pISSN: 2442-3386 eISSN: 2442-4293
Melek IT
Vol 11 No 1 Jan – Jun 2025, 55 - 66
INFORMATION TECHNOLOGY JOURNAL

OTOMATISASI KONFIGURASI *WAZUH* TERINTEGRASI *VIRUSTOTAL* MENGGUNAKAN *ANSIBLE* UNTUK MENDETEKSI DAN MEMPROTEKSI SERANGAN MALWARE

Muhammad Alvan Ekawan Putra*[1], I Putu Hariyadi[2], Khairan Marzuki[3]

[1]Ilmu Komputer, Universitas Bumigora, muhamadalvanekawanputra@gmail.com
[2]Ilmu Komputer, Universitas Bumigora, putu.hariyadi@universitasbumigora.ac.id
[3]Ilmu Komputer, Universitas Bumigora, khairan.marzuki@universitasbumigora.ac.id
*)Korespondensi: muhamadalvanekawanputra@gmail.com

Abstrak

Pesatnya perkembangan teknologi digital turut meningkatkan risiko *malware* yang dapat mengganggu sistem, mencuri data, dan menghambat operasional. *wazuh* hadir sebagai solusi *open-source* untuk mendeteksi aktivitas mencurigakan melalui analisis *log*, namun akurasinya dapat ditingkatkan dengan integrasi *virustotal* sebagai layanan verifikasi *file* mencurigakan. Tujuan utama dari penelitian ini adalah membangun *playbook ansible* untuk mengotomatisasi proses instalasi, konfigurasi *wazuh*, integrasi dengan *virustotal*, dan pengiriman notifikasi ke *telegram* secara *real-time*. Metode yang digunakan adalah pendekatan *Network Development Life Cycle (NDLC)* dengan tiga tahapan, yaitu: analisis, desain, dan *simulation prototyping*. Analisis dilakukan dengan studi literatur dan perbandingan penelitian terdahulu. Tahap desain mencakup perancangan jaringan, sistem, dan konfigurasi IP. Tahap simulasi dilakukan dalam lingkungan virtual menggunakan lima VM yang terdiri dari *server wazuh*, *server ansible*, dua *agent (windows dan linux*), serta satu VM tambahan untuk pengujian otomatisasi konfigurasi. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan mampu secara efektif mendeteksi tiga jenis *malware* utama (*trojan, ransomware, dan worm*), melakukan *active response*, dan mengirimkan peringatan secara otomatis melalui *telegram*. Proses instalasi dan konfigurasi dapat dilakukan dengan lebih cepat dan konsisten berkat *playbook ansible*. Sistem ini juga terbukti handal dan efisien dalam menangani serangan *malware* secara otomatis. active response terhadap *malware* yang berbeda, memiliki jeda waktu yang berbeda baik ketika terdeteksi maupun terhapus. Hal tersebut dipengaruhi oleh kualitas koneksi internet dan batas waktu penggunaan API. sehingga waktunya dapat berbeda-beda. Selain itu, penggunaan *ansible* terbukti mengurangi waktu konfigurasi secara signifikan serta meminimalkan potensi *human error* dalam proses *deployment* di lingkungan jaringan virtual. Kesimpulan dari penelitian ini adalah bahwa integrasi *wazuh* dan *virustotal* yang diotomatisasi konfigurasinya menggunakan *ansible* dapat meningkatkan efisiensi, akurasi, dan skalabilitas dalam deteksi serta proteksi terhadap *malware*. Otomatisasi ini tidak hanya mempercepat proses pengamanan, tetapi juga mengurangi kesalahan konfigurasi manual, sehingga memberikan solusi keamanan yang lebih tangguh dan dapat dikembangkan lebih lanjut.

Kata Kunci: *Otomatisasi, Wazuh, Ansible, Virustotal, Malware*

Abstract

The rapid advancement of digital technology has increased the risk of malware that can disrupt systems, steal data, and hinder operations. Wazuh emerges as an open-source solution capable of detecting suspicious activities through log analysis. However, its accuracy can be enhanced by integrating VirusTotal as a verification service for suspicious files. The main objective of this study is to develop an Ansible playbook to automate the installation process, Wazuh configuration, integration with VirusTotal, and real-time notification delivery to Telegram. The method used is the Network Development Life Cycle (NDLC) approach, consisting of three stages: analysis, design, and simulation prototyping. The analysis stage involves literature review and comparison of previous studies. The design stage covers network, system, and IP *configuration planning. The simulation is conducted in a virtual environment using five virtual machines consisting of a Wazuh server, Ansible server, two agents (Windows and Linux), and an additional VM for configuration automation testing. The results show that the developed system effectively detects three main types of malware (trojan, ransomware, and worm), performs active response, and sends alerts automatically via Telegram. Installation and configuration processes become faster and more consistent thanks to the Ansible playbook. The system also proves to be reliable and efficient in handling malware attacks automatically. Active response timing varies for each type of malware—both in detection and removal—affected by internet connection quality and API usage time limits, leading to time differences. Additionally, the use of Ansible significantly reduces configuration time and minimizes the potential for human error during deployment in the virtual network environment. The conclusion of this study is that the integration of Wazuh and VirusTotal with Ansible-based configuration automation improves efficiency, accuracy, and scalability*

Melek IT
Information Technology Journal. Vol 11 No 1 Jan - Jun 2025, 56 - 66

in malware detection and protection. This automation not only accelerates the security process but also reduces manual configuration errors, offering a more robust security solution that can be further developed.

Keywords: *Automation, Wazuh, Ansible, Virustotal, Malware*

I. PENDAHULUAN

Perkembangan teknologi informasi memberikan kemudahan sekaligus tantangan dalam hal keamanan sistem digital. Ancaman siber seperti *malware* termasuk *trojan*, *ransomware*, dan *worm* telah menjadi isu krusial yang memerlukan solusi cepat dan efektif [1]. Dalam era serangan siber yang semakin kompleks, penting bagi organisasi untuk memiliki sistem keamanan yang dapat secara proaktif mendeteksi dan merespons ancaman tersebut [2]. Deteksi dini dan penanganan yang cepat menjadi kebutuhan mendesak dalam sistem informasi modern. Keamanan dari sistem dan teknologi harus dilindungi untuk menjaga aset informasi dari serangan atau penyalahgunaan [3]. *wazuh* hadir sebagai *Host Intrusion Detection System* (*HIDS*) yang andal dalam memantau *log* sistem dan memberikan notifikasi peringatan. Namun dalam praktiknya, akurasi deteksi *file* mencurigakan dapat ditingkatkan dengan integrasi layanan pihak ketiga seperti *Virustotal* [4]. *Virustotal* menyediakan verifikasi *file* berbasis *hash* dengan dukungan lebih dari 70 mesin antivirus. Sayangnya, proses integrasi manual antara *wazuh* dan *Virustotal* tidak hanya memakan waktu tetapi juga rawan kesalahan konfigurasi [5].

Wazuh adalah platform keamanan yang terfokus pada deteksi ancaman dan pemantauan keamanan. Salah satu fitur utamanya adalah memiliki kemampuan untuk mengawasi dan memantau integritas *file* dalam *system* [6]. *Wazuh* efektif sebagai alat deteksi celah keamanan dan manajemen *log* khususnya terhadap serangan *DoS*[7]. Implementasi *Security Information and Event Management (SIEM)* berbasis *wazuh* merupakan solusi yang efektif dalam meningkatkan keamanan sistem informasi [8]. Meski tidak semua serangan terdeteksi akibat keterbatasan seperti *firewall cloud*, sistem ini tetap mampu memberikan perlindungan tambahan dengan dukungan visualisasi melalui *Kibana* dan *Elasticsearch* [9]. Mengembangkan sistem keamanan dengan menggabungkan *wazuh* dan IPS Kombinasi ini berhasil meningkatkan deteksi dan respon terhadap serangan seperti *DDoS* dan *Metasploit*, serta mampu mengurangi risiko melalui aturan pencegahan otomatis yang diterapkan *IPS* [10]. IPS berfungsi untuk memonitor lalu lintas jaringan, mengidentifikasi potensi ancaman, dan secara proaktif mencegah serangan sebelum menyebabkan kerusakan [11]. Memadukan *wazuh HIDS, Honeypot Dionaea*, dan *Cuckoo Sandbox* untuk mendeteksi aktivitas mencurigakan, menarik serangan, dan menganalisis perilaku *malware[12]*. *Intrusion Detection System (IDS)* adalah sistem yang memantau lalu lintas jaringan untuk aktivitas yang mencurigakan dan mengeluarkan peringatan ketika aktivitas tersebut ditemukan [13]. Meski deteksi lalu lintas jaringan masih terbatas, pendekatan ini memperkuat kemampuan identifikasi dan pemahaman terhadap ancaman siber [14].

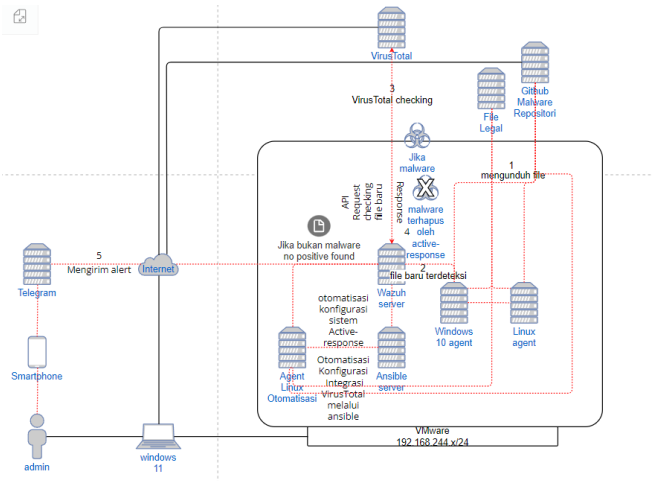
Untuk mengatasi hal tersebut maka penelitian ini mengembangkan sistem otomatisasi instalasi dan konfigurasi *wazuh* terintegrasi *virustotal* menggunakan *ansible*. Tujuan penelitian ini adalah menyederhanakan proses instalasi, integrasi, dan manajemen konfigurasi *wazuh-virustotal* secara efisien, konsisten, dan dapat diskalakan.

II. METODE

Metodologi penelitian yang digunakan pada penelitian ini adalah *NDLC (Network Development Life Cycle)*. Dari keenam tahap metode *NDLC* tersebut, penulis hanya menggunakan tiga tahapan saja yaitu tahap *analysis, design, simulation prototyping [15]*. Tujuan dari metode yang digunakan ini agar penulis dapat membuat penulisannya secara efisien dalam mengerjakannya.

2.1 Rancangan Sistem

Rancangan sistem yang digunakan pada penelitian ini seperti terliha pada gambar 1.



Gambar 1. Rancangan alur kerja sistem integrasi *Virustotal*

Alur kerja sistem pada gambar 1 menunjukan penggunaan *wazuh*, *ansible*, dan *virustotal* yang dijalankan dalam lingkungan virtual berbasis *Vmware workstations* dengan alamat jaringan 192.168.244.0/24. Proses dimulai saat *agent (Windows*

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 1 Jan – Jun 2025, 57 - 66
Melek IT
INFORMATION TECHNOLOGY JOURNAL

atau Linux) mengunduh *file malware* dari *GitHub* (langkah 1), lalu *wazuh server* mendeteksi *file* baru (langkah 2) dan mengirim permintaan ke *virustotal* untuk memeriksa reputasi *file* tersebut melalui *API* (langkah 3). Jika *virustotal* mengonfirmasi *file* sebagai *malware*, maka *wazuh* menjalankan *active-response* untuk menghapus *malware*, dan mengirimkan notifikasi ke *telegram* (langkah 4) agar admin dapat segera mengetahui insiden melalui *smartphone*. Seluruh konfigurasi dan integrasi dengan *virustotal* serta sistem *active-response* dapat dikonfigurasi otomatis menggunakan *ansible* pada agent *linux* otomatisasi. Diagram ini menunjukkan sistem keamanan jaringan yang terotomatisasi, responsif, dan terintegrasi penuh untuk mendeteksi serta menanggulangi *malware* secara *real-time*.

III. HASIL DAN PEMBAHASAN

Bagian ini memuat tentang pembahasan dari hasil instalasi, konfigurasi dan otomatisasi yang dilakukan pada *server* dan *agent* meliputi integrasi *virustotal*, *wazuh manager*, *linux agent*, *ansible server* dan *windows agent*.

3.1 Hasil Uji Coba

Hasil uji coba terdiri dari dua bagian yaitu verifikasi konfigurasi dan skenario uji coba.

3.2 Uji coba 3 (tiga) jenis *Malware*

Uji coba yang dilakukan yaitu mengunduh *file malware* pada *website* https://github.com/Da2dalus/The-MALWARE Repo dari setiap *agent* yaitu *windows* dan *linux*. Pada setiap *agent* telah terkonfigurasi *active response* yang terintegrasi dengan *virustotal* API sehingga dapat secara otomatis mendeteksi dan menghapus *file* yang terindikasi sebagai *malware*.

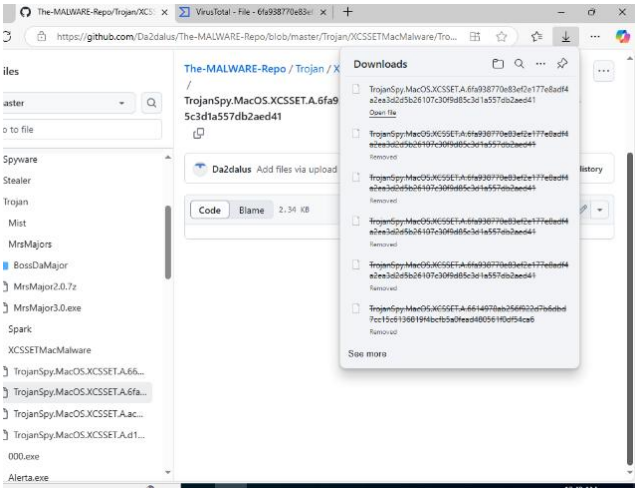
Skenario yang dilakukan adalah ketika setiap *wazuh agent* menambahkan atau mengubah *file* pada *agent* maka *wazuh server* akan membaca *hash value* pada *file* tersebut. Kemudian *wazuh server* akan mengirim *API request* ke *virustotal*. *Virustotal* akan mendeteksi *hash value* tersebut. Ketika nilai *hash* tersebut memiliki ancaman positif pada *database virustotal* maka *active response* yang telah dikonfigurasi pada *agent* akan otomatis menghapus *file* yang terdeteksi dan terindikasi *malware*.

Uji coba *malware* yang dilakukan adalah sebagai berikut:

a. Trojan

Trojan merupakan salah satu jenis *malware* (perangkat lunak berbahaya) yang menyamar sebagai program yang sah atau tidak berbahaya[16], tapi sebenarnya mengandung kode berbahaya yang dapat merusak, mencuri, atau mengambil alih sistem komputer tanpa sepengetahuan pengguna[17].

Uji coba yang dilakukan antara lain mengunduh *file trojan*, dan mengamati *alert* yang dikirimkan oleh *server* ke *telegram*, seperti terlihat pada gambar 1.



Gambar 2. Uji Coba Mengunduh *Trojan* Pada *Windows agent*

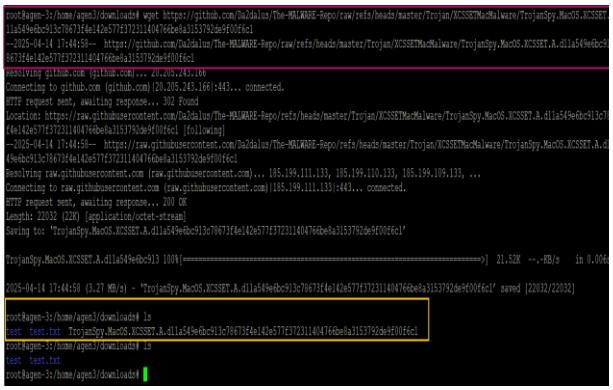
Terlihat pada gambar 2 yaitu *windows agent* melakukan pengunduhan *file trojan*. sedangkan *alert* yang dikirimkan ke *telegram* seperti terlihat pada gambar 3.

Gambar 3. *Alert* Pada *Bot Telegram*

Terlihat pada gambar 3 yaitu *alert* pada *bot telegram* yang menunjukkan *active response* dari

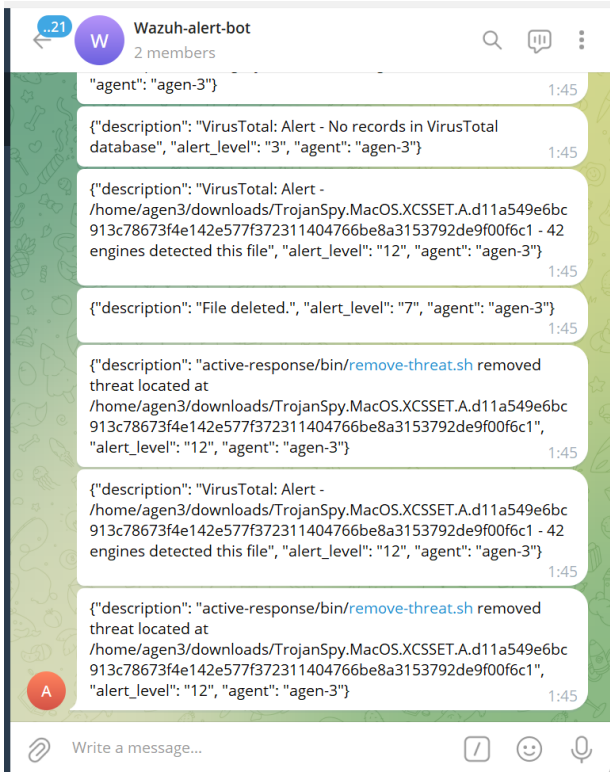
removethreat.exe telah berhasil mengatasi *file* yang terindikasi *malware* berjenis *trojan*.

Uji coba yang dilakukan pada *agent linux* seperti terlihat pada gambar 4.



Gambar 4. Ujicoba Mengunduh *Trojan* Pada *Linux agent*

Pada gambar 4 merupakan uji coba yang dilakukan dengan cara mengunduh *file trojan* menggunakan perintah *wget* dan disimpan pada direktori */home/agen3/downloads*. Terlihat *file* tersebut dapat diunduh kemudian dihapus secara otomatis oleh *active response*. *Alert* yang dikirimkan ke *telegram bot* terlihat pada gambar 5.



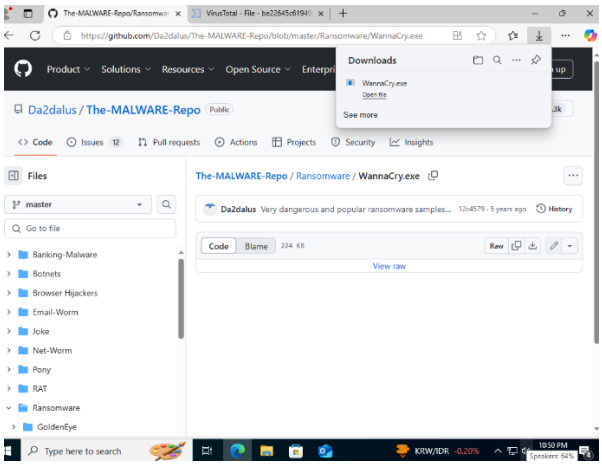
Gambar 5. *Alert Linux agent* Pada *Telegram*

Terlihat pada gambar 5 yaitu *alert* pada *bot telegram* yang menunjukkan *active response* dari *remove-threat.sh* telah berhasil mengatasi *file* yang terindikasi *malware* berjenis *trojan*.

b. Ransomware

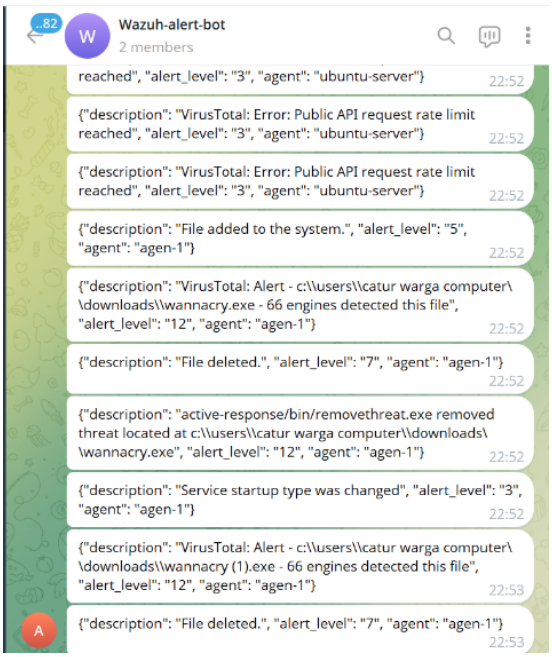
Ransomware adalah jenis *malware* (perangkat lunak berbahaya) yang dirancang untuk mengunci atau mengenkripsi data milik korban, sehingga korban tidak bisa mengaksesnya[18]. Setelah itu, penyerang biasanya akan menuntut tebusan (*ransom*) dalam bentuk uang (sering dalam *cryptocurrency* seperti *Bitcoin*) agar korban mendapatkan kunci untuk membuka akses kembali ke datanya[19].

Uji coba yang dilakukan antara lain mengunduh *file ransomware* dan mengamati *alert* yang dikirimkan oleh *server* ke *telegram bot*, seperti terlihat pada gambar 6.



Gambar 6. Uji Coba Unduh *File* Ransomeware Pada *agent Windows*

Pada gambar 6 uji coba yang dilakukan mengunduh salah satu *file ransomware* berjenis *wannacry* dapat terlihat pada gambar 6 *wannacry.exe* berhasil terunduh. Sedangkan *alert* yang dikirimkan ke *telegram* seperti terlihat pada gambar 7.

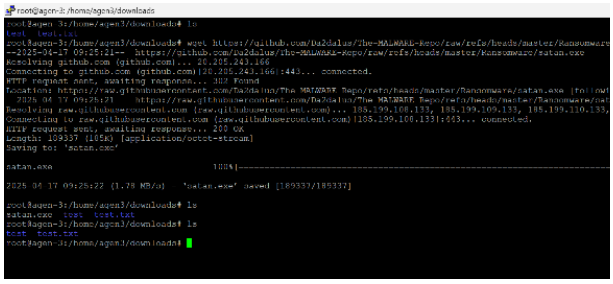


Gambar 7. *Alert Telegram Ransomware*

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 1 Jan – Jun 2025, 59 - 66
Melek IT
INFORMATION TECHNOLOGY JOURNAL

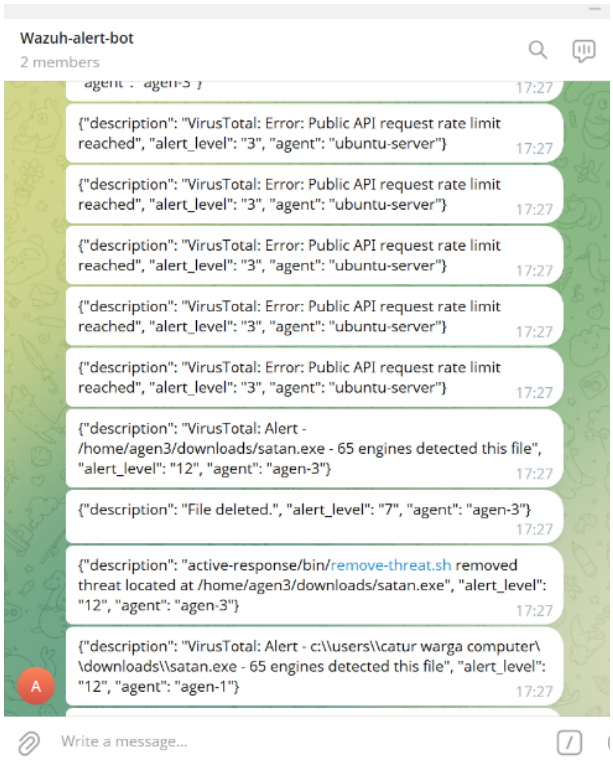
Terlihat pada gambar 7 yaitu *alert* pada *bot telegram* yang menunjukkan *active response* dari *removethreat.exe* telah berhasil mengatasi *file* yang terindikasi *malware* berjenis *ransomware wannacry*.

Selanjutnya *ransomware* akan diuji coba pada *wazuh agent linux* menggunakan jenis *ransomware* yang berbeda yaitu *ransomware satan*, seperti terlihat pada gambar 8.



Gambar 8. Uji Coba Unduh *Ransomware Satan*

Pada gambar 8 uji coba mengunduh *file ransomware* berjenis *satan* pada *agent linux*. Hasil eksekusi perintah *ls* pertama memperlihatkan bahwa *file satan.exe* berhasil tersimpan. Selanjutnya dalam beberapa detik, fitur *active response* akan melakukan pendeteksian dan penghapusan pada *file* tersebut. Hal ini dapat terlihat pada hasil eksekusi perintah *ls* kedua yaitu *file* tersebut sudah terhapus karena mengandung *ransomware* berjenis *satan*.

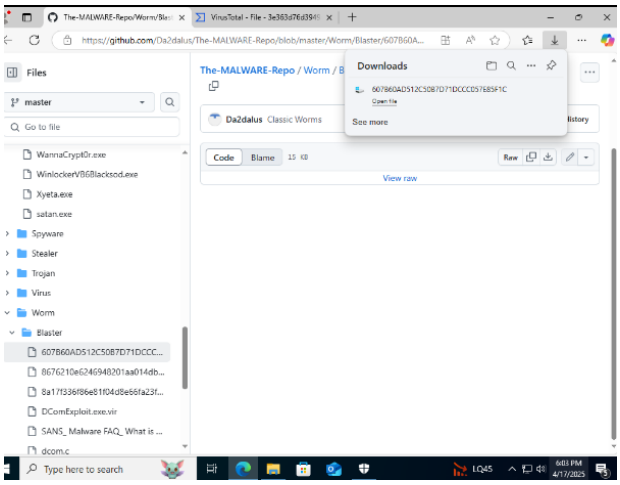


Gambar 9. *Alert* Pada *Bot Telegram*

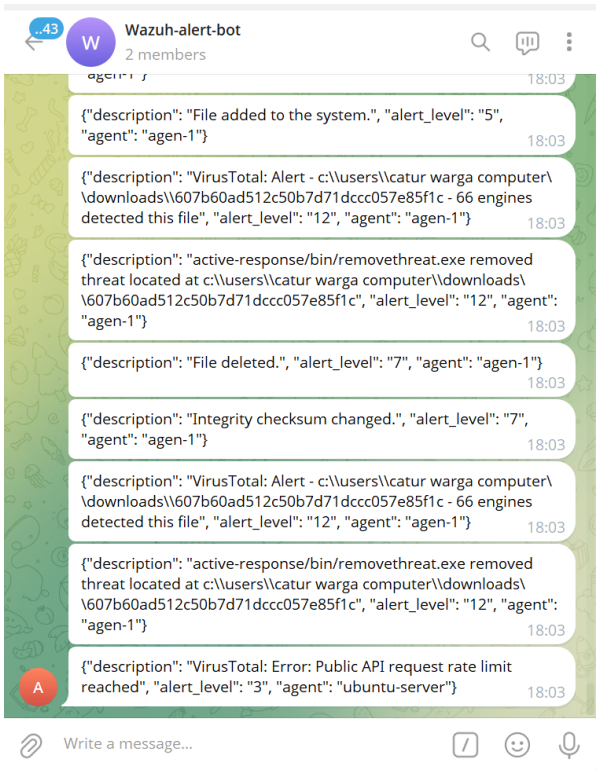
Pada gambar 9 dapat terlihat bahwa *active response remove-threat.sh* berhasil mendeteksi dan menghapus *file ransomware* berjenis *satan*.

c. *Worm*

Worm adalah jenis *malware* atau program berbahaya yang dapat menyebar sendiri secara otomatis dari satu komputer ke komputer lainnya tanpa bantuan pengguna dan tanpa perlu menempel pada *file* atau program lain seperti virus[20]. Uji coba mengunduh *file malware berjenis worm* yang dilakukan pada sistem operasi *windows* terlihat pada gambar 9.



Gambar 10. Uji Coba Mengunduh *File Malware* Berjenis *Worm* Pada *Windows agent*



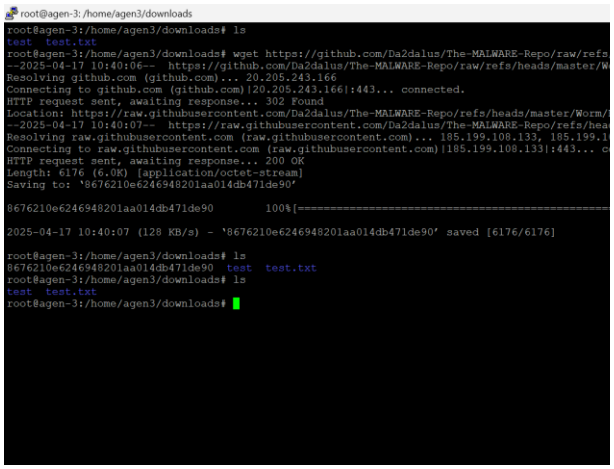
Gambar 11. *Alert* Pada *Bot Telegram*

Terlihat pada gambar 10 merupakan unduhan *file malware* berjenis *worm* pada *windows agent* berhasil masuk ke sistem *file windows*. *File* tersebut akan terdeteksi dan terhapus otomatis oleh *active response*

wazuh. Alert yang dikirimkan ke *telegram bot* seperti terlihat pada gambar 11.

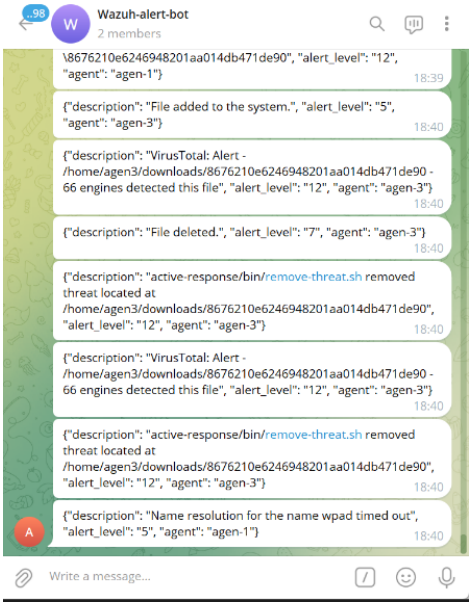
Terlihat pada gambar 11 yaitu *alert* pada *bot telegram* yang menunjukkan *active response* dari *removethreat.exe* telah berhasil mengatasi *file* yang terindikasi *malware* berjenis *worm*.

Pada *linux agent* dilakukan uji coba mengunduh file *malware* dengan jenis yang berbeda seperti terlihat pada gambar 12.



Gambar 12. Uji Coba Mengunduh *Malware Worm*

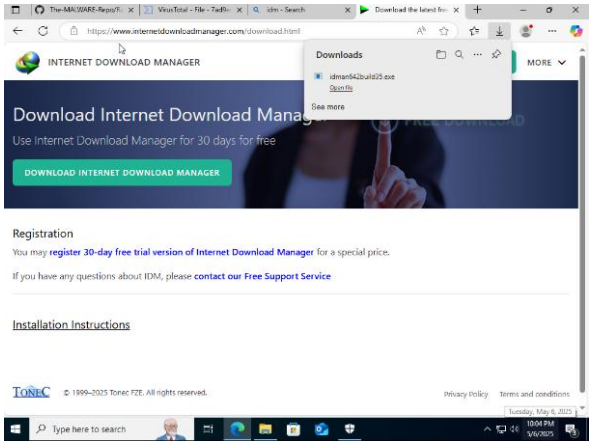
Terlihat pada gambar 12 merupakan hasil uji coba mengunduh *malware* berjenis *worm lovesan/blaster*. Hasil eksekusi perintah *ls* pertama menunjukkan bahwa *file* tersebut sudah berhasil masuk ke sistem. Namun hasil eksekusi perintah *ls* kedua yang dipicu beberapa detik berikutnya menunjukkan bahwa *file* tersebut terhapus. Hal ini menandakan bahwa *active response remove-threat.sh* berhasil mendeteksi dan menghapus *file* tersebut karena *file* tersebut terindikasi sebagai *malware*. Sedangkan hasil notifikasi berupa *alert* yang dikirimkan ke *telegram bot* seperti terlihat pada gambar 13.



Gambar 13. *Alert* Pada *Telegram Bot*

Terlihat pada gambar 13 pada *alert* telegram bot *active response* berhasil mendeteksi dan menghapus *file* tersebut menggunakan *remove-threat.sh.*

Untuk uji coba mengunduh *file* yang bukan terindikasi *malware* diperlihatkan pada gambar 14.



Gambar 14. Uji Coba Unduh *File* Legal

Pada gambar 14 terlihat uji coba mengunduh *file* dari aplikasi *Internet Download Manager* yang digunakan sebagai contoh untuk uji coba *file* legal. *Alert* yang dihasilkan seperti terlihat pada gambar 15.

Gambar 15. Alert Untuk *File* Bukan *Malware*

Terlihat pada gambar 15 merupakan *alert* yang diberikan ketika mengunduh *file* yang bukan terindikasi sebagai *malware*. *Alert* tersebut memberikan *no positives found* yang artinya tidak ada indikasi *malware* terhadap *file* tersebut.

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 1 Jan – Jun 2025, 61 - 66
Melek IT
INFORMATION TECHNOLOGY JOURNAL

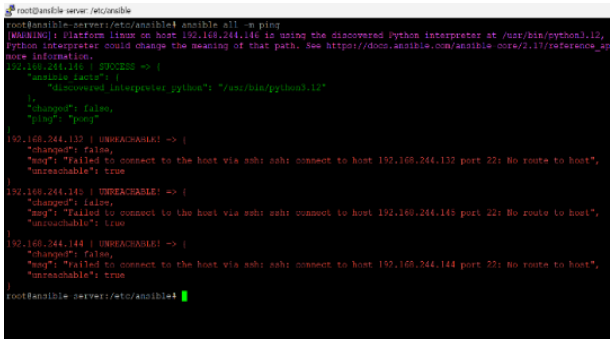
3.3 Hasil otomatisasi konfigurasi *wazuh agent* menggunakan *ansible*

Instalasi *wazuh agent* secara manual dapat memakan banyak waktu dan memungkinkan terjadinya *human error*. Penulis memanfaatkan *ansible* sebagai *tools* untuk mengotomatisasi konfigurasi penambahan *wazuh agent* pada *host* dengan sistem operasi *linux ubuntu server*.

Skenario uji coba yang dilakukan meliputi verifikasi koneksi menggunakan *ansible ad-hoc* ke seluruh *host* yang terdapat pada *inventory* menggunakan modul *ping* dan membuat serta menjalankan *playbook* ke *hosts* yang dijadikan sebagai *wazuh agent*.

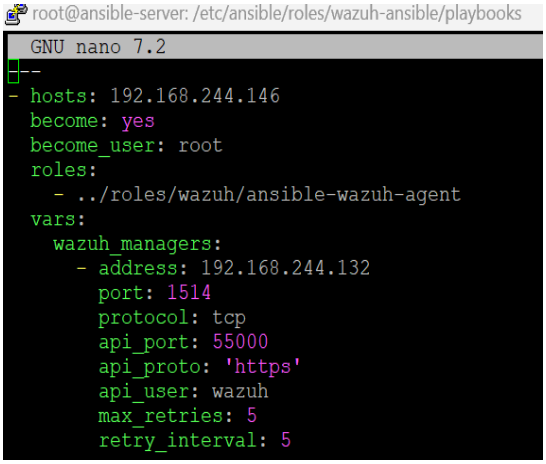
a) Hasil *ping ansible server* ke *hosts* sebagai target otomatisasi.

Langkah pertama yaitu *ping* ke semua *host* dengan perintah *ansible all -m ping* untuk mengetahui *ansible* dan *hosts* dapat terhubung dan berkomunikasi pada jaringan yang sama. Hasil dari eksekusi perintah tersebut seperti terlihat pada gambar 16.



Gambar 16. Hasil *Ping* Ke Semua *Hosts Ansible*

Terlihat pada gambar 16 bahwa hasil *ping* dari *ansible* ke *hosts* target dengan alamat IP 192.168.244.146 berhasil sehingga otomatisasi dapat dilakukan. Sedangkan luaran dari eksekusi ping ke IP dari *host* lainnya gagal dilakukan yaitu ditandai dengan pesan *unreachable*. Hal tersebut bermakna bahwa *vm* dari *host-host* tersebut belum aktif pada *vmware*.



Gambar 17. Playbooks Wazuh-Agent.Yml

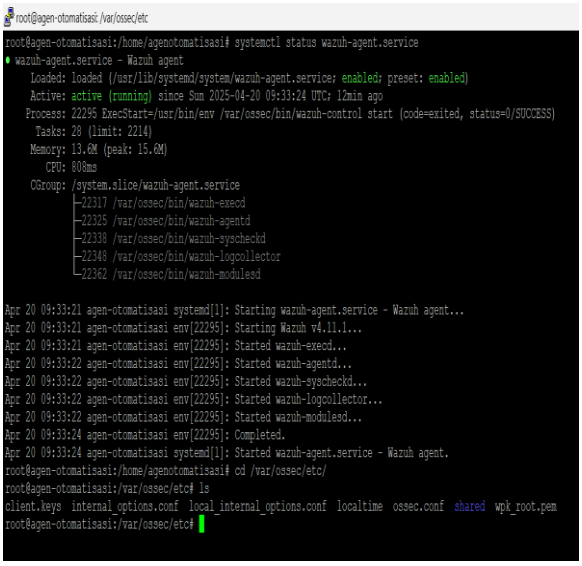
b) *Playbook wazuh agent*

Untuk melakukan otomatisasi maka *playbook* yang digunakan yaitu *wazuh-agent*.yml yang ada pada direktori /etc/*ansible*/roles/*wazuh-ansible*/*playbooks.*

Pada gambar 17 merupakan *playbook* yang akan dijalankan untuk otomatisasi menambah *agent* dengan perintah *ansible-playbook wazuh-agent*.yml -b -K.

Hasil menjalankan *playbook wazuh-agent.yml* seperti terlihat pada gambar 18.

Gambar 18. Hasil Menjalankan *Playbook wazuh-agent.yml*

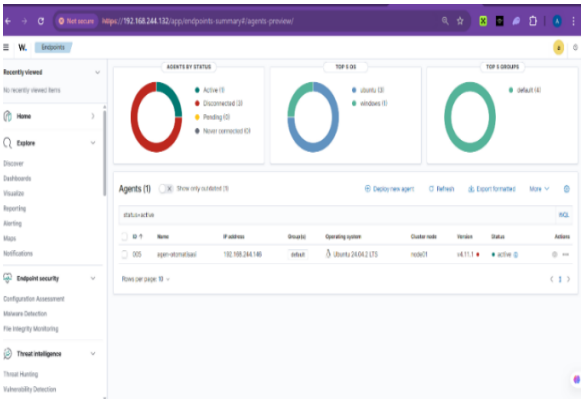


Gambar 19. Status *Wazuh Agent* Hasil Otomatisasi

Pada gambar 18 terlihat hasil eksekusi *playbook* untuk menjadikan *host* dengan IP 192.168.244.146 sebagai

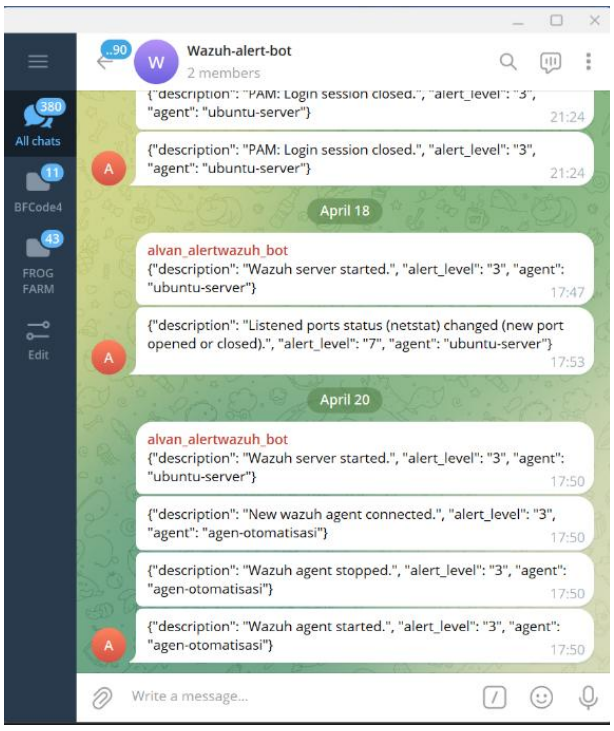
wazuh agent. Sedangkan hasil verifikasi dari status *wazuh agent* yang diterapkan menggunkan *playbook*, terlihat status *wazuh agent* terlihat pada gambar 19.

Pada gambar 19 terlihat status dari *service wazuh agent* yang telah aktif di salah satu *client* yang difungsikan sebagai *agent* yaitu *host* bernama *agen-otomatisasi* dengan alamat IP 192.168.244.146. Sedangkan hasil dari penambahan *agent* tersebut di *dashboard* dari *wazuh server* seperti terlihat pada gambar 20.



Gambar 20. Daftar *Wazuh Agent* Pada *Dashboard Wazuh Server*

Pada gambar 20 memperlihatkan hasil dari otomatisasi *wazuh agent* yang telah terdaftar pada *dashboard wazuh server* dengan alamat IP 192.168.244.146. Sedangkan hasil dari *telegram bot alert* yang dikirimkan seperti terlihat pada gambar 21.



Gambar 21. *Alert* Pada *Telegram Bot*

Pada gambar 21 terlihat *alert* dengan deskripsi *new wazuh agent connected* yang memiliki nama *agent agent*-otomatisasi. Hal tersebut menandakan bahwa *agent* baru telah terhubung ke *wazuh* sebagai dampak dari hasil eksekusi *playbook*.

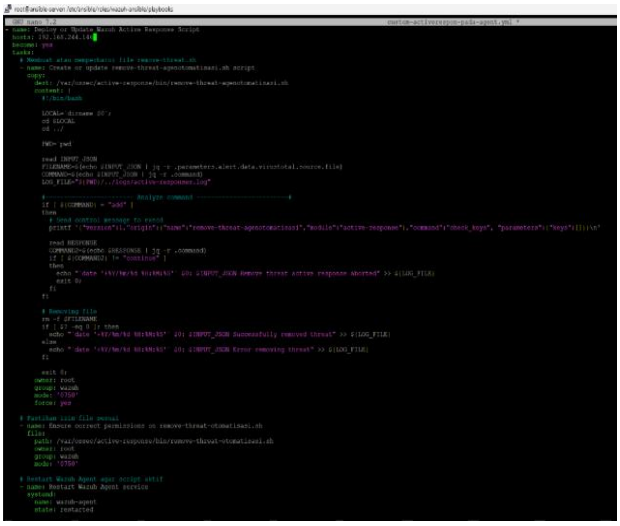
Berdasarkan hasil tersebut maka dapat disimpulkan baahwa otomatisasi mampu mempersingkat proses konfigurasi yang dilakukan secara berulang-ulang pada sistem manual sehingga menjadi lebih efisien. Dengan sekali eksekusi *playbook* pada *ansible server* maka keseluruhan konfigurasi yang harus dilakukan pada *server* dan *agent* dapat diterapkan.

3.4 Hasil konfigurasi integrasi *virustotal* secara otomatisasi menggunakan *ansible.*

Adapun hasil uji coba *playbook* yang diintegrasikan dengan *virustotal* adalah sebagai berikut:

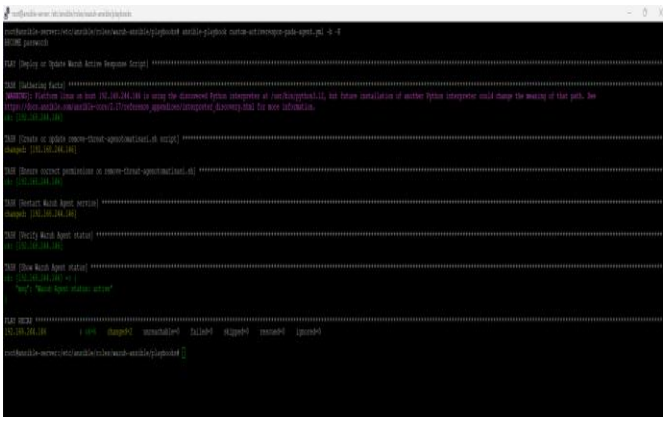
a. Uji coba *playbook* pada sisi *agent*

Playbook pada sisi *agent* yang akan digunakan yaitu *custom-activerespon-pada-agent.yml*, isi dari *playbook* terlihat pada gambar 22.



Gambar 22. *Playbook Active Response* Pada Sisi *agent*

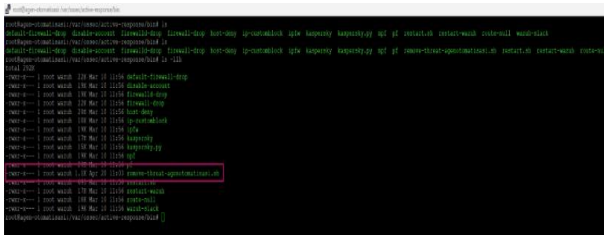
Pada gambar 22 terlihat *playbook* yang digunakan mengkonfigurasi *wazuh agent* dengan alamat IP pada 192.168.244.146. Hasil dari eksekusi *playbook* tersebut seperti terlihat pada gambar 23.



Gambar 23. Hasil *Playbook Active Response* Saat Di Jalankan

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 1 Jan – Jun 2025, 63 - 66
Melek IT
INFORMATION TECHNOLOGY JOURNAL

Pada gambar 23 terlihat bahwa hasil uji coba menjalankan *playbook active response* telah berhasil tanpa ada *error*. Hasil dari *playbook* setelah berhasil dijalankan seperti terlihat pada gambar 24.

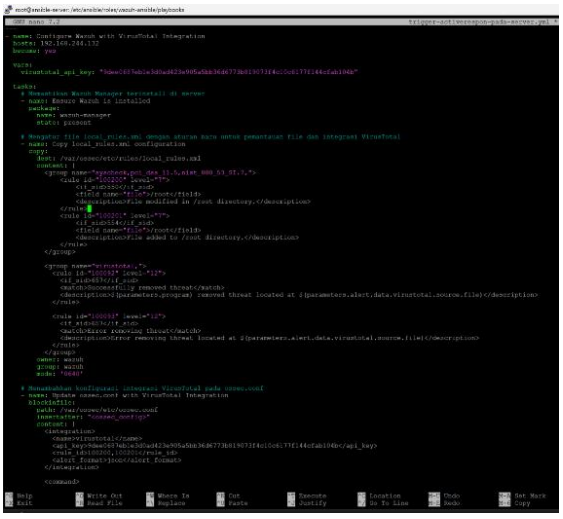


Gambar 24. Hasil Pada Otomatisasi Pada *agent*

Pada gambar 24 terlihat bahwa *file active response* telah berhasil dibuat menggunakan otomatisasi *ansible*.

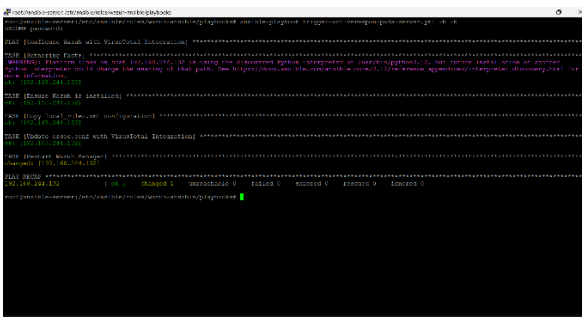
b. Uji coba *playbook* pada sisi *server*

Playbook yang digunakan pada sisi *server* yaitu *trigger-activerespon-pada-server.yml.* Cuplikan kode program pada *playbook* tersebut, seperti terlihat pada gambar 25.



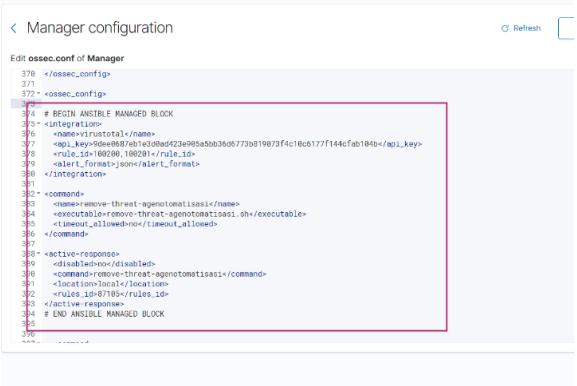
Gambar 25. *Playbook* Untuk Sisi *Server*

Pada gambar 25 merupakan *playbook* yang akan digunakan untuk sisi *server* yaitu target *hosts* 192.168.244.132 yang dimana merupakan IP dari *wazuh server*. Hasil dari *playbook* ketika dijalankan terlihat pada gambar 26.



Gambar 26. Hasil Menjalankan *Playbook* Sisi *Server*

Pada gambar 26 merupakan hasil menjalankan *playbook trigger-activerespon-pada-server.yml* terlihat tidak ada *error* saat menjalankan *playbook* yang dimana *task* dari *playbook* tersebut meliputi konfigurasi *local_rule.xml* dan *ossec.conf* untuk integrasi *virustotal* dan memicu *active response* yang telah dibuat pada sisi *agent*.

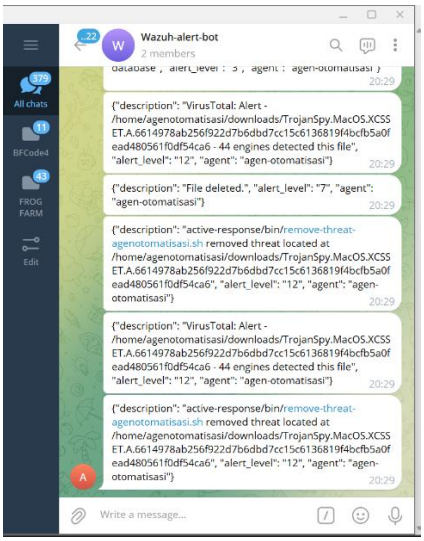


Gambar 27. Hasil Otomatisasi *Ossec.Conf*

Pada gambar 27 merupakan hasil konfigurasi yang ditambahkan melalui otomatisasi *ansible*, uji coba selanjutnya mengunduh *file malware* pada *agent* otomatisasi untuk memastikan bahwa otomatisasi yang dilakukan berhasil dan bekerja tanpa melakukan konfigurasi manual, adapun uji coba yang dilakukan terlihat pada gambar 28.

Gambar 28. Uji Coba Unduh *Malware* Setelah Otomatisasi

Pada gambar 28 terlihat uji coba yang dilakukan terkait mengunduh salah satu *malware* yaitu *trojan*. Hasil eksekusi perintah *ls* pertama menunjukkan *file* tersebut berhasil masuk ke sistem. Sedangkan hasil ekekusi *ls* kedua menunjukkan *file* telah terhapus secara otomatis. Hal tersebut menandakan bahwa uji coba konfigurasi secara otomatisasi telah berhasil. Sedangkan notifikasi berupa *alert* yang dikirimkan, seperti terlihat pada gambar 29.



Gambar 29. Alert Pada *Telegram* Setelah Otomatisasi

Pada gambar 29 terlihat bahwa *alert* yang diberikan pada *telegram remove-threat-agenotomatisasi.sh* yang sebelumnya di konfigurasi secara otomatisasi dengan *ansible* berhasil mendeteksi dan menghapus *file* yang terindikasi *malware*, ini menandakan semua *playbook* yang dijalankan telah berhasil seperti konfigurasi secara manual.

Dari hasil otomatisasi tersebut maka dapat disimpulkan bahwa semua *playbook* berhasil melakukan otomatisasi dan berjalan sesuai konfigurasi manual. Otomatisasi yang dilakukan sangat efisien dan tidak memerlukan melakukan konfigurasi dua arah baik dari *agent* ke *server* maupun sebaliknya. Diperlukan satu perintah melalui *ansible server* agar semua konfigurasi yang dilakukan secara manual dapat terselesaikan secara otomatis melalui *playbook* yang telah dikonfigurasi.

3.5 Hasil Analisis

Berdasarkan uji coba yang telah dilakukan maka dapat diperoleh hasil analisa sebagai berikut:

a. Konfigurasi yang dilakukan secara manual membutuhkan waktu yang terbilang cukup lama terutama konfigurasi integrasi *virustotal.*
b. 3 (tiga) jenis *malware* yang cukup berbahaya mampu diatasi oleh *active response*.
c. Otomatisasi hanya dapat dilakukan pada *agent* yang menggunakan sistem operasi berbasis *linux*.
d. Integrasi dengan *telegram bot* mampu memberikan *alert* lebih sederhana dan fleksibel untuk dianalisis.

Hasil analisis dari uji coba *active response* terhadap *malware,* seperti terlihat pada tabel 1.

Tabel 1. Analisa *active response* terhadap 3 jenis *malware*

Jenis *malware*	**Waktu *active-response* merespon *malware***	**Mendapatkan *alert***
Trojan	14 detik setelah terunduh	✓
Ransomware	15 detik setelah terunduh	✓
Worm	9 detik setelah teunduh	✓

Berdasarkan tabel 1 mak dapat disimpulkan bahwa active response terhadap malware yang berbeda, memiliki jeda waktu yang berbeda baik ketika terdeteksi maupun terhapus. Hal tersebut dipengaruhi oleh kualitas koneksi internet dan batas waktu penggunaan API. sehingga waktunya dapat berbeda-beda.

Tabel 2. Analisa Konfigurasi Yang Bekerja

Konfigurasi	**Kelebihan**	**Kekurangan**
Virustotal	Menganalisis *file* mencurigakan dan meningkatkan akurasi deteksi *malware* termasuk *zero-day threats*	Tergantung pada koneksi *internet* dan batas penggunaan *API*
Active response	Memberi respon otomatis terhadap ancaman, seperti menghapus *file* atau mencatat *log*, serta meningkatkan perlindungan sistem secara *real-time.*	Membutuhkan pengujian menyeluruh untuk mencegah *false positive* yang merusak sistem.
Otomatisasi *ansible*	Mengotomatisasi instalasi dan konfigurasi *wazuh* serta integrasi *virustotal.* Konsistensi konfigurasi di banyak *server*, menghemat waktu dan memudahkan penskalaan.	Memerlukan validasi konfigurasi setelah *deployment* untuk mencegah kesalahan konfi-gurasi. Hanya dapat mengotomatisasi otomatisasi pada sistem operasi *linux.*

Berdasarkan tabel 2 maka dapat disimpulkan kelebihan dan kekurangan dari konfigurasi yang telah di uji coba. Terlihat masih banyak kekurangan terutama otomatisasi yang hanya bisa dilakukan pada sistem

pISSN: 2442-3386 eISSN: 2442-4293
Vol 11 No 1 Jan – Jun 2025, 65 - 66
Melek IT
INFORMATION TECHNOLOGY JOURNAL

operasi *linux*. Selain itu masih dilakukan validasi secara manual setelah otomatisasi dijalankan guna mencegah kesalahan.

IV. PENUTUP

4.1. Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang dilakukan, dapat disimpulkan bahwa otomatisasi konfigurasi *wazuh* menggunakan *ansible* berhasil diterapkan dalam sistem keamanan jaringan, yang mempercepat proses instalasi dan konfigurasi *wazuh agent* serta integrasinya dengan layanan *virustotal*. Integrasi ini mampu mendeteksi *file* yang terindikasi sebagai *malware* berdasarkan hash *file* dan memberikan respons aktif secara otomatis. Sistem juga dilengkapi dengan notifikasi *real-time* melalui *telegram bot* yang meningkatkan kecepatan respon terhadap ancaman siber. Pengujian terhadap tiga jenis *malware* yaitu *trojan*, *ransomware*, dan *worm* menunjukkan bahwa sistem mampu mendeteksi dan menghapus *file* berbahaya secara efektif dan efisien. Selain itu, penggunaan *ansible* terbukti mengurangi waktu konfigurasi secara signifikan serta meminimalkan potensi *human error* dalam proses *deployment* di lingkungan jaringan virtual.

4.2. Saran

Adapun saran-saran yang dapat di berikan untuk pengembangan skripsi lebih lanjut adalah sebagai berikut:

a. Mengintegrasikan *wazuh* dengan sistem keamanan selain *virustotal*.
b. Mengembangkan sistem instalasi dan otomatisasi pada *wazuh agent* berbasis *windows*.
c. Mengujicoba pada lingkungan jaringan nyata (non-virtual) agar efektivitas sistem dapat dievaluasi dalam skala operasional yang lebih besar.

DAFTAR PUSTAKA

[1] E. Valdis Tjahjadi And B. Santoso, "Klasifikasi Malware Menggunakan Teknik Machine Learning," *Copyright @Balok*, Vol. 2, No. 1, 2023, [Online]. Available: Https://Www.Kaggle.Com/Datasets/Amauricio/Pe-Files-Malwares.

[2] 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," *Jurnal Informatika Dan Teknik Elektro Terapan*, Vol. 13, No. 1, Jan. 2025, Doi: 10.23960/Jitet.V13i1.5686.

[3] F. A. Saputra *Et Al.*, "Jurnal Informatika Terpadu Implementasi Wazuh Siem Untuk Manajemen Log Event Di Pesantren Teknologi Informasi Dan Komunikasi Jombang," *Jurnal Informatika Terpadu*, Vol. 10, No. 2, Pp. 146–155, 2024, [Online]. Available: Https://Journal.Nurulfikri.Ac.Id/Index.Php/Jit

[4] P. Sistem Pelayanan Dinas Kependudukan Dan Pencatatan Sipil Yesi Nurhana Dalimonthe, A. Dina Kalifia, S. Diwandari, F. Sains Dan Teknologi, And U. Teknologi Yogyakarta, "Pemanfaatan Api (Application Programming Interface) Untuk," *Jurnal Tekinkom*, Vol. 6, No. 2, P. 2023, Doi: 10.37600/Tekinkom.V6i2.1053.

[5] J. Misquitta And A. K, "A Comparative Study Of Malicious Url Detection: Regular Expression Analysis, Machine Learning, And Virustotal Api," Dec. 01, 2023. Doi: 10.21203/Rs.3.Rs-3685949/V1.

[6] B. Haryanto And D. W. Chandra, "Implementasi Wazuh Integritas File Untuk Perlindungan Keamanan Berdasarkan Aktivitas Log Di Btsi Uksw," *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, Vol. 5, No. 1, Pp. 183–192, Jan. 2024, Doi: 10.35870/Jimik.V5i1.447.

[7] A. Shafiyyah, G. F. Nama, And R. A. Pradipta, "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*, Vol. 12, No. 2, Apr. 2024, Doi: 10.23960/Jitet.V12i2.4074.

[8] M. Zulfikri, M. Syahrir, W. Kusuma, M. Z. Program, And S. T. Informasi, "Pelatihan Implementasi Security Event Monitoring Berbasis Wazuh/Siem Pada Aplikasi Command Center Pemerintah Provinsi Nusa Tenggara Barat," 2025.

[9] M. Dehan Pratama, F. Nova, And D. Prayama, "Wazuh Sebagai Log Event Management Dan Deteksi Celah Keamanan Pada Server Dari Serangan Dos." [Online]. Available: Http://Jurnal-Itsi.Org

[10] D. P. Penyebaran, M. Menggunakan, W. Denny, P. Widyantono, And W. Sulistyo, "Pemodelan Instrusion Prevention System Untuk Pendeteksi," 2023. [Online]. Available: Https://Journal-Computing.Org/Index.Php/Journal-Ita/Index

[11] A. Andika And R. Efendi, M.Kom, "Simulasi Dan Analisis Efektivitas Sistem Keamanan Jaringan Menggunakan Intrusion Prevention System (Ips) Berbasis Wazuh," *Jurnal Pendidikan Teknologi Informasi (Jukanti)*, Vol. 8, No. 1, Pp. 17–24, Apr. 2025, Doi: 10.37792/Jukanti.V8i1.1454.

[12] Y. Dwi *Et Al.*, "Analisis Malware Menggunakan Metode Analisis Statis Dan

Dinamis Untuk Pembuatan Ioc Berdasarkan Stix Versi 2.1."

[13] M. A. Fahrudi And I. M. Suartana, "Integrasi End-Point Security Berbasis Agent Dan Bot Messenger Untuk Deteksi Dan Monitoring Serangan Pada Web Server Secara Real-Time," *Journal Of Informatics And Computer Science*, Vol. 04, 2023.

[14] J. Elektro *Et Al.*, "Serta Analisis Malware Menggunakan Malware Analysis System Implementation Of The Local Network Security System Using A Honeypot Dionaea, And Ids, Also Malware Analysis Using Malware Analysis System."

[15] R. Danil Fajri And R. Djutalov, "Implementasi Jaringan Hotspot Menggunakan Mikrotik Untuk Rt Rw.Net Dengan Menggunakan Metode Network Development Life Cycle (Ndlc) Pada Kampung Kelapa Indah Tangerang." [Online]. Available: Https://Journal.Mediapublikasi.Id/Index.Php/Logic

[16] D. Suryono And D. W. Chandra, "Analisis Keamanan Jaringan Hardware Trojan Pada Iot," *Jurnal Teknik Informatika Dan Sistem Informasi*, Vol. 9, No. 4, 2022, [Online]. Available: Http://Jurnal.

[17] M. Rijal *Et Al.*, "Perbandingan Kinerja Metode Seleksi Fitur Untuk Mendeteksi Aktivitas Trojan Performance Comparison Of Feature Selection Methods For Detecting Trojan Activity."

[18] A. B. Siber, S. Negara, B. Siber, And J. S. Pramudito, "Metode Cepat Identifikasi Dan Mitigasi Malware Ransomware Ketika Terjadi Serangan Siber Ramadhan Ibrahim."

[19] T. A. Cahyanto, V. Wahanggara, D. Ramadana, And J. T. Informatika, "Analisis Dan Deteksi Malware Menggunakan Metode Malware Analisis Dinamis Dan Malware Analisis Statis."

[20] S. Sumarno, "Analisis Cara Kerja Sistem Deteksi Infeksi Worm Pada Komputer," *Metik Jurnal*, Vol. 7, No. 2, Pp. 93–100, Dec. 2023, Doi: 10.47002/Metik.V7i2.636.