Evaluasi Penerapan Pertahanan Proaktif Waf Dan Hids Terhadap Eksploitasi Kerentanan Aplikasi Web

Adi Suryadin, Kurniadin Abd Latif, Lilik Widyawati, Raisul Azhar, Muhamad Azwar

Universitas Bumigora, Mataram, Indonesia

Correspondence : *e-mail*: adisuryadin598@gmail.com

Abstrak

Keamanan aplikasi web menjadi semakin krusial dalam beberapa tahun terakhir, terutama terhadap serangan injeksi seperti SQL Injection (SQLi) dan Cross-Site Scripting (XSS). Berdasarkan laporan OWASP Top 10 tahun 2021, lebih dari 94% aplikasi web rentan terhadap serangan injeksi. Data Badan Siber dan Sandi Negara (BSSN) 2023 mencatat lebih dari 370 juta serangan siber di Indonesia, dengan sektor pemerintahan sebagai target utama. Penelitian ini mengevaluasi efektivitas Web Application Firewall (ModSecurity) dan Host-Based Intrusion Detection System (Wazuh) sebagai pertahanan proaktif. Eksperimen dilakukan di lingkungan virtual menggunakan pendekatan NDLC. Hasil menunjukkan sistem mencapai Detection Rate 100%, False Negative Rate 0%, dan False Positive Rate 4,2%, dengan akurasi 97,01%. Latensi respons pasif nol detik dan latensi respons aktif rata-rata satu detik, menandakan mitigasi hampir real-time. Dari sisi efisiensi, implementasi meningkatkan penggunaan CPU web server sebesar 25,9% dan RAM sebesar 18 MB, serta CPU Wazuh Server 9% dan RAM 53 MB. Kombinasi WAF dan HIDS efektif mendeteksi dan memblokir serangan SQLi dan XSS, termasuk varian tersamarkan. Integrasi ini direkomendasikan untuk sistem web dengan trafik menengah, dengan catatan pengaturan aturan dan manajemen log penting untuk menjaga efisiensi dan kestabilan jangka panjang.

***Kata kunci**: Pertahanan Proaktif, WAF, HIDS, SQL Injeksi, Cross-Site Scripting*

Abstract

Web application security has become increasingly critical, especially against injection attacks such as SQL Injection (SQLi) and Cross-Site Scripting (XSS). According to the OWASP Top 10 report in 2021, over 94% of web applications are vulnerable to injection attacks. Data from the National Cyber and Crypto Agency (BSSN) in 2023 recorded more than 370 million cyberattacks in Indonesia, with the government sector as the primary target. This study evaluates the effectiveness of a Web Application Firewall (ModSecurity) and Host-Based Intrusion Detection System (Wazuh) as proactive defenses. Experiments were conducted in a virtual environment using the NDLC approach. The results show that the system achieved a Detection Rate of 100%, False Negative Rate of 0%, and False Positive Rate of 4.2%, with an accuracy of 97.01%. The passive response latency was zero seconds, and the average active response latency was one second, indicating near real-time mitigation. In terms of efficiency, implementation increased web server CPU usage by 25.9% and RAM by 18 MB, as well as Wazuh Server CPU by 9% and RAM by 53 MB. The combination of WAF and HIDS effectively detects and blocks SQLi and XSS attacks, including obfuscated variants. This integration is recommended for web systems with moderate traffic, noting that rule configuration and log management are essential to maintain long-term efficiency and stability.

***Keywords**: Proactive Defense, WAF, HIDS, SQL Injection, Cross-Site Scripting*

1. Pendahuluan

Keamanan aplikasi web menjadi semakin krusial dalam beberapa tahun terakhir. Celah keamanan umum pada aplikasi web adalah Cross-Site Scripting (XSS) dan SQL Injection (SQLi)[1], yang kerap dimanfaatkan penyerang karena kesalahan pemrograman dan input yang tidak disanitasi. Data global menunjukkan XSS dan SQLi masih menempati peringkat atas sebagai ancaman kritis perangkat lunak. Dalam daftar CWE Top 25 tahun 2023, XSS menduduki urutan kedua dan SQLi urutan ketiga[2]. Sobola et al. (2020) melaporkan bahwa SQL Injection menyumbang lebih dari 72% dari semua serangan yang terdeteksi[1]. Sementara itu, OWASP Top 10 2021 mencatat bahwa 94% aplikasi web yang diuji mengandung kerentanan jenis injeksi (termasuk XSS)[3]. Tren ini menunjukkan bahwa selama lima tahun terakhir serangan berbasis injeksi, khususnya XSS dan SQLi, tetap menjadi ancaman utama terhadap keamanan aplikasi web.

Statistik insiden keamanan aplikasi web global dan nasional menggambarkan skala ancaman yang besar. Survei Cyberthreat Defense 2023 menyebut SQL Injection dan XSS sebagai dua dari serangan aplikasi web yang paling mengkhawatirkan para profesional keamanan[4]. Di Indonesia, Badan Siber dan Sandi Negara (BSSN) melaporkan terjadi 370,02 juta serangan siber pada tahun 2022, naik hampir 39% dari tahun sebelumnya[5]. Dari jumlah tersebut, sektor pemerintahan menjadi target utama dengan sekitar 284,09 juta serangan[5]. Hasil-hasil tersebut menunjukkan bahwa serangan aplikasi web (termasuk SQLi dan XSS) merupakan komponen signifikan dalam lanskap ancaman siber, baik di tingkat global maupun nasional.Selain

mekanisme perlindungan, akar masalah lain adalah minimnya penerapan prinsip *secure coding* selama pengembangan aplikasi web. Jayatilaka et al. (2022) mengidentifikasi bahwa banyak pelatihan keamanan tetap tidak kontekstual dan tidak efektif; sebagian besar pengembang merasa materi terlalu umum dan tidak relevan dengan kebutuhan mereka dalam menghadapi kerentanan seperti XSS dan SQLi[6]. Sebagai tambahan, Haug et al. (2022) menyarankan bahwa penggunaan feedback langsung dari alat *static application security testing* (SAST) di dalam lingkungan pengembangan (IDE) dapat meningkatkan kesadaran dan respons terhadap kesalahan kode keamanan secara otomatis, sehingga mengurangi kerentanan sejak fase awal implementasi[7].

Untuk menghadapi ancaman tersebut, solusi pertahanan proaktif digunakan sebagai lapisan tambahan keamanan. Web Application Firewall (WAF) adalah lapisan pertahanan pertama yang menyaring dan memantau lalu lintas HTTP menuju aplikasi web[1], [9]. Contoh WAF open-source yang banyak digunakan adalah ModSecurity, yang dapat dipasang sebagai modul pada server web atau proxy untuk memeriksa setiap permintaan HTTP[10]. ModSecurity, biasanya dikonfigurasi dengan OWASP Core Rule Set (CRS), mampu mendeteksi dan menolak serangan injeksi, XSS, dan eksploitasi kerentanan lain secara real-time[1]. Di sisi lain, Host-based Intrusion Detection System (HIDS) seperti Wazuh diinstal langsung di server atau endpoint untuk memantau aktivitas sistem. Wazuh adalah sistem SIEM/HIDS open-source yang mengumpulkan serta menganalisis log, peringatan intrusi, dan perubahan integritas file di host untuk mendeteksi ancaman dan anomali lokal[11]. Dengan demikian, WAF dan HIDS saling melengkapi: WAF fokus menyaring serangan dari tingkat jaringan, sementara HIDS mendeteksi tanda-tanda intrusi atau penyalahgunaan yang lolos ke dalam sistem host.

Validasi pendekatan *defense-in-depth* juga didukung oleh hasil empiris. Dalam sebuah kajian mengenai WAF berbasis pembelajaran mesin, Isiker & Sögükpinar (2021) melaporkan bahwa metode anomaly-based menggunakan SVM, Random Forest, dan algoritma lainnya dapat mencapai akurasi hingga 89% dan AUC 94% dalam mendeteksi serangan HTTP tren serangan termasuk SQLi dan XSS[8][12]. Hasil ini melengkapi studi lain di mana pendekatan kombinasi WAF + HIDS secara teori dapat meningkatkan deteksi terhadap serangan multi-langkah yang sering melewati satu lapisan perlindungan saja.

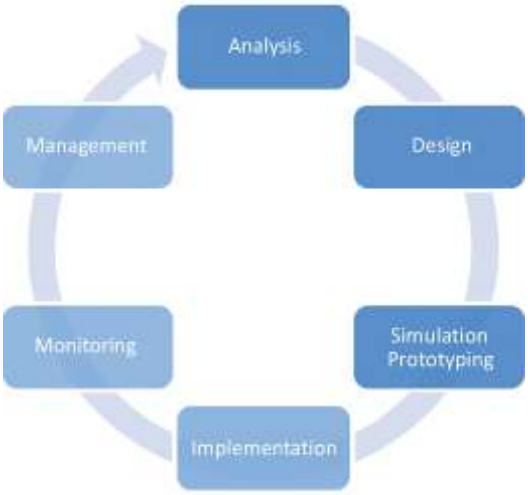
Meskipun memiliki keunggulan, penggunaan WAF dan HIDS secara terpisah masih menghadapi keterbatasan. Penelitian sebelumnya menunjukkan ModSecurity (CRS v3.2) dalam konfigurasi default dapat mendeteksi banyak serangan, namun sejumlah vektor tetap lolos tanpa terdeteksi atau tidak tercatat[1]. Misalnya, payload berbahaya dalam unggahan file dapat luput dari pemeriksaan ModSecurity. Dari sisi HIDS, evaluasi kinerja menemukan bahwa sistem Wazuh memiliki tingkat deteksi serangan yang rendah; dibandingkan dengan HIDS lain, Wazuh lebih efisien tetapi tetap mendeteksi hanya sebagian kecil percobaan serangan yang dilakukan[13]. Selain itu, sifat deteksi berbasis aturan Wazuh menyebabkan tingkat false positive yang tinggi dan kesulitan menyesuaikan diri pada serangan baru tanpa pembaruan ruleset[11]. Keterbatasan ini menegaskan perlunya pendekatan pertahanan berlapis (defense-in-depth), sehingga celah pada satu mekanisme dapat diisi oleh mekanisme lainnya.

Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi efektivitas kombinasi WAF(Modsecurity) dan HIDS(Wazuh) dalam mendeteksi serta merespons serangan SQL Injection dan XSS pada aplikasi web. Evaluasi dilakukan berdasarkan parameter Detection Rate, False Positive, False

Negative, waktu respons, serta efisiensi sumber daya. Penelitian ini diharapkan menjadi referensi dalam membangun sistem keamanan aplikasi web yang responsif, hemat sumber daya, dan mampu beroperasi secara andal dalam berbagai skenario serangan. Selain berkontribusi pada pengembangan keilmuan di bidang keamanan siber, penelitian ini juga memberikan manfaat langsung bagi masyarakat umum, terutama pengguna layanan daring, dengan meningkatkan kepercayaan terhadap keamanan data pribadi dan transaksi digital mereka. Penerapan sistem pertahanan yang efektif pada aplikasi web publik seperti layanan kesehatan, pendidikan, dan administrasi pemerintahan dapat meminimalisir risiko kebocoran data serta gangguan layanan yang merugikan masyarakat luas.

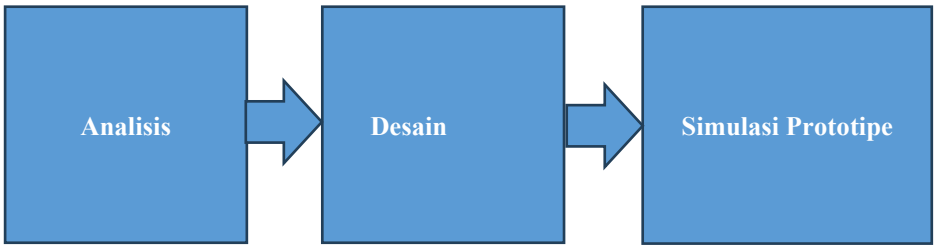
2. Metode Penelitian

Penelitian ini menggunakan pendekatan eksperimental, Pendekatan eksperimental selayaknya uji performa IDS sering digunakan dalam literatur keamanan siber, di mana metrik seperti True/False Positive Rate dan ROC curve menjadi bahan analisis utama[14], Pendekatan ini dipilih karena memungkinkan pengujian hipotesis dalam lingkungan virtual yang terkontrol, aman, dan dapat direplikasi, tanpa menimbulkan risiko pada sistem produksi. Untuk mendukung perancangan sistem, digunakan model pengembangan Network Development Life Cycle (NDLC), Kerangka NDLC sebagai siklus pendukung riset keamanan telah diterapkan pada sejumlah studi infrastruktur berbasis Linux/Firewall, sehingga cocok digunakan dalam desain sistem ini[15].



Gambar 1. Tahapan Metode NDLC

Dari enam tahapan utama NDLC, penelitian ini hanya menerapkan tiga tahap yang disesuaikan dengan fokus kajian, yaitu: Analisis, Desain, dan Simulasi Prototipe. Hal ini sejalan dengan penelitian sebelumnya yang menyatakan bahwa tahap Analisis, Desain, dan Simulasi Prototipe sudah cukup untuk digunakan dalam penelitian uji coba[16].



Gambar 2. Tahapan NDLC yang digunakan dalam penelitian

2.1. Tahap Analisis

Tahap Analisis merupakan fase yang bertujuan untuk mendefinisikan kebutuhan sistem secara menyeluruh, mencakup identifikasi permasalahan eksisting, keinginan pengguna, dan evaluasi topologi yang ada. Hasil dari fase ini menjadi dasar untuk menyusun spesifikasi sistem yang jelas sebelum memasuki tahap desain[17]

2.1.1. Spesifikasi Lingkungan Uji

Lingkungan pengujian terdiri dari empat mesin virtual dengan spesifikasi perangkat keras dan lunak seperti pada Tabel 1.

Table 1. Spesifikasi kebutuhan sistem

Peran	Spesifikasi Teknis	Komponen Perangkat Lunak	IP
Server Web Target	2 Vcpu, 4 GB RAM, 40 GB HDD	Ubuntu Server 22.04 LTS, Apache, MySQL, PHP, DVWA, ModSecurity, Wazuh Agent	172.20.51.140/24
Mesin Penyerang	1 Vcpu, 2 GB RAM, 80.1 GB HDD	Kali Linux (2024.3)	172.20.51.131/24
Klien Normal	N/A (Host PC)	Windows 11	DHCP
Server Manajemen HIDS	2 Vcpu, 6.1 GB RAM, 80 GB HDD	Ubuntu Server 22.04 LTS, Wazuh Manager	172.20.51.135/24

2.1.2. Matriks Parameter Evaluasi

Parameter evaluasi berdasarkan True Positive (TP), True Negative (TN), False Negative (FN), Detection Rate (DR), False Positive (FP), Response Time, dan Resource Usage. Definisi metrik ini mengacu pada konvensi baku dalam penelitian IDS modern [18], [19].

Untuk menilai performa sistem, digunakan sejumlah metrik evaluasi berbasis statistik yang merujuk pada hasil klasifikasi (TP, FP, TN, FN). Evaluasi dengan pendekatan ini memungkinkan analisis objektif terhadap keberhasilan deteksi, tingkat kesalahan klasifikasi, serta efisiensi waktu respons sistem. [18] secara rinci mengkaji metrik ini berdasarkan *confusion matrix* dari berbagai dataset intrusi global dan menyarankan agar metrik seperti DR, FPR, serta AUC digunakan secara bersamaan untuk analisis kinerja sistem keamanan informasi.

Perumusan masing-masing metrik evaluasi dijelaskan sebagai berikut:

Detection Rate (DR) menunjukkan persentase serangan yang berhasil dideteksi, dirumuskan sebagai:

$$DR = \frac{TP}{TP + FN} \, x \, 100\% \quad (1)$$

di mana TP 494dalah jumlah true positive (serangan yang berhasil dideteksi dan diblokir), dan FN 494dalah false negative (serangan yang gagal dideteksi).

False Negative Rate (FNR) adalah persentase serangan yang tidak terdeteksi, dihitung sebagai:

$$FPR = \frac{FP}{FP + TN} \, x \, 100\% \quad (2)$$

Sedangkan False Positive Rate (FPR) mengukur persentase permintaan sah yang salah diidentifikasi sebagai serangan:

$$FNR = \frac{FN}{TP + FN} \, x \, 100\% \quad (3)$$

Akurasi keseluruhan sistem didefinisikan sebagai:

$$AC = \frac{TP + TN}{TP + TN + FP + FN} \, x \, 100\% \quad (4)$$

Waktu respons (Response Time, RT) sistem dihitung dari selisih waktu antara aksi blokir dan deteksi serangan:

$$RT = \frac{1}{n} \sum_{i=1}^{n} ti \quad (5)$$

di mana ti adalah waktu respons pada pengujian ke-i.

Penggunaan sumber daya (Resource Usage, RU) dihitung sebagai perubahan persentase penggunaan penggunaan sumber daya berikut :

$$RU\ \% = \frac{RU_{setelah} - RU_{dasar}}{RU_{dasar}}\ x\ 100\% \quad (6)$$

2.2. Desain

Tahap Desain berfokus pada perancangan arsitektur sistem secara detail berdasarkan hasil analisis kebutuhan. Tahapan ini mencakup dua perspektif utama: (1) model konseptual, yang menjelaskan logika interaksi layanan dan sistem; (2) topologi arsitektur, yang menggambarkan penempatan komponen jaringan seperti firewall, server, dan segmen jaringan fisik. Hasil desain disusun sebagai blueprint formal dan dipresentasikan untuk validasi sebelum dilanjutkan ke fase berikutnya[20]

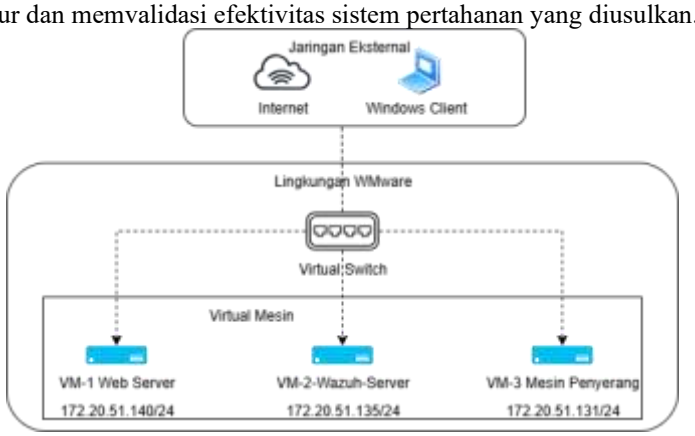
2.2.1. Model Konseptual

Sistem dirancang agar WAF mendeteksi serangan pada aplikasi web dan mencatat log. HIDS kemudian memonitor log tersebut dan memicu respons aktif untuk memblokir alamat IP penyerang.

Gambar 3. Model Alur Konseptual Sistem Pertahanan Proaktif

2.2.2. Topologi Arsitektur Sistem

Topologi pada gambar 4 dirancang untuk merepresentasikan skenario realistis di mana klaster server aplikasi menjadi target serangan yang berasal dari dua sumber berbeda, yaitu dari jaringan Internet publik dan dari perangkat penyerang yang beroperasi secara terkontrol. Lingkungan yang telah disiapkan ini menjadi dasar untuk melakukan seluruh skenario pengujian guna mengukur dan memvalidasi efektivitas sistem pertahanan yang diusulkan.



Gambar 4. Topologi Jaringan Sistem

2.3. Simulasi Prototipe

Desain skenario pengujian mengacu pada metode yang digunakan oleh Durmuşkaya dan Bay[12], yang menguji efektivitas WAF melalui ribuan permintaan HTTP terhadap aplikasi web rentan dengan proporsi input berbahaya dan normal yang seimbang. Berdasarkan pendekatan tersebut, penelitian ini melakukan pengujian terhadap total 87 permintaan (*payload*) yang mencakup serangan langsung, serangan tersamarkan, dan input normal.

Skenario pengujian sistem meliputi:

1. Pengiriman 20 serangan langsung (10 SQL Injection, 10 Cross-site Scripting) untuk menguji *Detection Rate (DR)*.
2. Pengiriman 47 input non-malicious yang mirip dengan pola serangan (*near miss legitimate*) untuk menguji *False Positive Rate (FPR)*.
3. Pengiriman 20 payload tersamarkan (obfuscated), masing-masing sepuluh untuk SQLi dan XSS, untuk menguji *False Negative Rate (FNR)*.

4. Mengukur latensi antara deteksi serangan dan eksekusi tindakan mitigasi. Dengan mencatat timestamp log akses sebelum dan setelaha log pemblokiran IP
5. Pengukuran dampak serangan terhadap performa sistem dilakukan dengan skenario *brute force* untuk menguji efisiensi sumber daya (CPU, RAM, dan disk I/O).
6. Perhitungan DR menggunakan persamaan (1), FPR dan FNR masing-masing menggunakan persamaan (3) dan (2). Parameter *Response Time* dihitung berdasarkan persamaan (5), sedangkan evaluasi penggunaan sumber daya mengikuti persamaan (6).
7. Menarik kesimpulan berdasarkan hasil pengujian

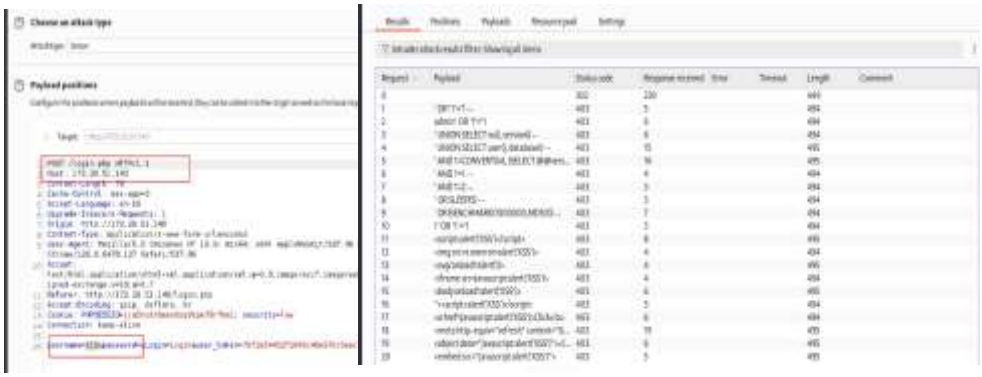
3. Hasil dan Pembahasan

Pada bagian ini, akan diuraikan secara rinci mengenai hasil dari simulasi sistem, proses pengujian, serta analisis terhadap data yang diperoleh. Pembahasan dalam bagian ini bertujuan untuk menjawab rumusan masalah yang telah ditetapkan dengan mengacu pada kerangka metodologi penelitian yang dijelaskan pada tahap 2.

3.1. Evaluasi Kualitas Deteksi

Evaluasi dilakukan dalam tiga skenario utama untuk mengukur *Detection Rate (DR)*, *False Positive Rate (FPR)*, dan *False Negative Rate (FNR)* menggunakan Burp Suite terhadap 87 permintaan (payload).

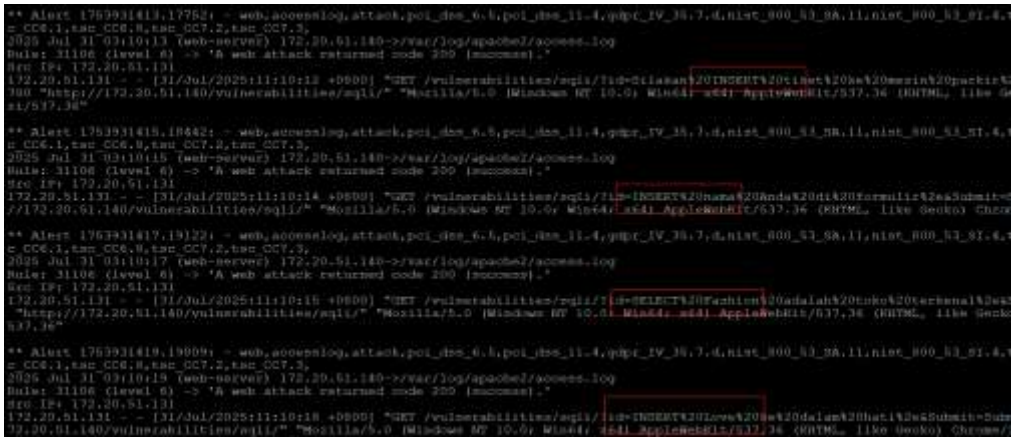
0. *Detection Rate (DR)*



Gambar 5. Konfigurasi dan hasil injeksi SQLi/XSS melalui Burp Suite

Gambar 5 menunjukkan pengiriman payload SQLi dan XSS menggunakan Burp Suite ke aplikasi target, dengan pola umum seperti ' OR 1=1 – dan <script>alert(1)</script>. WAF ModSecurity mendeteksi seluruh serangan dan memberi respons HTTP 403 Forbidden, sekaligus mengirim log ke Wazuh untuk pencatatan insiden, sehingga DR tercapai 100%.

b. *False Positive Rate (FPR)*

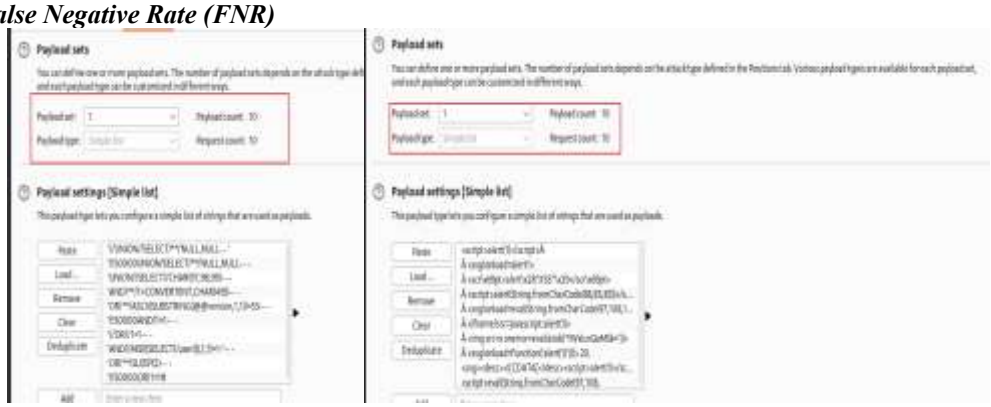


Gambar 6. Log Wazuh yang memicu alert pada input non-berbahaya

Meskipun WAF tidak memblokir input sah, HIDS Wazuh tetap menghasilkan alert pada beberapa kasus. Hal ini disebabkan rule berbasis signature yang mendeteksi keberadaan kata kunci

seperti SELECT atau INSERT meskipun digunakan dalam konteks normal. FPR sebesar 4,2% yang dihasilkan menunjukkan perlunya penyesuaian (rule tuning) untuk menghindari overload alert pada administrator. Kejadian ini menjadi bukti bahwa deteksi berbasis aturan memerlukan kalibrasi berkelanjutan agar tetap akurat.

c. ***False Negative Rate (FNR)***



Gambar 7. Payload XSS/SQLi tersamarkan

Gambar 7 menunjukkan contoh payload yang telah dimodifikasi (obfuscated) dengan teknik seperti encoding karakter, penggunaan whitespace berlebih, dan penggabungan operator. Tujuannya adalah menguji ketahanan sistem terhadap variasi serangan yang mencoba menghindari deteksi signature konvensional. Penggunaan payload ini mensimulasikan kondisi serangan canggih yang sering ditemukan pada skenario Advanced Persistent Threat (APT).

Tabel 2. Rekapitulasi Hasil Evaluasi Deteksi

Parameter Evaluasi	Total Uji	TP / FP	FN / TN	Nilai
DR	20	20 / –	0 / –	100%
FPR	47	2 / –	– / 45	4,26%
FNR	20	– / –	0 / –	0%
Accuracy	87	20 / 2	0 / 45	97,01%

Tabel diatas Merangkum hasil pengujian tiga parameter utama (DR, FPR, FNR) dan akurasi sistem, dengan hasil DR 100%, FNR 0%, FPR 4,26%, dan akurasi 97,01%.

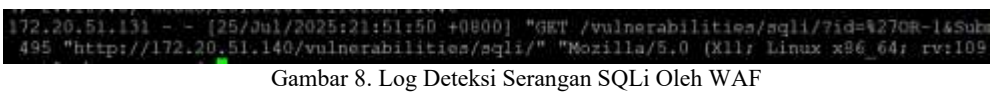
3.2. Evaluasi Waktu Respons Sistem

Evaluasi ini bertujuan untuk mengukur selisih waktu antara deteksi serangan oleh sistem dengan proses pemberian respons, baik secara pasif maupun proaktif. Pengujian pasif dilakukan dengan membandingkan waktu pencatatan log pada web server dan waktu pembuatan alert oleh HIDS. Pengujian proaktif dilakukan dengan mencatat perbedaan waktu antara deteksi serangan oleh WAF dan eksekusi pemblokiran IP otomatis oleh *active-response* Wazuh. Parameter waktu ini dianalisis untuk menilai kecepatan sistem dalam melakukan mitigasi terhadap ancaman secara efektif.

a. waktu respon pasif

memperlihatkan bahwa dalam semua percobaan, HIDS Wazuh mampu menghasilkan alert segera setelah log terdeteksi pada web server. Waktu deteksi pasif tercatat 0 detik, yang menandakan proses parsing log dan pencatatan alert berjalan hampir instan. Kemampuan ini penting untuk mendukung analisis insiden cepat oleh tim keamanan.

b. Waktu respon proaktif



Gambar 8. Log Deteksi Serangan SQLi Oleh WAF

Gambar 8 Menunjukkan timestamp pada log access.log ketika ModSecurity mendeteksi adanya pola serangan SQLi. Deteksi cepat pada tahap awal sangat krusial untuk mencegah eksekusi query berbahaya di database.

Gambar 9. Log Deteksi Serangan XSS Oleh WAF

Gambar 9 Menampilkan pencatatan deteksi XSS pada log server. Payload XSS yang diuji memanfaatkan injeksi script sederhana, namun langsung terdeteksi oleh signature ModSecurity.

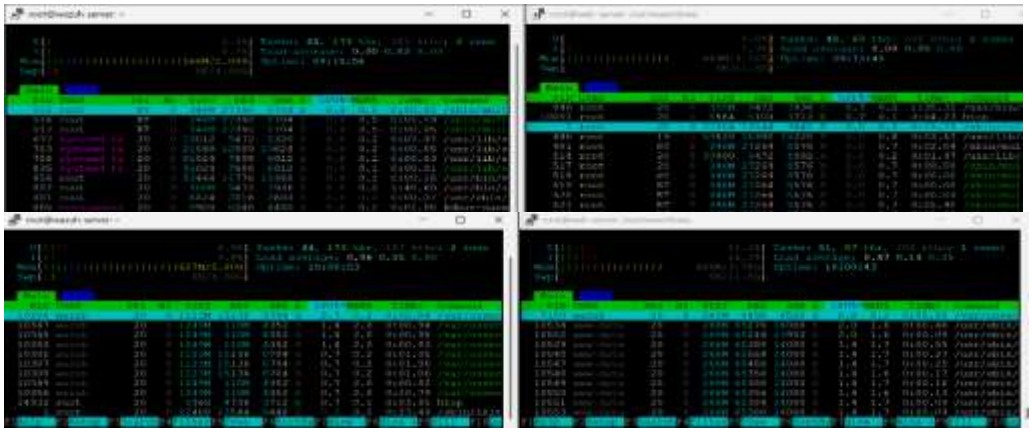
Tabel 3. Hasil Respons Pemblokiran

No	Jenis Serangan	T1 (Deteksi)	T2 (Blok IP)	Selisih
1	SQLi	21:51:50	21:51:51	1 detik
2	XSS	21:04:29	21:04:30	1 detik

Tabel diatas menunjukkan perbandingan waktu deteksi dan pemblokiran pada serangan SQLi dan XSS, masing-masing dengan selisih hanya 1 detik.

3.3. Evaluasi Penggunaan Sumber Daya

Evaluasi ini dilakukan untuk mengamati perubahan penggunaan sumber daya sistem selama proses serangan dan mitigasi berlangsung. Parameter yang dianalisis meliputi beban CPU, penggunaan memori (RAM), serta aktivitas I/O disk pada server web dan server Wazuh. Pengukuran dilakukan pada kondisi idle (sebelum serangan) dan saat serangan berlangsung, sehingga dapat diketahui dampak operasional dari penerapan WAF dan HIDS terhadap performa infrastruktur.



Gambar 10. Penggunaan CPU dan RAM saat idle dan saat serangan

Gambar 10 membandingkan beban CPU dan penggunaan RAM pada web server dalam kondisi idle dan saat terjadi serangan. Peningkatan CPU sebesar 25,9% masih berada dalam ambang toleransi untuk skenario trafik menengah. Hal ini menunjukkan bahwa penambahan lapisan keamanan tidak berdampak signifikan terhadap kinerja keseluruhan.

Tabel 4. Dampak Serangan terhadap Sumber Daya

Metrik	Server	Sebelum	Saat Serangan	Dampak
Load CPU	Web Server	0.00	0.47	+0.47
	Wazuh Server	0.06	0.06	Tidak signifikan
Penggunaan CPU	Web Server	1,3%	27,2%	+25,9%
	Wazuh Server	1,3%	10,3%	+9,0%

Konsumsi RAM	Web Server	451	469	+18 MB
	Wazuh Server	574	627	+53 MB
Aktivitas Disk	Wazuh Server	0,0	518,6	+518,6 Kb/s

Tabel diatas merangkuman hasil evaluasi sumberdaya sistem, dampak serangan pada CPU, RAM, dan I/O disk di kedua server (Web dan Wazuh).

3.4. Rekapitulasi Evaluasi Sistem

Bagian ini menyajikan rangkuman dari seluruh parameter evaluasi yang telah diuji, meliputi kualitas deteksi, waktu respons, dan penggunaan sumber daya sistem. Rekapitulasi dilakukan dengan menyusun data hasil pengujian ke dalam tabel ringkasan sehingga memudahkan pembaca dalam melihat kinerja keseluruhan sistem pertahanan proaktif berbasis WAF dan HIDS. Informasi ini digunakan sebagai dasar analisis efektivitas sistem dalam mengidentifikasi, merespons, dan memitigasi ancaman secara real-time tanpa mengganggu stabilitas operasional.

Tabel 5. Hasil Keseluruhan evaluasi

Tolok Ukur	Nilai	Kesimpulan
DR	100 %	Sangat baik, seluruh ancaman berhasil terdeteksi
FPR	4,2 %	Masih dapat ditoleransi, perlu tuning aturan untuk menurunkan lagi
FNR	0 %	Ideal, tidak ada serangan yang lolos undetected
Accuracy	97,01 %	Tingkat keakuratan klasifikasi sangat tinggi
Latency Respons Pasif	0 detik	Deteksi dan alert tercatat seketika oleh HIDS
Latency Respons Aktif	1 detik	Block-IP otomatis berjalan cepat, mendekati real-time
CPU Usage (Web Server)	+25,90 %	Beban masih dalam batas aman; perlu monitoring saat beban puncak
Disk I/O (Wazuh Server)	518,60 KB/s	Logging intensif selama serangan; pertimbangkan rotasi log atau buffering

Ringkasan akhir kinerja sistem: DR 100%, FPR 4,2%, FNR 0%, akurasi 97,01%, respon pasif 0 detik, respon aktif 1 detik, dengan dampak sumber daya yang masih terkendali.

Integrasi WAF ModSecurity dan HIDS Wazuh terbukti efektif sebagai sistem pertahanan proaktif terhadap SQL Injection dan XSS. Hasil pengujian menunjukkan nilai *Detection Rate* sebesar 100% dan *False Negative Rate* sebesar 0%, yang berarti seluruh ancaman, termasuk payload tersamarkan, berhasil dideteksi dan diblokir tanpa ada serangan yang lolos. Meskipun *False Positive Rate* mencapai 4,26%, nilai ini masih tergolong dapat ditoleransi untuk skenario simulasi, dan berpotensi diturunkan melalui penyesuaian aturan (*rule tuning*), Secara keseluruhan Accuracy sistem mencapai 97,01%. Dari sisi waktu respons, mekanisme deteksi pasif tercatat 0 detik, sedangkan respons aktif pemblokiran IP rata-rata hanya membutuhkan 1 detik, menunjukkan kemampuan mitigasi yang hampir real-time. Dampak terhadap sumber daya server juga relatif ringan, dengan peningkatan penggunaan CPU pada web server sebesar +25,90% dan lonjakan aktivitas I/O disk pada Wazuh server sebesar 518,60 kB/s selama proses logging intensif. Secara keseluruhan, sistem mampu memberikan perlindungan yang andal dengan kinerja yang stabil dalam skenario serangan yang diuji.

4. Kesimpulan

Penelitian ini membuktikan bahwa integrasi Web Application Firewall (ModSecurity) dan Host-Based Intrusion Detection System (Wazuh) mampu memberikan pertahanan proaktif yang efektif terhadap SQL Injection dan Cross-Site Scripting, termasuk varian tersamarkan. Sistem berhasil mendeteksi seluruh serangan secara real-time dengan dampak sumber daya yang masih dalam batas aman. Meski masih terdapat

false positive dan beban log yang cukup tinggi, kekurangan ini dapat diminimalkan melalui penyesuaian aturan dan manajemen log yang baik.

Daftar Pustaka

[1] T. D. Sobola, P. Zavarsky, and S. Butakov, "Experimental Study of ModSecurity Web Application Firewalls," in *Proceedings - 2020 IEEE 6th Intl Conference on Big Data Security on Cloud, BigDataSecurity 2020, 2020 IEEE Intl Conference on High Performance and Smart Computing, HPSC 2020 and 2020 IEEE Intl Conference on Intelligent Data and Security, IDS 2020*, 2020. doi: 10.1109/BigDataSecurity-HPSC-IDS49724.2020.00045.

[2] M. Tsiodra, S. Panda, M. Chronopoulos, and E. Panaousis, "Cyber Risk Assessment and Optimization: A Small Business Case Study," *IEEE Access*, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3272670.

[3] Synopsys, "OWASP Top 10 2021," 2021.

[4] Imperva, "What is OSI Model | 7 Layers Explained | Imperva," 2023.

[5] M. Lestari and T. Finaldin, "KERJA SAMA ANTARA INDONESIA DAN NEGARA-NEGARA DI ASIA TENGGARA MELALUI ASEAN REGIONAL FORUM DALAM BIDANG KEAMANAN SIBER," *Global Mind*, vol. 4, no. 2, 2023, doi: 10.53675/jgm.v4i2.987.

[6] V. Pikulin *et al.*, "Towards Developer-Centered Secure Coding Training," in *Proceedings - 2023 38th IEEE/ACM International Conference on Automated Software Engineering Workshops, ASEW 2023*, 2023. doi: 10.1109/ASEW60602.2023.00008.

[7] M. Haug, A. C. F. da Silva, and S. Wagner, "Towards Immediate Feedback for Security Relevant Code in Development Environments," in *Communications in Computer and Information Science*, 2022. doi: 10.1007/978-3-031-18304-1_4.

[8] M. Jaza Al Anzi and M. Abdul-Rahman Al Balwi, "International Journal of INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING Secure Software Development: Problems and Solutions." [Online]. Available: www.ijisae.org

[9] F. M. Alotaibi and V. G. Vassilakis, "Toward an SDN-Based Web Application Firewall: Defending against SQL Injection Attacks," *Future Internet*, vol. 15, no. 5, 2023, doi: 10.3390/fi15050170.

[10] H. Hardianto, "Analisis Cyber Crime handling pada Aplikasi Web dengan WAF ModSecurity," *PETIR*, vol. 16, no. 1, 2023, doi: 10.33322/petir.v16i1.1910.

[11] S. A. Chamkar, M. Zaydi, and Y. Maleh, "Improving Threat Detection in Wazuh Using Machine Learning Techniques," pp. 1–25, 2025.

[12] M. E. Durmuşkaya and S. Bayraklı, "Web application firewall based on machine learning models," *PeerJ Comput Sci*, vol. 11, p. e2975, Jul. 2025, doi: 10.7717/peerj-cs.2975.

[13] A. Willerton, "Evaluating the efficiency of Host-based Intrusion Detection Systems protecting web applications," no. June, 2022.

[14] L. Layman and W. Roden, "A Controlled Experiment on the Impact of Intrusion Detection False Alarm Rate on Analyst Performance," in *Proceedings of the Human Factors and Ergonomics Society*, 2023. doi: 10.1177/21695067231192573.

[15] R. T. Prabowo and M. T. Kurniawan, "Analisis dan Desain Keamanan Jaringan Komputer dengan Metode Network Development Life Cycle (Studi Kasus: Universitas Telkom)," *Jurnal Rekayasa Sistem & Industri*, vol. 2, no. 1, 2015.

[16] I. Kamu, M. T. Parinsi, M. W. Kuhu, and A. V. Mananggel, "Computer Network Design in Vocational School Using Network Simulator," *International Journal of Information Technology and Education*, vol. 2, no. 1, 2022, doi: 10.62711/ijite.v2i1.86.

[17] M. Anto, "Implementasi Jaringan Point to Multipoint Menggunakan Metode NDLC," *MULTINETICS*, vol. 8, no. 2, 2023, doi: 10.32722/multinetics.v8i2.5066.

[18] G. Kumar Ahuja and G. Kumar, "Evaluation metrics for intrusion detection systems-a study," *Evaluation*, vol. 2, no. 11, 2014.

[19] C. Anthony, W. Elgenaidi, and M. Rao, "Intrusion Detection System for Autonomous Vehicles Using Non-Tree Based Machine Learning Algorithms," *Electronics (Switzerland)*, vol. 13, no. 5, 2024, doi: 10.3390/electronics13050809.

[20] N. Nurdadyansyah and M. Hasibuan, "Perancangan Local Area Network Menggunakan NDLC Untuk Meningkatkan Layanan Sekolah," *Jurnal KONIK*, vol. 5, 2021.