Department of Digital Business
Journal of Artificial Intelligence and Digital Business (RIGGS)
Homepage: https://journal.ilmudata.co.id/index.php/RIGGS
Vol. 4 No. 2 (2025) pp: 97-111
P-ISSN: 2963-9298, e-ISSN: 2963-914X

Integration of Security Information and Event Management and WhatsApp Bot to Improve Server Security in Real-time

Lilik Widyawati[1], Yogi Surya Prana[2], Muhamad Azwar[3]
[123]Ilmu Komputer, Fakultas Teknik, Universitas Bumigora
lilikwidya@universitasbumigora.ac.id[1] , yogiyogi.prana43@gmail.com[2] , muhamadazwar@universitasbumigora.ac.id [3]

Abstrak

Kemajuan teknologi informasi yang pesat telah mendorong transformasi digital, namun juga membawa tantangan baru dalam hal keamanan siber, Ancaman seperti Cross-Site Scripting (XSS), SQL Injection, dan information disclosure menjadi semakin signifikan. Meskipun sistem Security Information and Event Management (SIEM) seperti Wazuh telah digunakan untuk mendeteksi dan merespons serangan, banyak dari sistem ini yang masih mengandalkan dashboard manual, yang mengurangi efisiensi dan waktu respons terhadap ancaman. Penelitian ini bertujuan untuk mengembangkan sistem monitoring serangan siber dengan mengintegrasikan Wazuh dengan WhatsApp Bot untuk memberikan pemberitahuan real-time, sehingga mempercepat respon terhadap insiden keamanan. Metode yang digunakan dalam penelitian ini adalah Network Development Life Cycle (NDLC), yang melibatkan tahapan analisis kebutuhan, desain sistem, simulasi prototipe, dan pengujian sistem. Pada tahap analisis, data mengenai teknologi terkini dalam manajemen keamanan siber dikumpulkan melalui studi literatur. Sistem yang dirancang mengintegrasikan Wazuh dengan WhatsApp Bot, memungkinkan pemberitahuan serangan dikirimkan langsung ke WhatsApp administrator. Proses implementasi dilakukan dengan mengonfigurasi Wazuh Server dan WhatsApp Bot, diikuti dengan pengujian untuk menilai keefektifan sistem. Penelitian ini menunjukkan bahwa integrasi antara Wazuh dan WhatsApp Bot berhasil meningkatkan efisiensi pemberitahuan serangan. Rata-rata waktu pemberitahuan setelah implementasi sistem adalah 52 hingga 58 detik, lebih cepat dibandingkan dengan waktu pemberitahuan lebih dari satu menit pada sistem sebelumnya. Selain itu, sistem ini juga memungkinkan pengelolaan konfigurasi active-response secara otomatis melalui WhatsApp, yang memungkinkan administrator untuk merespons ancaman tanpa harus membuka dashboard. Pengujian juga menunjukkan bahwa pemblokiran serangan seperti Brute Force, SQL Injection, dan XSS dapat dilakukan dengan lebih cepat, dengan rata-rata waktu pemblokiran mencapai 3 menit 6 detik untuk Brute Force dan 1 menit 43 detik untuk SQL Injection. Sistem ini mengurangi ketergantungan pada dashboard manual dan meningkatkan respons terhadap serangan dengan mengotomatiskan proses konfigurasi dan pemblokiran. Dengan demikian, sistem ini dapat berkontribusi pada peningkatan pengelolaan keamanan siber yang lebih efisien dan efektif.

Kata kunci: Keamanan Siber, SIEM, Wazuh, WhatsApp Bot, Monitoring Server.

1. Latar Belakang

Seiring perkembangan zaman kemajuan teknologi pada saat ini berkembang dengan sangat pesat sehingga membuat banyak perubahan yang sangat besar pada aspek kehidupan kita. Akses internet banyak di manfaatkan oleh masyarakat untuk berbagai macam kebutuhan salah satu contohnya adalah sebagai sarana informasi sehingga masyarakat dapat di permudah dalam mendapatkan informasi dari berbagai hal di dunia(Cecep Abdul Cholik, 2021). Hal ini dapat dimanfaatkan oleh banyak instansi untuk memanfaatkan internet agar dapat meningkatkan kinerja dan efektivitas dalam mencapai tujuan organisasinya. Salah satu contoh dalam pemanfaatan tersebut adalah penerapan aplikasi web yang dapat digunakan untuk berbagi informasi, penyimpanan data dan juga memberikan layanan. Dalam membangun aplikasi web di pastikan kita harus memiliki sebuah server yang berguna untuk menampung semua data-data yang akan di masukan atau di publikasi nantinya. Data-data yang di tampung dapat berupa teks, suara, Gambar dan masih banyak lagi sesuai dengan fitur yang akan di buat oleh pengembang web tersebut(Alan Aldo Powa et al., 2021) .

Di era digital saat ini, kemajuan teknologi informasi telah membawa perubahan besar dalam hampir setiap aspek kehidupan, mulai dari bisnis, pemerintahan, hingga kehidupan pribadi. Salah satu dampak terbesar dari

transformasi digital ini adalah meningkatnya ketergantungan pada aplikasi berbasis web dan sistem informasi yang terhubung ke internet. Sistem-sistem ini digunakan untuk menyimpan dan mengelola data penting, yang mencakup informasi pribadi, finansial, dan sensitif lainnya. Oleh karena itu, menjaga keamanan data dan sistem menjadi hal yang sangat penting untuk mencegah kebocoran data dan serangan siber.

Di balik kemudahan dalam mengaksesnya, internet juga memiliki resiko ancaman bagi instansi yang memegang data sensitif seperti data instansi, data karyawan, data pribadi seseorang dan lainnya karena tidak luput dari kejahatan siber sehingga beberapa data yang di pegang oleh instansi baik itu negeri ataupun swasta menjadi terancam dalam pencurian data(Fathur, 2020).

Namun, kemudahan akses dan penggunaan internet juga membuka celah yang signifikan bagi serangan siber. Menurut laporan Badan Siber dan Sandi Negara (BSSN) tahun 2024, ancaman siber yang sering terjadi pada aplikasi web meliputi Cross-Site Scripting (XSS), SQL Injection, dan information disclosure. Ancaman ini dapat mengekspos data pribadi, mengakses sistem tanpa izin, bahkan merusak sistem yang telah dibangun dengan investasi yang besar. Serangan-serangan ini semakin berkembang dari waktu ke waktu, baik dalam hal kompleksitas maupun frekuensinya, sehingga menambah tantangan bagi organisasi dalam menjaga keamanan informasi dan infrastrukturnya(Id-SIRTII /CC, 2023). Ancaman siber dari tahun ke tahun semakin meningkat sehingga banyak dari organisasi yang mengimplementasikan layanan berbasis internet ini harus mencari cara untuk mengamankan layanan pada aplikasinya sehingga dapat mencegah terjadinya kebocoran data tersebut demi untuk mengurangi kerugian yang akan ditimbulkan nantinya(Jeklin et al., 2016).

Untuk menghadapi masalah ini, banyak organisasi mengadopsi Security Information and Event Management (SIEM) sebagai solusi untuk mendeteksi, menganalisis, dan merespons ancaman siber secara lebih efisien. SIEM seperti Wazuh, yang berfungsi untuk mengumpulkan dan menganalisis data log dari berbagai perangkat dan aplikasi, memungkinkan administrator untuk mendeteksi aktivitas mencurigakan dan potensi serangan. Penelitian sebelumnya menemukan bahwa Security Information and Event Management (SIEM) merupakan solusi terbaik bagi teknologi yang dapat menyatukan fungsi-fungsi keamanan informasi dan manajemen insiden dengan mengumpulkan informasi keamanan yang diperoleh dari data log jaringan, aplikasi, dan perangkat keras ke dalam satu platform yang terpadu(Sheeraz et al., 2023). Penelitian Selanjutnya menjelaskan Wazuh akan membantu peneliti untuk mengumpulkan informasi-informasi yang di dapatkan dari hasil monitoring, mendeteksi adanya intrusi pada lalu lintas jaringan pada paket yang mencurigakan dan juga dapat memberikan respon jika terjadinya penyeranggan sehingga akan memudahkan admin jika terjadi serangan pada server(Kamil et al., 2024)

Namun, meskipun sistem SIEM seperti Wazuh mampu memberikan informasi yang komprehensif tentang status keamanan sistem, sebagian besar dari sistem ini masih bergantung pada antarmuka dashboard yang harus dipantau secara manual oleh administrator. Hal ini dapat menimbulkan keterlambatan dalam merespons ancaman, terutama jika administrator tidak dapat segera mengakses dashboard atau berada di lokasi yang berbeda. Selain itu, meskipun SIEM menawarkan solusi untuk mendeteksi dan mengelola insiden, banyak sistem yang tidak memberikan pemberitahuan secara langsung dan otomatis kepada administrator ketika terjadi serangan(Urbanelli et al., 2024). Proses manual dalam memantau dan merespons serangan memperlambat respons keamanan, sehingga memberikan kesempatan bagi penyerang untuk mengakses atau merusak data lebih lanjut. Oleh karena itu, dibutuhkan sistem yang tidak hanya mendeteksi ancaman tetapi juga dapat mengirimkan pemberitahuan secara real-time kepada administrator untuk mempercepat respons dan meminimalkan kerusakan.

Sebagai bagian dari solusi untuk meningkatkan efisiensi sistem SIEM, penelitian ini mengusulkan integrasi Wazuh dengan WhatsApp Bot. WhatsApp Bot akan mengirimkan notifikasi secara langsung kepada administrator ketika terjadi ancaman atau serangan pada server, memungkinkan pengelolaan yang lebih cepat dan lebih mudah. Dengan pendekatan ini, sistem dapat mengotomatisasi beberapa tugas yang sebelumnya dilakukan secara manual, seperti pengelolaan konfigurasi active-response terhadap serangan, yang memungkinkan administrator untuk merespons ancaman dengan lebih cepat dan efektif. Manfaat dari implementasi Whatsapp bot pada sistem Wazuh ini adalah dapat mempercepat dan mempermudah pengguna untuk menghubungkan server ke dalam Wazuh dan juga untuk melakukan monitoring server atau melakukan pemblokiran dari serangan serangan yang di kirimkan oleh cracker secara real-time melalui Whatsapp sehingga pengguna tidak perlu lagi untuk mengakses halaman Dashboard Wazuh. Dengan begitu pengguna akan lebih cepat dalam melakukan konfigurasi.

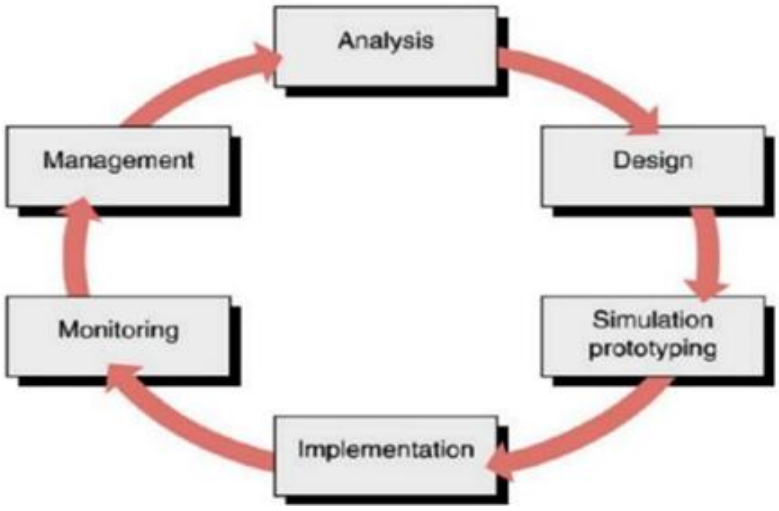
2. Metode Penelitian

2.1. Metode Pengembangan Sistem

Metode penelitian yang diterapkan dalam penelitian ini adalah Network Development Life Cycle (NDLC). NDLC merujuk pada pendekatan yang bergantung pada langkah-langkah pengembangan sebelumnya, termasuk perencanaan strategi bisnis, pengembangan aplikasi, dan analisis distribusi data. Terdapat 6 (enam) tahapan yang

DOI: https://doi.org/10.31004/riggs.v4i2.453

dimiliki oleh NDLC yaitu analysis, design, simulation prototyping, implementation dan monitoring serta management(Nirmalsari et al., 2023), metodologi Network Development Life Cycle (NDLC). NDLC merupakan sebuah siklus proses perancangan atau pengembangan suatu infrastruktur jaringan yang memungkinkan terjadinya pemantauan jaringan untuk mengetahui statistik dan kinerja jaringan(Miftahur Rahman et al., 2023).



Gambar 1 Metode NDLC

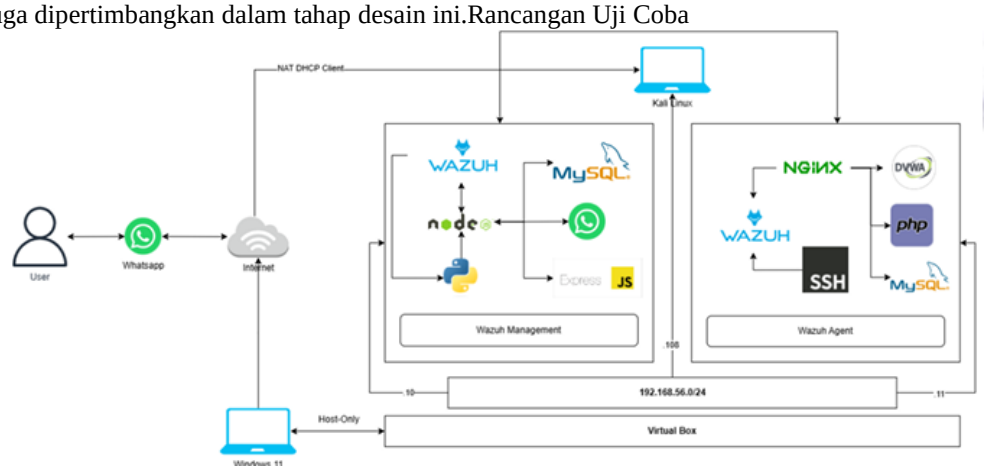
Penulis hanya menggunakan 3 (tiga) dari 6 (enam) tahapan pertama, yaitu:

1. *Analysis*

Tahap analisis, data terkait teknologi terkini dalam manajemen dan *monitoring* server dari serangan *cracker* dikumpulkan melalui studi literatur. Artikel ilmiah, buku, *paper*, dan sumber informasi lainnya digunakan sebagai referensi untuk memahami konsep- konsep terkait penggunaan *Security Information and Event Management* (SIEM) serta integrasi dengan bot *messenger* seperti *WhatsApp Bot.*

2. *Design*

Tahap desain melibatkan perancangan sistem monitoring keamanan server yang terintegrasi dengan *WhatsApp Bot* berdasarkan hasil analisis data. Rancangan ini mencakup perancangan uji coba, sistem integrasi bot, serta pengalokasian alamat IP. Selain itu, kebutuhan perangkat keras (*Hardware*) dan perangkat lunak (*Software*) yang diperlukan juga dipertimbangkan dalam tahap desain ini.Rancangan Uji Coba



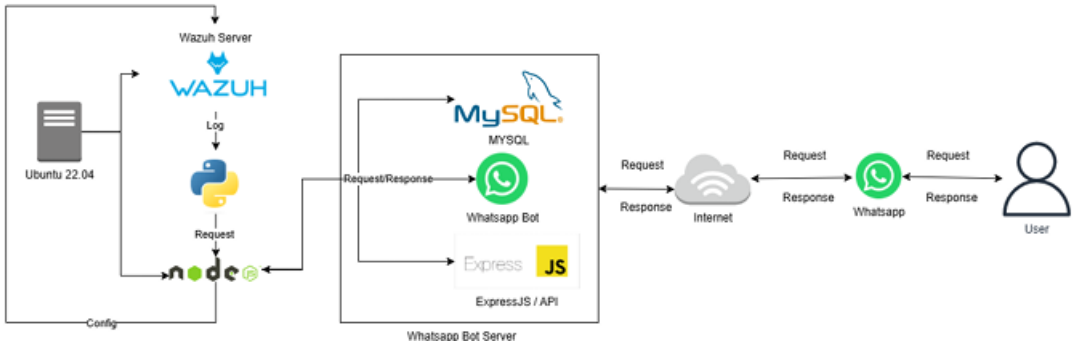
Gambar 2 Rancangan Uji Coba

Pada gambar 2. terlihat rancangan uji coba terdapat 1 laptop yang terinstal beberapa host virtual yang akan digunakan sesuai dengan fungsinya masing-masing. Terdapat 3 mesin yang di gunakan, yaitu Wazuh Agent

DOI: https://doi.org/10.31004/riggs.v4i2.453

dengan alamat IP 192.168.56.11/24 sebagai server yang menampung berbagai layanan Web Server, Wazuh Management dengan alamat IP 192.168.56.10/24 digunakan sebagai Wazuh Server yaitu server yang mengelola log-log dari Wazuh Agent yang telah dihubungkan sehingga Wazuh Server dapat mengidentifikasi suatu aktivitas yang mencurigakan pada server yang telah dihubungkan dan Wazuh Management juga menjalankan program Whatsapp BOT API yang digunakan untuk mengirimkan data log ke dalam Whatsapp sekaligus melakukan otomatisasi jika pengguna mengirimkan perintah sesuai yang di tentukan, Kali Linux dengan alamat IP 192.168.56.108/24 berperan untuk melakukan pengujian/penyerangan pada Wazuh Agent

A. Rancangan Sistem Integrasi BOT



Gambar 3 Rancangan Sistem Integrasi Bot

Pada gambar 3 Wazuh mengirimkan data alert berformat JSON ke dalam program integrasi Whatsapp Bot yang ada pada Wazuh Server kemudian data tersebut akan di kirimkan ke pada pengguna melalui Whatsapp Bot Server yang sudah di integrasikan pada Wazuh Server sebelumnya. Selain itu, pengguna juga mampu mengirimkan request untuk melakukan otomatisasi dari Whatsapp Bot ke dalam Wazuh Server untuk menambah atau menghapus konfigurasi active response.

B. Rancangan Pengalamatan IP

Untuk pengalamatan IP Address pada rancangan uji coba monitoring serangan server menggunakan Whatsapp Bot menggunakan tiga alamat IP class C yaitu 192.168.56.10/24, 192.168.56.11/24 dan 192.168.56.108/24. Detail alokasi pengalamatan IP Address pada masing-masing perangkat terlihat pada Tabel 1.

Tabel 1 IP Address

No.	Nama Perangkat	Alamat IP	*Subnetmask*
1	*wazuh-management*	192.168.56.10	255.255.255.0
2	*wazuh-agent*	192.168.56.11	255.255.255.0
3	*Kali Linux OS*	192.168.56.108	255.255.255.0

3. *Simulation Prototyping*

Implementasi sistem dilakukan dalam bentuk simulasi prototipe dengan menggunakan perangkat lunak virtualisasi. Pada tahap ini, dilakukan instalasi dan konfigurasi perangkat pendukung, uji coba, serta analisis hasil uji coba. Pengujian dilakukan dengan memanfaatkan lingkungan simulasi yang terkontrol, seperti penggunaan Virtual Box untuk menyiapkan lingkungan virtual dengan berbagai konfigurasi yang telah direncanakan sebelumnya.

A. Tahap Instalasi dan Konfigurasi

Pada tahap instalasi dan konfigurasi di bagi menjadi 2 tahap yaitu tahap instalasi dan konfigurasi *Host wazuh-management* dan *Host wazuh-agent.*

a. Instalasi dan Konfigurasi *Host wazuh-management*, Konfigurasi pada Host wazuh-management meliputi langkah pengaturan alamat IP pada interface enp0s3 dengan mengatur alamat IP menjadi static. Instalasi dan konfigurasi Wazuh Server meliputi konfigurasi pada ossec.config dan file integrasi Whatsapp Bot sebagai media monitoring pada sosial media Whatsapp.

DOI: https://doi.org/10.31004/riggs.v4i2.453

b. Konfigurasi pada *Host wazuh-agent* meliputi langkah pengaturan alamat IP pada *interface enp0s3* dengan mengatur alamat IP menjadi *static*. Instalasi *Wazuh Agent* meliputi perintah-perintah yang di masukan berdasarkan *command* yang di dapat pada *Wazuh Dashboard.*

B. Skenario Uji Coba

Pada tahap ini akan di lakukan uji coba terhadap sistem *Whatsapp Bot* dan *monitoring* menggunakan beberapa skenario uji coba diantaranya yaitu menambahkan *agent* baru ke dalam *Wazuh Server* melalui *Whatsapp Bot* berdasarkan data yang di kirimkan, menghubungkan *Agent* baru ke dalam *Wazuh Server* melalui perintah yang di berikan *Whatsapp Bot*, melihat status dari *agent*, menambahkan atau menghapus konfigurasi *Active Response* melalui otomatisasi *Whatsapp Bot* berdasarkan data yang dikirimkan, *monitoring* serangan SQL *Injection*, *Brute Force Attack SSH* dan *Cross Site Scripting* yang akan di kirimkan melalui *Whatsapp Bot* secara *real-time* dengan mengirimkan data-data yang telah di dapatkan dari hasil log yang di berikan oleh *Wazuh Server* dengan memberikan hasil uji coba serangan setelah dan sebelum di tambahkan konfigurasi *Active Response* yang telah di lakukan melalui *Whatsapp Bot* sehingga menampilkan perbedaan hasil serangan pada kedua sesi.

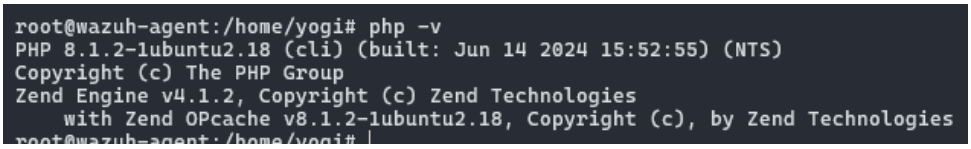
3. Hasil dan Pembahasan

3.1 Hasil Instalasi Packages pada *Wazuh Agent*

Wazuh Agent perlu di lakukan instalasi & konfigurasi menggunakan beberapa *packages* terlebih dahulu agar dapat melakukan uji coba penyerangan. Beberapa *packages* yang di instal digunakan sebagai pendukung dalam pengujian sistem pada server nantinya.

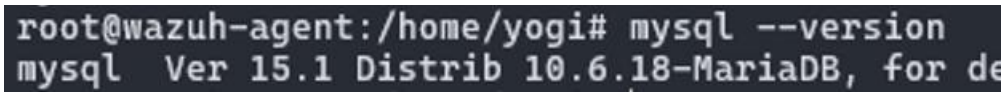
Gambar 4 Hasil Instalasi Nginx

Pada Gambar 4 merupakan hasil instalasi *Nginx* yang di gunakan sebagai web server pada server *Wazuh Agent* sehingga dapat menampung beberapa *file* web yang akan digunakan untuk uji coba penyerangan.



Gambar 5 Hasil instalasi PHP

Pada Gambar 5 merupakan hasil instalasi dari PHP yang digunakan untuk membaca *script* dari program DVWA yang sudah di sediakan pada halaman web.



Gambar 6 Hasil Instalasi *Mysql*

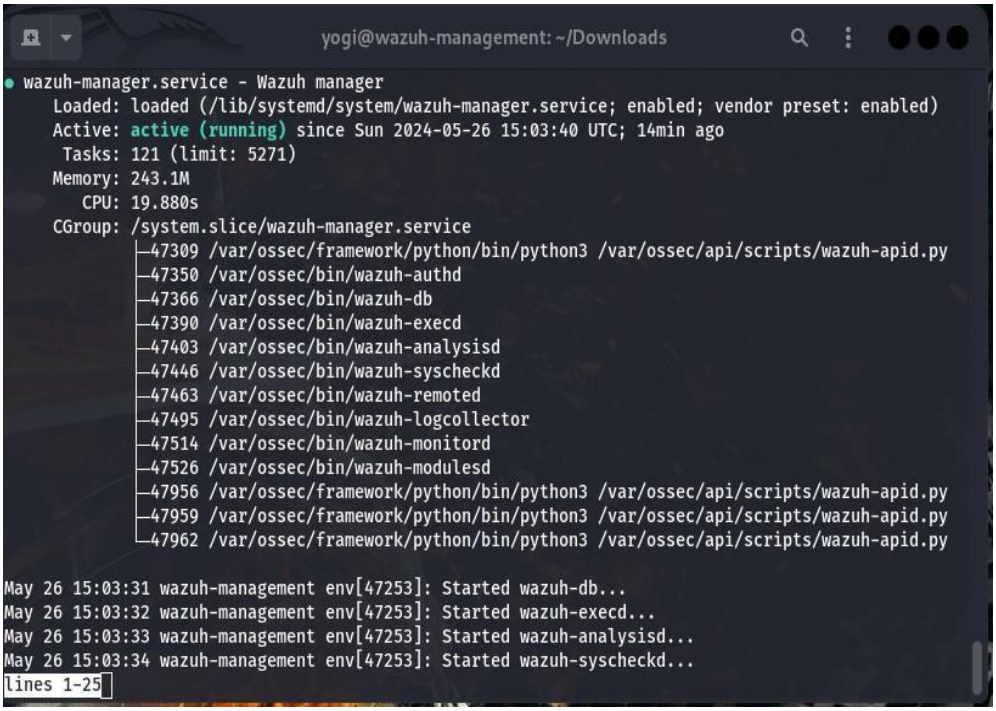
Pada Gambar 6 merupakan hasil instalasi dari *mysql* yang di gunakan sebagai database untuk menampung data-data dari program DVWA.

DOI: https://doi.org/10.31004/riggs.v4i2.453

Gambar 7 Hasil Instalasi DVWA

Pada Gambar 7 merupakan hasil instalasi dari program *Damn Vulnerable Web Application* (DVWA) yang sudah di lakukan. Terlihat halaman DVWA berhasil di akses dan dapat digunakan sebagai alat untuk melakukan uji coba serangan pada *SQL Injection* dan *Cross Site Scripting* (XSS).

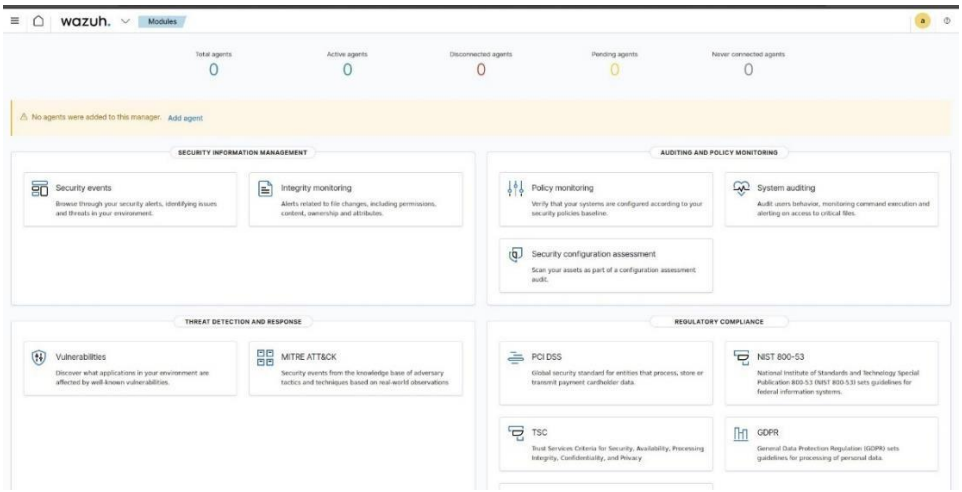
3.2 Hasil *Instalasi Wazuh Manager*



Gambar 8 Status *Wazuh Manager*

Terlihat pada Gambar 8 status dari *Wazuh Manager* telah aktif (*running*) sehingga dapat di simpulkan bahwa *Wazuh Manager* telah berjalan pada server.

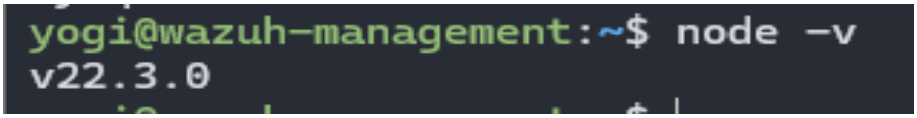
DOI: https://doi.org/10.31004/riggs.v4i2.453



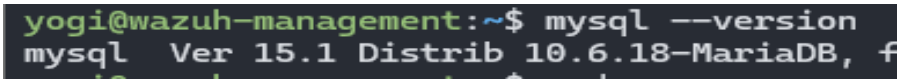
Gambar 9 Dashboard Wazuh

Terlihat pada Gambar 9 merupakan tampilan dari halaman *Wazuh Dashboard* yang merupakan halaman utama dari seluruh manajemen yang sudah di sediakan oleh Wazuh Manager. Halaman tersebut di gunakan untuk menjalankan semua fitur yang ada pada *Wazuh Manager.*

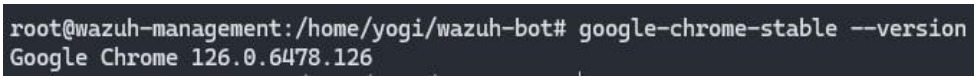
3.3 Instalasi & Konfigurasi *Whatsapp* API



Gambar 10 Package NodeJS



Gambar 11 *Package Mysql*



Gambar 12 Google Chrome

Gambar 13 *Package Python3*

Pada Gambar 10, Gambar 11, Gambar 12 dan Gambar 13 merupakan beberapa *package* yang di butuhkan untuk menjalankan program *Whatsapp Bot.*

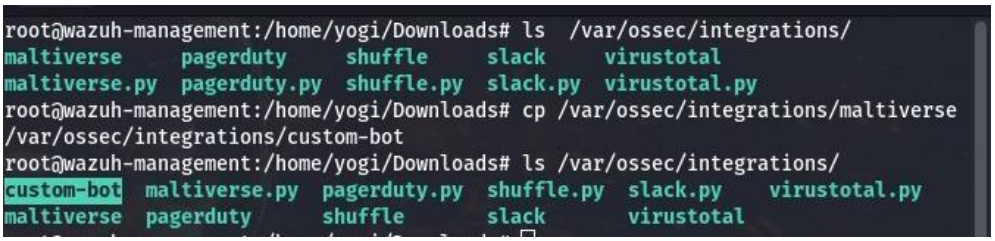
DOI: https://doi.org/10.31004/riggs.v4i2.453

Gambar 14 Program *Whatsapp Bot*

Terlihat pada Gambar 14 merupakan tampilan jika program *Whatsapp Bot* dijalankan. Saat pertama kali di jalankan program akan memberikan QR Code yang di gunakan untuk menghubungkan akun Whatsapp yang akan di jadikan *Whatsapp Bot*. File *Whatsapp Bot* dapat di unduh pada halaman : https://github.com/yogzsp/whatsapp-bot-wazuh.

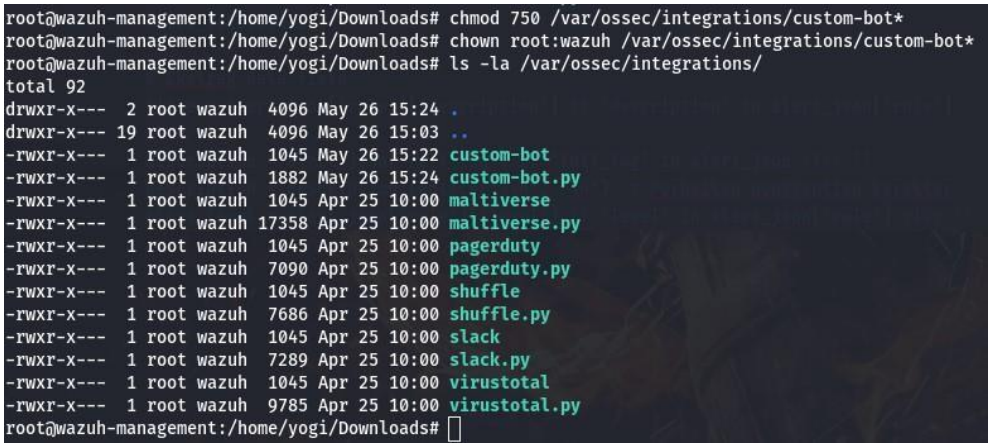
3.4. Membuat File Integrasi *Whatsapp Bot*

Langkah selanjutnya adalah membuat file konfigurasi *Whatsapp* Bot pada *Server Wazuh* Management agar data dari hasil *Monitoring Wazuh* dapat di integrasikan ke dalam *Whatsapp Bot.*



Gambar 15 Salin File Konfigurasi

Terlihat pada Gambar 15 adalah proses menyalin file konfigurasi lainnya menjadi file konfigurasi *custom-bot*. Hal ini di lakukan karena file *bash* dari seluruh konfigurasi untuk *integration* adalah sama.

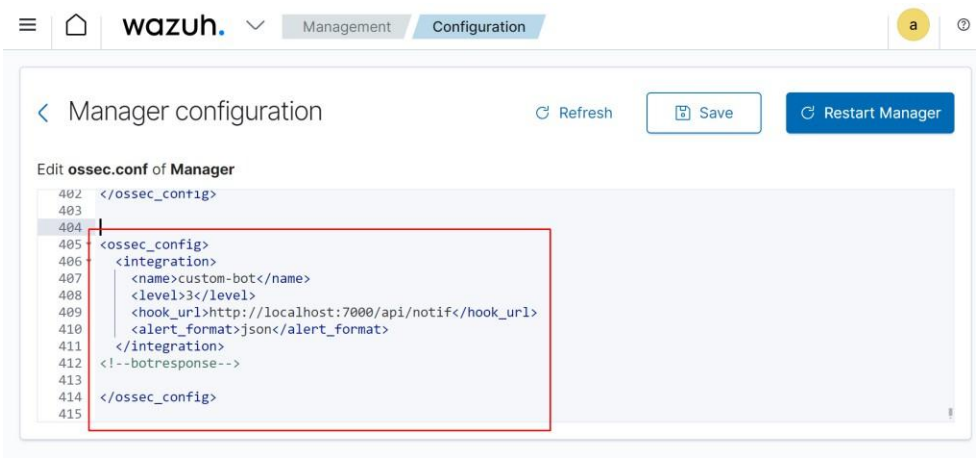


Gambar 16 Rubah Izin *File*

DOI: https://doi.org/10.31004/riggs.v4i2.453

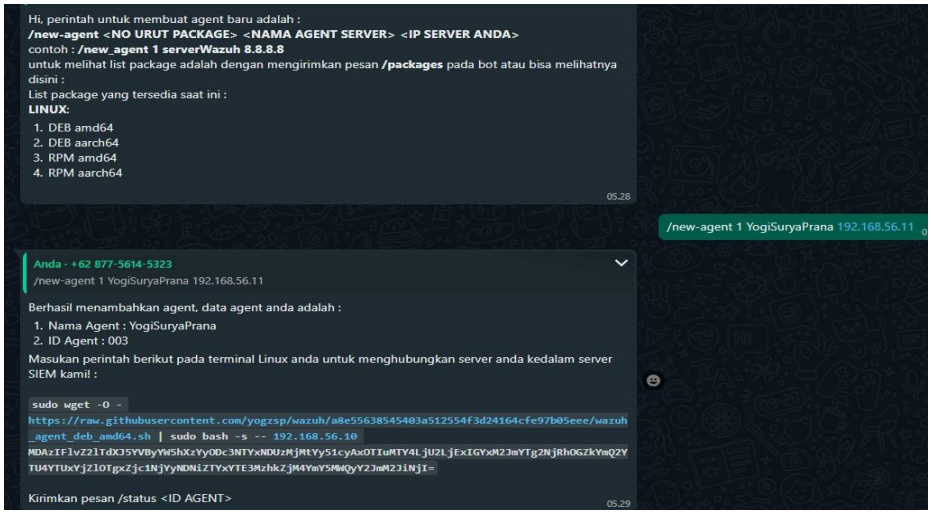
Gambar 16 merupakan langkah untuk mengubah izin dan kepemilikan dari file konfigurasi dan integrasi *Whatsapp Bot* yang memiliki nama dengan awalan *custom-bot*.

3.5. Mengkonfigurasi *Wazuh Management*



Gambar 17 Konfigurasi untuk *Whatsapp Bot*

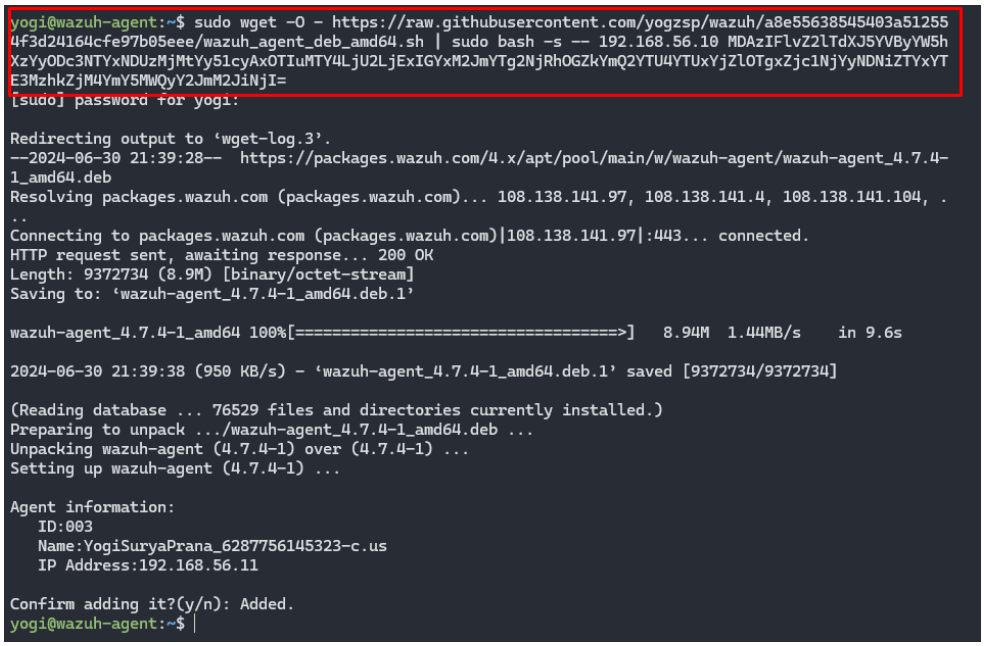
Gambar 17 merupakan hasil konfigurasi *ossec.conf* dari *Wazuh Manager* agar dapat mengintegrasikan data *alert* ke *Whatsapp Bot* sehingga *Whatsapp Bot* dapat memberikan notifikasi ancaman secara real-time kepada agent. Komentar *<!— botresponse-->* pada bagian paling bawah di gunakan untuk menandakan lokasi tempat menaruh konfigurasi *active-response* yang akan di buat bot nanti.



Gambar **18** Membuat *Agent* pada *Whatsapp Bot*

Terlihat pada Gambar 18 merupakan perintah yang digunakan untuk menambahkan *agent* baru pada *Wazuh Server*. Ketika perintah di kirimkan dengan benar maka *Whatsapp Bot* seperti alamat IP dan jenis *package linux* yang di gunakan maka *Whatsapp Bot* akan mengirimkan pesan berupa perintah yang harus di masukan ke dalam server *Agent*.

DOI: https://doi.org/10.31004/riggs.v4i2.453



Gambar 19 Instalasi *Wazuh Agent*

Pada Gambar 19 merupakan hasil dari instalasi *wazuh agent* menggunakan perintah yang di berikan oleh *Whatsapp Bot*. Setelah memasukkan perintah tersebut maka sistem akan melakukan instalasi *wazuh agent* sesuai dengan *package linux* yang di pilih.

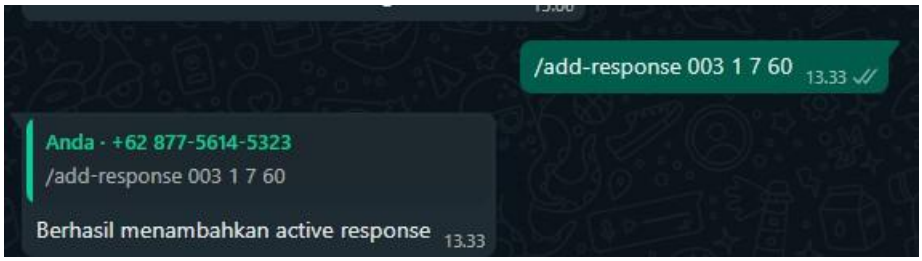
Gambar **20** Status *Wazuh Agent* Pada *Whatsapp Bot*

Terlihat pada Gambar 20 merupakan informasi status dari *agent* yang sudah di buat sebelumnya yang di tampilkan melalui *Whatsapp Bot* dengan menggunakan perintah "*/status <ID AGENT>"*.

3.5.1. Konfigurasi *active-response Wazuh Agent*

a. *Active-response* Anomali SSH

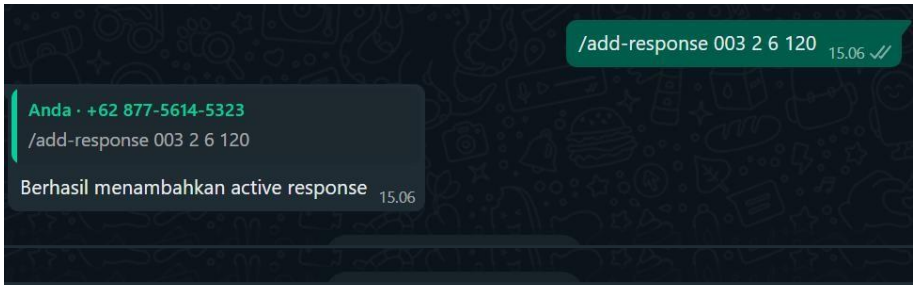
DOI: https://doi.org/10.31004/riggs.v4i2.453



Gambar 21 *Active-Response* anomali SSH

Perintah yang di masukan pada Gambar 21 merupakan perintah yang digunakan untuk menambahkan *active-response* terhadap serangan anomali SSH. Ketika *active-response* tersebut di aktifkan maka sistem akan memblokir alamat IP dari penyerang sehingga tidak dapat melanjutkan serangan seperti *Brute Force* SSH.

b. *Active-response SQL Injection & Cross Site Scripting (XSS)*

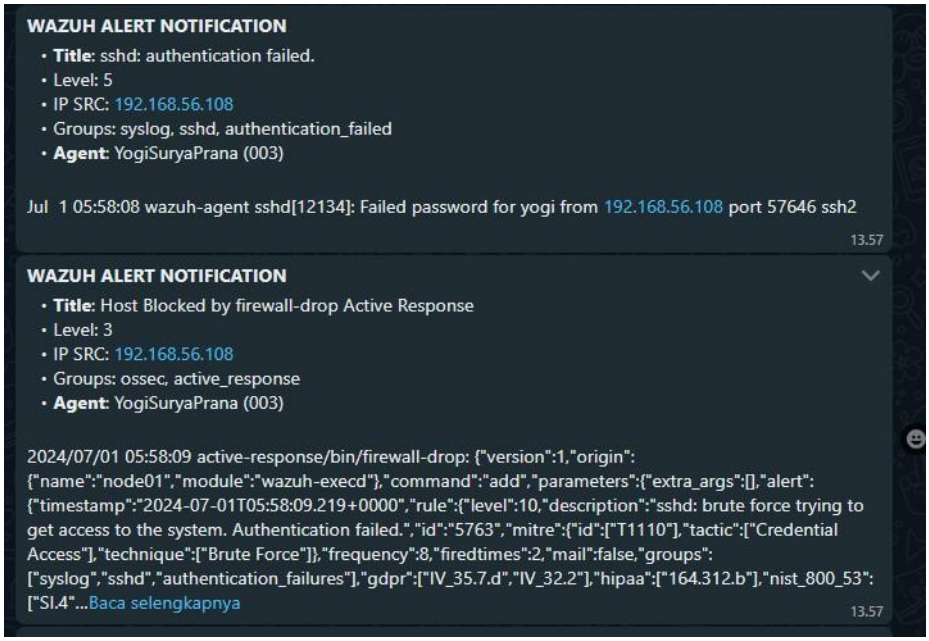


Gambar 22 *Active-Response SQL Injection & XSS*

Perintah yang di masukan pada Gambar 22 merupakan perintah yang digunakan untuk menambahkan *active-response* terhadap serangan pada web seperti SQL *Injection* dan *Cross Site Scripting* (XSS). Ketika *active-response* tersebut di aktifkan maka sistem akan memblokir alamat IP dari penyerang sehingga tidak dapat melanjutkan serangannya.

3.6. Uji Coba Serangan Setelah Konfigurasi Pada *Whatsapp Bot*

A. *Brute Force Attack*

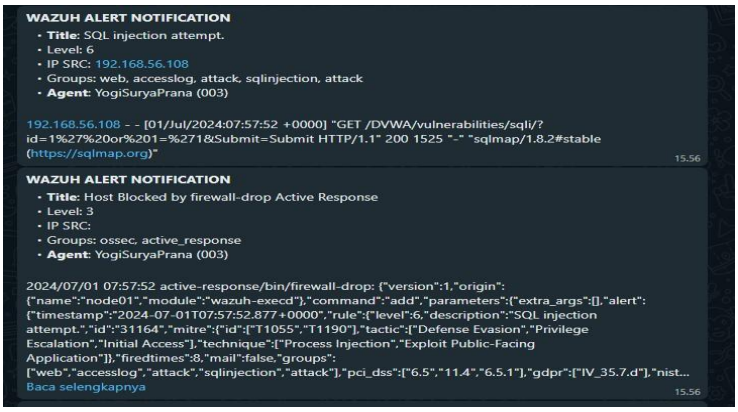


DOI: https://doi.org/10.31004/riggs.v4i2.453

Gambar 23 Respon Serangan *Brute Force*

Terlihat pada Gambar 23 Whatsapp Bot akan mengirimkan notifikasi berupa data berupa detail serangan, sumber agent yang mendapati serangan serta informasi tentang penyerang pada saat serangan Brute Force terjadi dan telah teridentifikasi oleh sistem Wazuh. Pada Gambar 23 Whatsapp Bot juga memberikan notifikasi tentang informasi pemblokiran terhadap penyerang.

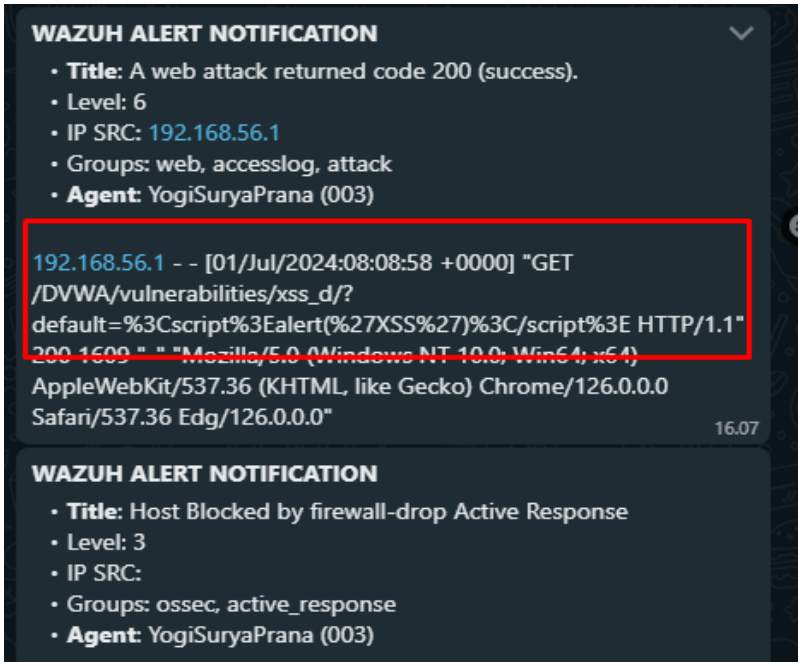
B. SQL *Injection*



Gambar **24** Respon Serangan SQL *Injection*

Terlihat pada Gambar 24 *Whatsapp Bot* akan mengirimkan notifikasi berupa data berupa detail serangan, sumber *agent* yang mendapati serangan serta informasi tentang penyerang pada serangan *SQL Injection* yang di mana pada keterangan yang di berikan penyerang berusaha memasukkan perintah yang bisa digunakan untuk memanipulasi kode dari *SQL* sehingga celah ini teridentifikasi oleh sistem *Wazuh*. Pada Gambar 24 *Whatsapp Bot* juga memberikan notifikasi tentang informasi pemblokiran terhadap penyerang.

C. *Cross Site Scripting (XSS)*



Gambar **25** Respon Serangan XSS

DOI: https://doi.org/10.31004/riggs.v4i2.453

Terlihat pada Gambar 25 *Whatsapp Bot* akan mengirimkan notifikasi berupa data berupa detail serangan, sumber *agent* yang mendapati serangan serta informasi tentang penyerang pada serangan *Cross Site Scripting* (XSS) yang di mana pada keterangan yang di berikan penyerang berusaha memasukkan perintah yang bisa digunakan untuk memanipulasi kode dari website berupa kode *javascript* sehingga celah ini teridentifikasi oleh sistem *Wazuh*. Pada Gambar 25 *Whatsapp Bot* juga memberikan notifikasi tentang informasi pemblokiran terhadap penyerang.

Berdasarkan serangan-serangan yang sudah di lakukan dapat di simpulkan bahwa *Whatsapp Bot* berhasil mengirimkan notifikasi berupa serangan dan juga pemblokiran secara *real-time*.

3.7. Hasil Analisa

Hasil uji coba yang sudah dilakukan akan dipaparkan pada tabel berikut dengan beberapa parameter yaitu jeda waktu penyerangan kemudian mendapatkan notifikasi sebelum diaktifkan respone, lalu tabel berikutnya yaitu jeda waktu penyerangan, keberhasilan serangan serta mendapatkan notifikasi ketika sudah diaktifkan respone. Berikut hasil analisis dari respon *Wazuh* dan *Whatsapp Bot* ketika melakukan uji coba penyerangan kepada server.

Tabel 2 Analisis serangan pada *Wazuh* sebelum menerapkan response

Jenis Serangan	Serangan ke	Jeda Waktu Penyerangan	Berhasil Di Serang	Mendapatkan Notifikasi
Brute Force SSH	1	53 Detik	✓	✓
	2	54 Detik	✓	✓
	3	50 Detik	✓	✓
SQL Injection	1	59 Detik	✓	✓
	2	58 Detik	✓	✓
	3	58 Detik	✓	✓
Cross Site Scripting (XSS)		-	✓	✓

Pada tabel 2. Analisis serangan pada Wazuh sebelum diaktifkan respone, peneliti menggabungkan 3 serangan didalam 1 tabel, pada serangan pertama yaitu Bruteforce SSH dilakukan sebanyak 3 kali uji coba, dari hasi 3 kali serangan uji coba rata-rata jeda waktu penyerangan 52 Detik atau kurang dari 1 menit untuk mendapatkan notifikasi bahwa telah terjadi penyerangan, begitu juga dengan serangan SQL Injection dari 3 kali ujicoba serangan rata-rata jeda waktu penyerangan 58 Detik atau kurang dari 1 menit untuk mendapatkan notifikasi bahwa telah terjadi penyerangan.

Tabel 3 Analisis serangan pada Wazuh setelah menerapkan response

Jenis Serangan	Serangan ke	Jeda Waktu Blokir	Berhasil Di Serang	Mendapatkan Notifikasi
Brute Force SSH	1	3 Menit 6 Detik	X	✓
	2	3 Menit 6 Detik	X	✓
	3	3 Menit 5 Detik	X	✓
SQL Injection	1	1 Menit 43 Detik	X	✓
	2	1 Menit 43 Detik	X	✓
	3	1 Menit 42 Detik	X	✓
Cross Site Scripting (XSS)		-	Berhasil Mengirimkan Request namun terkena blokir	✓

Pada tabel 3. Analisis serangan pada Wazuh setelah diaktifkan respone, peneliti menggabungkan 3 serangan didalam 1 tabel, pada serangan pertama yaitu Bruteforce SSH dilakukan sebanyak 3 kali uji coba,

DOI: https://doi.org/10.31004/riggs.v4i2.453

dari hasi 3 kali serangan uji coba tidak ada yang berhasil melakukan penyerangan dan rata-rata jeda waktu Blokir yaitu 3 menit 6 detik serta tetap mendapatkan notifikasi bahwa telah terjadi penyerangan dan sudah diblokir. Selanjutnya untuk serangan SQL Injection, dari 3 kali uji coba serangan tidak ada yang berhasil melakukan penyerangan dan rata-rata jeda waktu Blokir yaitu 1 menit 43 detik serta tetap mendapatkan notifikasi bahwa telah terjadi penyerangan dan sudah diblokir.
Hasil uji coba menunjukkan bahwa integrasi antara Wazuh dan WhatsApp Bot dapat meningkatkan efisiensi dalam merespons insiden keamanan dengan waktu pemberitahuan yang lebih cepat dan pengelolaan server yang lebih praktis. Penelitian ini memberikan solusi untuk mengatasi keterbatasan sistem SIEM yang belum mengintegrasikan notifikasi real-time yang mudah diakses oleh administrator, serta berkontribusi pada pengembangan sistem SIEM yang lebih adaptif dan efisien. Efisiensi yang dimaksud dalam konteks penelitian ini mencakup beberapa aspek penting:

Waktu Respons yang Lebih Cepat:
Sebelum implementasi WhatsApp Bot, sistem SIEM seperti Wazuh mengandalkan dashboard berbasis web yang harus dipantau secara manual oleh administrator. Proses ini membutuhkan waktu lebih lama untuk mendeteksi dan merespons ancaman, terutama jika administrator tidak berada di depan dashboard, dengan integrasi WhatsApp Bot, sistem mampu mengirimkan notifikasi serangan langsung ke ponsel administrator melalui aplikasi WhatsApp. Hasil pengujian menunjukkan bahwa notifikasi dapat diterima dalam waktu yang lebih cepat, ata-rata waktu pemberitahuan untuk serangan seperti Brute Force, SQL Injection, dan XSS adalah sekitar 52 detik hingga 58 detik, yang jauh lebih cepat dibandingkan metode manual yang mengharuskan admin membuka dashboard untuk mengecek status serangan.

Automatisasi Pengelolaan Konfigurasi:
Salah satu fitur utama dari sistem yang dikembangkan adalah kemampuannya untuk mengotomatisasi beberapa konfigurasi yang sebelumnya dilakukan secara manual melalui Wazuh Dashboard. Misalnya, administrator dapat menggunakan WhatsApp Bot untuk menambah atau menghapus konfigurasi active-response, seperti pemblokiran IP penyerang atau penyesuaian pengaturan pada Wazuh. Dengan otomatisasi ini, waktu yang diperlukan untuk merespons serangan berkurang signifikan, dan administrator dapat segera mengatasi ancaman tanpa harus masuk ke dashboard atau melakukan konfigurasi secara manual. Hal ini juga mengurangi potensi kesalahan manusia dalam pengelolaan konfigurasi keamanan.

Pemantauan dan Notifikasi Secara Real-time:
Sistem WhatsApp Bot yang diintegrasikan dengan Wazuh memungkinkan pemantauan serangan secara lebih efisien, administrator tidak lagi harus menunggu untuk membuka dashboard atau menunggu laporan periodik mengenai status server, notifikasi yang diterima di WhatsApp memberikan informasi langsung mengenai jenis serangan, sumber penyerangan, serta respons yang telah diambil (misalnya pemblokiran IP). Hal ini memungkinkan administrator untuk segera mengambil tindakan lebih cepat dan lebih tepat.

Peningkatan Efektivitas Pengelolaan Keamanan:
Efisiensi juga tercermin dari peningkatan efektivitas pengelolaan keamanan yang dihasilkan, dengan otomatisasi pemberitahuan dan respon terhadap insiden, waktu yang diperlukan untuk mendeteksi dan merespons ancaman dapat diminimalkan. Hal ini berpotensi mengurangi kerusakan atau kebocoran data yang diakibatkan oleh serangan siber.

4. Kesimpulan

Berdasarkan hasil uji coba yang dilakukan, penelitian ini menunjukkan bahwa integrasi antara Wazuh dan WhatsApp Bot berhasil meningkatkan efisiensi dalam deteksi dan respons terhadap serangan siber. Sebelum implementasi sistem, waktu yang dibutuhkan untuk menerima notifikasi mengenai serangan lebih dari satu menit, namun dengan sistem baru, notifikasi dapat diterima dalam waktu rata-rata 52 hingga 58 detik, yang menunjukkan peningkatan kecepatan pemberitahuan secara signifikan. Selain itu, sistem ini memungkinkan administrator untuk melakukan pengelolaan konfigurasi secara otomatis melalui WhatsApp Bot, seperti menambah atau menghapus active-response terhadap ancaman, yang sebelumnya harus dilakukan secara manual melalui dashboard Wazuh. Dengan demikian, proses pengelolaan menjadi lebih cepat dan efisien, mengurangi waktu yang dibutuhkan untuk merespons serangan. Penerapan fitur active-response dalam sistem ini juga terbukti efektif dalam mengurangi keberhasilan serangan. Dalam uji coba, serangan Brute Force, SQL Injection, dan XSS yang sebelumnya berhasil menerobos server, kini dapat diblokir dengan rata-rata waktu pemblokiran yang lebih cepat setelah konfigurasi active-response diaktifkan. Sebagai contoh, untuk serangan Brute Force SSH, tidak ada serangan yang berhasil setelah pemblokiran diterapkan, sementara untuk serangan SQL Injection

DOI: https://doi.org/10.31004/riggs.v4i2.453

dan XSS, sistem mampu memblokir penyerang sebelum mereka dapat mengakses data sensitif. Secara keseluruhan, integrasi WhatsApp Bot dengan Wazuh memberikan solusi yang lebih praktis dan efisien dalam pengelolaan keamanan server. Sistem ini memungkinkan administrator untuk mendapatkan informasi serangan secara real-time dan langsung bertindak untuk mencegah kerusakan lebih lanjut tanpa harus mengakses dashboard secara manual. Penelitian ini membuktikan bahwa sistem yang dikembangkan dapat meningkatkan efisiensi, efektivitas, dan keamanan secara keseluruhan dalam memantau dan mengelola ancaman siber.

Daftar Pustaka

Alan Aldo Powa, Johannis E. Kaawoan, & Fanley N. Pangemanan. (2021). Pemanfaatan Teknologi Dan Informasi di Dinas Komunikasi Dan Informatika Statistik Dan Persandian di Kabupaten Minahasa Tenggara. *Governance*, *1*(2), 1. https://ejournal.unsrat.ac.id/index.php/governance/article/view/35164%0Ahttps://ejournal.unsrat.ac.id/index.php/governance/article/viewFile/35164/32924

Cecep Abdul Cholik. (2021). Teknologi Informasi, ICT,. *Jurnal Fakultas Teknik*, *2*(2), 39–46.

Fathur, M. (2020). Tanggung Jawab Tokopedia Terhadap Kebocoran Data Pribadi Konsumen (Tokopedia's Responsibility for the Leakage of Consumers Personal Data). *Procceding: Call for Paper 2nd National Conference on Law Studies: Legal Development Towards A Digital Society Era*, 43–60. http://jurnal.unissula.ac.id/index.php/PH/article/view/1476

Id-SIRTII /CC. (2023). Lanskap Keamanan Siber Indonesia. *Id-SIRTII /CC*, *70*, 1–107.

Jeklin, A., Bustamante Farías, Ó., Saludables, P., Para, E., Menores, P. D. E., Violencia, V. D. E., Desde, I., Enfoque, E. L., En, C., Que, T., Obtener, P., Maestra, G. D. E., & Desarrollo, E. N. (2016). Implementasi Security Information and Event Management (Siem) Untuk Deteksi Dan Analisa Insiden Keamanan Pada Web Server. *Correspondencias & Análisis*, *15018*, 1–23.

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*. *2*, 80–92.

Miftahur Rahman, Ravi Budi Handwika, & Ahadini Izzatus Zahro. (2023). Penerapan Model Network Development Life Cycle (NDLC) Pada Infrastruktur Jaringan Internet Kantor Desa Kemiri. *Jurnal Publikasi Teknik Informatika*, *2*(3), 37–47. https://doi.org/10.55606/jupti.v2i3.1790

Nirmalsari, P., Fahmi, H., Fadli, S., Basuki Rahmat Praya Mataram, J., Praya, K., Lombok Tengah, K., & Tenggara Bar, N. (2023). Implementasi Metode Network Development Life Cycle Pada Rancang Bangun Jaringan Wireless Berbasis Mikrotik. *Jtmei)*, *2*(3), 72–87. https://doi.org/

Sheeraz, M., Paracha, M. A., Haque, M. U., Durad, M. H., Mohsin, S. M., Band, S. S., & Mosavi, A. (2023). Effective Security Monitoring Using Efficient SIEM Architecture. *Human-Centric Computing and Information Sciences*, *13*(September). https://doi.org/10.22967/HCIS.2023.13.023

Urbanelli, A., Frisiello, A., Bruno, L., & Rossi, C. (2024). The ERMES chatbot: A conversational communication tool for improved emergency management and disaster risk reduction. *International Journal of Disaster Risk Reduction*, *112*(August), 104792. https://doi.org/10.1016/j.ijdrr.2024.104792

DOI: https://doi.org/10.31004/riggs.v4i2.453