *input box*, dengan cara membatasinya di kode program.
   b. Filter *input* yang dimasukkan oleh *user*, terutama penggunaan tanda kutip tunggal (*Input Validation*).
   c. Matikan atau sembunyikan pesan-pesan *error* yang keluar dari SQL Server yang berjalan.
   d. Matikan fasilitas-fasilitas standar seperti *Stored Procedures, Extended Stored Procedures* jika memungkinkan.
   e. Berikan akses basis data minimum yang diperlukan untuk aplikasi.
2. .htaccess Information Leak
   file htaccess dapat digunakan untuk mengubah konfigurasi perangkat lunak server Apache untuk mengaktifkan/menonaktifkan fungsionalitas tambahan dan fitur yang ditawarkan oleh perangkat lunak server Apache. Bila peretas mampu mengakses file .htaccess, maka peretas mampu untuk mengubah konfigurasi server Apache. Untuk solusinya dengan memastikan bahwa file .htaccess tidak dapat diakses secara publik.
3. Directory Browsing
   Directory Browsing yang tidak dinonaktifkan memungkinkan peretas untuk melihat daftar direktori pada aplikasi, daftar direktori biasanya dapat mengungkap skrip tersembunyi termasuk *file* yang dapat diakses untuk membaca informasi penting. Untuk solusinya harus menonaktifkan Directory Browsing dengan cara menambahkan kode options-indexes pada file .htaccess.
4. Vulnerable JS Library

Library jquery masih menggunakan versi lama sehingga memiliki beberapa kerentanan sehingga perlu dilakukan *upgrade* ke versi terbaru.

5. Absence of Anti-CSRF Tokens
   Cross-Site Request Forgery (CSRF) juga dikenal sebagai Session Riding atau One-Click Attack. Serangan ini adalah jenis serangan eksploitasi berbahaya terhadap pengguna aplikasi web. Serangan ini telah terdaftar sebagai yang ke-7 yang paling dapat dieksploitasi di antara 10 Serangan Web teratas. CSRF adalah serangan yang memungkinkan penyerang melakukan permintaan HTTP sembarang POST/GET yang tidak sah atas nama korban yang saat ini diautentikasi ke situs web. Ada banyak jenis serangan web yang yang dilakukan dengan menggunakan teknik CSRF ini, mulai dari yang tidak begitu berbahaya sampai yang berbahaya. Untuk mencegah terjadinya serangan CSRF, maka perlu menggunakan tokens Anti-CSRF.
6. Cookie No HttpOnly Flag
   Cookie telah di setting tanpa HttpOnly Flag, yang memungkinkan dapat diakses oleh JavaScript. Jika peretas mampu memasukkan kode serangan, mak cookie akan dapat diakses dan dikirim ke situs lain. Untuk pencegahannya sebaiknya dengan memastikan mengatur HTTPOnly Flag digunakan pada semua cookie.
7. Cookie Without SameSite Attribute
   SameSite Attribute adalah tindakan pencegahan yang efektif untuk pemalsuan *request cross-site*, *cross-site script inclusion* dan *timing attacks*. Untuk pencegahan perlu men-*setting* SameSite Attribute ke *lax* atau sebaiknya *tight* untuk semua cookie.
8. Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)
   Web Server memberikan informasi melalui "X-Powered-By" HTTP Response Header Field(s) sehingga memungkinkan peretas mampu untuk mengidentifikasi *framework/component* aplikasi dan kemungkinan *component* memiliki kerentanan. Untuk solusinya dengan memastikan web server menyembunyikan "X-Powered-By" HTTP Response Header.
9. X-Content-Type-Options Header Missing
   X-Content-Type-Options Header tidak distel ke ‘nosniff yaitu penanda yang digunakan oleh server untuk menunjukan tipe MIME yang diiklankan dalam *header* tidak boleh diubah atau diikuti tidak ada, sehingga dapat menyebabkan tidak adanya perlindungan Cross-Origin Read Blocking (CORB) untuk file HTML, TXT, JSON dan XML (tidak termasuk gambar SVG / svg + xml). Untuk pencegahan dengan memastikan aplikasi menetapkan X-Content-Type-Options Header dengan tepat. X-Content-Type-Options Header diatur ke ‘nosniff’ dan diatur untuk semua halaman web.
10. Content-Type Header Missing
    Tidak ditemukan Content-Type Header sehingga perlu dilakukan setting Content-Type Header disetiap halaman web.
11. Information Disclosure - Suspicious Comments
    Kolom komentar dapat saja berisi komentar mencurigakan yang dikirimkan oleh peretas. Untuk pencegahan dengan menghapus semua komentar yang mencurigakan.
12. Loosely Scoped Cookie
    Lingkup domain yang diterapkan pada cookie menentukan domain yang dapat mengaksesnya, maka perlu selalu mengarahkan lingkup cookie ke FQDN (Fully Qualified Domain Name).
13. Timestamp Disclosure – Unix
    Informasi tanggal dan waktu web server terlihat secara publik, maka perlu memastikan bahwa informasi timestamp tidak mempengaruhi aplikasi bila terjadi serangan.

D. KESIMPULAN DAN SARAN

1. Kesimpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah :

a. Aplikasi elearning UMMAT memiliki 13 jenis kerentanan, yaitu *SQL Injection, .htaccess Information Leak, Directory Browsing, Vulnerable JS Library, Absence of Anti-CSRF Tokens, Cookie No HttpOnly Flag, Cookie Without SameSite Attribute, Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s), X-Content-Type-Options Header Missing, Content-Type Header Missing, Information Disclosure - Suspicious Comments, Loosely Scoped Cookie, Timestamp Disclosure – Unix.*
b. Level resiko yang ada pada kerentanan aplikasi elearning UMMAT terdiri dari 1 kasus dalam resiko tinggi, 3 kasus dalam resiko menengah, 5 kasus dalam resiko rendah dan 4 kasus yang bersifat informasi.
c. Terdapat beberapa kerentanan yang ada pada aplikasi elearning UMMAT yang termasuk dalam 10 isu resiko keamanan aplikasi, yaitu Injection, CSRF dan Cross Site Scripting.
d. Kerentanan yang beresiko tinggi adalah SQL Injection dan merupakan kerentanan yang paling banyak dimanfaatkan peretas untuk melakukan serangan pada aplikasi.

2. Saran

a. Perlu dilakukan pengujian dengan menggunakan metode lain, misalnya Information System Security Assessment Framework (ISSAF) atau kombinasi dengan metode OWASP.
b. Perlu dilakukan pengujian yang lebih mendalam untuk mendapatkan hasil pengujian yang lebih detil.
c. Pengujian sebaiknya dilakukan secara berkala untuk memperkecil kemungkinan terjadinya serangan yang akan mengakibatkan kerusakan yang parah.

DAFTAR RUJUKAN

[1]. Guntoro, Loneli Costaner, & Musfawati. 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)

[2]. G. Guntoro & M. Fikri, 2018, “Perancangan Aplikasi Single Sign-On Menggunakan Autentikasi Gambar,” Digit. Zo. J. Teknol. Inf. dan Komun., vol. 9, no. 1, pp. 12–21, 2018.

[3]. Imam Riadi, Rusydi Umar & Tri Lestari, 2020, “Analisis Kerentanan Serangan *Cross Site Scripting* (XSS) pada Aplikasi Smart Payment Menggunakan Framework OWASP”, JISKA (Jurnal Informatika Sunan Kalijaga), Universitas Ahmad Dahlan

[4]. Sunardi, 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), 135–143. https://doi.org/10.14569/IJACSA.2019.0101118

[5]. Syarifudin, I, 2018, “*Pentesting dan Analisis Keamanan Web Paud Dikmas”*, *April*.

[6]. Putri ZS, 2020, “Kejahatan Siber Naik 4 Kali Lipat Selama Pandemi”, https://tekno.kompas.com/read/2020/10/12/07020007/kejahatan-siber-di-indonesia-naik-4-kali-lipat-selama-pandemi.