

Desain dan Prototipe Simulasi Sistem *Active Response* Berbasis Wazuh Untuk Deteksi dan Mitigasi Serangan *Malware* Terintegrasi *Threat Intelligence*

Mulyana¹⁾, I Putu Hariyadi²⁾, Ondi Asroni³⁾

^{1,2,3}Universitas Bumigora, Indonesia

Email: mulyanayani123@gmail.com¹, putu.hariyadi@universitasbumigora.ac.id²,
ondi817@gmail.com³

Abstract: *Malware is a growing cyber threat designed to operate covertly to steal data and even take control of a system without the owner's knowledge. This research aims to design and implement a Wazuh-based Active Response system integrated with Threat Intelligence to automatically detect and mitigate malware attacks, using the Network Development Life Cycle (NDLC) method with the stages of Analysis, Design, and Prototyping Simulation, utilizing Ubuntu Server as the Wazuh Server, Windows Server 2019 as the Wazuh Agent, File Integrity Monitoring (FIM), and the VirusTotal API. The test results show that the Detection Rate and Mitigation Success Rate reach 100% on .exe files and compressed files that have been extracted, with a Response Time of seconds and a False Positive Rate of 0%. However, the system has not been able to detect malware in compressed files before they are extracted due to hash scanning limitations. The system has proven effective in accelerating mitigation and increasing the success of handling malware threats. Further research is recommended to add other Threat Intelligence sources, variations of Wazuh Agent, and testing in production environments.*

Abstrak : *Malware menjadi ancaman siber yang terus berkembang dan dirancang untuk beroperasi secara tersembunyi guna mencuri data hingga mengambil alih kendali sistem tanpa sepengetahuan pemiliknya. Penelitian ini bertujuan merancang dan mengimplementasikan sistem Active Response berbasis Wazuh terintegrasi Threat Intelligence untuk mendeteksi dan memitigasi serangan malware secara otomatis, menggunakan metode Network Development Life Cycle (NDLC) dengan tahapan Analysis, Design, dan Simulation Prototyping, memanfaatkan UbuntuServer sebagai Wazuh Server, Windows Server 2019 sebagai Wazuh Agent, File Integrity Monitoring (FIM), dan API VirusTotal. Hasil pengujian menunjukkan Detection Rate dan Mitigation Success Rate mencapai 100% pada file .exe dan file terkompresi yang telah diekstrak, dengan Response Time hitungan detik dan False Positive Rate 0%, namun sistem belum mampu mendeteksi malware dalam file terkompresi sebelum diekstrak akibat keterbatasan pemindaian hash. Sistem terbukti efektif mempercepat mitigasi dan meningkatkan keberhasilan penanganan ancaman malware. Penelitian selanjutnya disarankan menambahkan sumber Threat Intelligence lain, variasi Wazuh Agent, serta pengujian pada lingkungan produksi.*

Keywords : *Active response, Malware, Threat Intelligence, VirusTotal, Wazuh.*

PENDAHULUAN

Malware menjadi salah satu ancaman siber yang terus berkembang dan dihadapi oleh setiap organisasi saat ini. Beberapa kategori *malware* seperti *spyware*, *Remote access trojan* (RAT) dan *stealer* dirancang dan beroperasi secara tersembunyi guna mencuri data, menjadi mata-mata hingga mengambil alih kendali sistem tanpa sepengetahuan pemiliknya (Damanik et al., 2023) (Na'im & Kurniawan, 2026). Dari penelitian (Manzoor et al., 2024) menegaskan bahwa organisasi *modern* membutuhkan solusi

keamanan yang mampu mendeteksi dan merespons ancaman secara *real-time*. Kemudian (Ilca et al., 2023) menambahkan bahwa *malware modern* secara aktif dirancang untuk menghindari deteksi konvensional, menjadikan pendekatan berbasis *Security Information and Event Management* (SIEM) yang terintegrasi dengan *Cyber Threat Intelligence* (CTI) sebagai kebutuhan mendasar. Hadirnya *wazuh* sebagai *platform SIEM open-source* dengan kemampuan *Host Intrusion Detection System* (HIDS), *File Integrity monitoring* (FIM), dan *active response* dalam satu ekosistem yang terintegrasi (Putra et al., 2025) (Haryanto & Chandra, 2024).

Efektivitas *wazuh* dalam mendeteksi dan memitigasi serangan siber juga dibuktikan dalam beberapa penelitian seperti, (Shafiyyah et al., 2024) berhasil mengimplementasikan *Wazuh* pada sistem keamanan jaringan Universitas Lampung Way kanan menggunakan metode PPDIOO dan membuktikan kemampuan *active response* dalam merespons serangan secara otomatis. Kemampuan *wazuh* dan *active response* juga ditunjukkan pada penelitian yang dilakukan oleh (Djamalyanto et al., 2025) dan (Chamkar et al., 2025) yaitu mampu memblokir 100% serangan *File Upload Vulnerability*, *Remote Code Execution*, dan *Webshell* pada web server. Sedangkan integrasi *Wazuh* dengan *VirusTotal* berhasil mendeteksi *trojan* dalam 14 detik, *ransomware* 15 detik, dan *worm* 9 detik secara otomatis (Putra et al., 2025). Namun, (Hidayat et al., 2025) menemukan bahwa *Wazuh standalone* hanya mencapai akurasi 19,70% dan *recall* 16,26% dalam mendeteksi *malware*, dan baru meningkat menjadi 96,3% setelah diintegrasikan dengan *platform CTI* berbasis MISP.

Penelitian-penelitian terdahulu masih terdapat kesenjangan yakni belum adanya sistem yang secara mencakup *Threat Intelligence* sebagai komponen inti pengambilan keputusan *active response* untuk deteksi *malware*. Sebagian besar penelitian hanya berfokus pada jenis ancaman tertentu tanpa memanfaatkan *Threat Intelligence* secara analitis sebagai dasar mitigasi otomatis. Diharapkan tersedia sistem yang mampu mendeteksi, memverifikasi melalui *threat intelligence*, dan memitigasi ancaman *malware* secara otomatis dan akurat.

Berdasarkan permasalahan tersebut, penelitian ini merancang prototipe simulasi sistem *active response* berbasis *Wazuh* untuk deteksi dan mitigasi serangan *malware* terintegrasi *threat intelligence*. Pengujian sistem dilakukan menggunakan dua metode, yaitu tanpa *active response* dan dengan *active response*, terhadap tiga jenis *malware* (RAT, *spyware*, dan *stealer*) dalam format *file .exe* serta *.zip*. Keberhasilan sistem tersebut kemudian diukur berdasarkan enam parameter utama, yaitu *Detection Rate*, *Response Time*, *Threat Intelligence Accuracy*, *Alert Level*, *Mitigation Success Rate*, dan *False Positive Rate*.

Penelitian ini diharapkan menjadi referensi untuk desain prototipe simulasi sistem keamanan berbasis *open-source* yang dapat diadaptasi sekaligus memperkaya literatur integrasi SIEM dengan *Threat Intelligence* untuk penanganan ancaman *malware* secara aktif dan terukur.

METODE

Penelitian ini menggunakan pendekatan eksperimental dengan metodologi *Network Development Life Cycle* (NDLC), yang mencakup tiga tahapan yaitu *Analysis*, *Design*, dan *Simulation Prototyping* (Prayitno & Yasir, 2025) (Kristiyanto et al., 2024). Tahap *Analysis* dilakukan melalui studi literatur terhadap 15 penelitian terdahulu yang membahas integrasi Wazuh dengan *Threat Intelligence* dan *Active Response*, untuk mengidentifikasi kesenjangan penelitian sebagaimana telah diuraikan pada bagian Pendahuluan.

Pada tahap *Design*, sistem dirancang menggunakan dua *virtual machine* yang dihosting pada layanan *virtual private server* (VPS) IDCloudHost dalam satu jaringan privat (VPC) beralamat 10.7.26.0/24, yaitu UbuntuServer sebagai Wazuh Server (IP publik 210.79.191.219, IP privat 10.7.26.3) dan Windows Server 2019 sebagai Wazuh Agent (IP publik 103.55.38.45, IP privat 10.7.26.4). Spesifikasi perangkat yang digunakan ditunjukkan pada Tabel 1.

Tabel 1. Spesifikasi Perangkat Penelitian	
Perangkat	Spesifikasi
VPS UbuntuServer (Wazuh Server)	4 vCPU, 8 GB RAM, 50 GB Storage
VPS Windows Server 2019 (Wazuh Agent)	2 vCPU, 4 GB RAM, 50 GB Storage
Laptop Peneliti	AMD Ryzen 3, RAM 12 GB, SSD 500 GB

Alur kerja sistem dimulai ketika sampel malware, diperoleh dari GitHub (format .exe) dan MalwareBazaar (format .zip), diunduh ke direktori Downloads pada Windows Agent. Fitur *File Integrity Monitoring* (FIM) pada Wazuh mendeteksi kemunculan file baru tersebut dan memicu rule 100201, yang kemudian mengirim alert ke Wazuh Manager (Kurniawan et al., 2024) (Kamil et al., 2024). Selanjutnya, integrator Wazuh mengirimkan hash file ke VirusTotal API sebagai sumber *Threat Intelligence* untuk verifikasi reputasi file. Apabila VirusTotal mengonfirmasi file sebagai malware (rule 87105), sistem menjalankan *Active Response* berupa skrip remove-threat.exe pada Agent untuk menghapus file secara otomatis, yang dikonfirmasi melalui rule 553. Seluruh proses ini ditampilkan secara real-time pada Wazuh Dashboard.

Tahap *Simulation Prototyping* terdiri atas instalasi dan konfigurasi lingkungan kerja (Wazuh Server, Wazuh Agent, integrasi VirusTotal, serta konfigurasi *Active Response*), pelaksanaan pengujian, dan analisis hasil (Fahmi et al., 2025). Pengujian dilakukan terhadap tiga jenis malware, *Remote Access Trojan* (RAT), *spyware*, dan *stealer*, dalam dua format file (.exe dan .zip), dengan dua kondisi (dengan dan tanpa *Active Response*), masing-masing diulang lima kali, ditambah satu skenario kontrol menggunakan file bukan malware (installer dan Notepad++). Keberhasilan sistem diukur menggunakan enam parameter sebagaimana disajikan pada Tabel 2.

Tabel 2. Parameter Pengujian Sistem

Parameter	Definisi	Rumus Evaluasi
Detection Rate	Kemampuan sistem mendeteksi file malware yang masuk ke <i>endpoint</i>	$(\text{Jumlah malware terdeteksi} / \text{Total sampel diuji}) \times 100\%$
Response Time	Waktu sejak file terdeteksi FIM hingga Active Response selesai dieksekusi	Timestamp AR selesai – Timestamp deteksi FIM (detik)
Threat Intelligence Accuracy	Tingkat akurasi VirusTotal berdasarkan jumlah <i>engine</i> yang mendeteksi	$(\text{Jumlah engine mendeteksi} / \text{Total engine VirusTotal}) \times 100\%$
Alert Level	Tingkat keparahan <i>alert</i> berdasarkan skala <i>rule.level</i> Wazuh	Level 1–6 = Low; 7–11 = Medium; 12+ = High
Mitigation Success Rate	Keberhasilan Active Response menghapus file malware terdeteksi	$(\text{Jumlah file berhasil dihapus} / \text{Total file terdeteksi malware}) \times 100\%$
False Positive Rate	Kesalahan sistem mendeteksi file aman sebagai malware	$(\text{Jumlah file normal salah terhapus} / \text{Total file normal diuji}) \times 100\%$

Analisis hasil dilakukan dengan membandingkan performa sistem pada kondisi dengan dan tanpa *Active Response* terhadap keenam parameter tersebut, guna mengevaluasi efektivitas integrasi Threat Intelligence dalam mempercepat dan meningkatkan keberhasilan mitigasi serangan malware.

HASIL DAN PEMBAHASAN

Pengujian terhadap tiga jenis *malware* (RAT, *Spyware*, *Stealer*) dalam format .exe dilakukan sebanyak lima kali replikasi pada dua kondisi, yaitu dengan dan tanpa *Active Response*.

Tabel 3. Rata-Rata Hasil Pengujian File .exe dengan AR (5 Replikasi)

Parameter	<i>RAT</i>	<i>Spyware</i>	<i>Stealer</i>
<i>Detection Rate</i>	100%	100%	100%
<i>Response Time</i>	7.6 detik	9.4 detik	6.0 detik
<i>TI Accuracy</i>	85.83%	2.90%	90.27%
<i>Alert Level</i>	12 (<i>High</i>)	12 (<i>High</i>)	12 (<i>High</i>)
<i>Mitigation Success Rate</i>	100%	100%	100%
<i>False Positive Rate</i>	0%	0%	0%

Berdasarkan Tabel 3, sistem dengan *Active Response* aktif berhasil mencapai *Detection Rate* dan *Mitigation Success Rate* sebesar 100% untuk ketiga jenis *malware*, dengan *Alert Level* konsisten di tingkat 12 (*High*) dan *False Positive Rate* 0%. *Response Time* berlangsung cepat, berkisar antara 6,0–9,4 detik, dengan *Spyware* secara konsisten membutuhkan waktu terlama, diduga terkait proses

verifikasi *VirusTotal* terhadap sampel dengan jumlah engine pendeteksi yang sedikit. *TI Accuracy* bervariasi signifikan antar jenis malware, dari 2,90% pada *Spyware* hingga 90,27% pada *Stealer*, mengindikasikan keterbatasan basis data *VirusTotal* dalam mengenali varian *Spyware* yang diuji.

Tabel 4. Rata-Rata Hasil Pengujian File .exe tanpa AR (5 Replikasi)

Parameter	<i>RAT</i>	<i>Spyware</i>	<i>Stealer</i>
<i>Detection Rate</i>	100%	100%	100%
<i>Response Time</i>	Tidak Terukur	Tidak Terukur	Tidak Terukur
<i>TI Accuracy</i>	85.83%	10.60%	90.27%
<i>Alert Level</i>	12 (<i>High</i>)	12 (<i>High</i>)	12 (<i>High</i>)
<i>Mitigation Success Rate</i>	0%	0%	0%
<i>False Positive Rate</i>	0%	0%	0%

*Rata-rata *TI Accuracy Spyware* pada kondisi tanpa AR sedikit lebih tinggi (10,60%) akibat satu nilai luar (*outlier*) 41,42% pada replikasi pertama, empat replikasi lainnya konsisten di 2,89%.

Berdasarkan Tabel 4, tanpa *Active Response*, sistem tetap mampu mendeteksi seluruh malware dengan *Detection Rate* 100% dan *Alert Level* 12 (*High*), membuktikan bahwa kemampuan deteksi FIM dan verifikasi *VirusTotal* tidak bergantung pada status *Active Response*, terkonfirmasi dari nilai *TI Accuracy* *RAT* dan *Stealer* yang identik dengan Tabel 3 (85,83% dan 90,27%). Namun, karena tidak ada mekanisme eksekusi otomatis, *Response Time* tidak dapat diukur dan *Mitigation Success Rate* tertahan di angka 0% untuk ketiga jenis *malware*, sehingga seluruh file berbahaya tetap tersimpan di endpoint tanpa tindakan penghapusan. *False Positive Rate* tetap 0%, menunjukkan akurasi deteksi tidak terganggu oleh nonaktifnya *Active Response*.

Perbandingan Tabel 3 dan Tabel 4 menegaskan bahwa perbedaan mendasar antara kedua kondisi hanya terletak pada tahap mitigasi, *Mitigation Success Rate* melonjak dari 0% menjadi 100% ketika *Active Response* diaktifkan, sementara kemampuan deteksi (*Detection Rate*, *Alert Level*, *TI Accuracy*) tetap konsisten di kedua kondisi.

Pengujian file .zip dilakukan pada dua kondisi tambahan, yaitu sebelum dan sesudah file diekstrak, untuk mengevaluasi kemampuan sistem terhadap file terkompresi.

Tabel 5. Rata-Rata Hasil Pengujian File .zip Setelah ekstrak dengan AR (5 Replikasi)

Parameter	<i>RAT</i>	<i>Spyware</i>	<i>Stealer</i>
<i>Detection Rate</i>	100%	100%	100%
<i>Response Time</i>	5.0 detik	6.4 detik	5.2 detik
<i>TI Accuracy</i>	62.98%	52.06%	28.48%
<i>Alert Level</i>	12 (<i>High</i>)	12 (<i>High</i>)	12 (<i>High</i>)
<i>Mitigation Success Rate</i>	100%	100%	100%
<i>False Positive Rate</i>	0%	0%	0%

Berdasarkan Tabel 5, begitu file .zip diekstrak, sistem dengan *Active Response* aktif langsung menunjukkan performa optimal. *Detection Rate* dan *Mitigation Success Rate* mencapai 100% untuk ketiga jenis *malware*, dengan *Alert Level* konsisten di tingkat 12 (*High*) dan *False Positive Rate* 0%. *Response Time* tercatat cukup cepat, berkisar antara 5,0–6,4 detik, sedikit lebih lambat dibanding file .exe kemungkinan karena proses ekstraksi menambah satu tahapan sebelum FIM dapat mendeteksi file hasil ekstrak. Nilai *TI Accuracy* bervariasi cukup lebar, dari 28,48% pada *Stealer* hingga 62,98% pada *RAT*, mengindikasikan bahwa sampel *malware* dalam format arsip yang baru diekstrak belum tentu memiliki cakupan pengenalan yang sama di basis data *VirusTotal* dibanding format .exe langsung.

Tabel 6. Rata-Rata Hasil Pengujian File .zip Setelah ekstrak tanpa AR (5 Replikasi)

Parameter	<i>RAT</i>	<i>Spyware</i>	<i>Stealer</i>
<i>Detection Rate</i>	100%	100%	100%
<i>Response Time</i>	Tidak terukur	Tidak terukur	Tidak terukur
<i>TI Accuracy</i>	66.28%	56.89%	37.78%
<i>Alert Level</i>	12 (<i>High</i>)	12 (<i>High</i>)	12 (<i>High</i>)
<i>Mitigation Success Rate</i>	0%	0%	0%
<i>False Positive Rate</i>	0%	0%	0%

Berdasarkan Tabel 6, tanpa *Active Response*, sistem tetap mampu mendeteksi seluruh *malware* pasca-ekstraksi dengan *Detection Rate* 100% dan *Alert Level* 12 (*High*), membuktikan bahwa kemampuan deteksi FIM dan verifikasi *VirusTotal* tidak bergantung pada status *Active Response*. Namun, karena tidak ada mekanisme eksekusi otomatis, *Response Time* tidak dapat diukur dan *Mitigation Success Rate* tertahan di angka 0% pada ketiga jenis *malware*, artinya seluruh file berbahaya tetap tersimpan di endpoint tanpa ada tindakan penghapusan. *False Positive Rate* tetap 0%, menunjukkan akurasi deteksi tidak terganggu oleh nonaktifnya *Active Response*.

Tabel 7. Rata-Rata Hasil Pengujian File .zip Sebelum ekstrak dengan & tanpa AR (5 Replikasi)

Parameter	<i>RAT</i>	<i>Spyware</i>	<i>Stealer</i>
<i>Detection Rate</i>	0%	0%	0%
<i>Response Time</i>	Tidak terukur	Tidak terukur	Tidak terukur
<i>TI Accuracy</i>	<i>No positives</i>	<i>No positives</i>	<i>No positives</i>
<i>Alert Level</i>	3 (<i>Low</i>)	3 (<i>Low</i>)	3 (<i>Low</i>)
<i>Mitigation Success Rate</i>	0%	0%	0%
<i>False Positive Rate</i>	0%	0%	0%

Berdasarkan Tabel 7, sebelum file .zip diekstrak, sistem sama sekali tidak mampu mendeteksi kandungan *malware* di dalamnya, dengan *Detection Rate* 0% pada seluruh replikasi untuk ketiga jenis

malware, terlepas dari status *Active Response* aktif atau nonaktif. *VirusTotal* hanya mampu memindai hash dari file arsip secara keseluruhan dan menghasilkan status *No positives*, sehingga *Alert Level* hanya mencapai level 3 (*Low*), jauh di bawah ambang batas yang memicu *Active Response*. Akibatnya, *Response Time* tidak terukur dan *Mitigation Success Rate* tetap 0% pada kedua kondisi, karena memang tidak ada ancaman yang berhasil diidentifikasi untuk dimitigasi. Temuan ini menjadi keterbatasan utama sistem yang diidentifikasi dalam penelitian.

Perbandingan antara Tabel 5, Tabel 6, dan Tabel 7 menegaskan dua temuan utama pada pengujian file .zip. Pertama, kemampuan deteksi sistem terhadap *malware* dalam file .zip sepenuhnya bergantung pada status ekstraksi, bukan pada status *Active Response*, dibuktikan oleh *Detection Rate* yang identik (0% pada Tabel 7, 100% pada Tabel 5 dan 6) antara kondisi dengan dan tanpa AR pada tahap ekstraksi yang sama. Kedua, *Active Response* hanya berpengaruh pada tahap mitigasi setelah file diekstrak. *Mitigation Success Rate* mencapai 100% pada Tabel 5 namun tetap 0% pada Tabel 6, meskipun *Detection Rate* keduanya sama-sama sempurna. Dengan demikian, keterbatasan sistem murni terletak pada tahap verifikasi hash *VirusTotal* terhadap file arsip yang belum diekstrak (Tabel 7), bukan pada mekanisme *Active Response* itu sendiri.

Tabel 8. Hasil Pengujian File Bukan Malware

Parameter	<i>Installer</i>	Notepad++
<i>Detection Rate</i>	0%	0%
<i>Response Time</i>	Tidak terukur	Tidak terukur
<i>TI Accuracy</i>	<i>No positives</i>	<i>No positives</i>
<i>Alert Level</i>	3 (<i>Low</i>)	3 (<i>Low</i>)
<i>Mitigation Success Rate</i>	0%	0%
<i>False Positive Rate</i>	0%	0%

Berdasarkan Tabel 8, pengujian kontrol menggunakan dua file *non-malware* dengan kondisi *Active Response* aktif menghasilkan *Detection Rate* 0% pada keduanya, menandakan sistem berhasil menyaring file tersebut sebagai aman dan tidak keliru mengklasifikasikannya sebagai ancaman. Hal ini diperkuat oleh *TI Accuracy* yang menunjukkan status *No Positive* dan *Alert Level* yang tetap berada di tingkat minimal 3 (*Low*). Karena tidak ada indikasi bahaya yang terpicu, *Active Response* tidak mengeksekusi tindakan apa pun sehingga *Response Time* tidak terukur dan *Mitigation Success Rate* berada di 0% (tidak relevan diukur, bukan kegagalan sistem). *False Positive Rate* tercatat sempurna 0%, membuktikan sistem mampu membedakan file aman dari file berbahaya tanpa memicu alarm palsu.

Secara keseluruhan, integrasi *Active Response* dengan *Threat Intelligence* terbukti meningkatkan kemampuan mitigasi sistem *Wazuh* secara signifikan. Sistem tanpa *Active Response* tetap mampu

mendeteksi seluruh *malware* pada format .exe dan .zip yang telah diekstrak (*Detection Rate* 100%), namun sepenuhnya gagal memitigasi ancaman (*Mitigation Success Rate* 0%), sehingga file *malware* tetap berada di endpoint dan berisiko dieksekusi oleh pengguna. Sebaliknya, sistem dengan *Active Response* mampu menghapus seluruh file *malware* secara otomatis dalam hitungan detik, konsisten dengan temuan (Djamalyanto et al., 2025) dan (Putra et al., 2025) yang juga melaporkan efektivitas *Active Response* berbasis Wazuh dalam memitigasi ancaman secara *real-time*.

Namun, penelitian ini juga mengonfirmasi keterbatasan yang belum banyak dibahas pada penelitian terdahulu, yaitu ketidakmampuan sistem mendeteksi *malware* di dalam file terkompresi sebelum diekstrak, karena *VirusTotal* hanya mampu memindai *hash* file arsip secara keseluruhan. Temuan ini melengkapi kesenjangan yang diidentifikasi pada bagian Pendahuluan, di mana penelitian sebelumnya belum menguji secara khusus pengaruh format file terhadap efektivitas deteksi berbasis *hash scanning*. Variasi *TI Accuracy* yang lebar antar jenis *malware*, khususnya rendahnya akurasi pada sampel *Spyware* juga mengindikasikan bahwa efektivitas sistem sangat bergantung pada cakupan basis data *Threat Intelligence* yang digunakan, sejalan dengan catatan (Hidayat et al., 2025) bahwa integrasi sumber CTI tambahan (seperti MISP) dapat meningkatkan presisi deteksi secara substansial dibanding mengandalkan satu sumber saja. Pengujian kontrol pada file bukan *malware* turut memperkuat validitas sistem, membuktikan bahwa peningkatan sensitivitas deteksi tidak mengorbankan akurasi klasifikasi (*False Positive Rate* tetap 0% di seluruh skenario).

KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem *Active Response* berbasis Wazuh yang terintegrasi *Threat Intelligence* melalui beberapa penyesuaian mandiri, meliputi pembuatan *custom rule* 100200 dan 100201 untuk memonitor direktori *Downloads* secara spesifik, konfigurasi integrasi *VirusTotal* menggunakan *rule_id* sebagai pemicu pemindaian, serta pengembangan skrip *remove-threat.py* yang dikompilasi menjadi *remove-threat.exe* sebagai agen mitigasi otomatis pada *endpoint* Windows Server 2019. Hasil pengujian terhadap tiga jenis *malware*, *Remote Access Trojan* (RAT), *Spyware*, dan *Stealer* pada dua format file menunjukkan bahwa sistem dengan *Active Response* aktif mampu mencapai *Detection Rate* dan *Mitigation Success Rate* sebesar 100% dengan *Response Time* berkisar antara 1 hingga 11 detik serta *False Positive Rate* 0%. Sebaliknya, sistem tanpa *Active Response* tetap mendeteksi seluruh *malware* dengan *Detection Rate* 100% namun *Mitigation Success Rate* bernilai 0% karena tidak ada mekanisme penghapusan otomatis, sehingga file *malware* tetap tersimpan di disk. Pengujian pada file terkompresi .zip mengungkap keterbatasan mendasar sistem. Sebelum file diekstrak, *VirusTotal* hanya mampu membaca nilai *hash* file secara keseluruhan sehingga kandungan *malware* di dalamnya tidak terdeteksi (*Alert Level* hanya mencapai level 3/Low), sedangkan

setelah diekstrak sistem kembali mampu mendeteksi dan memitigasi malware secara penuh (*Detection Rate* dan *Mitigation Success Rate* 100%, *Alert Level* 12/High). Dengan demikian, integrasi *Active Response* dengan *Threat Intelligence* terbukti efektif meningkatkan kemampuan mitigasi otomatis dibandingkan sistem tanpa *Active Response*, sekaligus mengonfirmasi kesenjangan penelitian yang diidentifikasi pada bagian Pendahuluan, yaitu belum tersedianya sistem yang menjadikan *Threat Intelligence* sebagai komponen inti pengambilan keputusan *Active Response* secara menyeluruh, termasuk pada kondisi file terkompresi. Penelitian ini memiliki beberapa keterbatasan. Pertama, sistem hanya menggunakan *VirusTotal* sebagai satu-satunya sumber *Threat Intelligence*, sehingga cakupan deteksi bergantung sepenuhnya pada basis data dan mekanisme pemindaian *hash* platform tersebut, terbukti dari rendahnya akurasi deteksi pada sampel *Spyware* dan ketidakmampuan memindai file .zip sebelum diekstrak. Kedua, pengujian hanya dilakukan pada satu jenis *Wazuh Agent* (Windows Server 2019), sehingga efektivitas sistem pada sistem operasi atau endpoint lain belum dapat dipastikan. Ketiga, penelitian dilakukan pada lingkungan simulasi berbasis VPS, bukan pada lingkungan produksi dengan volume trafik dan variasi ancaman yang lebih kompleks. Bagi penelitian selanjutnya, disarankan menambahkan sumber *Threat Intelligence* lain di luar *VirusTotal*, seperti MISP atau AbuseIPDB, guna memperluas cakupan deteksi khususnya terhadap malware dalam file terkompresi sebelum diekstrak. Disarankan pula menguji sistem pada variasi *Wazuh Agent* dengan sistem operasi berbeda, serta melakukan pengujian pada lingkungan produksi untuk mengevaluasi kinerja sistem dalam kondisi operasional yang sesungguhnya.

UCAPAN TERIMA KASIH

Penulis menyampaikan terima kasih kepada Program Studi Teknologi Informasi, Fakultas Teknik, Universitas Bumigora atas dukungan institusional selama penelitian berlangsung. Terima kasih khusus disampaikan kepada Dosen Pembimbing I dan Dosen Pembimbing II atas bimbingan, arahan, dan masukan yang berharga dalam penyusunan penelitian ini. Penulis juga mengucapkan terima kasih kepada Yayasan Khouw Kalbe “Beasiswa Bestari” yang telah memberikan dukungan dana hingga pengembangan kapasitas bagi setiap *awardee*.

DAFTAR PUSTAKA

- Chamkar, S. A., Zaydi, M., Maleh, Y., & Gherabi, N. (2025). Improving Threat Detection in Wazuh Using Machine Learning Techniques. In *Journal of Cybersecurity and Privacy* (Vol. 5, Nomor 2, hal. 34). <https://doi.org/10.3390/jcp5020034>
- Damanik, A. R., Seta, H. B., Universitas, D., Nasional, P., & Jakarta, V. (2023). Analisis Trojan Dan Spyware Menggunakan Metode Hybrid Analysis. *Jurnal Ilmiah MATRIK*, 25(1), 89–97.

<https://doi.org/10.33557/jurnalmatrik.v25i1.2327>

- Djamalyanto, M., Widyawati, L., Husain, H., & Hariyadi, I. P. (2025). Implementasi Security Information and Event Management Untuk Mencegah Serangan Deface Pada Server Terintegrasi Telegram. *Melek IT: Information Technology Journal*, 11(1), 31–42. <https://doi.org/10.30742/melekitjournal.v11i1.398>
- Fahmi, R. N., Hartono, R., & Anwar, D. S. (2025). Integrasi Wazuh Siem Dengan Modsecurity Dan Virus Total Menggunakan Nist Framework Untuk Mendeteksi Serangan Website. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(4), 6578–6586. <https://doi.org/10.36040/jati.v9i4.13804>
- Haryanto, B., & Chandra, D. W. (2024). Implementasi Wazuh Integritas File untuk Perlindungan Keamanan Berdasarkan Aktivitas Log di BTSI UKSW. *Jurnal Indonesia: Manajemen Informatika dan Komunikasi*, 5(1), 183–192. <https://doi.org/10.35870/jimik.v5i1.447>
- Hidayat, M. R. T., Widiyasono, N., & Gunawan, R. (2025). Optimasi Deteksi Malware Pada Siem Wazuh Melalui Integrasi Cyber Threat Intelligence Dengan Misp Dan Dfir-Iris. *Jurnal Informatika dan Teknik Elektro Terapan*, 13(1), 588–601. <https://doi.org/10.23960/jitet.v13i1.5686>
- Ilca, L. F., Lucian, P., & Balan, T. C. (2023). Enhancing Cyber-Resilience for Small and Medium-Sized Organizations with Prescriptive Malware Analysis, Detection and Response. *Sensors*, 23(15). <https://doi.org/10.3390/s23156757>
- Kamil, A., Rizaludin, D., & Tsalitsatun, A. (2024). Implementasi Wazuh FIM (File Integrity Monitoring) untuk Perlindungan Keamanan Sistem Informasi pada Unit Kegiatan Mahasiswa di Universitas Trunojoyo Madura. *Sains Data Jurnal Studi Matematika dan Teknologi*, 2(2), 80–92. <https://doi.org/10.52620/sainsdata.v2i2.127>
- Kristiyanto, Y., Sugiyono, Perdana, C. A., & Sarah, S. (2024). Pengembangan Keamanan Komputer Dengan Teknologi Thin Client Dan Domain Controller Dengan Metode Network Development Life Cycle (NDLC). *Jurnal Manajemen Informatika Jayakarta*, 4(3), 258–265. <https://doi.org/10.52362/jmijayakarta.v4i3.1531>
- Kurniawan, C., Triayudi, A., Technology, I., & Nasional, U. (2024). File Integrity Monitoring as a Method for Detecting and Preventing Web Defacement Attacks. *JOIN (Jurnal Online Informatika)*, 9(2), 276–285. <https://doi.org/10.15575/join.v9i2.1326>
- Manzoor, J., Waleed, A., Jamali, A. F., & Masood, A. (2024). Cybersecurity on a budget: Evaluating security and performance of open-source SIEM solutions for SMEs. *PLOS ONE*, 19(3), 1–24. <https://doi.org/10.1371/journal.pone.0301183>
- Na'im, M. R., & Kurniawan, Y. (2026). Otomatisasi Respons pada Sistem Keamanan Siber Terintegrasi Threat Intelligence pada Pusat Data Pemerintah Kota Tangerang dengan Metode PPDIOO. *SUDO Jurnal Teknik Informatika*, 5(2), 188–201. <https://doi.org/10.56211/sudo.v5i2.1742>
- Prayitno, M. H., & Yasir, M. (2025). Peran Metode Network Development Life Cycle (NDLC) pada Implementasi Failover Base Transceiver Station. *Innovative: Journal Of Social Science Research*, 5(2), 2146–2158. <https://doi.org/10.31004/innovative.v5i2.18512>
- Putra, M. A. E., Hariyadi, I. P., & Marzuki, K. (2025). Otomatisasi Konfigurasi Wazuh Terintegrasi

Virustotal Menggunakan Ansible Untuk Mendeteksi Dan Memproteksi Serangan Malware. *MELEK IT :Information TECHNOLOGY JOURNAL*, 11(1), 55–66. <https://doi.org/10.30742/melekitjournal.v11i1.403>

Shafiyyah, A., Nama, G. F., & Pradipta, R. A. (2024). Implementasi Wazuh Menggunakan Metode PPDIOO di Sistem Keamanan Jaringan PSDKU Universitas Lampung Waykanan Sebagai Deteksi Dan Respon Serangan Siber. *Jurnal Informatika dan Teknik Elektro Terapan (JITET)*, 12(2), 970–982. <https://doi.org/10.23960/jitet.v12i2.4074>