Jurnal Ilmiah
Teknologi Sistem
Informasi

Jitsi

Vol. 6 (2025) No. 4

https://jurnal-itsi.org/index.php/jitsi; E-ISSN: 2722-4600; ISSN: 2722-4619
DOI: 10.62527/jitsi.6.4.523

Implementasi Prototipe SIEM Berbasis Wazuh pada Website dengan Pengujian FIM dan Threat Hunting

Nurmi Hidayasari[#], Mansur[#], Kasmawi[#], Zuliar Efendi[#]

[#] *Departemen of Informatics Engineering, Politeknik Negeri Bengkalis, 28712, Indonesia*
E-mail: nurmihidayasari[at]polbeng.ac.id

A B S T R A C T S

This study implements a Wazuh-based Security Information and Event Management (SIEM) prototype to enhance security monitoring for a web application. The architecture uses two VPS instances: a web server as the log source equipped with a Wazuh Agent, and a monitoring server running the Wazuh Manager and Dashboard for event analysis and visualization. The evaluation combines threat hunting and File Integrity Monitoring (FIM) using several test scenarios: OWASP ZAP scanning, XSS, SQL injection (login-form testing and automated sqlmap attacks), and SSH brute force using hydra. The results show that Wazuh successfully detects XSS via rule 31105 (level 6) and sqlmap-based SQL injection via rule 31106 (level 6) because the attack patterns are clearly recorded in the web access logs. SSH brute force is strongly detected by rule 5763 (level 10), indicating repeated failed login attempts. In addition, FIM records file changes such as added and modified files (e.g., rules 554/550); however, it may generate noise when monitoring dynamic directories. The SQL injection attempt through the login form does not produce a specific SQL injection alert, suggesting limitations in log visibility/format and the need for decoder/ruleset tuning. Overall, Wazuh is effective for log-based security monitoring, while detection quality depends on log completeness, rule configuration, and FIM scope.

A B S T R A K

Penelitian ini mengimplementasikan prototipe Security Information and Event Management (SIEM) berbasis Wazuh untuk meningkatkan pemantauan keamanan pada aplikasi web. Arsitektur dibangun menggunakan dua VPS, yaitu server web sebagai sumber log dan Wazuh Agent, serta server monitoring sebagai Wazuh Manager dan Dashboard untuk analisis dan visualisasi event. Evaluasi dilakukan melalui skenario threat hunting dan File Integrity Monitoring (FIM), meliputi scanning menggunakan OWASP ZAP, serangan XSS, SQL injection (uji pada form login dan menggunakan sqlmap), serta brute force SSH menggunakan hydra. Hasil pengujian menunjukkan Wazuh mampu mendeteksi XSS melalui rule 31105 (level 6) dan SQL injection menggunakan sqlmap melalui rule 31106 (level 6) karena pola serangan tercatat jelas pada access log web server. Serangan brute force SSH terdeteksi kuat melalui rule 5763 (level 10) yang mengindikasikan percobaan login gagal berulang. Selain itu, FIM berhasil mencatat perubahan file seperti penambahan dan modifikasi (misalnya rule 554/550), namun berpotensi menimbulkan noise jika direktori yang dipantau bersifat dinamis. Percobaan SQL injection melalui form login tidak menghasilkan alert SQL injection yang spesifik, yang mengindikasikan keterbatasan visibilitas/format log dan kebutuhan tuning decoder/ruleset. Secara umum, Wazuh efektif untuk monitoring keamanan berbasis log, tetapi kualitas deteksi sangat dipengaruhi oleh kelengkapan sumber log, konfigurasi rule, dan pengaturan FIM.

Manuscript received Dec 23, 2024; revised Nov 24, 2025. accepted Dec 26, 2025 Date of publication Dec 31, 2025. International Journal, JITSI : Jurnal Ilmiah Teknologi Sistem Informasi licensed under a Creative Commons Attribution-Share Alike 4.0 International License

Keywords / Kata Kunci — ***SIEM; Wazuh; threat huntin; file integrity monitoring (FIM)***

CORRESPONDING AUTHOR

Nurmi Hidayasari
Departemen of Informatics Engineering, Politeknik Negeri Bengkalis, 28712, Indonesia
Email: nurmihidayasari[at]polbeng.ac.id

1. PENDAHULUAN

Aplikasi web saat ini menjadi komponen utama layanan digital dan sekaligus target yang sering diserang karena terekspos ke internet, memiliki banyak titik masukan (input), serta bergantung pada konfigurasi server dan komponen pihak ketiga. Risiko yang biasa dapat terjadi pada aplikasi web, seperti injection, kelemahan autentikasi, malware hingga teknik advanced persistent threat (APT) yang menyasar sistem secara berkelanjutan[1][2]. Dalam praktiknya, banyak pengelola sistem masih bersifat reaktif, artinya investigasi baru dilakukan setelah terjadi gangguan, sementara data log tersebar di berbagai sumber (web server, sistem operasi, aplikasi) sehingga sulit dikumpulkan secara cepat.

Bahkan saat ini banyak instansi atau organisasi yang belum aware terhadap kebutuhan sistem deteksi dini maupun sistem yang bisa memantau kejadian pada aplikasi dan jaringan yang mereka gunakan. Sehingga sering terjadi sistem yang tiba-tiba diserang atau data yang dicuri tanpa diketahui oleh pemilik sistem dan tidak ada persiapan yang dirancang apabila ada kejadian tidak sah yang terjadi. Hal itu berdampak tidak baik bagi instansi atau organisasi.

Salah satu pendekatan yang bisa digunakan adalah implementasi sistem Security Information and Event Management (SIEM). NIST menempatkan SIEM sebagai bagian dari pendekatan centralized logging yang melibatkan komponen analisis log dan penyimpanan log untuk mendukung pemantauan dan investigasi[3]. SIEM berfungsi untuk mengumpulkan, menganalisis, dan memvisualisasikan log keamanan dari berbagai perangkat dalam jaringan. SIEM merupakan sistem monitoring yang dapat mendeteksi serangan dan respon suatu sistem keamanan melalui analisis log dari berbagai event yang berasal dari sumber data secara realtime[4][5][6]. Namun, solusi SIEM komersial umumnya memerlukan biaya tinggi, sehingga belum banyak digunakan di lingkungan akademik atau instansi dengan sumber daya terbatas.

Sebagian besar solusi SIEM komersial seperti Splunk, IBM QRadar, dan ArcSight memiliki biaya lisensi yang tinggi dan kompleksitas implementasi yang cukup berat, sehingga tidak selalu cocok diterapkan di institusi pendidikan atau organisasi skala kecil–menengah. Dalam konteks ini, solusi berbasis sumber terbuka (open-source) seperti Wazuh menjadi alternatif yang layak, karena mampu menyediakan sebagian besar fungsi SIEM dengan biaya yang rendah dan fleksibilitas tinggi.

Wazuh menyediakan fitur SIEM untuk pengumpulan dan pemantauan log keamanan sekaligus berperan sebagai sistem deteksi intrusi berbasis host (endpoint) yang mendukung analisis log, pemeriksaan integritas file, pemantauan registri Windows, deteksi rootkit, peringatan berbasis waktu, respons aktif, serta integrasi dengan Elastic Stack untuk visualisasi dan analisis data; meskipun relatif ramah bagi organisasi dengan sumber daya terbatas, Wazuh tetap memiliki tantangan pada kompleksitas konfigurasi dan pemeliharaan, namun telah terbukti efektif untuk pemantauan keamanan real-time pada lingkungan cloud (AWS EC2) dengan ratusan alert dalam periode pemantauan 24 jam[7][8][9][10]. Wazuh efektif mendeteksi dan merespons ancaman potensial, seperti SQL Injection, brute force dan Dos Attack. Visualisasi log memberikan manfaat dalam hal efisiensi dan efektivitas dalam menangani insiden keamanan[11].

Pada sisi host, pemantauan integritas file (File Integrity Monitoring/FIM) penting untuk mendeteksi perubahan tidak sah pada file atau direktori penting (misalnya direktori aplikasi web) dengan membandingkan baseline dan memunculkan peringatan saat terjadi penambahan, modifikasi, atau penghapusan file[12]. Selain itu, threat hunting mendorong pendekatan proaktif dengan menelusuri indikator ancaman dari data yang dikumpulkan (log/event) untuk menemukan aktivitas berbahaya yang luput dari mekanisme otomatis.

Sejumlah penelitian terdahulu telah mengkaji bahwa Wazuh dapat dimanfaatkan sebagai platform monitoring atau SIEM berbasis open-source. Namun, fokus pengujian dan pendekatan deteksinya masih beragam. Misalnya, Pratama, dkk. membangun sistem monitoring menggunakan Wazuh untuk mendeteksi serangan DoS dengan memanfaatkan Suricata sebagai sumber deteksi, namun dari lima percobaan hanya dua yang terdeteksi[7]. Haryanto dan Chandra pengujian berfokus pada pemantauan integritas file di lingkungan BTSI UKSW dan menunjukkan bahwa Wazuh mampu mengidentifikasi kejadian mencurigakan terhadap file serta memberikan notifikasi kepada administrator[13].

Shafiyyah mengimplementasikan Wazuh sebagai server monitoring dengan agen pada server jaringan laboratorium dan melakukan simulasi serangan seperti DoS, hasilnya Wazuh dapat mendeteksi serangan, tetapi tidak efektif ketika DoS dilakukan pada skala kecil dengan intensitas rendah, dengan jumlah rata-rata koneksi di bawah 4000 koneksi http[14]. Aditya, dkk, mengimplementasikan SIEM menggunakan Wazuh untuk monitoring keamanan server. Hasilnya, Wazuh dapat mendeteksi serangan pada web server namun umumnya pada tingkat

kerentanan rendah; DDoS terdeteksi, sedangkan SQL injection tidak terdeteksi, sehingga disarankan pengujian dengan variasi serangan lain seperti DoS dan brute force[15].

Berdasarkan penelitian-penelitian tersebut, masih terbatas kajian yang secara spesifik memantau respons deteksi Wazuh pada serangan aplikasi web melalui threat hunting berbasis log web server, sekaligus mengombinasikannya dengan File Integrity Monitoring (FIM) dalam satu rangkaian pengujian. Penelitian sebelumnya lebih dominan mengevaluasi serangan jaringan seperti DoS[7][14], atau berfokus pada integritas file dan notifikasi[13], sementara aspek serangan aplikasi web (misalnya XSS, SQL injection), aktivitas pemindaian dengan tools seperti OWASP ZAP serta brute force dan bagaimana Wazuh merespons indikatornya melalui log akses web belum banyak diulas secara terpadu. Oleh karena itu, penelitian ini mengimplementasikan prototipe SIEM berbasis Wazuh pada aplikasi web dan menguji respons threat hunting serta FIM pada skenario OWASP ZAP, XSS, SQL injection, dan brute force, dengan keluaran utama berupa event/alert beserta informasi pendukung seperti waktu kejadian dan tingkat ancaman. Target atau Wazuh Agen yang akan digunakan Adalah website Jurusan TI Polbeng yang disalin dan dihosting.

Peneliti berharap prototipe yang dikembangkan ini dapat memberikan solusi SIEM yang terjangkau dan dapat diterapkan di lingkungan Pendidikan, layanan desa, maupun organisasi kecil hingga menengah. Penelitian ini diharapkan dapat memberikan kontribusi bagi pengembangan keamanan sistem informasi di lingkungan pendidikan dan referensi tambahan yang berguna bagi pengembang dan administrator jaringan dalam membangun sistem keamanan yang baik.

2. METODOLOGI PENELITIAN

Metode Penelitian yang digunakan dalam penelitian ini yakni Research and Development (R&D). Metode ini memungkinkan peneliti untuk merancang, mengembangkan serta mengevaluasi prototipe. Selain itu, pendekatan R&D dipilih karena memungkinkan peneliti untuk tidak hanya menghasilkan inovasi perangkat, tetapi juga melakukan iterasi perancangan hingga prototipe memenuhi kriteria keamanan, fleksibilitas, dan kemudahan penggunaan. Metode ini juga mendukung integrasi Fuzzy Sugeno sebagai mekanisme pengambilan keputusan pada sistem, sehingga status pintu dapat ditentukan secara adaptif sesuai kondisi yang dapat diukur.

Metodologi penelitian ini berfokus pada implementasi prototipe SIEM berbasis Wazuh dan pengujian respons deteksi melalui *threat hunting* berbasis log serta *File Integrity Monitoring* (FIM) pada aplikasi web. Tahapan penelitian secara keseluruhan dapat dilihat pada Gambar 1.

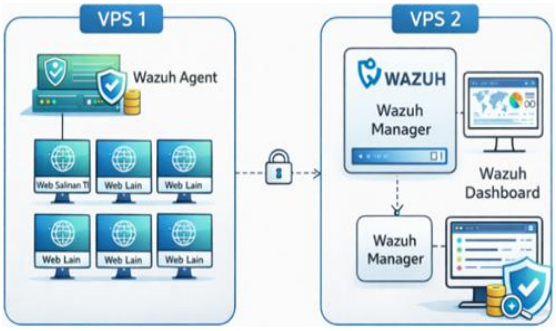
GAMBAR 1. Tahapan Penelitian[16]

2.1. Planning

Pada tahap ini ditetapkan tujuan dan ruang lingkup penelitian, yaitu mengimplementasikan SIEM berbasis Wazuh untuk memantau keamanan aplikasi web melalui *threat hunting* dan *File Integrity Monitoring* (FIM). Peneliti menentukan kebutuhan lingkungan uji (server terpusat untuk Wazuh Manager dan server aplikasi web untuk Wazuh Agent), menyiapkan perangkat dan akses yang diperlukan, serta menyusun rencana skenario pengujian yang akan digunakan, meliputi *scanning* dengan OWASP ZAP, XSS, SQL injection, brute force, dan skenario perubahan file untuk uji coba FIM. Wazuh Agen yang digunakan sebagai target pengujian adalah salinan Web Jurusan TI Polbeng yang dipublik.

2.2. Design

Pada tahap desain, arsitektur sistem dirancang dengan model manager–agent, termasuk alur pengumpulan log dari server aplikasi web menuju server terpusat dan mekanisme penampilan alert pada dashboard. Peneliti menetapkan sumber log yang dikumpulkan (misalnya access log web server) dan menentukan direktori aplikasi yang dipantau oleh FIM, termasuk parameter event yang diamati (added/modified/deleted) serta pengecualian direktori/file bila diperlukan untuk mengurangi *noise*. Selain itu, rancangan prosedur pengujian disusun, seperti endpoint target, parameter serangan, dan indikator keberhasilan berupa munculnya event/alert pada Wazuh.



GAMBAR 2. Rancangan Arsitektur SIEM Wazuh

Pada Gambar 2. Dapat dilihat arsitektur SIEM berbasis Wazuh yang digunakan dalam penelitian ini, dengan memanfaatkan dua buah VPS. VPS 1 berperan sebagai *endpoint* yang dipasangi Wazuh Agent dan menampung beberapa layanan website, termasuk website target yaitu web salinan Jurusan TI, sehingga aktivitas serta log web server dapat dikumpulkan untuk kebutuhan pemantauan. Data log yang dikirimkan dari VPS 1 mencakup informasi akses web (misalnya URL, metode, dan waktu akses) serta event perubahan file pada direktori aplikasi melalui File Integrity Monitoring (FIM). Selanjutnya, VPS 2 berfungsi sebagai Wazuh Manager sekaligus dashboard monitoring yang menerima data dari agent, melakukan pemrosesan berdasarkan *decoder* dan *rules*, lalu menampilkan event serta *alert* keamanan secara terpusat. Dengan pemisahan peran ini, proses pemantauan menjadi lebih terstruktur karena pengumpulan data dilakukan pada sisi server aplikasi, sedangkan analisis dan visualisasi dilakukan pada server monitoring.

2.3. Deployment

Pada tahap implementasi, Wazuh Manager dipasang dan dikonfigurasi pada server terpusat sebagai pusat pemrosesan log dan pengelolaan alert, kemudian Wazuh Agent diinstal pada server aplikasi web dan diverifikasi konektivitasnya. Selanjutnya dilakukan konfigurasi pengumpulan log web server agar dapat diproses oleh Wazuh (decoder dan rules berjalan) serta aktivasi FIM pada direktori aplikasi web. Setelah sistem siap, seluruh skenario pengujian dijalankan, serta uji perubahan file untuk menghasilkan event FIM, sambil memastikan data *event* dan *alert* tercatat pada dashboard.

2.4. Evaluation

Tahap evaluasi dilakukan dengan mengamati dan merekap respons deteksi Wazuh untuk setiap skenario pengujian, termasuk informasi waktu kejadian, deskripsi alert, tingkat ancaman (rule level), sumber log, dan indikator payload pada URL/log. Peneliti memverifikasi bahwa FIM menghasilkan event yang sesuai ketika file ditambah atau dimodifikasi, serta menilai apakah skenario *threat hunting* menghasilkan alert yang relevan berdasarkan log web server. Hasil evaluasi kemudian akan dirangkum untuk mendukung pembahasan, diikuti penarikan kesimpulan mengenai kemampuan Wazuh dalam meningkatkan visibilitas dan mendukung deteksi dini ancaman pada aplikasi web

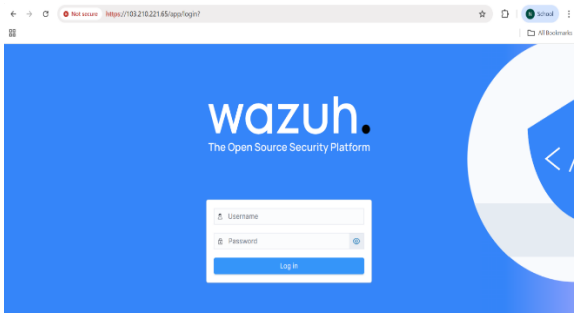
3. HASIL DAN PEMBAHASAN

3.1. Hasil Implementasi Arsitektur SIEM Wazuh

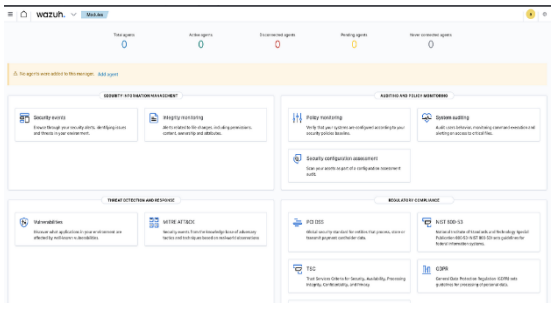
Implementasi sistem dilakukan menggunakan dua VPS, yaitu VPS 1 sebagai server aplikasi web yang dipasangi Wazuh Agent dan menampung beberapa website termasuk target (web salinan Jurusan TI), serta VPS 2 sebagai server terpusat yang menjalankan Wazuh Manager dan dashboard monitoring. Alur data menunjukkan bahwa event dan log dari VPS 1 terkirim ke VPS 2 dan dapat ditampilkan pada dashboard dalam bentuk daftar event serta visualisasi. Pda tahap ini dilakukan dengan menginstal dan mengonfigurasi Wazuh Manager, Wazuh Agent, serta dashboard visualisasi.

3.1.1. Konfigurasi Wazuh Manager

Wazuh manager yang berfungsi sebagai sistem monitoring dipasang pada VPS yang disewa, dengan spesifikasi: OS: Ubuntu 22.04 LTS, CPU: 2 core, RAM: 4 GB, Storage: 20 GB. Wazuh manager yang sudah di-*install*, dapat diakses pada browser dengan memasukkan URL: *https://103.210.221.65*. Gambar tampilan awal Wazuh Manager dapat dilihat pada Gambar 3. Dan Gambar 4.



GAMBAR 3. Tampilan Awal Wazuh Manager



GAMBAR 4. Halaman Utama Wazuh Manager

3.1.2. Konfigurasi Wazuh Agent

Tahap ini adalah proses konfigurasi wazuh agen, yang merupakan bagian dari wazuh yang dipasang pada endpoint. Pada penelitian ini Wazuh Agent adalah aplikasi website Jurusan Teknik Informatika (TI) yang dibuat salinannya dan dimasukkan pada VPS lain, dengan domain yang terdaftar *www.wazuhagenti.id*. Sebelum

menghubungkan domain dengan VPS Agen, dilakukan konfigurasi aapanel ada VPS. Gunakan perintah di bawah ini untuk menginstall aapanel secara otomatis, perintah ini didapatkan dari website resmi nya *https://www.aapanel.com/new/download.html*.

GAMBAR 5. Perintah *Install* aapanel

Setelah proses *install* sesuai dengan Gambar 5. selesai, maka akan diberikan kredensial aapanel, yang dapat digunakan untuk masuk ke dashboardnya. Setelah proses konfigurasi dan pemasangan selesai, maka akan muncul halaman utama dari aapanel yang digunakan untuk konfigurasi website Wazuh Agen. Selanjutnya, proses penambahan website *www.wazuhagenti.id* pada aapanel tersebut. Dapat dilakukan dengan memilih menu Website dan Add Site. Pada Gambar 6. dapat dilihat web salinan TI yang dimaksud sudah berhasil ditambahkan.

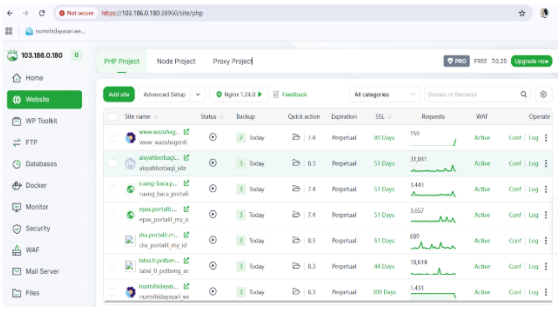
3.1.3. Setting FIM dan Threat Hunting

Setelah memastikan salinan web Jurusan TI dapat diakses, langkah selanjutnya adalah instalasi wazuh agent di Wazuh Manager. Konfigurasi dasar wazuh manager, dengan mangakses file konfigurasi utama yang terletak pada: /var/ossec/etc/ossec.conf. Kemudian tambahkan *script* di bawah ini pada file konfigurasi tersebut, *script* ini dapat disesuaikan.

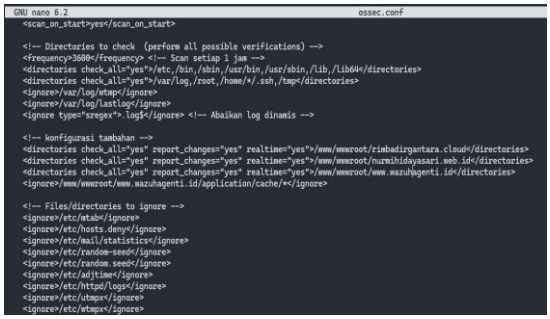
Script ossec.conf

```
<directories check_all="yes" report_changes="yes"
  realtime="yes">/www/wwwroot/wazuhagenti.id</directories>
```

Selanjutnya dapat melakukan konfigurasi FIM, prosesnya dapat dilihat pada Gambar 7.



GAMBAR 6. Daftar Web pada VPS Wazuh Agen



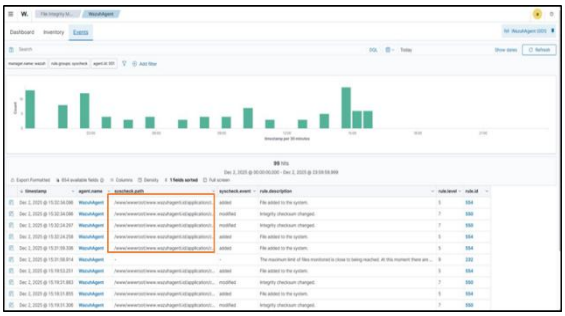
GAMBAR 7. Script Konfigurasi File Integrity Monitoring

Tag **<ignore>** bisa ditambahkan untuk mengecualikan direktori yang dianggap tidak perlu dipantau. Selanjutnya, dapat dilakukan konfigurasi **Threat Hunting** dengan menggunakan perintah berikut:

Script Threat Hunting

```
<localfile>
  <log_format>apache</log_format>
  <location>/www/wwwlogs/wazuhagenti.id</location>
</localfile>
```

GAMBAR 8. Konfigurasi Threat Hunting



GAMBAR 9. Tampilan Log Pengujian FIM

3.2. Hasil Pengujian FIM

Pengujian FIM, dilakukan dengan memindahkan beberapa file penting yang berada pada direktori utama sistem. Jika pengujian berhasil ditandai adanya *event* baru pada direktori domain di halaman /app/file- integrity-monitoring

Pada Gambar 9. dapat dilihat sistem Wazuh berhasil mendeteksi seluruh aktivitas perubahan file yang terjadi pada direktori aplikasi web yang dimonitor, yaitu pada path: */www/wwwroot/www.wazuhagenti.id/application/*. Berdasarkan hasil visualisasi di dashboard Wazuh – Events, terlihat bahwa setiap perubahan file direkam dan dikirimkan ke Wazuh Manager sebagai sebuah event dengan informasi detail, meliputi jenis perubahan (syscheck_event), deskripsi aturan (rule_description), tingkat keparahan (rule_level), serta waktu kejadian (timestamp).

Dari hasil pengujian, Wazuh menghasilkan beberapa jenis alert, antara lain:

- Event "added"
  Menunjukkan bahwa terdapat file baru yang ditambahkan ke dalam direktori yang dipantau. Hal ini tercermin pada rule ID 554 dengan deskripsi “File added to the system.”
- Event "modified"
  Mengindikasikan adanya perubahan isi atau atribut file yang sudah ada sebelumnya. Alert ini muncul melalui rule ID 550 dengan deskripsi “Integrity checksum changed.”, yang menandakan bahwa hash file telah berubah akibat modifikasi.
- Event terkait waktu pemindaian
  Rule ID 232 muncul saat Wazuh mendeteksi bahwa scan interval mendekati batas waktu maksimum, sebagai bagian dari mekanisme internal FIM untuk menjaga konsistensi pemantauan

Secara keseluruhan, grafik histogram pada dashboard menunjukkan distribusi jumlah event perubahan file sepanjang hari pengujian. Banyaknya event yang tercatat membuktikan bahwa modul FIM pada Wazuh berfungsi dengan baik dalam:

- Mengidentifikasi perubahan file secara real time.
- Mengirimkan alert berdasarkan tingkat keparahan.
- Merekam seluruh aktivitas pada direktori yang dipantau.

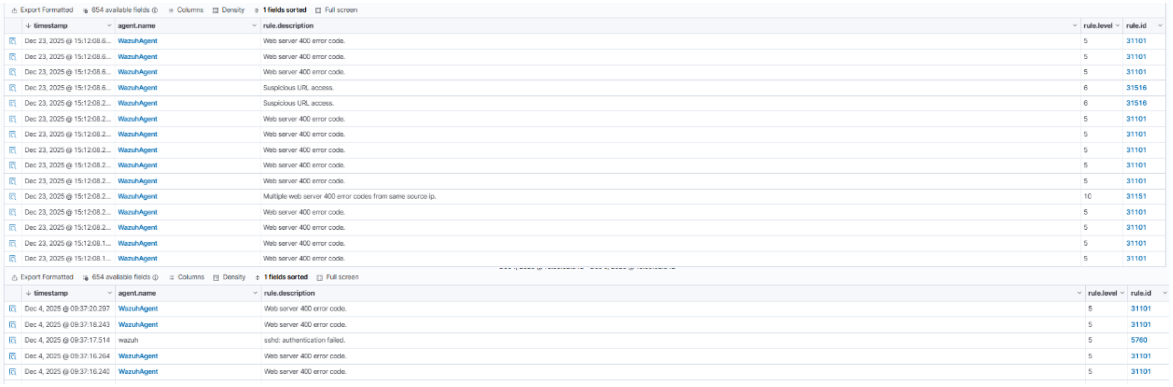
Hasil ini mengonfirmasi bahwa konfigurasi FIM di Wazuh berhasil berjalan sesuai rencana dan mampu memberikan visibilitas menyeluruh terhadap perubahan file yang berpotensi menjadi indikator ancaman keamanan.

3.3. Hasil Pengujian Threat Hunting

Pengujian *Threat Hunting* adalah kegiatan yang direncanakan untuk menguji apakah, log sudah terkumpul dengan benar (dari server, aplikasi, firewall, endpoint, dll), aturan deteksi/korlasi di SIEM (misal Wazuh) bekerja, analis mampu mencari (*hunting*) jejak serangan di *log*, bukan hanya menunggu *alert* otomatis.

3.3.1. Scanning OWASP ZAP

Pengujian dilakukan dengan memasukkan URL target pada *tools* ZAP dan melakukan *sanning* seperti biasa.



GAMBAR 10. Tampilan *Log* Serangan OWASP ZAP

Pada Gambar 10. Dapat dilihat log hasil skenario pemindaian menggunakan OWASP ZAP, Wazuh mencatat lonjakan event pada access log web berupa ‘Web server 400 error code’ (rule id 31101) dan ‘Suspicious URL access’ (rule id 31516). Selain itu, Wazuh memunculkan alert ‘Multiple web server 400 error codes from same source ip’ (rule id 31151, level 10) yang menunjukkan banyaknya permintaan tidak valid dari satu sumber dalam interval singkat. Pola ini konsisten dengan karakteristik aktivitas *scanning* otomatis yang dilakukan OWASP ZAP[17].

3.3.2. Serangan XSS

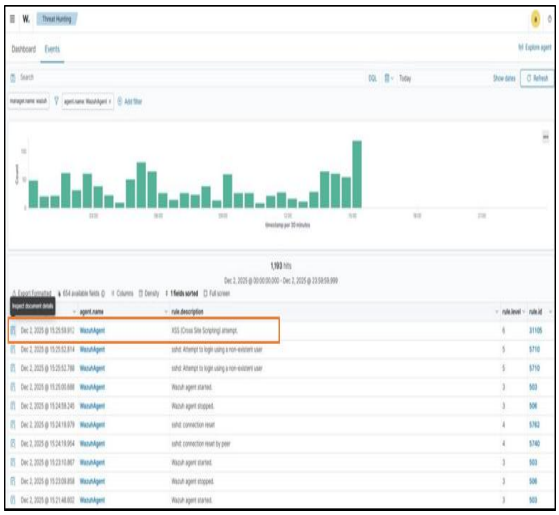
Pengujian dilakukan dengan serangan XSS sederhana, dengan memasukkan *script/ payload* pada URL web, yaitu <script>alert("testing")</script>.

GAMBAR 11. Serangan XSS Sederhana

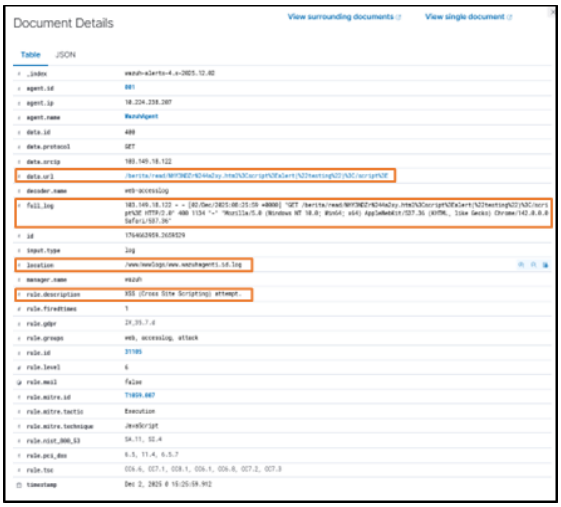
Pada Gambar 11, sistem menampilkan respons berupa pesan error "*The URL you submitted has disallowed characters.*", yang menunjukkan bahwa framework aplikasi menolak karakter berbahaya tersebut dan mencegah payload XSS dieksekusi pada sisi klien. Walaupun serangan tidak berhasil dijalankan pada browser, aktivitas ini tetap meninggalkan jejak di web server access log, sehingga dapat dimanfaatkan sebagai sumber data dalam proses Threat Hunting.

Gambar 12. memperlihatkan tampilan dashboard Wazuh pada menu Threat Hunting – Events, yang menunjukkan bahwa sistem berhasil mengidentifikasi aktivitas percobaan XSS dari agent. Kemudian pada daftar event terlihat sebuah alert dengan deskripsi: "XSS (Cross Site Scripting) attempt." Alert tersebut memiliki tingkat keparahan 6, yang menunjukkan bahwa percobaan XSS dikategorikan sebagai ancaman menengah. Hal ini membuktikan bahwa mekanisme *log analysis* dan *ruleset* Wazuh bekerja dengan baik dalam mendeteksi pola serangan berbasis *script injection* meskipun serangan tersebut tidak dieksekusi secara penuh pada sisi aplikasi.

Grafik histori event pada bagian atas dashboard juga memperlihatkan adanya peningkatan jumlah log pada waktu percobaan serangan, sehingga mendukung analisis pola aktivitas mencurigakan berbasis *timeline*.



GAMBAR 12. Tampilan Dashboard Wazuh pada Menu Threat Hunting-Events

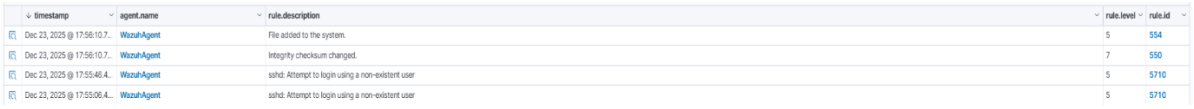


GAMBAR 13. Detail Serangan XSS yang Terdeteksi Hunting-Events

Gambar 13. menampilkan *Document Details* dari serangan XSS yang berhasil terdeteksi oleh Wazuh, di mana data.url memperlihatkan payload berbahaya yang dikirimkan ke endpoint */berita/read/...html%3Cscript%3Ealert(%22testing%22)%3C/script%3E*. Payload tersebut ditampilkan dalam format *URL-encoded* sehingga mempermudah proses penelusuran dan analisis. Event ini berasal dari parsing access log web server yang ditunjukkan oleh decoder.name: web-accesslog, sementara full_log merekam lengkap permintaan HTTP GET yang masuk, termasuk alamat IP sumber (103.149.18.122), *timestamp* kejadian, *user-agent* (browser), serta URL payload yang dikirimkan. Lokasi file log yang dianalisis juga tercatat pada location, yaitu /www/wwwlogs/www.wazuhagenti.id.log. Wazuh mengklasifikasikan kejadian ini sebagai percobaan XSS melalui rule.description "XSS (Cross Site Scripting) attempt." dengan rule.id 31105 dan rule.level 6 yang merupakan rule bawaan untuk deteksi XSS. Selain itu, Wazuh memetakan aktivitas ini ke framework MITRE ATT&CK dengan *tactic* Execution dan *technique* JavaScript, yang menunjukkan bahwa Wazuh tidak hanya mendeteksi input berbahaya, tetapi juga memberikan klasifikasi serangan, tingkat ancaman, serta pemetaan ke kerangka analisis lanjutan untuk mendukung proses investigasi.

3.3.3. Serangan SQL Injection

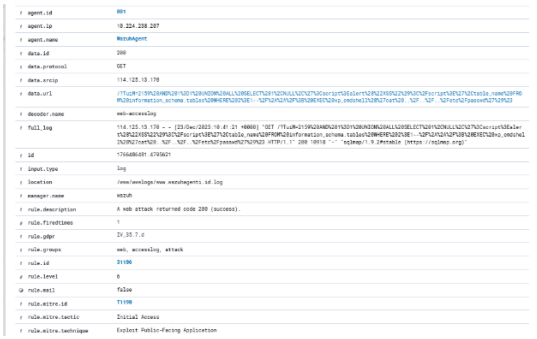
Pengujian ini dilakukan dengan dua metode, yaitu percobaan SQL Injection pada form login dan menggunakan *tools* sqlmap.



timestamp	agent.name	rule.description	rule.level	rule.id
Dec 23, 2025 @ 17:56:10.7...	WazuhAgent	File added to the system.	5	554
Dec 23, 2025 @ 17:56:10.7...	WazuhAgent	Integrity checksum changed.	7	550
Dec 23, 2025 @ 17:55:46.4...	WazuhAgent	sshd: Attempt to login using a non-existent user	5	5710
Dec 23, 2025 @ 17:55:06.4...	WazuhAgent	sshd: Attempt to login using a non-existent user	5	5710

GAMBAR 14. Tampilan *Log* Serangan SQL Injection pada Form Login

Pada Gambar 14. dapat dilihat event yang muncul tidak menunjukkan SQL injection pada form login, tetapi menunjukkan dua hal lain yang terjadi pada waktu pengujian, yaitu aktivitas FIM dan percobaan login SSH yang tidak sah. Pada saat pengujian input pada form *login* (termasuk percobaan SQL injection), Wazuh mencatat event FIM berupa penambahan dan perubahan file (rule id 554 dan 550). Event ini mengindikasikan adanya aktivitas pembaruan file aplikasi seperti *session/cache* akibat proses autentikasi. Selain itu, tercatat event terpisah dari layanan SSH (rule id 5710) berupa percobaan login menggunakan user tidak valid, yang tidak berkaitan langsung dengan proses login web.

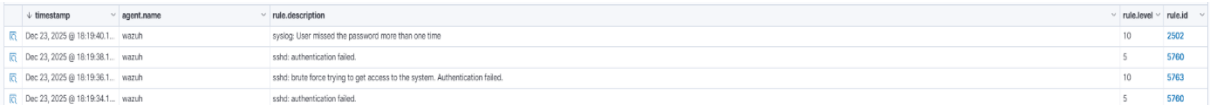


GAMBAR 15. Detail Serangan SQL Injection dengan *Tools* sqlmap

Pada Gambar 15. dapat dilihat Wazuh mendeteksi serangan berbasis access log web (decoder web-accesslog) dengan rule id 31106 (level 6) dan deskripsi '*A web attack returned code 200*'. Log menunjukkan request berisi pola UNION ALL SELECT yang mengarah ke enumerasi information_schema.tables serta user-agent sqlmap, sehingga mengindikasikan adanya pengujian SQL injection secara otomatis dari sumber IP tertentu menggunakan *tools* sqlmap.

3.3.4. Serangan Brute Force

Serangan dilakukan dengan menggunakan *tools* hydra. Meskipun serangan tidak berhasil mendapatkan data kredensial, namun Wazuh tetap mencatat log serangan yang dapat dilihat pada Gambar 16.

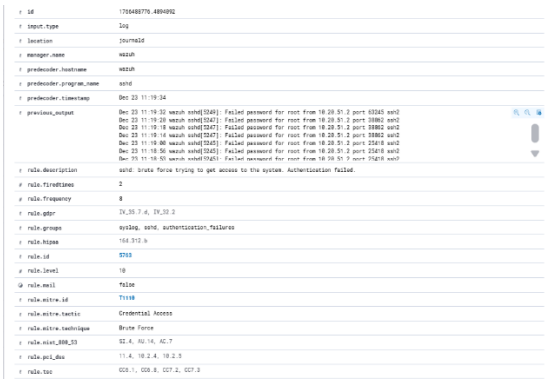


timestamp	agent.name	rule.description	rule.level	rule.id
Dec 23, 2025 @ 18:19:40.1...	wazuh	syslog: User missed the password more than one time	10	2502
Dec 23, 2025 @ 18:19:38.1...	wazuh	sshd: authentication failed.	5	5760
Dec 23, 2025 @ 18:19:36.1...	wazuh	sshd: brute force trying to get access to the system. Authentication failed.	10	5763
Dec 23, 2025 @ 18:19:34.1...	wazuh	sshd: authentication failed.	5	5760

GAMBAR 16. Tampilan *Log* Serangan Brute Force

Pada skenario brute force, Wazuh mencatat beberapa alert autentikasi SSH, termasuk 'sshd: authentication failed' (rule id 5760, level 5) dan alert agregasi 'sshd: brute force trying to get access to the system. Authentication failed' (rule id 5763, level 10). Selain itu muncul event 'User missed the password more than one time' (rule id 2502, level 10) yang menunjukkan terjadinya kesalahan kata sandi berulang dalam periode pengujian."

Pada Gambar17. menunjukkan detail log rule id 5763 yang mendeteksi percobaan autentikasi SSH berulang yang menargetkan akun root. Event tercatat pada decoder sshd dengan pesan 'Failed password for root' yang berasal dari sumber IP 10.20.51.2.



GAMBAR 17. Detail *Rule* Id 5763

Wazuh memunculkan alert 'sshd: brute force trying to get access to the system. Authentication failed' (rule id 5763) dengan tingkat ancaman level 10, serta menampilkan rangkaian percobaan login gagal secara berurutan pada bagian previous_output.

3.4 Pembahasan

Pada Tabel 1. menunjukkan bahwa Wazuh paling efektif mendeteksi serangan berbasis web ketika pola serangan tercatat jelas pada access log dan sesuai dengan decoder/rule bawaan (seperti, XSS dan sqlmap). Sementara itu, pada kasus input SQL injection melalui form login, alert yang muncul lebih dominan pada FIM/SSH sehingga perlu analisis lebih lanjut.

TABEL 1. Rekapitulasi Hasil Pengujian

Skenario	Metode/Tools	Sumber log	Decoder/ Modul	Rule utama (ID–Level)	Status deteksi	Indikator bukti utama
Scanning	OWASP ZAP	Access log web server	web-accesslog	(tergantung rule web scan)	Terdeteksi (indikasi *scanning*)	Lonjakan request otomatis pola akses tidak wajar, banyak respons error (4xx dalam waktu singkat
XSS	Payload XSS sederhana	/www/wwwlogs /www.wazuhagenti.id.log	web-accesslog	31105 – level 6	Terdeteksi	Payload tertangkap (URL-encoded), request GET terekam, mapping MITRE ATT&CK tercantum
SQL Injection (form login)	Input/payload di form login	Event yang muncul dominan bukan SQLi web	FIM + sshd	554/550 (FIM), 5710 (SSH)	Tidak spesifik SQLi (yang muncul: FIM/SSH)	Event FIM (file added/modified) dan even SSH invalid user; tidak muncul rule SQLi web pada skenario ini
SQL Injection (sqlmap)	sqlmap otomatis	Access log web server	web-accesslog	31106 – level 6	Terdeteksi	Request berisi pola SQL (mis. UNION), enumeras information_schema, user agent sqlmap
Brute force	Hydra (SSH)	Auth log/sshd event	sshd	5760–5, 5763–10, 2502–10	Terdeteksi	Rangkaian failed passwor berulang; detail previous_output menunjukkan percobaan berurutan
FIM	Create/modify file pada direktori dipantau	Direktori dipantau pada host web	syscheck/ FIM	554, 550, 232	Terdeteksi	File baru terdeteksi, checksum berubah, informasi interval scan tercatat

Pada Tabel 2. Dapat dilihat bahwa perbedaan keberhasilan deteksi Wazuh dipengaruhi oleh kelengkapan serta format log yang dipantau dan kesesuaian decoder/ruleset, misalnya payload SQL injection pada form login tidak memunculkan alert SQLi karena parameter tidak tercatat dengan baik, sementara aktivitas sqlmap lebih mudah terdeteksi karena meninggalkan pola request yang eksplisit. Tabel ini juga menegaskan bahwa FIM berpotensi menghasilkan informasi berlebih apabila direktori yang dipantau bersifat dinamis (cache/session), serta scanning dapat menimbulkan peningkatan event yang perlu dikendalikan melalui threshold.

TABEL 2. Hasil Temuan Pengujian

Temuan	Dugaan Penyebab Teknis	Dampak	Perbaikan yang Diusulkan
SQLi via form tidak terdeteksi	payload tidak match ruleset / tidak terekam jelas di access log / endpoint login berbeda log format	Blind spot deteksi	pastikan log_format tepat; cek endpoint login terekam; tambah custom rule untuk pola tertentu
FIM memunculkan event saat uji login	direktori dipantau berisi session/cache	false alarm/noise	tambah <ignore>; monitor folder kritikal saja
ZAP memicu 400 burst	request otomatis & invalid path	alert scanning jelas	jadikan baseline pola scanning; buat rule korelasi per IP

4. KESIMPULAN

Berdasarkan implementasi dan pengujian yang dilakukan, Wazuh mampu membantu pemantauan keamanan sistem melalui analisis log dan fitur File Integrity Monitoring (FIM). Hasil uji menunjukkan bahwa aktivitas scanning (OWASP ZAP) dapat dikenali dari pola permintaan yang meningkat dalam waktu singkat dan banyaknya respons error. Percobaan XSS terdeteksi melalui rule 31105 (level 6) dan serangan SQL injection menggunakan sqlmap terdeteksi melalui rule 31106 (level 6) karena pola serangan tercatat jelas pada access log. Pada pengujian brute force SSH, Wazuh mendeteksi percobaan login berulang dengan tingkat peringatan tinggi melalui rule 5763 (level 10). Namun, percobaan SQL injection melalui form login tidak menghasilkan alert SQL injection yang spesifik, yang menunjukkan bahwa deteksi sangat bergantung pada kelengkapan dan format log yang dipantau serta kesesuaian decoder dan ruleset. Pengujian FIM berhasil mencatat perubahan file (misalnya rule 554/550), tetapi dapat menimbulkan permintaan yang berlebih jika direktori yang dipantau bersifat dinamis seperti cache atau session. Secara umum, Wazuh efektif untuk monitoring, tetapi hasil deteksi akan lebih optimal jika dilakukan

penyesuaian sumber log, aturan deteksi, dan cakupan direktori FIM agar peringatan lebih akurat dan mudah dianalisis.

Penelitian ini berbeda dari penelitian sebelumnya karena pengujian Wazuh dilakukan pada beberapa skenario sekaligus, yaitu threat hunting dari log akses web (scanning menggunakan OWASP ZAP, XSS, dan SQL injection), brute force SSH, serta File Integrity Monitoring (FIM) pada direktori website dalam satu prototipe SIEM. Selain itu, web agent yang digunakan tidak berasal dari mesin virtual atau website rentan yang sengaja dirancang seperti DVWA, melainkan website nyata yang terdaftar pada domain dan dioperasikan pada VPS. Temuan yang didapat dari penelitian ini penting sebagai acuan penerapan SIEM berbasis open-source yang lebih terarah, sekaligus sebagai dasar tuning sumber log, aturan deteksi, dan cakupan FIM agar pemantauan lebih akurat dan mudah dianalisis.

Untuk penelitian selanjutnya, pengujian dapat diperluas dengan menambah variasi skenario dan payload agar hasil evaluasi lebih kuat dan mendekati kondisi lebih nyata. Misalnya, pengujian SQL injection tidak hanya menggunakan sqlmap, tetapi juga mencakup tipe error-based, boolean-based, dan time-based, termasuk serangan melalui metode POST pada form login agar terlihat apakah parameter dapat terekam dan terdeteksi dengan baik. Pengujian XSS juga dapat diperluas ke stored XSS dan DOM-based XSS, serta divalidasi pada beberapa endpoint berbeda. Selain itu, pengujian dapat dilakukan dalam kondisi trafik normal bercampur trafik serangan untuk mengukur kemampuan sistem membedakan aktivitas wajar dan anomali.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada Politeknik Negeri Bengkalis (Polbeng) sebagai pemberi dana melalui program Penelitian Internal Polbeng. Penulis juga menyampaikan apresiasi kepada Pusat Penelitian dan Pengabdian kepada Masyarakat (P3M) Polbeng atas dukungan administrasi dan fasilitasi kegiatan penelitian.

REFERENSI

[1] OWASP, "OWASP Top Ten 2021," 2021, [Online]. Available: https://owasp.org/www-chapter-minneapolis-st-paul/download/20211216_OWASP-MSP_OWASP_Top_Ten_2021.pdf

[2] W. Stallings, Network Security Essentials : Applications and Standards, Fourt. Pearson Education, Inc., publishing, 2011.

[3] K. Kent and M. Souppaya, "Guide to Computer Security Log Management," Nist Spec. Publ., 2006, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-92.pdf

[4] M. R. Kamal and M. A. Setiawan, "Deteksi Anomali dengan Security Information and Event Management (SIEM) Splunk pada Jaringan UII," Automata, vol. 2, no. 2, pp. 1–6, 2021.

[5] I. Kotenko and A. Chechulin, "Attack Modeling and Security Evaluation in SIEM Systems," Int. Trans. Syst. Sci. Appl., vol. 8, no. December, pp. 129–147, 2012.

[6] A. Vazão, L. Santos, M. B. Piedade, and C. Rabadão, "SIEM open source solutions: A comparative study," Iber. Conf. Inf. Syst. Technol. Cist., vol. 2019-June, 2019, doi: 10.23919/CISTI.2019.8760980.

[7] M. D. Pratama, F. Nova, and D. Prayama, "Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos," JITSI J. Ilm. Teknol. Sist. Inf., vol. 3, no. 1, pp. 1–7, 2022, doi: 10.30630/jitsi.3.1.59.

[8] A. Alhafidz and D. Haryanto, "Sistem Operasi Monitoring Server Menggunakan WAZUH," Karimah Tauhid, vol. 3, no. 10, pp. 11513–11517, 2024, doi: 10.30997/karimahtauhid.v3i10.15090.

[9] A. Alanda, H. . Mooduto, and R. Hadi, "Real-time Defense Against Cyber Threats: Analyzing Wazuh's Effectiveness in Server Monitoring," JITCE (Journal Inf. Technol. Comput. Eng., vol. 7, no. 2, pp. 56–62, 2023, doi: 10.25077/jitce.7.2.56-62.2023.

[10] M. R. T. Hidayat, N. Widiyasono, and R. Gunawan, "Optimasi Deteksi Malware Pada Siem Wazuh Melalui Integrasi Cyber Threat Intelligence Dengan Misp Dan Dfir-Iris," J. Inform. dan Tek. Elektro Terap., vol. 13, no. 1, 2025, doi: 10.23960/jitet.v13i1.5686.

[11] F. A. Saputra, T. R. Dharmawan, and A. Rustianto, "Implementasi Wazuh SIEM Untuk Manajemen Log Event di Pesantren Teknologi Informasi dan Komunikasi Jombang," J. Inform. Terpadu, vol. 6, no. 1, pp. 29–37, 2024, [Online]. Available: https://journal.nurulfikri.ac.id/index.php/JIT

[12] Wazuh, "Wazuh documentation: Ruleset," Wazuh Documentation. [Online]. Available: https://documentation.wazuh.com/current/user-manual/ruleset/index.html

[13] B. Haryanto and D. W. Chandra, "Implementasi Wazuh Integritas File untuk Perlindungan Keamanan Berdasarkan Aktivitas Log di BTSI UKSW," J. Indones. Manaj. Inform. dan Komun., vol. 5, no. 1, pp. 183–192, 2024, doi: 10.35870/jimik.v5i1.447.

[14] A. Shafiyyah, "Implementasi Sistem Keamanan Jaringan Di Psdku Universitas Lampung Waykanan Menggunakan Server Wazuh Untuk Deteksi Dan Respon Serangan Siber," Skripsi, 2024.

[15] R. Aditya, Y. Muhyidin, and D. Singasatia, "Implementasi Security Information And Event Management (SIEM) Untuk Monitoring Keamanan Server Menggunakan Wazuh," J. Ris. Sist. Inf. dan Tek. Inform., vol. 2, no. 5, pp. 137–144, 2024.

[16] M. Ramli and B. Soewito, "Monitoring dan Evaluasi Keamanan Jaringan Dengan Pendekatan System Information and Security Management (SIEM)," Fakt. Exacta, vol. 16, no. 1, pp. 50–56, 2023, doi: 10.30998/faktorexacta.v16i1.16534.

[17] OWASP, "ZAP Documentation." [Online]. Available: https://www.zaproxy.org/