JITET (Jurnal Informatika dan Teknik Elektro Terapan)
Vol. 13 No. 1, pISSN: 2303-0577 eISSN: 2830-7062 http://dx.doi.org/10.23960/jitet.v13i1.5686

OPTIMASI DETEKSI MALWARE PADA SIEM WAZUH MELALUI INTEGRASI CYBER THREAT INTELLIGENCE DENGAN MISP DAN DFIR-IRIS

Muhamad Ropi Taofiq Hidayat[*], Nur Widiyasono[2], Rohmat Gunawan[3]

[1,2,3] Program Studi Informatika, Universitas Siliwangi, Jl. Mugarsari Kel.Mugarsari Kec. Tamansari Kota Tasikmalaya 46191

Received: 13 Desember 2024
Accepted: 14 Januari 2025
Published: 20 Januari 2025

Keywords:
CTI;
DFIR-IRIS;
Malware Detection;
MISP,
SIEM Wazuh.

Corespondent Email:
207006003@student.unsil.ac.id

Abstrak. Ancaman siber terus meningkat seiring kemajuan teknologi informasi, dengan malware sebagai salah satu bentuk ancaman utama yang mengeksploitasi celah keamanan. Security Information and Event Management (SIEM) seperti WAZUH menjadi solusi efektif untuk mendeteksi dan merespons ancaman siber. Namun, performa deteksi malware oleh WAZUH standalone masih terbatas, dengan akurasi rendah (19,70%) dan recall rendah (16,26%). Penelitian ini bertujuan mengoptimalkan deteksi malware melalui integrasi WAZUH dengan Cyber Threat Intelligence (CTI) menggunakan Malware Information Sharing Platform (MISP) dan DFIR-IRIS. Hasil penelitian menunjukkan bahwa integrasi MISP meningkatkan presisi deteksi ancaman (96,3%), meskipun recall (62,9%) dan akurasi (63,1%) masih menunjukkan adanya ancaman yang terlewat. Penambahan DFIR-IRIS memungkinkan respons insiden secara real-time, meningkatkan efisiensi mitigasi. Kombinasi MISP dan DFIR-IRIS memperkuat kemampuan deteksi dan respons SIEM secara signifikan, memberikan solusi yang lebih efektif dan menyeluruh dalam menghadapi ancaman siberkngkat.

Abstract. *Cyber threatscontinue to increase as information technology advances, with malware asone of the main forms of threats that exploit security holes. SecurityInformation and Event Management (SIEM) such as WAZUH is an effective solution to detect and respond to cyber threats. to detect and respond to cyber threats. However, the malware detection performance by WAZUH standalone is still limited, with low accuracy (19.70%) and low recall (16.26%). (16.26%). This research aims to optimize malware detectiondetection through the integration of WAZUH with Cyber Threat Intelligence (CTI) using the Malware Information Sharing Platform (MISP) and DFIR-IRIS. Research results showed that the integration of MISP improved the precision of threat detection (96.3%), although recall (62.9%) and accuracy (63.1%) still showed that there were threats that were missed.that were missed. The addition of DFIR-IRIS enables real-time incident response, improving mitigation efficiency.response, improving mitigation efficiency. The combination of MISP and DFIR-IRIS significantly strengthens SIEM's detection and response capabilities, providing a more effective and comprehensivea more effective and comprehensive solution to cyber threat..*

1. PENDAHULUAN

Ancaman serangan siber semakin meningkat seiring dengan pesatnya perkembangan teknologi informasi. Salah satu bentuk ancaman siber adalah *malicious software* (*malware*), yakni kode berbahaya yang dirancang untuk mengeksploitasi celah keamanan pada sistem operasi, situs web, aplikasi, maupun jaringan. Celah keamanan (*vulnerability*) merupakan kesalahan dalam kode atau konfigurasi yang membuka peluang bagi pihak tidak bertanggung jawab untuk melakukan eksploitasi yang merugikan [1]. Dalam era serangan siber yang semakin kompleks, penting bagi organisasi untuk memiliki sistem keamanan yang dapat secara proaktif mendeteksi dan merespons ancaman tersebut.

Salah satu solusi efektif dalam keamanan siber adalah Security Information and Event Management (SIEM). SIEM memungkinkan pengumpulan, analisis, dan respons terhadap data keamanan dari berbagai sumber. Efektivitas SIEM dapat ditingkatkan dengan mengintegrasikan Cyber Threat Intelligence (CTI), yang menyediakan informasi mendalam dan terkini tentang ancaman siber, termasuk indikator kompromi (IOC), taktik, teknik, dan prosedur (TTP) yang digunakan oleh pelaku ancaman. Menurut [2]. CTI membantu memprediksi dan mencegah serangan dengan memanfaatkan data ancaman siber yang dianalisis secara strategis. CTI juga berperan dalam mengurangi false positives, sehingga fokus dapat diberikan pada ancaman yang benar-benar signifikan.

Salah satu platform CTI yang banyak digunakan adalah *Malware Information Sharing Platform* (MISP). MISP memungkinkan organisasi untuk menyimpan, berbagi, dan menerima informasi tentang malware, ancaman, dan kerentanan secara terstruktur. Aplikasi ini telah digunakan secara luas di berbagai sektor, seperti keuangan, perawatan kesehatan, telekomunikasi, pemerintah, dan teknologi, untuk memperkuat analisis ancaman. [3].

SIEM berfungsi sebagai sistem pemantauan dan analisis keamanan yang bekerja secara real-time atau melalui history log. Sistem ini mengumpulkan log dari berbagai perangkat, seperti firewall, server, dan endpoint, untuk mendeteksi pola aktivitas mencurigakan yang dapat mengindikasikan serangan malware [4]. Salah satu perangkat lunak SIEM berbasis open-source yang populer adalah Wazuh. Wazuh dapat berjalan pada sistem operasi Windows, Linux, dan macOS, dan berfungsi untuk mengumpulkan log dari host endpoint. Log tersebut kemudian dianalisis di server Wazuh untuk menentukan apakah aktivitas tersebut normal atau mencurigakan [5].

DFIR IRIS merupakan platform open-source yang mendukung investigasi dan respons insiden keamanan siber. DFIR IRIS memungkinkan pengelolaan data insiden secara terorganisir dengan menyediakan fitur pengumpulan, visualisasi, dan analisis data yang terstruktur. Platform ini memfasilitasi investigasi terperinci, termasuk pencatatan data log dan analisis insiden siber yang komprehensif, sehingga meningkatkan kemampuan deteksi dan respons terhadap ancaman [6]. Integrasi DFIR IRIS dengan SIEM seperti Wazuh memberikan solusi yang lebih kuat untuk mendeteksi ancaman secara real-time dan memastikan analisis data yang lebih baiks.

Penelitian sebelumnya telah menunjukkan efektivitas Wazuh dalam pertahanan real-time terhadap ancaman siber, termasuk deteksi integritas file dan malware [7], [8]. Namun, terdapat kelemahan dalam deteksi malware, di mana Wazuh memerlukan integrasi lebih lanjut dengan threat sharing dan manajemen log untuk meningkatkan kemampuan deteksinya. Penelitian lain menyoroti pentingnya *log management* serta *incident response* dan berbagi ancaman untuk memperluas cakupan deteksi serangan [9], [10], [11].

Berdasarkan permasalahan yang ada, penelitian ini bertujuan untuk mengoptimalkan penggunaan SIEM Wazuh dalam deteksi malware dengan menerapkan CTI. Pendekatan ini dilakukan melalui integrasi SIEM Wazuh dengan MISP dan DFIR IRIS untuk meningkatkan efektivitas deteksi ancaman dan respons keamanan siber.

2. TINJAUAN PUSTAKA

2.1 Isi **Security Information and Event Management (SIEM)**

Menurut [12] *SIEM* adalah sebuah aplikasi software yang berfungsi mengumpulkan informasi dan event terkait dengan keamanan sebuah jaringan (*WAN* maupun *LAN*). *SIEM* merupakan gabungan dari sebuah produk yang sebelumnya terpisah, yaitu *Security Information Management (SIM)* dan *Security Event Management (SIEM*). Produk dari *SIEM*, bisa berupa peralatan,software ataupun service dan dapat digunakan untuk membuat log data keamanan dan men-generate report sesuai dengan keadaan yang terjadi.

2.2 Cyber Threat Intelligence

Menurut [2] *Cyber Threat Intelligence* (CTI) dapat digambarkan sebagai proses pengambilan berbagai informasi tentang serangan siber, memahami bagaimana hal itu terjadi dan maknanya. Hal tersebut membantu untuk memprediksi dan mencegah serangan siber. Juga disebut sebagai intelijen ancaman, ini adalah metode untuk memperingatkan organisasi berdasarkan data yang diperoleh dari berbagai sumber yang telah dianalisis. Sumber umum pelanggaran data adalah malware, ancaman orang dalam, rekayasa sosial, kredensial yang lemah dan dicuri, kerentanan aplikasi, konfigurasi yang salah, dan kesalahan dari pengguna. Kecerdasan ancaman itu sendiri merupakan evolusi dalam proses pengamanan data, file, dan infrastruktur.

Menurut [13] kecanggihan dalam serangan menyebabkan kemajuan dalam pengumpulan informasi dari berbagai sumber untuk melindungi aset dalam suatu lingkungan. Pertukaran intelijen ancaman memberikan dukungan yang kuat untuk menghadapi serangan siber di era baru.

2.3 Wazuh

Wazuh adalah sebuah aplikasi *open source* yang secara resmi didefinisikan sebagai *host-based intrusion detection system* (HIDS) [5]. *Wazuh* memiliki keunggulan yakni memiliki fitur yang lebih banyak dibandingkan aplikasi serupa.

Wazuh mengintegrasikan berbagai fungsi yang sebelumnya terpisah menjadi satu agen dan platform arsitektur terpadu. Sistem ini menawarkan perlindungan keamanan untuk berbagai lingkungan, termasuk cloud publik, cloud privat, dan pusat data. Wazuh menyediakan analisis korelasi secara real-time dengan respons yang aktif, granular, dan mencakup tindakan perbaikan untuk menjaga keamanan endpoint. Selain itu, Wazuh mendukung analisis log, pemantauan integritas file, pemantauan registry Windows, deteksi rootkit, peringatan berbasis waktu, serta respons aktif yang berlangsung secara real-time[5].

2.4 Malware Information Sharing Platform (MISP)

MISP merupakan singkatan dari *Malware Information Sharing Platform*, adalah platform intelijen ancaman sumber terbuka, platform ini dikembangkan terutama oleh *CIRCL*, di antara kontributor lainnya [14] *MISP* digunakan untuk menyimpan, membagikan, dan menerima informasi tentang malware, ancaman, dan kerentanan secara terstruktur.

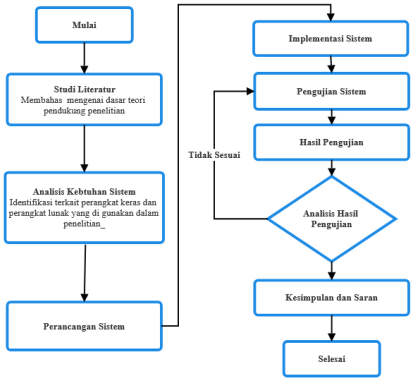
2.5 Digital Forensics and Incident Response Information Sharing (DFIR) IRIS

DFIR IRIS (*Digital Forensics and Incident Response Information Sharing*) adalah platform open-source yang dirancang untuk mendukung respons insiden keamanan siber melalui kolaborasi, analisis bukti digital, dan manajemen informasi ancaman [6]. Platform ini memfasilitasi pengumpulan, analisis, dan berbagi data insiden keamanan untuk meningkatkan efisiensi respons terhadap ancaman siber. DFIR IRIS menawarkan fitur manajemen insiden, pengelolaan bukti digital, dan integrasi dengan alat keamanan lainnya seperti MISP, sehingga memungkinkan respons lebih cepat dan terstruktur terhadap ancaman keamanan. DFIR IRIS juga mendukung dokumentasi untuk pelaporan dan kepatuhan terhadap regulas

3. METODE PENELITIAN

3.1 Tahapan Penelitian

Tahapan Penelitian ini secara keseluruhan disajikan menggunakan diagram alur penelitian. Diagram alur penelitian ini dapat di lihat pada gambar 3.1



Gambar 3. 1 Diagram Alur Penelitian

Berdasarkan *Gambar 3.1,* diagram alur penelitian merupakan prosedur untuk melakukan penelitian yang dimulai dari tahap studi literature tahap analisis kebutuhan sistem, tahap perancangan, tahap implemetasi, dan tahap analisis hasil pengujianStudi Literatur

Studi literatur merupakan tahapan untuk memenuhi kebutuhan data, melakukan eksplorasi konsep dan teori yang berkaitan dengan *SIEM*, log analisis dan CTI. Studi literatur yang dilakukan diperoleh dari berbagai sumber diantaranya yaitu jurnal, e-book, artikel dan lain sebagainya. Kajian studi literature menggunakan aplikasi *Harzing's Publish or Perish*. Indikator keberhasilan dari tahapan ini adalah memahami domain penelitan, mengdentifikasi peluang penelitian serta menetapkan tujuan dan pertanyaan penelitian, luaran dari tahapan ini adalah Bab I pendahuluan dan Bab II tinjauan Pustaka.

3.2 Analisis Kebutuhan Sistem

Analisis Kebutuhan sistem bertujuan untuk mengetahui elemen apa saja yang di butuhkan dalam melaksanakan penelitian ini untuk mendapatkan gambaran umum sistem. Kebutuhan sistem yang digunakan dalam penelitian ini terbagi menjadi dua jenis, yaitu kebutuhan *Software* dan *Hardware :*

Tabel 3. 1 Kebutuhan Hardware

Requirement	Hardware yang di gunakan
2 core Processor	Processor Intel Core I3-5005U
8 Gb Ram	10Gb Storage
50 gb storage	500Gb HDD

Tabel 3.1 merupakan kebutuhan perangkat keras yang dibutuhkan oleh sistem, dan juga hardware yang akan di gunakan pada penelitian.

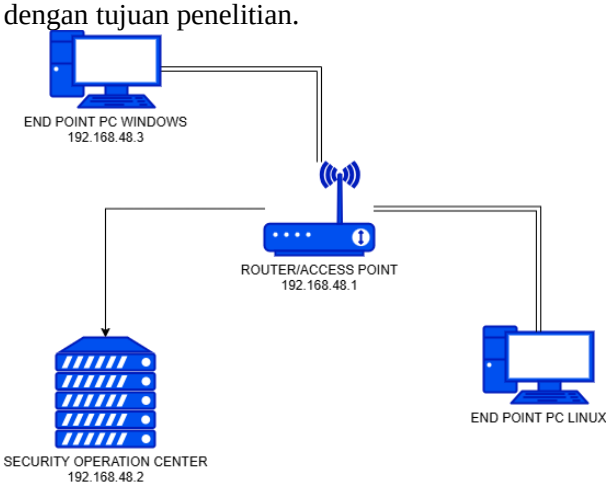
Tabel 3. 2 Kebutuhan Software

Operating System	Microsoft Windows 10 Pro, Linux: Ubuntu Desktop LTS (22.04),
Security Monitoring	*WAZUH* 4.9
Incident Response Mangement	DFIR IRIS
Threat Intelligence	*MISP* – Malware Information Sharing Platform

Tabel 3.2 merupakan kebutuhan perangkat lunak yang dibutuhkan oleh sistem, perangkat lunak apa saja yang akan di gunakan pada penelitian ini. Tabel ini mencakup empat kategori utama, yaitu Operating System yang mencakup Windows 10 Pro dan Linux Ubuntu 20.04, Security Monitoring menggunakan *WAZUH* 4.7x, Log Analytics dengan Graylog 6.03, dan Threat Intelligence menggunakan platform berbagi informasi malware (*MISP*).

3.3 Perancangan Sistem

Perancangan sistem dilakukan untuk membuat desain perencanaan arsitektur sistem yang akan dibangun dapat berjalan sesuai dengan tujuan penelitian.



Gambar 3. 2 Topologi Sistem

Gambar 3.2 merupakan rencana topologi sistem yang akan di bangun terdiri dari 1 buah ruter, 1 buah SOC dan 2 buah PC end point

dengan masing masing pc menggunakan sistem operasi windows 11 dan ubuntu desktop.

3.4 Implemetasi Sistem

Implementasi sistem merupakan proses penelitan, pada tahap ini hasil perancangan sistem di implementasikan sesuai dengan desain yang telah dibuat. Instalasi dan konfigurasi dilakukan sesuai referensi dokumentasi dari masing-masing software yang digunakan.

Gambar 3. 3 Gambaran Cara Kerja Sistem

Gambar 3.3 merupakan gambaran sistem yang akan di implentasikan. Sistem yang dibuat menggunakan os ubuntu desktop sebagai sistem operasinya, *WAZUH* sebagai *SIEM*, *DFIR-IRIS* sebagai *incident response* dan *MISP* sebagai *cyber threat intelligence sharing platform*. Ketiga aplikasi tersebut di integrasikan sehingga membuat sistem *threat hunting incident response* otomatis.

3.5 Pengujian Sistem

Pengujian sistem merupakan proses lanjutan dari implementasi sistem yang kemudian akan diuji sesuai skenario pengujian yang telah ditentukan. Pengujian yang akan dilakukan yaitu membandingkan *SIEM WAZUH* Standar dan *SIEM WAZUH* yang telah di integrasikan dengan *MISP* dan *DFIR-Iris*.

Tahapan pengujian dalam penelitian ini melibatkan beberapa skenario untuk menilai efektivitas deteksi *malware* melalui integrasi berbagai komponen. Pengujian akan menggunakan sampel *malware* dari repositori *github.com/ThatSINEWAVE*, yang dideploy pada host untuk memicu deteksi. Tiga skema akan diuji secara bertahap, yaitu deteksi malware oleh *WAZUH* secara mandiri, integrasi *WAZUH* dengan *MISP*, dan integrasi *WAZUH* dengan *MISP* serta DFIR-IRIS untuk *incident response*. Setiap skema ini akan dievaluasi berdasarkan beberapa metrik, yaitu True Positive (TP), False Positive (FP), False Negative (FN), dan True Negative (TN).

3.6 Analisis Hasil Pengujian

Berdasarkan hasil pengujian maka dilakukan evaluasi hasil pengujian dengan menghitung akurasi, presisi, recall dan F1-score untuk mengetahui keefektifan sistem. Rumus yang digunakan meliputi:

$$Accuracy = \frac{TP+TN}{TP+TN+FP\ FN} \text{`} \quad (1)$$

Akurasi mengukur seberapa sering sistem membuat prediksi yang benar, baik dalam mendeteksi kejadian positif (TP) maupun negatif (TN), dari semua prediksi yang dibuat.

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Presisi menunjukkan seberapa banyak dari semua yang dideteksi sebagai positif oleh sistem benar-benar positif, sehingga mencerminkan tingkat keakuratan deteksi positif.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Recall (True Positive Rate / Sensitivitas) mengukur seberapa baik sistem dapat menemukan semua kejadian positif yang benar-benar ada.

$$F1 - Score = 2\,.\frac{Precision.Recall}{Precision+Recall} \quad (4)$$

F1 Score adalah rata-rata harmonis dari presisi dan recall, memberikan gambaran seimbang tentang performa sistem terutama ketika terdapat ketidakseimbangan antara data positif dan negatif

Kesimpulan yang diambil dari hasil pengujian sistem dapat menjadi data untuk dijadikan sebagai kualitas hasil pengujian

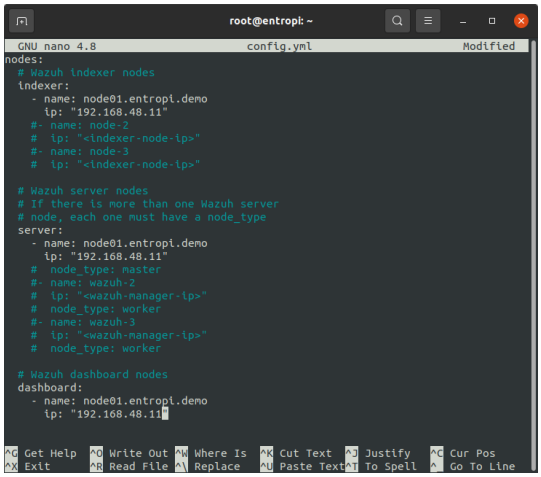
4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Implementasi sistem terbagi menjadi beberapa tahapan yaitu sebagai berikut :

4.1.1. Konfigurasi dan Instalasi *WAZUH*

Konfigurasi untuk instalasi *WAZUH* dilakukan pada file *config.yml* dan ditunjukan pada *Gambar 4.1* untuk mengedit file *config.yml* dapat dilakukan dengan menggunakan *vim* ataupun *nano.*

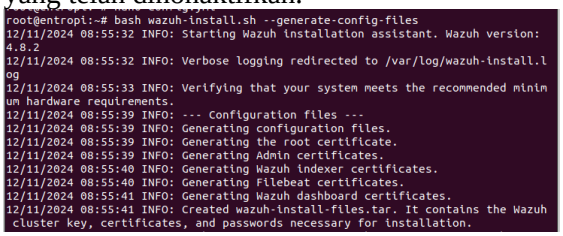


Gambar 4. 1 Konfigurasi *WAZUH*

Gambar 4.1 merupakan konfigurasi instalasi *WAZUH* pada gambar tersebut terdiri dari tiga bagian utama, yaitu pengaturan node untuk indexer, server, dan dashboard. Pada bagian indexer, hanya terdapat satu node yang aktif dengan nama *node01.entropi.demo* dan alamat IP 192.168.48.11. Node ini bertanggung jawab untuk mengindeks data yang diterima dari agen *WAZUH*. Terdapat juga beberapa entri tambahan untuk node indexer lain yang dinonaktifkan, kemungkinan sebagai persiapan untuk menambahkan lebih banyak node indexer di masa mendatang.

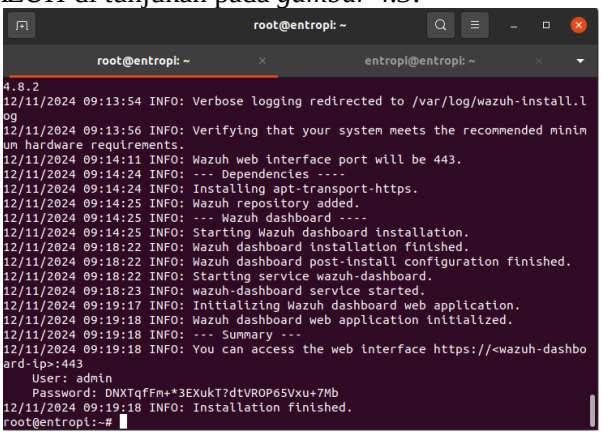
Bagian server *WAZUH* mengatur node server yang mengelola komunikasi dengan agen-agen *WAZUH* yang terpasang di sistem endpoint. Node *node01.entropi.demo*, dengan IP 192.168.48.11, ditetapkan sebagai server tipe master, yang berarti ini adalah node utama yang mengelola server *WAZUH*. Terdapat juga opsi untuk menambahkan lebih banyak node server dengan tipe worker, yang akan mendistribusikan beban kerja dari master server, tetapi entri ini dinonaktifkan.

Bagian terakhir adalah konfigurasi untuk dashboard *WAZUH*, yang mengelola antarmuka pengguna berbasis web untuk memantau sistem. Seperti bagian sebelumnya, node yang digunakan untuk dashboard adalah *node01.entropi.demo* dengan IP yang sama (192.168.48.11). Ini menunjukkan bahwa satu node server menangani semua fungsi penting, yaitu sebagai indexer, server utama (master), dan dashboard, yang mungkin cocok untuk lingkungan dengan jumlah agen yang tidak terlalu besar. Ada kemungkinan file ini bisa diperluas untuk mendukung infrastruktur yang lebih besar dengan menambahkan node lain yang telah dinonaktifkan.



Gambar 4. 2 Generate konfigurasi *WAZUH*

Tahap berikutnya yaitu instalasi *WAZUH* untuk melakukan instalasi dengan cara mengenerate config file dengan menggunakan bash *WAZUH-install.sh.* File tersebut akan mengenerate konfigurasi yang tadi telah di buat kemudian melakukan instalasi *WAZUH* sesuai konfigurasi yang telah di buat dan tunggu prosesnya hingga selesai proses instalasi *WAZUH* di tunjukan pada *gambar 4.3*.



Gambar 4. 3 Proses instalasi *WAZUH*

Gambar 4.3 menunjukan proses instalasi *WAZUH* telah selesai dan *WAZUH* dan *WAZUH* dapat di akses dengan menggunakan alamt ip host atau alamat ip yang digunakan pada saat konfigurasi.

Hasil dan Diskusi menyusun 60-70% dari naskah. Bagian ini adalah bagian utama dari artikel penelitian. Hasil harus meringkas atau menyoroti temuan daripada memberikan hasil dan analisis rinci. Berisi hasil yang diambil dari analisis data dan/atau hasil uji hipotesis dan hanya menyediakan data yang mendukung pembahasan. Bagian ini meliputi tabel dan grafik yang

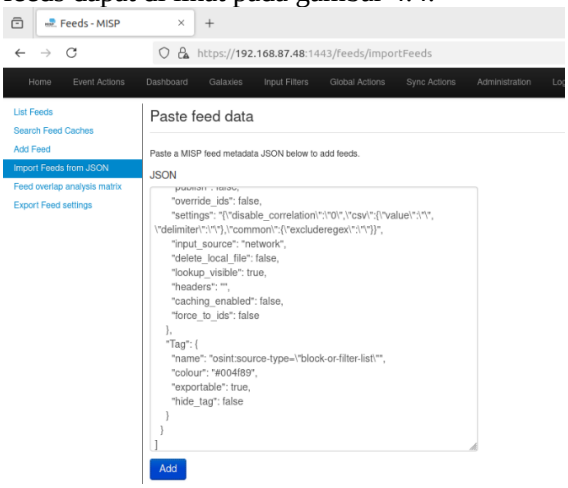
4.1.2. Instalasi *MISP*

Instalasi MISP dapat di lakukan dengan menggunakan docker, file docker yang di gunakan dapat langsung di clone dengan menggunakan git, https://github.com/MISP/MISP.git setelah

melakukan cloning terhadap repositori git kemudian edit file .env dan docker-compose.yml. konfigurasi yang di ubah adalah Base_URL dan juga port dari MISP agar tidak bentrok dengan port yang digunakan oleh WAZUH dashboard ubah menjadi 'https://alamat ip:1443'.

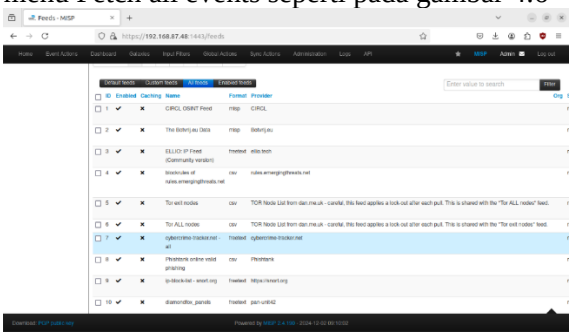
Instalasi MISP bisa dilakukan dengan cara memanggil docker compose build untuk membangun image dari MISP docker, kemudian untuk menjalankan docker tersebut bisa memanggil docker compose up.

Feeds pada *MISP* adalah sumber informasi ancaman (*threat intelligence*) yang diperoleh dari berbagai penyedia eksternal atau internal. Feeds digunakan untuk memperbarui basis data ancaman *MISP* secara otomatis sehingga pengguna dapat selalu mendapatkan informasi terkini yang relevan untuk mendeteksi, mencegah, dan merespons serangan siber. Penambahan feeds pada *MISP* dapat di lakukan dengan masuk ke menu action, feeds lalu pilih import feeds from json. Feeds yang digunakan adalah feeds dari repository *MISP* dalam folder app/files/feed-metadata/default.json. proses penambahan feeds dapat di lihat pada gambar 4.4.



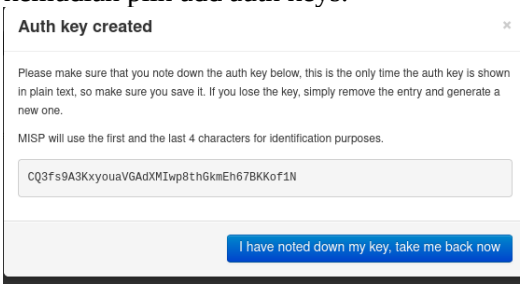
Gambar 4. 5 Penambahan Feeds pada MISP

Gambar 4.5 menunjukan data JSON telah diisi, dengan menekan tombol ***"Add"*** untuk menambahkan feed ke dalam *MISP*. Feed ini kemudian akan tersedia untuk diaktifkan dan disinkronisasi, memungkinkan *MISP* untuk mengimpor informasi ancaman terbaru dari feed tersebut. Proses ini penting dalam meningkatkan kemampuan organisasi untuk tetap up-to-date dengan ancaman siber terbaru. Proses berikutnya yaitu mengkatifkan feeds yang akan digunakan dengan masuk ke menu list event, kemudian pilih semua event dan aktifkan sinkronasi feeds dengan memilih menu Fetch all events seperti pada gambar 4.6



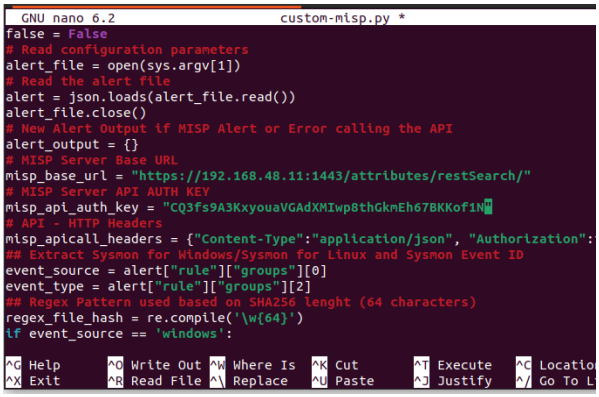
Gambar 4. 64 Feeds MISP

Integrasi *MISP* dengan *WAZUH* dapat di lakukan dengan menggunakan API milik mips yang dapat di akses dengan menggunakan auth keys. Auth keys tersebut berfungsi sebagai mekanisme otentikasi untuk mengakses API *MISP*. Auth keys tersebut dapat di buat pada menu list auth keys kemudian pilh add auth keys.



Gambar 4. 7 Auth key MISP

Gambar 4.7 Menunjukan bahwa auth key berhasil di buat. Tahapan berikutnya adalah menggunakan auth key tersebut untuk integrasi dengan *WAZUH*. Integrasi dapat dilakukan dengan menggunakan custom file custom integration. Custom-*MISP*.py yang di gunakan berasal dari repository github.com/ OpenSecureCo. *Script* tersebut berfungsi untuk menhubungkan *WAZUH* dengan *MISP*, script tersebut memungkinkan *WAZUH* untuk mengambil data serangan dari *MISP*. Konfigurasi dari file custom-*MISP*.py yang perlu di ubah hanya base url dan memasukan auth key yang telah di generate pada *MISP*. contoh konfigurasi nya dapat di lihat pada gambar 4.8

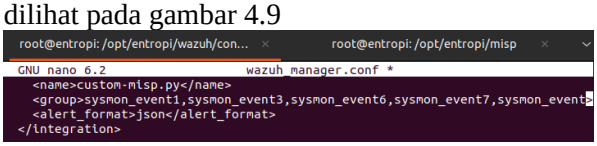


Gambar 4. 8 Konfigurasi custom-MISP

Gambar tersebut menunjukkan isi dari skrip custom-*MISP*.py yang digunakan untuk mengintegrasikan *WAZUH* dengan *MISP*. Skrip ini berfungsi untuk mengambil data ancaman dari *MISP* melalui API, memprosesnya, dan menyimpannya sebagai alert yang dapat digunakan oleh *WAZUH*. Pada bagian awal skrip, terdapat konfigurasi utama seperti *MISP* Base URL dan API Key yang digunakan untuk mengakses platform *MISP*. URL https://192.168.48.11:1443/attributes/restSearch/ menunjukkan endpoint API yang digunakan untuk mencari atribut ancaman di *MISP*.

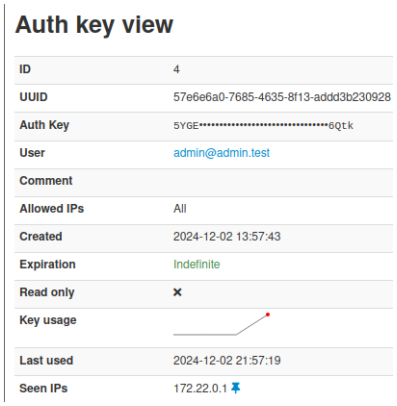
Selain itu, skrip ini mengatur HTTP Headers dengan format JSON untuk melakukan otentikasi API menggunakan API key (Cq3f...kof1N). Ada pula pengaturan terkait sumber data ancaman (event source) untuk mendeteksi log dari Windows/Sysmon berdasarkan pola hash SHA256. Skrip ini juga memanfaatkan regex (regular expressions) untuk mengenali atribut ancaman dengan panjang hash 64 karakter. Fungsi utamanya adalah menyesuaikan data dari *MISP* dengan kebutuhan analisis log di *WAZUH*, memungkinkan deteksi ancaman berbasis IOC secara otomatis dan terstruktur.

Script custom-*MISP*.py dipindahkan ke folder /var/ossec/integrations, agar *WAZUH* dapat menggunakan script tersebut berikan permission chown root:*WAZUH* chmod 750. Tujuan diberikan permission tersebut adalah untuk mengatur kepemilikan dan izin akses file atau direktori secara keselurhan. Tahap berikutnya memasukan konfigurasi pada blok integrasi pada file konfigurasi Ossec.conf. blok konfigurasi yang di tambahkan dapat dilihat pada gambar 4.9



Gambar 4. 9 Integrasi Pada WAZUH

Berdasarkan gambar blok konfigurasi integrasi <integration> pada *WAZUH*'s ossec.conf digunakan untuk mendefinisikan cara *WAZUH* menangani log tertentu dan menghubungkannya dengan skrip eksternal seperti custom-*MISP*.py. Bagian <name> menunjukkan nama integrasi, yaitu custom-*MISP*, yang akan menjalankan skrip khusus untuk mengirim data log ke sistem *MISP*. Parameter <group> menentukan jenis log atau event yang diproses, seperti Sysmon Event IDs (misalnya, sysmon_event1, sysmon_event3) dan syscheck, yang digunakan untuk mendeteksi aktivitas penting seperti pembuatan proses atau perubahan file. Selain itu, parameter <alert_format> menentukan bahwa data log akan dikirim dalam format JSON, yang mudah diolah oleh sistem eksternal. Setelah konfigurasi ini diterapkan, layanan *WAZUH* harus direstart agar perubahan aktif. Buka kembali menu list auth key pada *MISP* untuk melihat apakah integrasi *WAZUH* dan *MISP* sudah berhasil atau belum jika berhasil maka dapat dilihat pada gambar 4.10



Gambar 4. 50 Integrasi Berhasil

Berdasarkan gambar tersebut dapat dilihat pada seen Ips dan key usage bahwa *WAZUH* berhasil mengakses auth key milik *MISP*.

4.1.3. Instalasi dan Konfigurasi DFIR - IRIS

Proses konfigurasi dan integrasi *WAZUH* dengan *DFIR-Iris* dimulai dengan instalasi DFIR-Iris, untuk instalasi menggunakan docker. File docker yang digunakan berasal dari repositori github milik DFIR IRIS. Konfigurasi file docker-compose.yml dan file env yang di ubah hanyalah port dari iris.webapp dari 443 menjadi 8443 agar port tidak bertabrakan dengan *WAZUH* dashboard. Instalasi dilakukan dengan memanggil command docker compose build dan docker compose up, kemudian tunggu hingga proses instalasi selesai seperti yang di tunjukan pada gambar 4.11.

```
root@entropi:/opt/entropi/iris-web# docker compose up -d
WARN[0000] /opt/entropi/iris-web/docker-compose.yml: the attribute `version` is
obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 5/5
 ✔ Container iriswebapp_rabbitmq  Start...                          0.7s
 ✔ Container iriswebapp_db        Started                           0.7s
 ✔ Container iriswebapp_app       Started                           1.6s
 ✔ Container iriswebapp_nginx     Started                           2.8s
 ✔ Container iriswebapp_worker    Started                           2.7s
root@entropi:/opt/entropi/iris-web#
```

Gambar 4. 61 Instalasi DFIR-Iris

Proses instalasi telah selesai tahap berikutnya adalah integrasi DFIR-Iris dengan *WAZUH* agar dapat menampilkan alert dari *WAZUH*. Integrasi dilakukan dengan menambahakan custom rule python dengang menggunakan file custom-iris.py yang berasal dari repository github.com/nateuribe/*WAZUH*-IRIS-integration. Konfigurasi integrasi mengikuti bawaan dan hanya mengubah alert source nya menjadi url dari *WAZUH* seperti pada gambar 4.12.

```
 GNU nano 6.2                        custom-iris.py *
# Generate request
# Reference: https://docs.dfir-iris.org/_static/iris_api_reference_v2.0.1.html#
payload = json.dumps({
    "alert_title": alert_json.get("rule", {}).get("description", "No Descriptio
    "alert_description": alert_details,
    "alert_source": "Wazuh",
    "alert_source_ref": alert_json.get("id", "Unknown ID"),
    "alert_source_link": "https://192.168.48.11/app/wazuh",  # Replace with act
    "alert_severity_id": severity,
    "alert_status_id": 2,  # 'New' status
    "alert_source_event_time": alert_json.get("timestamp", "Unknown Timestamp"),
    "alert_note": "",
    "alert_tags": f"wazuh,{alert_json.get('agent', {}).get('name', 'N/A')}",
    "alert_customer_id": 1,  # '1' for default 'IrisInitialClient'
    "alert_source_content": alert_json  # raw log
})

# Send request to IRIS
response = requests.post(hook_url, data=payload, headers={"Authorization": "Bea
```

Gambar 4. 7 Script Integrasi DFIR-IRIS

Gambar tersebut menunjukkan potongan kode Python dalam file bernama custom-iris.py, yang dirancang untuk mengintegrasikan *WAZUH* dengan platform DFIR-IRIS melalui API. Kode ini digunakan untuk mengirimkan data peringatan (alert) yang dihasilkan oleh *WAZUH* ke DFIR-IRIS. Proses dimulai dengan pembuatan payload dalam format JSON, yang mencakup informasi seperti judul peringatan (alert_title), deskripsi (alert_description), sumber peringatan (alert_source), tingkat keparahan (alert_severity_id), waktu kejadian (alert_source_event_time), serta konten log (alert_source_content). Setelah payload dibuat, skrip mengirimkan data tersebut ke endpoint API IRIS menggunakan metode HTTP POST. Header permintaan mencakup token otentikasi yang diperlukan untuk mengakses API IRIS secara aman. Konfigurasi ini memungkinkan otomatisasi pengiriman peringatan dari *WAZUH* ke IRIS untuk dianalisis lebih lanjut, mempercepat respons terhadap insiden keamanan.

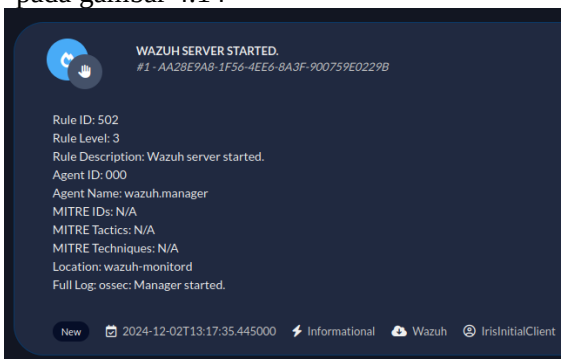
Script custom-*MISP*.py yang telah diedit dipindahkan ke folder /var/ossec/integrations, agar *WAZUH* dapat menggunakan script tersebut berikan permission chown root:*WAZUH* chmod 750. Tujuan diberikan permission tersebut adalah untuk mengatur kepemilikan dan izin akses file atau direktori secara keselurhan. Tahap berikutnya memasukan konfigurasi pada blok integrasi pada file konfigurasi Ossec.conf. blok konfigurasi yang di tambahkan dapat dilihat pada gambar 4.13.

```
<integration>
  <name>custom-iris.py</name>
  <hook_url>https://iriswebapp_nginx:8443/alerts/add</hook_url>
  <level>6</level>
  <group>ossec,syslog,syscheck,authentication_failed,pam,pfsense,suricata,mis
  <api_key>oqDuFP6DwkkXO4vDb_vV4oDQ-FPM1-p1mrE_e8kUEg6aUqAoLNwFGBC8vEeBDZ8Lvl
  <alert_format>json</alert_format>
</integration>
```

Gambar 4. 8 Konfigurasi WAZUH untuk integrasi DFIR-iris

Gambar tersebut menunjukkan konfigurasi integrasi custom-iris.py dalam file konfigurasi *WAZUH*, khususnya di dalam elemen <integration>. Konfigurasi inimenunjukan beberapa parameter penting diatur untuk memungkinkan *WAZUH* mengirimkan alert langsung ke platform DFIR IRIS melalui sebuah webhook yang telah disediakan. Bagian <name> menunjukkan nama skrip integrasi yang digunakan, yaitu custom-iris.py. Parameter <hook_url> memuat URL webhook yang ditujukan untuk menerima alert yang dikirimkan dari *WAZUH* ke IRIS. Tingkat log atau prioritas alert yang akan diteruskan didefinisikan melalui <level>, di mana angka 6 menunjukkan bahwa alert dengan level tersebut atau lebih tinggi akan diteruskan. Bagian <group> menentukan kategori atau jenis log yang diintegrasikan, termasuk ossec, syslog, syscheck, dan lainnya,

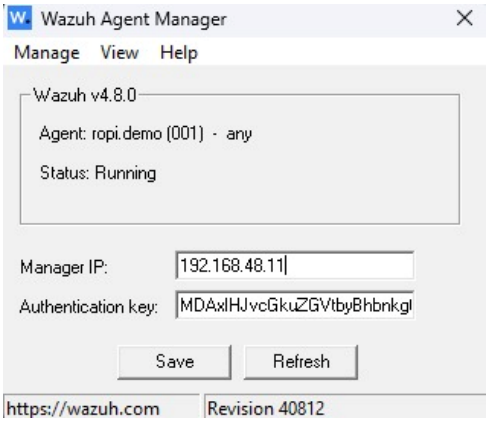
yang mencakup berbagai sumber data seperti firewall atau IDS. Selanjutnya, elemen <api_key> menyimpan token autentikasi yang diperlukan untuk otorisasi komunikasi antara *WAZUH* dan IRIS. Akhirnya, <alert_format> menentukan format data yang dikirim, yaitu json, untuk memastikan bahwa data alert dapat diproses dengan benar oleh sistem IRIS. Proses integrasi telah selesai jika alert *WAZUH* telah muncul pada laman DFIR-Iris seperti pada gambar 4.14



Gambar 4. 94 Alert WAZUH Pada DFIR-IRIS

4.1.4. Deploy WAZUH Agent Pada Host

Proses deploy *WAZUH* Agent pada host dimulai dengan menginstal agent sesuai sistem operasi host, seperti menggunakan perintah apt pada Ubuntu atau yum pada CentOS. Setelah instalasi, konfigurasi dilakukan dengan menambahkan alamat IP dan port *WAZUH* Server untuk memastikan konektivitas antara agent dan server. Terakhir, koneksi diverifikasi agar *WAZUH* Manager dapat mulai mengumpulkan log dan data keamanan dari host, mendukung deteksi ancaman secara terpusat dan real-time, untuk instalasi *WAZUH* agent pada sistem operasi windows bisa melalu aplikasi GUI, yang dapat di download pada website resmi *WAZUH* seperti yang ditunjukan pada gambar 4. 15



Gambar 4. 10 Deploy WAZUH Agent

Gambar tersebut menunjukkan antarmuka *WAZUH* Agent Manager versi 4.8.0, yang merupakan komponen penting dari platform *WAZUH*. *WAZUH* Agent adalah perangkat lunak yang diinstal pada endpoint untuk memantau dan mengirim data log keamanan ke *WAZUH* Manager. Dalam antarmuka ini, terlihat bahwa agen bernama ropi.demo (001) sedang dalam status Running, menandakan bahwa agen ini aktif dan berhasil terhubung ke manajer dengan alamat IP 192.168.48.11. Selain itu, terdapat Authentication Key, yang digunakan untuk otentikasi aman antara agen dan manajer, memastikan bahwa komunikasi antar perangkat dalam sistem tetap terenkripsi dan valid. Fitur seperti tombol "Save" dan "Refresh" memberikan opsi untuk menyimpan konfigurasi atau memperbarui status terkini.

Antarmuka ini memungkinkan untuk memantau dan mengelola agen *WAZUH* dengan mudah, yang merupakan langkah penting dalam implementasi sistem deteksi dan respons keamanan berbasis SIEM.

WAZUH agent pada windows harus melakukan konfigurasi Sysmon terlebih dahulu karena *MISP* memanfaatkan log Sysmon pada *WAZUH* untuk mengambil dan mendeteksi pola ancaman lebih detail dari aktifitas endpoint atau *WAZUH* agent.

4.2. Pengujian Sistem

Bagian pengujian sistem ini mencakup tiga skenario utama: pengujian *WAZUH*, pengujian integrasi *WAZUH* dengan *MISP*, dan pengujian integrasi *WAZUH*, DFIR-iris, dan *MISP*. Setiap pengujian dilakukan menggunakan sampel malware dari repositori github.com/ThatSINEWAVE, dengan jumlah

malware 130 yang di-deploy pada windows endpoint untuk memicu deteksi oleh sistem. Pengujian pertama, *WAZUH* diuji sebagai detektor tunggal. Pengujian kedua melibatkan integrasi dengan *MISP* untuk memanfaatkan data Threat Intelligence dalam mendeteksi ancaman. Pada skenario ketiga, DFIR-Iris ditambahkan untuk manajemen respon insiden dengan menerima alert serangan dari *WAZUH* sehingga respon terhadap insiden serangan lebih baik.

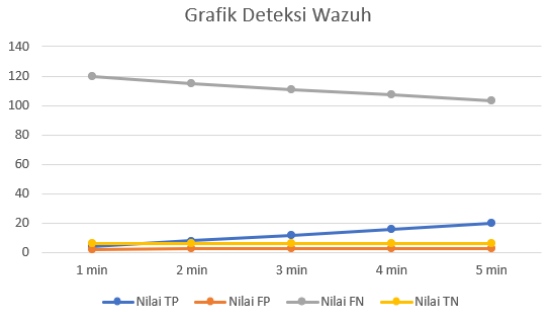
4.2.1. Pengujian WAZUH

Skenario pengujian pertama yaitu menguji *WAZUH*. Deteksi malwar pada *WAZUH* memanfaatkan integrasi bawaan dari virus total. API dari virus total memiliki keterbatasan melakukan lookup terhadap deteksi serangan sebanyak 4 look up per menit dan maksimal 500 per hari. Sehingga didapatkan hasil pengujian dari deteksi malware menggunakan *WAZUH* dapat dilihat pada tabel 4.1 berikut

Tabel 4. 1 Hasil Pengujian Wazuh

Pengujian	1 min	2 min	3 min	4 min	5 min
Nilai TP	4	8	12	16	20
Nilai FP	2	3	3	3	3
Nilai FN	120	115	111	107	103
Nilai TN	6	6	6	6	6

Pengujian dilakuan selama 5 menit untuk proses ekstraksi atau penambahan malware dilakukan pada menit pertama. Hasil pengujian menujukan pada menit pertama *WAZUH* dengan menggunakan API virus total hanya mampu mendeteksi malware sebanyak 4, hal tersebut dikarenakan keterbatasan API yang hanya memungkinkan melakukan lookup sebanaya 4 per menit. Peningkatan terjadi pada 5 menit berikutnya *WAZUH* mampu mendeteksi malware sebanyak 20. Hasil tersebut membuktikan bahwa lookup dari API virus total berpengaruh terhadap deteksi malware tersebut. Bentuk Grafik dari hasil ada pada gambar 4.16



Gambar 4. 116 Grafik Pengujian Wazuh

4.2.2. Pengujian WAZUH dan MISP

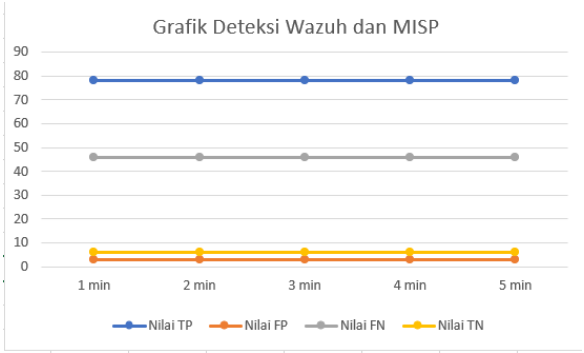
Dalam skenario ini, *WAZUH* diintegrasikan dengan *MISP* untuk memanfaatkan data Threat Intelligence yang disediakan oleh *MISP*. Proses deteksi dilakukan sama seperti pada pengujian *WAZUH*, tetapi dengan peningkatan data ancaman untuk akurasi lebih tinggi. *MISP* mampu mengirim Alert ke *WAZUH* apabila ada event Sysmon yang berjalan kemudian hasil dari file integriti management atau deteksi file malware yang di ekstrak di cocokan dengan data yang ada pada *MISP*.

Hasil pengujian menggunakan *MISP* yang dilakukan selama 5 menit dapat dilihat pada tabel di berikut:

Tabel 4. 2 Hasil Pengujian Wazuh+MISP

Pengujian	1 min	2 min	3 min	4 min	5 min
Nilai TP	78	78	78	78	78
Nilai FP	3	3	3	3	3
Nilai FN	46	46	46	46	46
Nilai TN	6	6	6	6	6

Hasil pengujian menunjukan bahwa integrasi *WAZUH* dan *MISP* memiliki nilai TP yang lebih baik dari pengujian sebelumnya tetapi untuk menit berikutnya tidak memiliki perubahan dan hasilnya tetap sama. Hasil tersebut disebabkan karena untuk pengambilan data API *MISP* harus ada event dari Sysmon sehingga jiga event Sysmon tidak ada yang terdeteksi lagi maka koneksi terhadap API akan mengalami error. Hasil pengujian dengan skenario dalam bentuk grafik ada pada gambar 4.17



Gambar 4. 17 Grafik Pengujian Wazuh dan MISP

4.2.3. Pengujian WAZUH, DFIR-Iris dan MISP

Pada tahap ini, *WAZUH*, *MISP*, dan DFIR-IRIS diintegrasikan untuk melakukan deteksi malware. DFIR-IRIS berfungsi sebagai managemen respon insiden, yang meneruskan alert serangan dari *WAZUH* mulai dari level 6. Alert yang di tampilkan pada DFIR-IRIS lebih mudah di baca dibandingkan yang ada pada *WAZUH* dashboard. Pengiriman alert dari *WAZUH* ke DFIR-IRIS tidak mengalamai hambatan atau mampu berjalan secara realtime. Alert tersebut siap di gunakan untuk membuar report dari incident response. Contoh alert yang bisa terkirim oleh *WAZUH* ke dfir iris ada pada gambar 4.18

Gambar 4. 128 Alert Pada DFIR-IRIS

4.3. Analisis Hasil Pengujian

Setelah seluruh pengujian, hasil dari tiga skenario dibandingkan untuk menentukan skema yang paling efektif dalam mendeteksi malware. Setiap skema dianalisis berdasarkan akurasi, *F1-score* yang diukur dari waktu *deploy* malware hingga terdeteksi oleh sistem.

Hasil evaluasi sistem dari pengujian sebelumnya ada pada tabel 4.3

Tabel 4. 3 Hasil Evaluasi Pengujian

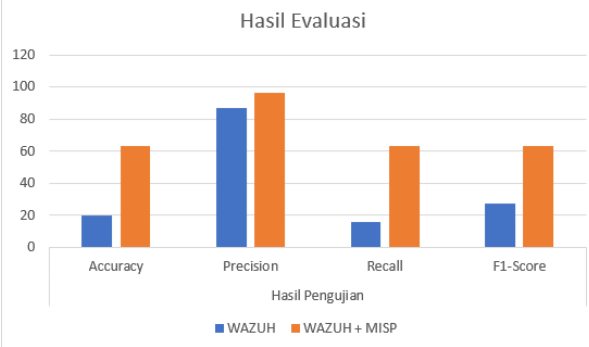
	Hasil Pengujian

Pengujian	Accuracy	Precision	Recall	F1-Score
WAZUH	19,70 %	86,96 %	16,26 %	27,40 %
WAZUH + MISP	63,16 %	96,30 %	62,90 %	62,90 %

Dari hasil pengujian, Akurasi sangat rendah (19,70%), menunjukkan bahwa sistem standalone Wazuh memiliki keterbatasan dalam mendeteksi ancaman dengan benar. Meski presisi cukup tinggi (86,96%), recall sangat rendah (16,26%), menandakan bahwa banyak ancaman yang sebenarnya ada tidak terdeteksi. F1-Score hanya 27,40%, mencerminkan performa keseluruhan yang kurang optimal karena keterbatasan API yang di gunakan dari virus total, tetapi dengan berjalanya waktu selama beberapa menit deteksi mengalami peningkatan secara perlahan.

Integrasi Wazuh dengan MISP menunjukkan presisi yang tinggi (96.3%), yang berarti hampir semua ancaman yang terdeteksi sebagai positif memang benar ancaman. Namun, tingkat Recall rendah (62.9%) menunjukkan bahwa sistem masih belum mampu mendeteksi semua ancaman yang ada. F1-Score sebesar 76.1% mencerminkan keseimbangan antara presisi dan recall, dengan akurasi keseluruhan sistem sekitar 63.1%, namun hasil tersebut tidak memiliki perubahan pada beberpa menit berikutnya dikarenakan koneksi dengan MISP memiliki kendala.

Hasil ini menunjukkan bahwa integrasi MISP dapat meningkatkan kualitas deteksi ancaman, meskipun masih perlu dilakukan peningkatan pada kemampuan sistem untuk mendeteksi ancaman positif yang terlewatkan (False Negative). Hasil pengujian dalam bentuk grafik ada pada gambar 4.19



Gambar 4. 19 Grafik Hasil Evaluasi

Integrasi dengan DFIR-IRIS memungkinkan penerimaan alert secara real-time, yang sangat membantu dalam mempercepat respons terhadap insiden keamanan. Dengan kemampuan ini, sistem dapat memberikan notifikasi dan detail ancaman secara langsung ke tim keamanan, sehingga tindakan mitigasi dapat dilakukan dengan lebih cepat dan efektif. Hal ini memperkuat kesiapan sistem dalam menghadapi ancaman keamanan yang dinamis.

5. KESIMPULAN

a. Wazuh sebagai sistem SIEM standalone memiliki akurasi sangat rendah (19,70%) dengan recall rendah (16,26%). Hal ini menunjukkan keterbatasan sistem dalam mendeteksi ancaman secara menyeluruh, meskipun presisi cukup tinggi (86,96%). F1-Score sebesar 27,40% mencerminkan performa keseluruhan yang kurang optimal, terutama disebabkan oleh keterbatasan API VirusTotal yang digunakan.
b. Integrasi dengan MISP secara signifikan meningkatkan kualitas deteksi ancaman. Presisi tinggi (96,3%) menunjukkan bahwa hampir semua ancaman yang terdeteksi sebagai positif adalah ancaman sebenarnya. Namun, recall sebesar 62,9% dan akurasi 63,1% mengindikasikan bahwa sistem masih melewatkan sejumlah ancaman (False Negative). Kendala pada koneksi dengan MISP memengaruhi stabilitas dan hasil deteksi, terutama pada pengujian jangka waktu tertentu
c. Penambahan DFIR-IRIS memungkinkan penerimaan alert secara real-time, yang mempercepat respons insiden keamanan. Kemampuan ini memberikan manfaat besar dalam meningkatkan kesiapan dan efektivitas sistem untuk menangani ancaman secara langsung, sehingga mitigasi dapat dilakukan dengan lebih cepat dan terarah.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pihak-pihak terkait yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

[1] R. D. Hapsari and K. G. Pambayun, "ANCAMAN CYBERCRIME DI INDONESIA: Sebuah Tinjauan Pustaka Sistematis," *Jurnal Konstituen*, vol. 5, no. 1, 2023, doi: 10.33701/jk.v5i1.3208.

[2] A. Roberts, "Cyber Threat Intelligence – What Does It Even Mean?," in *Cyber Threat Intelligence: The No-Nonsense Guide for CISOs and Security Managers*, Berkeley, CA: Apress, 2021, pp. 17–36. doi: 10.1007/978-1-4842-7220-6_2.

[3] S. Gillard, D. P. David, A. Mermoud, and T. Maillart, "Efficient collective action for tackling time-critical cybersecurity threats," *J Cybersecur*, vol. 9, no. 1, 2023, doi: 10.1093/cybsec/tyad021.

[4] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, 2021, doi: 10.3390/s21144759.

[5] Wazuh, "The Open Source Security Platform | Wazuh," Web. Accessed: May 13, 2024. [Online]. Available: https://documentation.wazuh.com/current/index.html

[6] P. Briand, R. Rafati, and A. C. Team, "Incident Response Information Sharing with DFIR IRIS: Enhancing Cybersecurity Investigations," *Threat Intelligence Lab*, 2023, [Online]. Available: https://blog.dfir-iris.org

[7] A. Alanda, H. A. Mooduto, and R. Hadi, "JITCE (Journal of Information Technology and Computer Engineering) Real-time Defense Against Cyber Threats: Analyzing Wazuh's Effectiveness in Server Monitoring," *JITCE*, pp. 56–62, 2023, doi: 10.25077/jitce.7.02.56-62.2023.

[8] D. P. Widyatono and W. Sulistyo, "Pemodelan Instrusion Prevention System Untuk Pendeteksi Dan Pencegahan Penyebaran Malware Menggunakan Wazuh," *Journal of Information Technology Ampera*, vol. 4, no. 1, pp. 113–127, 2023, [Online]. Available: https://journal-computing.org/index.php/journal-ita/index

[9] M. Alexandru STAN, "Automation of Log Analysis Using the Hunting ELK Stack," 2021.

[10] S. E. Jeon *et al.*, "An Effective Threat Detection Framework for Advanced Persistent Cyberattacks," *Computers, Materials and*

Continua, vol. 75, no. 2, 2023, doi: 10.32604/cmc.2023.034287.
[11] R. Fernandes, S. Bugla, P. Pinto, and A. Pinto, "On the Performance of Secure Sharing of Classified Threat Intelligence between Multiple Entities," *Sensors*, vol. 23, no. 2, Jan. 2023, doi: 10.3390/s23020914.
[12] IBM, "What is Security Information and Event Management (SIEM)?," IBM. Accessed: May 13, 2024. [Online]. Available: https://www.ibm.com/id-en/topics/siem
[13] S. Abu, S. R. Selamat, A. F. M. Ariffin, and R. Yusof, "Cyber Threat Intelligence – Issue and Challenges," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, pp. 371–379, 2018, [Online]. Available: https://api.semanticscholar.org/CorpusID:4882915
[14] MISP, "MISP - Malware Information Sharing Platform and Threat Sharing - The Open Source Threat Intelligence Platform," MISP. [Online]. Available: https://www.misp-project.org/