IMPLEMENTASI *SECURITY INFORMATION AND EVENT MANAGEMENT* (SIEM) UNTUK DETEKSI DAN ANALISA INSIDEN KEAMANAN PADA WEB SERVER

Muhammad Sofiyan Hadi; Devi Afriyantari Puspa Putri
Prodi Teknik Informatika, Fakultas Komunikasi dan Informatika, Universitas Muhammadiyah Surakarta

Abstrak

Website merupakan salah satu bentuk teknologi yang muncul dari perkembangan akses jaringan internet. Kemudahan yang ditawarkan pada website banyak dimanfaatkan oleh instansi maupun perusahaan sebagai sarana untuk berbagi informasi dan meningkatkan layanan. Hal ini karena website mampu menampilkan informasi teks, grafik, serta suara dari manapun melalui jaringan internet. Dibalik kemudahan tersebut, terdapat risiko ancaman keamanan siber dalam penggunaan teknologi berbasis internet karena dapat diakses dari manapun dan oleh siapapun, termasuk *threat actor* yang ingin mencuri informasi sensitif atau mengambil alih sistem. Dalam penelitian ini, penulis mencoba mengatasi permasalahan tersebut melalui implementasi SIEM menggunakan Wazuh yang dikombinasikan dengan Suricata serta terintegrasi dengan Telegram untuk mendeteksi adanya insiden keamanan pada web server lalu menganalisa insiden tersebut untuk mengetahui kebenarannya. Metode yang digunakan dalam penelitian ini adalah metode eksperimental dengan tahapan yang terdiri dari analisa kebutuhan, desain sistem, implementasi, pengujian, dan evaluasi. Berdasarkan hasil pengujian beberapa teknik serangan web server, sistem mampu mendeteksi adanya insiden keamanan terhadap web server kemudian meneruskan log insiden keamanan yang terdeteksi menuju ke *central engine* SIEM Wazuh sehingga dapat memunculkan *security alert* untuk dianalisa dari Wazuh *dashboard*. Hasil dari analisa *security alert* menunjukkan bahwa *security alert* yang muncul merupakan insiden yang benar terjadi dan bukan merupakan *false positive.*

Kata Kunci: keamanan siber, SIEM, IDS, wazuh, *monitoring*.

Abstract

Website is a form of technology that emerged from the development of internet network access. The convenience offered on the website is widely used by agencies and companies as a means to share information and improve services. This is because the website is able to display text, graphic and sound information from anywhere via the internet network. Behind this convenience, there is a risk of cybersecurity threats in the use of internet-based technology because it can be accessed from anywhere and by anyone, including threat actors who want to steal sensitive information or take over the system. In this study, the authors try to overcome these problems

through the implementation of SIEM using Wazuh combined with Suricata and integrated with Telegram to detect security incidents on web servers and then analyze these incidents to find out the truth. The method used in this study is an experimental method with stages consisting of needs analysis, system design, implementation, testing, and evaluation. Based on the test results of several web server attack techniques, the system is able to detect security incidents against the web server and then forward the detected security incident logs to the SIEM Wazuh central engine so that it can generate security alerts for analysis from the Wazuh dashboard. The results of the security alert analysis show that the security alerts that appear are true incidents and are not false positives.

Keywords: cybersecurity, SIEM, IDS, wazuh, monitoring.

1. PENDAHULUAN

Kemajuan teknologi informasi dan komunikasi yang sangat pesat saat ini telah membawa perubahan besar di berbagai aspek kehidupan masyarakat. Akses internet yang berkembang cukup luas memberikan kemudahan dalam melakukan komunikasi ke berbagai tujuan dengan jangkauan yang sangat luas (Yuwono, 2022). Hal tersebut mendorong instansi maupun perusahaan untuk memanfaatkan internet agar dapat meningkatkan kinerja dan efektivitas dalam mencapai tujuan organisasi. Salah satu bentuk dari pemanfaatan tersebut berupa penerapan aplikasi web menggunakan web server untuk berbagi informasi, memberikan layanan, dan menyimpan data (Fahrudi & Suartana, 2023). Aplikasi web atau sering disebut *website* merupakan salah satu teknologi yang banyak digunakan untuk keperluan bisnis, pemerintahan, pendidikan, dan sebagainya. *Website* adalah situs di internet yang menawarkan teks, grafik, suara, dan sumber data animasi yang disimpan pada web server dan ditransmisikan melalui *hypertext transfer protocol* (HTTP) (Priyawati et al., 2022).

Dibalik kemudahan dalam penggunaan aplikasi web yang dapat diakses darimana pun, terdapat risiko ancaman keamanan siber yang cukup besar dimana aplikasi web dapat menjadi target serangan oleh penjahat siber yang ingin mencuri informasi sensitif atau mengambil alih sistem (Sudarmilah et al., 2014). Berdasarkan laporan tahunan monitoring keamanan siber BSSN tahun 2021, terdapat 332 aduan siber dengan target sektor pemerintah daerah, pemerintah pusat, ekonomi digital, IIVN, dan lainnya. Terdapat 3 jenis aduan siber tertinggi yang dilaporkan berdasarkan jumlah kasus sesuai pembagian sektor dimana jenis tersebut merupakan kerentanan yang dapat ditemukan pada aplikasi web diantaranya adalah *SQL Injection*, *cross-site scripting*

(XSS), dan *information disclosure* (BSSN, 2022). Selain kerentanan yang terlapor pada BSSN, masih terdapat kerentanan lain yang dapat mengancam keamanan pada aplikasi web seperti *security misconfiguration, sensitive data exposure, cross-site request forgery (CSRF), bad authentication, insecure direct object references* (Guntoro et al., 2020). Semakin banyak organisasi yang mengimplementasikan teknologi berbasis internet, maka ancaman serangan siber juga semakin meningkat karena peningkatan *attack surface*. Oleh karena itu, organisasi perlu mengambil tindakan proaktif maupun reaktif untuk mengurangi ancaman serangan siber guna menghindari kerugian lebih lanjut (Ariffin et al., 2022).

Masalah keamanan tersebut dapat diatasi dengan pendekatan reaktif yaitu penerapan sistem keamanan yang mampu mendeteksi dan menganalisa insiden keamanan pada aplikasi web secara cepat dan efektif. Salah satu solusi yang dapat digunakan adalah dengan mengimplementasikan *Security Information and Event Management (SIEM)*. SIEM adalah sistem yang dapat digunakan untuk mengelola log yang dihasilkan dari berbagai sumber data seperti *endpoint*, perangkat jaringan, maupun *firewall* namun untuk kasus serangan siber umumnya data log didapatkan dari IDS (*Intrusion Detection System*). IDS merupakan perangkat keras/lunak yang digunakan sebagai sensor untuk monitoring dan mendeteksi adanya intrusi pada lalu lintas jaringan lalu menghasilkan *alert* berupa log apabila terdapat paket mencurigakan pada lalu lintas jaringan. Salah satu perangkat lunak IDS yang open source dan merupakan *next-gen* dari perangkat-perangkat IDS yang ada saat ini adalah Suricata (Zain et al., 2023). Log merupakan kumpulan informasi atau data peristiwa seperti *timestamp, source IP address, destination IP address*, dsb yang dapat dimanfaatkan untuk membantu proses investigasi terhadap serangan siber dengan melakukan analisa pada log tersebut (Rokhmah & Utomo, 2020). Dalam hal ini, SIEM dapat menampilkan data log dalam bentuk visualisasi sehingga dapat difungsikan sebagai *live monitoring* lalu lintas jaringan secara *real-time* dan memudahkan dalam melakukan analisa (Muhammad et al., 2023). Salah satu aplikasi SIEM yang bersifat *open source* dan dapat digunakan untuk deteksi serta analisa insiden serangan siber adalah Wazuh. Adapun referensi dari penelitian sebelumnya dalam bidang serupa yang penulis gunakan dalam penelitian ini sebagai bahan acuan penelitian. Berikut merupakan referensi penelitian sebelumnya yang memiliki keterkaitan dengan penelitian ini dapat dilihat pada Tabel 1.

Tabel 1. Referensi penelitian sebelumnya

Nama peneliti	**Judul Penelitian**	**Subyek penelitian**	**Software yang digunakan**	**Hasil Penelitian**
(Khotimah et al., 2022)	Implementasi *Security Information And Event Management* (SIEM) Pada Aplikasi Sms Center Pemerintah Daerah Provinsi Nusa Tenggara Barat	Aplikasi SMS Gateway	Wazuh	Wazuh Agent pada SMS Center dapat menyimpan log yang dihasilkan aplikasi SMS Center dan log dapat dianalisis melalui Wazuh *dashboard.* Sistem SIEM dapat membantu tim dalam mengetahui dan menanggapi serangan yang ada pada aplikasi SMS Center.
(Stephani et al., 2020)	Implementasi dan Analisa Keamanan Jaringan IDS (*Intrusion Detection System*) Menggunakan Suricata Pada Web Server	Web Server	- OPNSense - Suricata - Slowloris - Dirbuster - Skipfish	Implementasi IDS berhasil mendeteksi dan melakukan blok terhadap serangan sesuai dengan *rule* lalu menghasilkan *alert*
(Olalere et al., 2019)	*Performance Analysis of Security Information*	Web Server	- ArcSight - AlienVault - Splunk ES - Loghrythm	Berdasarkan analisis yang dilakukan peneliti, ke-7

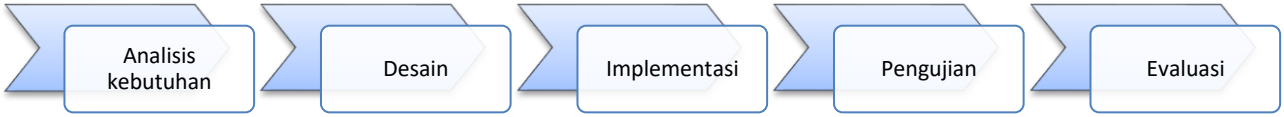
Nama peneliti	Judul Penelitian	Subyek penelitian	Software yang digunakan	Hasil Penelitian
	and Event Management Solutions for Detection of Web-			SIEM berfungsi dengan baik dalam mendeteksi *web attack*. Namun keseluruhan SIEM
	Based Attacks		- McAfee - SolarWinds - QRadar	memiliki kelebihan dan kekurangan masing-masing tergantung kebutuhan yang spesifik.
(Pratama et al., 2022)	Wazuh sebagai *Log Event Management* dan Deteksi Celah Keamanan pada Server dari Serangan Dos	- Ubuntu Server (LAMP) - Windows Server	- Wazuh - Filebeat - ELK Stack - Suricata - Hping3	Suricata berhasil mendeteksi adanya serangan DoS kemudian *alert* diteruskan ke Wazuh agar ditampilkan pada *web interface* Wazuh. Alert dari Wazuh berhasil dikirm ke *administrator* melalui e-mail dan berhasil terintegrasi dengan *platform* Virus Total.

Berdasarkan referensi dari penelitian yang sudah dilakukan sebelumnya, penulis menyusun penelitian ini dengan pendekatan yang berbeda. Adapun perbedaan penelitian ini dengan penelitian yang sebelumnya yaitu pada subyek yang diteliti berfokus terhadap deteksi insiden keamanan pada web server baik dari sisi aplikasi web maupun jaringan sedangkan penelitian sebelumnya berfokus pada sisi *endpoint*, jaringan, dan aplikasi SMS center. Dalam penelitian ini, sistem SIEM menggunakan Wazuh yang diintegrasikan dengan Telegram sedangkan pada penelitian sebelumnya belum terdapat integrasi Telegram pada sistem SIEM. Penggunaan Suricata pada penelitian sebelumnya belum menggunakan SIEM untuk visualisasi log melainkan melalui OPNSense.

Tujuan dari penelitian ini adalah untuk mengimplementasikan SIEM dengan Wazuh untuk deteksi dan analisa insiden keamanan pada web server. Wazuh dikombinasikan dengan Suricata sebagai IDS yang diinstal pada web server untuk mendeteksi adanya intrusi pada lalu lintas jaringan khususnya yang mengarah ke web server. Wazuh akan menghasilkan alert ketika *rules* yang dibuat dipicu oleh kondisi tertentu lalu mengirimkan notifikasi melalui Telegram. Dengan mengimplementasikan sistem SIEM Wazuh yang terintegrasi dengan Telegram, diharapkan dapat membantu *administrator* untuk mengetahui apa yang sedang terjadi pada web server sehingga dapat mendeteksi lebih awal jika terdapat insiden, mempermudah dalam manajemen insiden, dan membantu dalam proses digital forensik.

2. METODE

Perkembangan teknologi informasi yang semakin canggih memberikan banyak manfaat bagi masyarakat. Seiring perkembangan teknologi informasi, berbagai ancaman insiden keamanan siber juga terus berkembang dan terjadi di lingkungan masyarakat seperti adanya kebocoran data, phising, akses illegal, *Distributed Denial of Service* (DDoS), dan sebagainya (Nakao, 2018). Oleh karena itu, diperlukan pendekatan secara reaktif dalam mendeteksi dan mencegah terjadinya insiden keamanan siber, salah satunya adalah dengan penerapan *continual monitoring* pada jaringan maupun *endpoint* (Niu et al., 2022). Metode yang digunakan dalam penyusunan penelitian ini adalah metode eksperimental. Metode eksperimental merupakan metode penelitian yang digunakan untuk mengetahui pengaruh perlakuan tertentu terhadap subyek yang sedang diteliti melalui pengujian dengan kondisi yang berbeda – beda dan terkendalikan (Jaedun, 2011). Adapun diagram alir penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Diagram Alir Penelitian

2.1 Analisis Kebutuhan

Analisis kebutuhan berguna untuk menentukan kebutuhan apa yang diperlukan dalam penelitian. Berdasarkan tujuan penelitian yaitu mengimplementasikan SIEM dengan Wazuh untuk deteksi dan analisa insiden keamanan pada web server, maka dibutuhkan

dua buah server yang masing-masing digunakan sebagai server untuk SIEM Wazuh dan web server yang terpasang IDS untuk mendeteksi adanya intrusi pada web server. Menurut dokumentasi resmi dari Wazuh, rekomendasi spesifikasi hardware untuk 1-25 *agent* adalah 4 vCPU, 8 GB RAM, dan 50 GB *storage* dengan sistem operasi Amazon Linux 2; CentOS 7, 8; Red Hat Enterprise Linux 7, 8, 9; Ubuntu 16.04, 18.04, 20.04, 22.04 (*Quickstart · Wazuh Documentation*, n.d.). Dalam hal ini, penulis menggunakan *Virtual Private Server* (VPS) dari salah satu *cloud provider* regional dengan spesifikasi server masing-masing dapat dilihat pada Tabel 2.

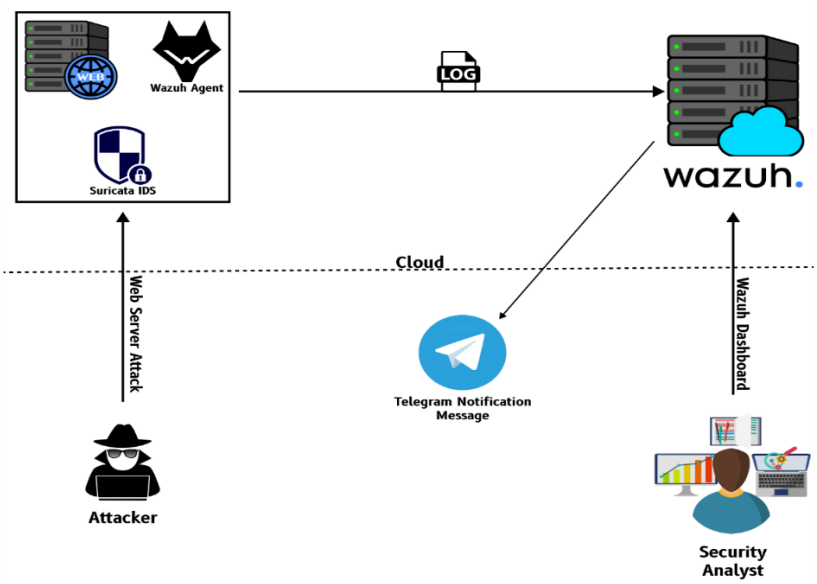
Tabel 2. Spesifikasi server

	SIEM Server (Wazuh)	**Web Server**
Perangkat keras	-4 *Core* CPU -12 GB RAM -60 GB *Storage*	-4 *Core* CPU -8 GB RAM -40 GB *Storage*
Perangkat lunak	-Ubuntu 22.04 LTS -Wazuh *Component* (*Indexer*, *Manager*, dan *Dashboard*)	-Ubuntu 22.04 LTS -Docker -Wazuh Agent -Suricata IDS

Selain server, dibutuhkan perangkat yang dapat melakukan *remote Secure Shell* (SSH) ke VPS untuk keperluan konfigurasi, dapat mengakses *dashboard* Wazuh untuk keperluan analisa, dan dapat melakukan penyerangan ke Web Server untuk keperluan pengujian. Dalam hal ini, penulis menggunakan perangkat laptop HP Notebook - 14-cm0xxx. Penulis menggunakan Virtual Box untuk menjalankan Kali Linux yang digunakan untuk keperluan pengujian serangan ke web server karena sudah tersedia tools yang dapat membantu dalam melakukan pengujian serangan.

2.2 Desain

Desain merupakan tahapan kedua setelah analisis kebutuhan yang berguna untuk mengetahui gambaran dari alur sistem agar dapat memperjelas bagaimana cara kerja dari sistem yang diimplementasikan. Adapun skema sistem dari penelitian dapat dilihat pada Gambar 2.



Gambar 2. Skema sistem SIEM

Berdasarkan skema pada Gambar 2, penulis menerapkan IDS Suricata pada web server guna mendeteksi adanya intrusi yang dilancarkan oleh *attacker*. Suricata mendeteksi intrusi berdasarkan berkas *rules* yang terdapat dalam direktori "*/etc/suricata/rules"* dengan ekstensi *.rules*. Apabila terdapat paket yang mengarah ke web server terdeteksi oleh *rules* Suricata, maka *alert* akan terpicu kemudian Suricata akan menghasilkan log berdasarkan *alert* dan secara *default* disimpan dalam dua bentuk berkas yaitu *fast.log* dan *eve.json*.

Log *alert* dari Suricata selanjutnya dikirim ke Wazuh server bersama dengan log lain dari web server seperti log aplikasi web atau *syslog* melalui Wazuh agent yang dipasang pada web server. Log Suricata yang dikirim adalah berkas *eve.json* dengan format *Javascript Object Notation* (JSON) agar dapat diproses oleh Wazuh server. Wazuh server bertindak sebagai manajemen agen dan *dashboard* sistem *monitoring* baik *file integrity*, *intrusion*, maupun log. Sedangkan Wazuh agent merupakan perangkat lunak yang dipasang pada *endpoint* untuk melakukan pembacaan sistem, pengumpulan log serta mengirimkan log ke Wazuh server (Pratama et al., 2022). Data log yang telah dikirim ke Wazuh server akan di-*decode* oleh *decoder* untuk mengekstrak setiap informasi yang terdapat pada log seperti *timestamp, hostname, source ip,* dsb. Informasi tersebut kemudian divisualisasikan melalui Wazuh *dashboard* yang dapat diakses oleh *administrator* atau tim *IT security* sehingga dapat dilakukan analisa terhadap data informasi tersebut untuk menemukan korelasi antar data, membantu proses digital forensik, maupun menentukan kebenaran terjadinya

insiden (*true positive*).

Wazuh memiliki fitur *alert* untuk mendeteksi mendeteksi serangan, intrusi, penyalahgunaan perangkat lunak, masalah konfigurasi, kesalahan aplikasi, malware, rootkit, anomali sistem, atau pelanggaran kebijakan keamanan berdasarkan *ruleset* yang dibuat. Dalam hal ini, penulis membuat *ruleset* untuk mendeteksi serangan web server berdasarkan log yang dihasilkan Suricata sehingga serangan yang terdeteksi akan muncul dalam visualisasi Wazuh *dashboard* agar selanjutnya dapat dilakukan analisa lebih lanjut.

Dalam upaya mendeteksi serangan terhadap web server, seorang *administrator* memiliki tanggung jawab untuk memantau, menganalisis, dan melaporkan apabila terdapat insiden keamanan yang terjadi sehingga *administrator* harus siap 24/7 dalam melakukan monitoring SIEM karena insiden keamanan dapat terjadi kapan saja (Agyepong et al., 2023). Penulis memanfaatkan fitur dari Wazuh yaitu *Integrator* untuk menghubungkan Wazuh dengan Telegram agar ketika *alert* pada Wazuh terpicu maka Wazuh akan mengirimkan notifikasi melalui Telegram Bot secara *realtime*.

2.3 Implementasi

Tahap ketiga merupakan implementasi dari desain skema yang sudah dibuat sebelumnya. Dalam tahap ini penulis melakukan perancangan sistem meliputi instalasi dan konfigurasi terhadap perangkat lunak pada masing-masing VPS sesuai dengan skema pada Gambar 2.

2.3.1 Suricata

Suricata diinstal pada web server dengan mengikuti dokumentasi yang terdapat pada *website* resmi Suricata. Konfigurasi dasar yang diterapkan pada Suricata meliputi penentuan alamat IP dari jaringan yang terhubung dengan web server, penentuan *network interface* yang akan digunakan untuk memantau lalu lintas data, dan penentuan *ruleset* yang akan digunakan untuk mendeteksi adanya intrusi yang mengarah ke web server.

2.3.2 Wazuh

Dalam proses instalasi Wazuh terdapat 3 komponen yang perlu diinstall agar Wazuh dapat digunakan. Sebelum penginstalan komponen Wazuh, diperlukan penambahan repositori Wazuh ke daftar repositori pada server agar komponen dapat diinstal melalui baris perintah di terminal. Konfigurasi yang diterapkan pada Wazuh meliputi

konfigurasi *ruleset* untuk membuat *alert* sesuai dengan kondisi yang diperlukan dan integrasi Wazuh ke Telegram. Berikut merupakan komponen Wazuh yang perlu diinstal dapat dilihat pada Tabel 3.

Tabel 3. Komponen Wazuh

Komponen	**Deskripsi**	**Baris perintah instalasi**
Wazuh Indexer	Mesin analitik dan pencarian teks lengkap yang sangat skalabel.	*apt-get -y install wazuh-indexer*
Wazuh Server	Komponen utama yang menyertakan Wazuh manager dan Filebeat. Wazuh manager mengumpulkan dan analisis data dari Wazuh agent, memicu alert saat terdeteksi ancaman/anomali. Filebeat mengirim alert dan event ke Wazuh indexer dengan aman.	*apt-get -y install wazuh-manager* *apt-get -y install filebeat*
Wazuh Dashboard	Sebagai antarmuka pengguna web untuk visualisasi dan analisis data.	*apt-get -y install wazuh-dashboard*

2.3.3 Wazuh Agent

Wazuh agent diinstal pada web server sebagai pengumpul log dan pengirim log menuju Wazuh server. Penulis menggunakan opsi instal dan registrasi *agent* secara otomatis melalui Wazuh *dashboard*. Konfigurasi yang diterapkan pada Wazuh agent adalah penambahan *script* untuk membaca log dari Suricata.

2.4 Pengujian

Tahap keempat merupakan tahap pengujian terhadap sistem yang telah dirancang untuk mendeteksi dan memberikan notifikasi ketika terdapat insiden keamanan pada web server. Dalam penelitian ini, pengujian dilakukan dengan beberapa jenis serangan yang sering ditujukan terhadap web server.

Website yang digunakan untuk pengujian merupakan *broken web application* atau web yang sengaja dibuat untuk keperluan pengujian serangan web. *Website* tersebut memiliki berbagai jenis kerentanan yang dapat dieksploitasi dan dapat disesuaikan tingkat keamanan dari website tersebut mulai dari *low, medium, dan high*. Dalam tahap pengujian ini, mencakup pembahasan mengenai hasil pengujian serangan terhadap ketiga tingkatan keamanan web tersebut. Berikut merupakan jenis-jenis serangan yang dilakukan penulis untuk skenario pengujian sistem:

2.4.1 SQL Injection

SQL Injection merupakan jenis serangan yang ditujukan ke web server dengan

melakukan injeksi *SQL Query* berbahaya sehingga dapat memanipulasi logika dalam *syntax* SQL yang akan dieksekusi oleh aplikasi web. Serangan *SQL Injection* memungkinkan penyerang untuk mencuri maupun mengubah data yang tersimpan di dalam database sehingga dapat menimbulkan kerugian terhadap organisasi maupun pengguna aplikasi web (Anugrah et al., 2022). *SQL Injection* termasuk ke dalam kategori *Injection* yang menempati posisi ke-3 pada OWASP Top 10.

2.4.2 Cross-site Scripting

Cross-site Scripting atau disingkat XSS merupakan jenis serangan dimana penyerang menyisipkan kode berbahaya pada halaman web dari sisi klien yang biasanya berupa kode Javascript. Kode berbahaya yang disisipkan memungkinkan penyerang untuk mendapatkan akses yang lebih besar dari halaman yang diakses hingga dapat melakukan *account takeover* apabila kode tersebut dieksekusi oleh aplikasi web (Suroto & Asman, 2021). *Cross-site Scripting* juga termasuk ke dalam kategori *Injection* yang menempati posisi ke-3 pada OWASP Top 10.

2.4.3 Directory Traversal

Directory Traversal merupakan jenis serangan yang memungkinkan penyerang untuk mengakses direktori maupun berkas yang seharusnya tidak boleh untuk diakses. Serangan ini disebabkan oleh program yang menggunakan *directory path string* untuk mengambil atau mengunduh file tanpa memeriksa bahwa *path string* merujuk ke *file* yang benar sehingga rentan dimanipulasi oleh penyerang untuk merujuk ke *path* yang lain (Flanders, 2019).

2.4.4 Denial of Service (DoS)

Denial of Service atau disingkat DoS merupakan jenis serangan dimana penyrang membanjiri server dengan permintaan yang sangat banyak sehingga menghabiskan *resource* dari server tersebut. Akibatnya server tidak dapat menjalankan tugasnya dengan baik dan tidak bisa menangani permintaan yang masuk sehingga mengakibatkan penolakan layanan dari server (*denial of service*) (Hijriyannto & Ulum, 2021).

2.4.5 Port Scanning

Port Scanning adalah jenis serangan yang digunakan penyerang untuk mendeteksi *port-port* yang terbuka pada suatu server (Amarudin, 2018). *Port Scanning* merupakan salah satu teknik *reconnaissance* yang memungkinkan penyerang untuk mendapatkan

semua informasi tentang target termasuk sistem informasi, layanan, dan kerentanan yang mungkin ada di jaringan.

2.5 Evaluasi

Tahap evaluasi merupakan tahap pembahasan mengenai performa dari sistem SIEM yang telah dirancang dan diuji dengan beberapa skenario. Dalam tahap evaluasi terdapat beberapa poin yang akan dibahas terkait hasil dari pengujian sistem pada masing-masing tingkat keamanan dari website mulai dari *low*, *medium*, dan *high*. Serangan DoS dan Port Scanning dalam penelitian ini hanya dibahas dalam sekali serangan. Hal ini disebabkan karena DoS dan Port Scanning merupakan jenis serangan yang berada pada tingkat jaringan atau *network-level attack* sehingga fitur tingkat keamanan web tidak akan berpengaruh.

Poin yang dibahas dalam tahap ini adalah kemampuan sistem dalam mendeteksi insiden dan memunculkan *security alert* pada SIEM, kemampuan sistem dalam mengirim notifikasi melalui Telegram ketika *security alert* pada SIEM terpicu, dan analisa *security alert* yang dihasilkan oleh sistem untuk mengetahui apakah *alert* yang dihasilkan benar-benar sebuah insiden atau bukan.

3. HASIL DAN PEMBAHASAN

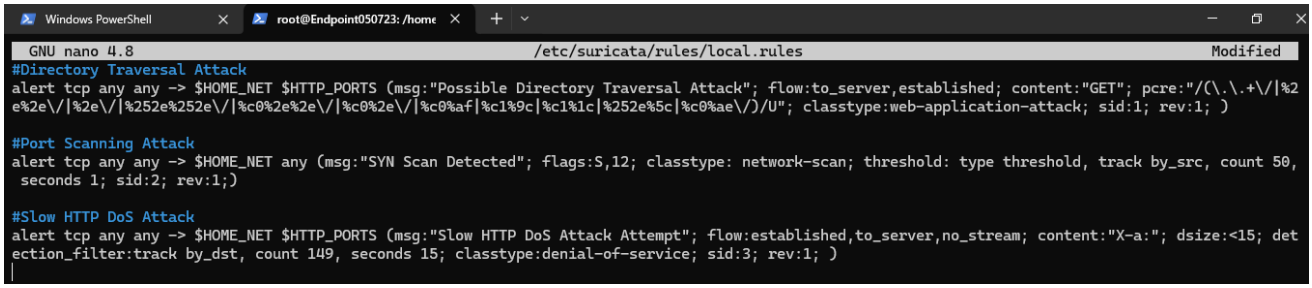
3.1 Hasil Implementasi

Berdasarkan alur penelitian, implementasi merupakan proses perancangan sistem yang terdiri dari instalasi dan konfigurasi terhadap perangkat lunak meliputi Suricata, Wazuh SIEM, dan Wazuh Agent pada masing-masing VPS sesuai dengan skema pada Gambar 2.

3.1.1 Suricata

Konfigurasi Suricata terdiri dari penambahan *ip HOME_NET* yang merupakan *network address* dari web server yaitu [10.116.132.0/24], penentuan *network interface* dari web server yaitu *ens3*, serta penerapan *ruleset* yang telah disediakan secara *default* oleh Suricata yaitu Emerging Threats Open *ruleset*. Konfigurasi diterapkan dengan memodifikasi *file* konfigurasi Suricata yang terletak pada direktori "*/etc/suricata/suricata.yml*". Penulis menyadari bahwa Emerging Threats Open *ruleset* belum maksimal dalam mendeteksi serangan web sehingga penulis menambahkan *custom rule* pada penelitian ini untuk menambah akurasi dari pendeteksian. *Custom rule*

dapat dilihat pada Gambar 3.



Gambar 3. Custom rule Suricata

3.1.2 Wazuh SIEM

Konfigurasi Wazuh terdiri dari konfigurasi *ruleset* untuk memicu *alert* dan integrasi dengan Telegram. Konfigurasi *ruleset* menggunakan bawaan *default* dari Wazuh yang dimodifikasi sehingga *security alert* yang muncul pada Wazuh dashboard lebih informatif dan memudahkan analisa.

Integrasi Wazuh dengan Telegram memanfaatkan *tools integrator* yang tersedia pada Wazuh. *Integrator* memungkinkan pengguna untuk mengintegrasikan sistem *alert* dengan API (*Application Programming Interface*) dari perangkat lunak eksternal melalui sebuah *script*. Konfigurasi *intergator* telegram terdiri dari penambahan komponen blok integrasi pada *file* konfigurasi Wazuh manager yang terletak pada direktori "*/var/ossec/etc/ossec.conf*" dan penambahan *script custom-telegram.py* pada direktori "*/var/ossec/integrations*".

3.1.3 Wazuh Agent

Konfigurasi yang diterapkan pada Wazuh agent merupakan penambahan blok *script* pada *file* konfigurasi Wazuh agent yang terletak pada direktori "*/var/ossec/etc/ossec.conf*". Penambahan blok *script* berfungsi agar Wazuh agent dapat membaca *file* log dari Suricata yaitu *eve.json* lalu mengirimkannya ke *central engine* atau server Wazuh untuk diproses lebih lanjut.

3.2 Hasil Pengujian

3.2.1 Pengujian SQL Injection

Pengujian SQL Injection dilakukan menggunakan *tools* sqlmap dengan perintah *sqlmap -u http://ip_address/sqli_1.php?title= --cookie="cookie;security_level=0" --dump*. Parameter (*-u*) menunjukan URL dari target beserta parameter *query* yang rentan, parameter (*--cookie*) menunjukan HTTP *cookie header value* dari sesi login serta level *security* web saat ini, dan parameter (*--dump*) menginstruksikan sqlmap untuk

melakukan ekstraksi data dari *database* target. Hasil dari pengujian SQL Injection pada kerentanan tingkat *low* berhasil mendapatkan *database user* beserta *password*-nya, sedangkan serangan pada kerentanan tingkat *medium* dan *high* gagal salah satu alasannya karena terdapat proteksi berupa WAF (*Web Application Firewall*).

3.2.2 Pengujian XSS

Pengujian XSS dilakukan dengan menyisipkan *payload* berupa kode Javascript pada *form input* yang terdapat pada aplikasi web. Dalam hal ini, penulis membuat *payload* XSS sederhana dengan kode sebagai berikut: *<script>alert(document.cookie)</script>*. Ketika *payload* tersebut di-*submit* pada *form input* kemudian dieksekusi oleh aplikasi web maka akan muncul *alert* Javascript pada web browser berisikan cookie browser yang menandakan serangan tersebut berhasil. Hasil pengujian XSS pada kerentanan tingkat *low* dan *medium* berhasil memunculkan *alert* berisikan *cookie browser*, sedangkan serangan pada kerentanan tingkat *high payload* tidak tereksekusi salah satu alasannya karena *payload* tersaring oleh proteksi WAF.

3.2.3 Pengujian Directory Traversal

Pengujian Directory Traversal dilakukan dengan memanipulasi *path* atau jalur menuju *file* yang diakses oleh sistem. Dalam hal ini, penulis memanipulasi parameter "*page"* pada url *http://ip_address/directory_traversal_1.php?page=messages.txt* dengan *payload* sebagai berikut: *../../../../etc/passwd*. Hasil pengujian Directory Traversal pada kerentanan tingkat *low* berhasil mengakses *file* passwd linux dan menampilkannya pada browser, sedangkan pada kerentanan *medium* dan *high* gagal mengakses *file* passwd karena terdeteksi kemudian diblok oleh WAF.

3.2.4 Pengujian Denial of Service (DoS)

Penulis menggunakan serangan DoS tipe Slow HTTP dimana penyerang mengirimkan permintaan HTTP yang sah ke web server dalam jumlah yang besar untuk mengganggu koneksi HTTP. Pengujian serangan DoS dilakukan menggunakan *tools* Slowloris yang dijalankan melalui terminal Kali linux dengan perintah "*slowloris -p 80 ip_address"*.

3.2.5 Pengujian Port Scanning

Pengujian *port scanning* dilakukan menggunakan *tools* Nmap yang dijalankan melalui terminal Kali linux dengan perintah sebagai berikut: *nmap -sS ip_address*. Dalam perintah tersebut penulis menggunakan parameter (*-sS*) yang artinya teknik port scanning SYN Scan.

3.3 Hasil Evaluasi Sistem

Hasil evaluasi didapatkan setelah sistem diuji dengan berbagai skenario serangan web untuk mengetahui performa dari sistem SIEM. Hasil evaluasi sistem dapat dilihat pada Tabel 4.

Tabel 4. Hasil evaluasi sistem SIEM

No	**Uji Serangan**	**Deteksi SIEM**			**Notifikasi Telegram**
		Low	***Medium***	***High***	
1	SQL Injection	Terdeteksi	Terdeteksi	Terdeteksi	Berhasil
2	XSS	Terdeteksi	Terdeteksi	Terdeteksi	Berhasil
3	Directory Traversal	Terdeteksi	Terdeteksi	Terdeteksi	Berhasil
4	Denial of Service	Terdeteksi			Berhasil
5	Port Scanning	Terdeteksi			Berhasil

Tabel 5 menunjukkan bahwa sistem SIEM mampu mendeteksi adanya insiden serangan pada web server dari sisi *application level* berupa serangan SQL Injection, XSS, dan Directory Traversal maupun dari sisi *network level* berupa DoS dan Port Scanning kemudian mengirim notifikasi ke Telegram dengan baik.

3.4 Hasil Analisa Alert

Analisa *alert* yang dihasilkan oleh SIEM berguna untuk mengetahui kebenaran dari insiden keamanan yang terdeteksi apakah merupakan *false positive* atau bukan. *False Positive* merupakan situasi dimana terdapat *alert* yang muncul pada SIEM namun sebenarnya tidak terjadi insiden yang sesuai dengan *alert* tersebut. Dalam penelitian ini, penulis melakukan analisa terhadap *security alert* yang dihasilkan oleh SIEM dengan beberapa data log yang terkandung pada *security alert* tersebut.

3.4.1 Alert SQL Injection

Serangan SQL injection memicu 2 *security alert* yang berbeda namun masih mengindikasikan adanya percobaan SQL injection. Berdasarkan analisa, penulis

menemukan data *http user agent* pada log dengan *value* sqlmap beserta versinya sehingga hal tersebut menandakan bahwa insiden tersebut benar terjadi dan bukan *false positive* karena berasal dari *tools* sqlmap.

3.4.2 Alert XSS

Serangan XSS memicu sebuah *security alert* yang mengindikasikan adanya percobaan XSS menggunakan *tag script* yang terdeteksi pada URI (*Uniform Resource Identifier*) dari *http request*. Penulis menganalisa data *http url* yang terdapat pada log dan menemukan adanya *payload* XSS menggunakan *tag script* pada parameter "*firstname"* sehingga bisa disimpulkan bahwa insiden tersebut benar terjadi dan bukan *false positive*.

3.4.3 Alert Directory Traversal

Serangan Directory Traversal memicu sebuah *security alert* yang mengindikasikan adanya percobaan Directory Traversal. Penulis menganalisa data *http url* yang terdapat pada log dan menemukan adanya *payload* Directory Traversal yang mencoba mengakses *file* passwd linux sehingga bisa disimpulkan bahwa insiden tersebut benar terjadi dan bukan *false positive*.

3.4.4 Alert DoS

Serangan DoS memicu sebuah *security alert* yang mengindikasikan adanya serangan DoS bertipe *slow http*. *Rules* deteksi slow http DoS pada Suricata diatur untuk terpicu ketika terdapat *http request* yang mengandung payload dengan ukuran kurang dari 15 byte dan mengandung string "X-a:" sebanyak 149 kali dalam rentang waktu 15 detik. Dalam percobaan DoS pada penelitian ini terdapat beberapa *security alert slow http* DoS yang muncul dalam waktu yang sama sehingga dapat ditarik kesimpulan bahwa memang benar terjadi percobaan *slow http DoS attack*.

3.4.5 Alert Port Scanning

Serangan port scanning memicu sebuah *security alert* yang mengindikasikan adanya percobaan *SYN scan*. *Rules* deteksi SYN scan pada Suricata diatur untuk terpicu ketika terdapat 50 paket SYN yang masuk dalam satu detik lalu memunculkan *security alert* pada SIEM. Dalam percobaan *SYN scan* pada penelitian ini terdapat beberapa *security alert SYN scan* yang muncul dalam waktu yang sama sehingga dapat ditarik kesimpulan bahwa memang benar terjadi percobaan port scanning dengan teknik SYN scan.

4. PENUTUP

4.1 Kesimpulan

Berdasarkan hasil evaluasi dan analisa yang telah dilakukan penulis, dapat ditarik kesimpulan bahwa penerapan sistem SIEM Wazuh yang dikombinasikan dengan Suricata mampu mendeteksi intrusi pada web server secara real-time dan membantu dalam menganalisis adanya insiden keamanan berdasarkan log sehingga respon terhadap insiden siber dapat menjadi lebih cepat dan efisien. Dalam penelitian ini, terdapat beberapa uji serangan yang berhasil dideteksi berkat penggunaan sistem SIEM. Hal ini menunjukkan bahwa SIEM cukup efektif dalam menjaga keamanan web server dari ancaman siber. Selain itu, sistem SIEM berhasil diintegrasikan dengan aplikasi Telegram dan berjalan dengan baik sehingga mampu memberikan notifikasi kejadian insiden keamanan langsung melalui pesan Telegram.

4.2 Saran

Penulis memberikan saran terhadap penelitian berikutnya yang serupa untuk mengembangkan sistem agar mampu digunakan untuk *Threat Intelligence* melalui integrasi dengan beberapa *Threat Intelligence feeds* seperti VirusTotal, Open Threat Exchange (OTX), Shodan, ThreatConnect, dan yang lainnya. Dengan demikian, sistem mampu mengidentifikasi ancaman insiden keamanan lebih awal sehingga tim *IT security* dapat menyusun rencana respons keamanan yang lebih baik seperti isolasi *host* yang terinfeksi atau memblokir alamat IP berbahaya.

DAFTAR PUSTAKA

Agyepong, E., Cherdantseva, Y., Reinecke, P., & Burnap, P. (2023). A systematic method for measuring the performance of a cyber security operations centre analyst. *Computers & Security*, *124*, 102959. https://doi.org/10.1016/J.COSE.2022.102959

Amarudin. (2018). Analisis Dan Implementasi Keamanan Jaringan Pada Mikrotik Router Menggunakan Metode Port Knocking. *Seminar Nasional Sains Dan Teknologi 2018*, 1–7.

Anugrah, F. T., Ikhwan, S., Gusti, J., Studi, P., Telekomunikasi, T., Elektro, D., Teknologi, I., & Purwokerto, T. (2022). Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection. *Techné : Jurnal Ilmiah Elektroteknika*, *21*(2), 199–210. https://doi.org/10.31358/TECHNE.V21I2.320

Ariffin, M. A. M., Darus, M. Y., Haron, H., Kurniawan, A., Muliono, Y., & Pardomuan, C. R. (2022). Deployment of Honeypot and SIEM Tools for Cyber Security

Education Model in UITM. *International Journal of Emerging Technologies in Learning*, *17*(20), 149–172. https://doi.org/10.3991/ijet.v17i20.32901

BSSN. (2022). *Laporan Tahunan Monitoring Keamanan Siber 2021*. https://cloud.bssn.go.id/s/Lyw8E4LxwNiJoNw

Fahrudi, M. A., & Suartana, I. M. (2023). 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 (JINACS)*, *04*, 275–282. https://ejournal.unesa.ac.id/index.php/jinacs/article/view/50723

Flanders, M. (2019). A Simple and Intuitive Algorithm for Preventing Directory Traversal Attacks. *ArXiv Preprint ArXiv:1908.04502*. https://arxiv.org/abs/1908.04502v1

Guntoro, G., Costaner, L., & Musfawati, M. (2020). Analisis Keamanan Web Server Open Journal System (OJS) menggunakan Metode ISSAF dan OWASP (Studi Kasus OJS Universitas Lancang Kuning). *JIPI (Jurnal Ilmiah Penelitian Dan Pembelajaran Informatika)*, *5*(1), 45–55. https://doi.org/10.29100/JIPI.V5I1.1565.G707

Hijriyannto, B., & Ulum, F. (2021). Perbandingan Penerapan Metode Pengamanan Web Server menggunakan Mod Evasive dan Ddos Deflate terhadap Serangan Slow Post. *Journal of Engineering, Computer Science and Information Technology (JECSIT)*, *1*(2), 88–92. https://doi.org/10.33365/JECSIT.V1I1.11

Jaedun, A. (2011). *Metodologi Penelitian Eksperimen*. 0–12. http://staff.uny.ac.id/sites/default/files/pengabdian/drs-amat-jaedun-mpd/metode-penelitian-eksperimen.pdf

Khotimah, H., Bimantoro, F., Silas Kabanga, R., Bagus, I., & Widiartha, K. (2022). *Implementasi Security Information and Event Management (SIEM) pada Aplikasi SMS Center Pemerintah Daerah Provinsi Nusa Tenggara Barat*. *3*(2), 213–219. http://begawe.unram.ac.id/index.php/JBTI/

Muhammad, A. R., Sukarno, P., & Wardana, A. A. (2023). Integrated Security Information and Event Management (SIEM) with Intrusion Detection System (IDS) for Live Analysis based on Machine Learning. *Procedia Computer Science*, *217*, 1406–1415. https://doi.org/10.1016/J.PROCS.2022.12.339

Nakao, K. (2018). Proactive cyber security response by utilizing passive monitoring technologies. *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 1–1. https://doi.org/10.1109/icce.2018.8326061

Niu, J., Alroobaea, R., Baqasah, A. M., & Kansal, L. (2022). Implementation of network information security monitoring system based on adaptive deep detection. *Journal of Intelligent Systems*, *31*(1), 454–465. https://doi.org/10.1515/JISYS-2022-0032/ASSET/GRAPHIC/J_JISYS-2022-0032_FIG_007.JPG

Olalere, M., Ndunagu, J., Abdulhamid, S. M., & Odey, P. (2019). Performance Analysis

of Security Information and Event Management Solutions for Detection of Web-Based Attacks. *Proceedings of the Cyber Secure Nigeria 2019 Conference*, 39–47.

Pratama, M. D., Nova, F., & Prayama, D. (2022). Wazuh sebagai Log Event Management dan Deteksi Celah Keamanan pada Server dari Serangan Dos. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, *3*(1), 1–7. https://doi.org/10.30630/jitsi.3.1.59

Priyawati, D., Rokhmah, S., & Utomo, I. C. (2022). Website Vulnerability Testing and Analysis of Website Application Using OWASP. *International Journal of Computer and Information System (IJCIS)*, *3*(3), 142–147. https://doi.org/10.29040/IJCIS.V3I3.90

Quickstart · Wazuh documentation. (n.d.). Retrieved March 23, 2023, from https://documentation.wazuh.com/current/quickstart.html

Rokhmah, S., & Utomo, I. C. (2020). Binary Log Analysis on MySQL to Help Investigation Process Against Database Privillege Attacks. *International Journal of Computer and Information System (IJCIS)*, *1*(1), 11–15. https://doi.org/10.29040/IJCIS.V1I1.7

Stephani, E., Nova, F., & Asri, E. (2020). Implementasi dan Analisa Keamanan Jaringan IDS (Intrusion Detection System) Menggunakan Suricata Pada Web Server. *JITSI : Jurnal Ilmiah Teknologi Sistem Informasi*, *1*(2), 67–74. https://doi.org/10.30630/JITSI.1.2.10

Sudarmilah, E., Pratisti, W. D., Fadlilah, U., & Giwangkoro, G. G. (2014). Website Vulnerability Scan for Information System of Toddler's Growth and Development. *The Asian Conference on Society, Education & Technology 2014*.

Suroto, S., & Asman, A. (2021). Ancaman terhadap Keamanan Informasi oleh Serangan Cross-Site Scripting (XSS) dan Metode Pencegahannya. *Zona Komputer: Program Studi Sistem Informasi Universitas Batam*, *11*(1), 11–19. https://doi.org/10.37776/ZK.V11I1.658

Yuwono, D. T. (2022). Analysis Performance Intrusion Detection System in Detecting Cyber-Attack on Apache Web Server. *IT Journal Research and Development*, *6*(2), 169–178. https://doi.org/10.25299/itjrd.2022.7853