Implementasi dan Analisis Web Application Firewall Menggunakan Proxy Untuk Mengidentifikasi Serangan Siber Pada Web Berstandar OWASP

1st Rama Wijaya Shiddiq
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
ramawijayas.techdr7@gmail.com

2nd Nyoman Bogi
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
aditya@telkomuniversity.ac.id

3rd Sofia Naning
Fakultas Teknik Elektro
Universitas Telkom
Bandung, Indonesia
sofiananing@telkomuniversity.ac.id

***Abstrak* — Banyak pengembang aplikasi web yang kurang memperhatikan keamanan aplikasi web, sehingga website mereka sering dieksploitasi oleh para hacker. Hasil Web Security Report SiteLock pada tahun 2022 menunjukkan tingginya frekuensi serangan dan akses bot ke website. Serangan pada tahun 2022 meningkat drastis dibandingkan dengan tahun 2020. Dua jenis kerentanan utama yang ditemukan adalah Cross Site Scripting (XSS) dan SQL Injection. Untuk mengatasi masalah ini, telah dirancang sebuah Web Application Firewall (WAF) berbasis proxy. Untuk mendeteksi dan memblokir alamat IP berbahaya, serta mencatat setiap alamat IP yang mencoba melakukan serangan. Evaluasi serangan terhadap XSS, SQL Injection, dan LFI menunjukkan efektivitas plugin ini dengan tingkat perlindungan masing-masing 100%, 97%, dan 74%, berdasarkan standar dari OWASP cheat sheet.**

***Kata kunci*— WAF, OWASP, Kerentanan**

I. PENDAHULUAN

Pelaku kejahatan siber biasanya menyerang suatu sistem keamanan informasi yang memuat data-data penting dan rahasia. Biasanya penyerangan yang sering terjadi dilakukan terhadap aplikasi web. Banyak pengembang aplikasi web yang kurang memperhatikan sisi keamanan aplikasi web sehingga banyak dieksploitasi oleh para hacker. Menurut hasil Web Security Report SiteLock pada tahun 2022, didapatkan 172 serangan dalam sehari untuk satu website dan website diakses oleh bot sebanyak 2306 kali per minggu. Bot digunakan oleh hacker untuk mencari kelemahan situs web. Jumlah serangan di tahun 2022 meningkat sebanyak 210% dibandingkan tahun 2020. Terdapat 2 jenis kerentanan tertinggi yang tercatat yaitu Cross Site Scripting (XSS) sebanyak 1 juta halaman website, dan SQL Injection sebanyak 332 ribu halaman website. Untuk mengatasi berbagai permasalahan terkait keamanan aplikasi web diperlukan suatu sistem yang dapat mencegah serangan berbahaya. WAF atau Web Application Firewall adalah perangkat keamanan yang digunakan untuk melindungi aplikasi web dari serangan jaringan yang tidak sah. WAF menganalisis lalu lintas jaringan yang masuk ke aplikasi web dan mengeliminasi lalu lintas yang tidak diinginkan atau merusak sebelum lalu lintas tersebut sampai ke aplikasi. WAF dapat digunakan sebagai sistem yang dapat mencegah serta mendeteksi serangan SQL injection, Cross Site Scripting (XSS), dan Local File Inclusion (LFI) dengan menggunakan detection rules yang telah ditetapkan untuk dapat memblokir akses bagi penyerang ke dalam website [2].

OWASP atau Open Web Application Security Project adalah organisasi nirlaba yang bertujuan untuk membantu para pengembang perangkat lunak dalam membangun aplikasi web yang aman. OWASP menyediakan berbagai sumber daya keamanan web yang berguna, termasuk dokumentasi, tools, dan proyek open source yang dapat digunakan oleh para pengembang untuk membangun aplikasi web yang aman. Salah satu bentuk penilaian tingkat resiko kerentanan keamanan aplikasi berbasis website adalah OWASP Risk Rating Methodology. Langkah besar dalam mengukur tingkat resiko adalah menentukan dampak buruk yang dihasilkan dari analisa kerentanan [4].

II. KAJIAN TEORI

A. Web Application Firewall

Penggunaan Web Application Firewall berfungsi untuk menyaring permintaan data yang masuk ke aplikasi web dan melakukan penolakan atau pemblokiran terhadap data yang mencurigakan atau berbahaya. Web Application Firewall mengirimkan data kembali ke aplikasi web apabila data yang dikirimkan sudah sesuai dengan aturan firewall [6].

B. Proxy

Proxy meerupakan sebuah server yang mempunyai peran untuk meneruskan permintaan dari user ke server lainnya yang berada di internet. Sehingga dengan adanya proxy komputer dapat terhubung dengan komputer lainnya melalui internet. Umumnya proxy server digunakan sebagai pengaman jaringan komputer pribadi yang terhubung ke jaringan publik, sehingga dapat dilakukan monitoring terhadap paket yang keluar dan masuk [4].

C. OWASP

Menurut ("Who is the OWASP Foundation?") OWASP adalah singkatan dari Open Web Application Security Project yang merupakan sebuah organisasi yang berfokus pada keamanan pada software. Seluruh tools, dokumen, maupun forum terbuka bagi semua pihak yang tertarik untuk memperbaiki keamanan pada aplikasi. OWASP menyediakan beberapa dokumen untuk membantu para developer untuk membuat software dan website yang aman.

III. METODE

A. Studi Literatur

Tahap awal adalah tahap sebelum penelitian dilakukan dengan mengidentifikasi masalah beberapa masalah yang timbul. Permasalahan yang timbul dari fakta-fakta yang ada adalah seiring dengan meningkatnya pengguna internet di Indonesia, maka semakin meningkat pula serangan siber yang ditujukan serangannya kepada website yang dimiliki oleh perusahaan/organisasi yang berada di Indonesia. Banyak website digunakan oleh perusahaan/organisasi untuk melakukan transaksi maupun pelayanan untuk para klien. Serangan yang terjadi dapat menyebabkan bocornya informasi data diri dari klien maupun perusahaan/organisasi yang tersimpan didalam server.

B. Tahap perancangan sistem

Pada tahap ini, membuat rancangan sesuai dengan solusi yang ditemukan pada tahap sebelumnya untuk menjawab dari permasalahan yang timbul. Permasalahan yang timbul dari fakta-fakta yang ditemukan adalah peningkatan pengguna internet membuat semakin banyaknya serangan siber yang terjadi di Indonesia terutama serangan yang menuju website. Oleh karena itu, solusi yang dapat diimplementasikan oleh peneliti adalah membuat *web application firewall* berdasarkan rule proxy. Sehingga dapat mencegah serangan yang akan dilakukan ke server dan akan melakukan pencatatan IP address yang melakukan serangan serta memberikan PoC secara otomatis ke database jika terdapat serangan yang masuk.

C. Tahap Pengujian Sistem dan Analisa

Pada Pada tahap ini, terdapat proses eksploitasi dan akan memiliki dua kondisi untuk melihat efektifitas penggunaan *web application firewall*. Kondisi pertama adalah aplikasi web yang rentan akan diserang tanpa menggunakan *web application firewall* proxy. Kondisi kedua adalah aplikasi web yang rentan akan diserang dengan menggunakan *Web Application Firewall* proxy. Dari proses ini akan diperoleh hasil eksperimen yang akan diolah pada tahap analisis. Dengan adanya pengujian maka dapat mengetahui seberapa jauh sistem yang dibuat sudah memenuhi kebutuhan.

D. Tahap Analisis

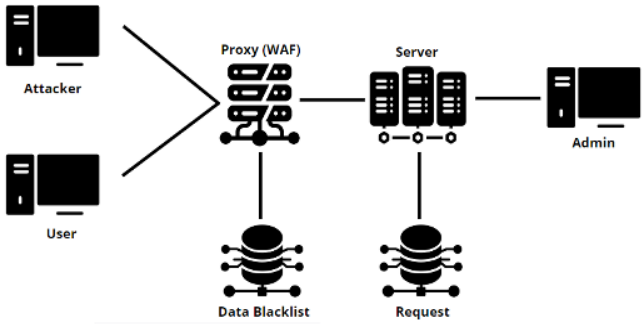
Pada tahap ini, proses analisis dilakukan terhadap hasil percobaan yang telah dilakukan pada tahap pengujian. Analisis kerentanan menggunakan standar kerentanan CVE. Analisis dilakukan untuk mengetahui efektivitas kinerja rule proxy *web application firewall* dalam melindungi aplikasi berbasis web dengan melakukan analisis kuantitatif berdasarkan *vulnerability* dan *threat* dengan hasil akhir dari analisis kuantitatif berupa perhitungan efektivitas *web application firewall* serta secara kualitatif berdasarkan *vulnerability* dan *threat* dengan hasil akhir mengetahui tingkat efektivitas *web application firewall* berdasarkan kerentanannya.

E. Tahap Kesimpulan

Pada Hasil akhir dari penelitian berupa laporan penelitian, dengan langkah terakhir penarikan kesimpulan yang diperoleh dari hasil analisis penelitian yang dilakukan dengan menghubungkan parameter *web application firewall* menggunakan proxy dengan kemampuan eksploit.

IV. HASIL DAN PEMBAHASAN

Pada bagian ini akan dijelaskan tentang implementasi dari WAF *(web application firewall)* yang mana sistem terdiri dari *attacker, user, proxy, data blacklist, server, request* dan admin. Berikut merupakan desain keamanan mengenai sistem yang akan dibuat pada gambar 4.1.



GAMBAR 4.1
(Blok Diagram WAF)

Pada Gambar 4.1 menjelaskan bahwa setiap permintaan masuk ke aplikasi *web* akan diteruskan melalui WAF sebagai *proxy*. Dengan adanya WAF berbasis *proxy*, *request* yang mengandung anomali akan dicek menggunakan *proxy*. Didalam *proxy*, IP *address* yang melakukan *request* akan dicatat dan disimpan. Apabila IP *address* tersebut melakukan serangan maka akan diblokir dan IP *address* tersebut akan dicatat. Sehingga jika IP yang sudah tercatat mencoba melakukan serangan kembali maka akan langsung terblokir. Setelah itu akan dilakukan *generate* secara otomatis ke database dari serangan yang masuk. Dalam pengidentifikasian penelitian mengambil sample SQL Injection, Cross-site Scripting, dan Local File Inclusion. Sample diambil karena mengacu pada OWASP Top 10 dimana serangan yang sering terjadi ada pada nomor 1 yaitu Injection yang dimana kita mengambil SQL Injection, nomor 7 yaitu Cross-Site Scripting, dan Local File Inculsion. SQL Injection sering di gunakan untuk melakukan pencurian sebuah data pada sebuah perusahaan, Cross Site Scripting biasanya di gunakan untuk melakukan Phising atau mencoba melakukan eksekusi command pada server, dan Local File Inclusion mengeksekusi file local yang sensitif.

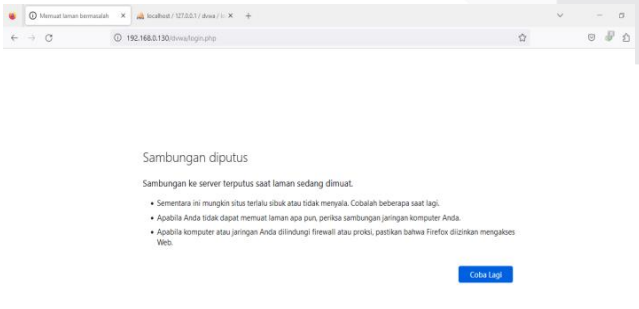
Pada bagian ini akan dijelaskan tentang langkah untuk melakukan pengujian WAF pada DVWA. Pertama jalankan *script* WAF menggunakan *vscode*, setelah *script WAF*

berhasil dijalankan buka aplikasi *web* DVWA pada *browser*. Atur DVWA *security* pada *level low* agar dapat dieksploitasi. Setelah mengatur *level* keamanan DVWA, proses pengujian bisa dilakukan dengan memilih jenis kerentanan yang tersedia. Pengujian dilakukan dengan memasukkan banyak *payload* ke *field* yang disediakan. Jika koneksi ke DVWA terputus maka artinya WAF berhasil mendeteksi *payload* sebagai *script* yang berbahaya dan melakukan pemblokiran pada IP yang mencoba menyerang. IP yang terblokir dapat dilihat pada *database* DVWA. *Payload* serangan akan dicoba satu per satu untuk menilai seberapa baik efektivitas WAF.

SQL Injection adalah salah satu ancaman paling umum yang menyerang sistem *database*. *SQL Injection* dapat terjadi karena kurangnya validasi input dalam aplikasi berbasis web yang menyebabkan penyerang menyuntikkan input string untuk mengubah atau memanipulasi SQL dengan sukses [9]. Dalam penelitian ini menggunakan *payload* serangan yang diperoleh OWASP *cheat sheet* untuk mengeksploitasi kerentanan *SQL Injection.* Pengujian serangan *SQL Injection* pada aplikasi web DVWA dengan mematikan dan menghidupkan *web application firewall* sepertipada gambar 4.2 dan 4.3 dibawah ini.

GAMBAR 4.2
(Tampilan serangan *SQL Injection* tanpa WAF)

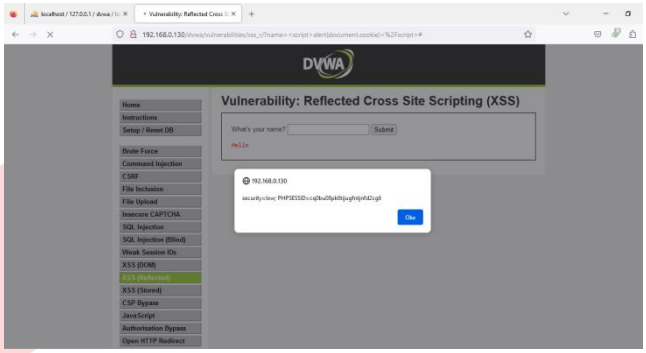
Pada bagian gambar 4.2 telah dilakukan pengujian serangan SQL *Injection* pada web DVWA dengan memasukan query 1. Dapat dilihat setelah dimasukan angka 1 attacker bisa mendapatkan informasi mengenai ID, username, dan surname dari website tersebut. Informasi database yang didapatkan tergantung dari jenis query yang kita inputkan pada kolom sql injection tersebut.



GAMBAR 4.3
(Tampilan pemblokiran *SQL Injection* oleh WAF)

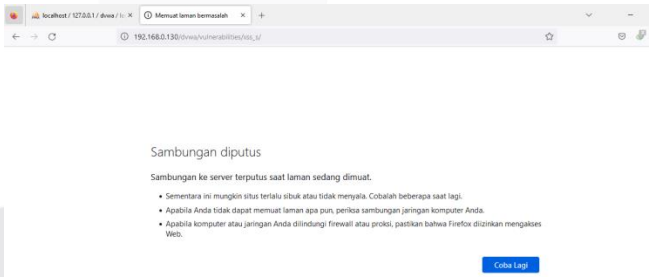
Pada bagian gambar 4.3 merupakan tampilan blokir dari *web application firewall.* Secara otomatis, proxy akan melakukan pemblokiran terhadap request tersebut, dan akan melakukan pencatatan ke database dan menyimpan IP address tersebut sehingga tidak bisa mengakses kembali.

Pada XSS (Cross-Site Scripting) adalah serangan pada aplikasi web yang memungkinkan penyerang menyisipkan skrip berbahaya (script) ke dalam halaman web yang dilihat oleh pengguna lain. XSS dapat berdampak pada web yaitu pencurian informasi pengguna, penggantian konten halaman web, pengalihan pengguna ke situs phishing, atau menjalankan skrip berbahaya di browser pengguna.



GAMBAR 4.4
(Tampilan serangan *XSS* tanpa WAF)

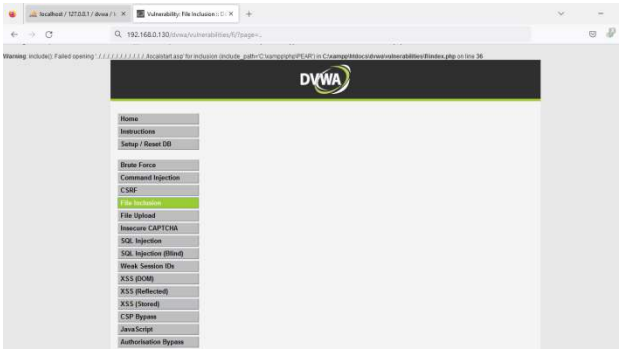
Pada bagian gambar 4.4 telah dilakukan pengujian serangan Cross-site Scripting pada web DVWA dengan memasukan serangan dari XSS payload. Dapat dilihat setelah dimasukan XSS bisa mendapatkan informasi cookie dari website tersebut. Dengan mendapatkan cookie, attacker bisa mendapatkan akses yang tidak sah, penggunaan identitas baru, pencurian data pribadi, bahkan bisa *take over account* dari website tersebut.



GAMBAR 4.5
(Tampilan pemblokiran *XSS* oleh WAF)

Pada bagian gambar 4.5 merupakan tampilan blokir dari *web application firewall.* Secara otomatis, proxy akan melakukan pemblokiran terhadap request tersebut, dan akan melakukan pencatatan ke database dan menyimpan IP address tersebut sehingga tidak bisa mengakses kembali.

Local File Inclusion (LFI) adalah kerentanan keamanan pada aplikasi web yang memungkinkan penyerang menyertakan atau mengakses file lokal yang ada pada server yang rentan. Serangan ini terjadi ketika aplikasi web tidak memvalidasi dengan benar input pengguna yang digunakan untuk menyertakan file lokal. Hal ini memungkinkan penyerang untuk menyertakan file yang tidak seharusnya diakses oleh pengguna biasa, termasuk file konfigurasi, file log, atau bahkan file sistem yang penting.



GAMBAR 4.6
(Tampilan serangan *LFI* tanpa WAF)

Pada bagian gambar 4.6 telah dilakukan pengujian serangan Local File Inclusion pada web DVWA dengan memasukan perintah atau file lokal yang tidak aman dalam konten halaman web. Dapat dilihat setelah dimasukan perintah LFI, attacker bisa mendapatkan informasi sensitif dari file direktori yang terdapat dalam website tersebut.

GAMBAR 4.7
(Tampilan pemblokiran *LFI* oleh WAF)

Pada bagian gambar 4.7 merupakan tampilan blokir dari *web application firewall*. Secara otomatis, proxy akan melakukan pemblokiran terhadap request tersebut, dan akan melakukan pencatatan ke database dan menyimpan IP address tersebut sehingga tidak bisa mengakses kembali.

A. Analisis Hasil Pengujian

Hasil pengujian *web application firewall* menggunakan daftar 100 payload yang digunakan untuk melakukan serangan. Hasil evaluasi serangan SQL Injection, XSS dan LFI dapat dihitung berdasarkan tingkat efektivitas sistem dalam menangkal *payload* dengan perhitungan sebagai berikut:

$$Efektivitas = \frac{Total\ Refused\ SQL\ Injection\ Payload}{Total\ SQL\ Injection\ Payload} \times 100\%$$

$$Efektivitas = \frac{97}{100} \times 100\% = 97\%$$

Pada hasil pengujian tingkat efektivitas sistem dalam menangkal *payload* SQL *Injection* didapatkan hasil 97 *payload* berhasil terdeteksi sebagai serangan SQL *Injection* sehingga sistem memblokir IP yang melakukan percobaan *login* dengan *payload* SQL *Injection*. Terdapat 3 *payload* SQL *Injection* yang tidak berhasil terdeteksi yaitu *payload* OR 1=1, ORDER BY 1--, ORDER BY 1. Hasil evaluasi serangan XSS dapat dihitung berdasarkan tingkat efektivitas sistem dalam menangkal payload dengan perhitungan sebagai berikut:

$$Efektivitas = \frac{Total\ Refused\ XSS\ Payload}{Total\ XSS\ Payload} \times 100\%$$

$$Efektivitas = \frac{100}{100} \times 100\% = 100\%$$

Dari hasil pengujian tingkat efektivitas sistem dalam menangkal *payload* XSS didapatkan hasil 100 *payload* berhasil terdeteksi sebagai serangan XSS sehingga sistem memblokir IP yang melakukan percobaan *login* dengan *payload* XSS. Hasil evaluasi serangan LFI dapat dihitung berdasarkan tingkat efektivitas sistem dalam menangkal payload dengan perhitungan sebagai berikut:

$$Efektivitas = \frac{Total\ Refused\ LFI\ Payload}{Total\ LFI\ Payload} \times 100\%$$

$$Efektivitas = \frac{74}{100} \times 100\% = 74\%$$

Dari hasil pengujian tingkat efektivitas sistem dalam menangkal payload LFI didapatkan hasil 74 payload berhasil terdeteksi sebagai serangan LFI sehingga sistem memblokir IP yang melakukan percobaan untuk mengakses file lokal dengan payload LFI. Terdapat 26 payload LFI yang tidak berhasil terdeteksi. Dari perhitungan evaluasi serangan XSS, SQL Injection, dan LFI sistem ini mampu melakukan pengecekan dengan tingkat efektivitas pada XSS sebesar 100%, pada SQL Injection sebesar 97%, dan pada LFI sebesar 74% berdasarkan standar yang dimiliki oleh OWASP pada OWASP cheat sheet

V. KESIMPULAN

Web Application Firewall menggunakan rule proxy mampu melindungi aplikasi web dari tiga serangan yaitu serangan SQL Injection, Cross-site Scripting dan Local File Inclusion. Tingkat efektivitas Web Application Firewall proxy dalam melindungi aplikasi web yang rentan mampu blokir serangan SQL Injection sebesar 97%, Cross-site Scripting 100% dan Local File Inclusion 74% dari masing masing payload berjumlah 100 serangan dengan tingkat risiko level high.

REFERENSI

[1] S. Ali and T. Nadeem Malik, “Intrusion Detection and Prevention against Cyber Attacks for an Energy Management System,” Mehran University Research Journal of Engineering and Technology, vol. 41, no. 1, p. 202, doi:

10.22581/muet1982.2201.20.

[2] H. Alnabulsi, M. R. Islam, and Q. Mamun, "Detecting SQL injection attacks using SNORT IDS," in Asia- Pacific World Congress on Computer Science and Engineering, APWC on CSE 2014, Institute of Electrical and Electronics Engineers Inc., 2014. doi: 10.1109/APWCCSE.2014.7053873.

[3] Md. A. Islam and Md. M. Islam, "A Novel Signature- Based Traffic Classification Engine To Reduce False Alarms In Intrusion Detection Systems," International journal of Computer Networks & Communications, vol.7, no. 1, pp. 63–80, Jan. 2015, doi: 10.5121/ijcnc.2015.7105

[4] I. Gede, W. Bangga, and S. M. Ladjamuddin, "SIMULASI SNORT SEBAGAI ALAT PENDETEKSI INTRUSI PADA WEB DAMN VULNERABLE WEB APPLICATION," Jurnal Rekayasa Informasi, vol. 11, no. 2, 2022.

[5] D. T. Yuwono, "Analysis Performance Intrusion Detection System in Detecting Cyber-Attack on Apache Web Server," IT Journal Research and Development, pp. 169–178, Feb. 2022, doi: 10.25299/itjrd.2022.7853..

[6] R. Muwardi, H. Gao, H. U. Ghifarsyam, M. Yunita, A. Arrizki, and J. Andika, "Network Security Monitoring System Via Notification Alert," Journal of Integrated and Advanced Engineering (JIAE), vol. 1, no. 2, pp. 113– 122, Nov. 2021, doi: 10.51662/jiae.v1i2.22.

[7] E. Gunadhi and A. Sudrajat, "PENGAMANAN DATA REKAM MEDIS PASIEN MENGGUNAKAN KRIPTOGRAFI VIGÈNERE CIPHER," 2016. [Online]. Available: http://jurnal.sttgarut.ac.id

[8] R. M. Muhammad, I. Dyah Irawati, and M. Iqbal, "Integrated Security System Implementation for Network Intrusion," 2021.

[9] R. M. Muhammad, I. Dyah Irawati, and M. Iqbal, "Integrated Security System Implementation for Network Intrusion," 2021.

[10] N. Novi and Z. Zaini, "Secure Socket Layer untuk Keamanan Data Rekam Medis Tumor Otak pada Health Information System," JURNAL NASIONAL TEKNIK ELEKTRO, vol. 6, no. 3, p. 137, Jul. 2017, doi: 10.25077/jnte.v6n3.405.2017.