Digital Transformation Technology (Digitech) | e-ISSN: 2807-9000
Volume 6, Nomor 1, Maret 2026
https://doi.org/10.47709/digitech.v6i1.8802

Pengujian Sistem Pemantauan Keamanan Jaringan Berbasis Suricata dan Elastic Stack

Albert Sembiring[1*], Alfredo Juliansa Sitohang[2]
[1,2]Universitas Bakrie, Indonesia
[1]albert.sembiring@bakrie.ac.id, [2]alfredo.sihotang123@gmail.com

Riwayat Artikel:
Diajukan: 22 Juni 2026
Disetujui: 28 Juli 2026
Dipublikasikan: 28 Juli 2026

Kata Kunci:
Suricata; Elastic Stack; Sistem Deteksi Intrusi; SQL Injection; Cross-Site Scripting

Abstrak

Instansi ABC membentuk tim Security Operation Center (SOC) untuk menyelenggarakan pengamanan jaringan, yang salah satu tugasnya adalah memantau lalu lintas jaringan. Tim SOC mengalami kesulitan dalam mendeteksi dan menganalisis anomali lalu lintas sehingga membutuhkan sistem pemantauan keamanan jaringan yang dapat mendeteksi anomali lalu lintas yang berpotensi menjadi serangan siber. Anomali lalu lintas dan serangan siber dapat dideteksi menggunakan Intrusion Detection System (IDS) dengan metode deteksi berbasis signature. Penelitian ini menguji sistem pemantauan keamanan jaringan yang terdiri atas IDS Suricata dan Security Information and Event Management (SIEM), yang mencakup Elasticsearch, Filebeat, dan Kibana dari Elastic Stack. Hasil penelitian menunjukkan bahwa sistem pemantauan keamanan jaringan berbasis Suricata dan Elastic Stack dapat mendeteksi serangan siber berupa SQL injection dan Cross-Site Scripting (XSS), serta menghasilkan visualisasi dalam bentuk Dashboard dan Discover sehingga memudahkan tim SOC dalam mendeteksi dan menganalisis anomali lalu lintas serta serangan siber. Penelitian ini juga menunjukkan bahwa penerapan aturan bawaan Suricata dengan penambahan aturan kustom lebih efektif dalam mendeteksi berbagai serangan SQL injection dan XSS dibandingkan dengan hanya menerapkan aturan bawaan Suricata. Kontribusi penelitian ini adalah memberikan bukti empiris bahwa integrasi Suricata dan Elastic Stack dengan kombinasi aturan bawaan dan aturan kustom mampu meningkatkan efektivitas deteksi serangan SQL injection dan XSS sekaligus mendukung proses pemantauan serta analisis keamanan jaringan oleh tim SOC.

PENDAHULUAN

Pemanfaatan jaringan komputer dan aplikasi web dalam mendukung proses bisnis menyebabkan volume lalu lintas, jumlah perangkat, dan aktivitas pengguna yang harus diawasi semakin meningkat. Kondisi tersebut menuntut organisasi untuk memiliki kemampuan pemantauan keamanan yang dapat memberikan visibilitas terhadap aktivitas mencurigakan sejak dini. Instansi ABC telah membentuk tim Security Operation Center (SOC), yang salah satu tugasnya adalah memantau anomali lalu lintas jaringan. Namun, tim SOC masih mengalami kesulitan dalam mendeteksi dan menganalisis anomali lalu lintas yang berpotensi menjadi serangan siber. Tanpa dukungan sistem pemantauan yang terintegrasi, proses identifikasi insiden dapat berlangsung lebih lambat dan menyulitkan tim dalam menentukan tindak lanjut yang tepat (Al Hilmi & Khujaemah, 2022).

Ancaman terhadap aplikasi web menjadi perhatian penting karena serangan dapat memengaruhi kerahasiaan, integritas, dan ketersediaan layanan teknologi informasi. Berdasarkan tren aduan siber tahun 2021 yang diterima oleh Direktorat Operasi Keamanan Siber BSSN, terdapat tiga jenis serangan pada sektor pemerintahan pusat, yaitu SQL injection sebanyak 36 aduan, Cross-Site Scripting (XSS) sebanyak 15 aduan, dan Information Disclosure sebanyak 7 aduan. SQL injection memanfaatkan masukan pada aplikasi untuk memanipulasi perintah basis data, sedangkan XSS menyisipkan skrip berbahaya ke dalam halaman web yang kemudian dijalankan pada sisi pengguna. Serangan-serangan tersebut dapat merusak layanan teknologi informasi dan sistem jaringan Instansi ABC sehingga diperlukan mekanisme deteksi yang mampu mengenali pola serangan pada lalu lintas jaringan (Badan Siber dan Sandi Negara, 2022; Anugrah et al., 2022; Sijabat & Evo, 2023).

Secara konseptual, Intrusion Detection System (IDS) digunakan untuk memantau aktivitas sistem atau jaringan, mencocokkan lalu lintas dengan karakteristik serangan, serta menghasilkan peringatan ketika ditemukan aktivitas yang mencurigakan. Pada metode deteksi berbasis signature, efektivitas deteksi sangat dipengaruhi oleh ketersediaan dan ketepatan aturan yang digunakan. Suricata merupakan perangkat lunak IDS/IPS sumber terbuka yang dapat menerapkan aturan bawaan maupun aturan kustom untuk mendeteksi pola serangan. Selanjutnya, Security Information and Event Management (SIEM) berfungsi mengumpulkan, menyimpan, menganalisis, dan memvisualisasikan log keamanan secara terpusat. Elastic Stack mendukung fungsi tersebut melalui Elasticsearch sebagai mesin penyimpanan dan pencarian data, Filebeat sebagai pengirim log, serta Kibana sebagai sarana analisis dan visualisasi (Díaz-Verdejo et al., 2022; Sijabat & Evo, 2023; Zain et al., 2023).

Beberapa penelitian terdahulu telah membahas penerapan IDS dan Suricata dalam pemantauan keamanan jaringan. Al Hilmi dan Khujaemah (2022) mengembangkan sistem IDS sumber terbuka yang ringan untuk mendeteksi aktivitas port scanning dan memberikan catatan serta notifikasi kepada administrator jaringan. Anugrah et al. (2022) menerapkan Suricata sebagai Intrusion Prevention System untuk menghadapi SQL injection dan menunjukkan bahwa proses deteksi bergantung pada kecocokan paket terhadap signature rules; aturan dengan penggunaan kode ASCII sebagai kata kunci dinilai efektif untuk mengenali serangan yang diuji. Kedua penelitian tersebut menegaskan pentingnya pemilihan dan penyusunan aturan dalam meningkatkan kemampuan deteksi sistem berbasis signature.

Penelitian lain menekankan pentingnya pengelolaan log dan visualisasi insiden. Sijabat dan Evo (2023) merancang SIEM berbasis Elastic Stack untuk menganalisis, memantau, mendeteksi, dan menyimpan informasi peristiwa keamanan dari aplikasi web yang rentan. Zain et al. (2023) mengintegrasikan IDS Suricata dengan ELK Stack untuk mendeteksi aktivitas web mining serta menguji fungsi sistem, waktu respons, tingkat deteksi, dan penggunaan memori. Sementara itu, Yudhianto (2023) menerapkan Elastic Stack sebagai SIEM untuk meningkatkan analisis log dan deteksi kejahatan siber pada lingkungan e-government. Hasil penelitian-penelitian tersebut menunjukkan bahwa integrasi mesin deteksi dengan pengelolaan log terpusat dapat membantu administrator memahami peristiwa keamanan secara lebih cepat dan terstruktur.

Berdasarkan ruang lingkup penelitian terdahulu tersebut, sebagian penelitian berfokus pada port scanning, SQL injection sebagai satu jenis serangan, insiden aplikasi web secara umum, aktivitas web mining, atau analisis log pada e-government. Masih diperlukan pengujian yang secara bersamaan membandingkan kemampuan aturan bawaan Suricata dengan kombinasi aturan bawaan dan aturan kustom terhadap SQL injection serta beberapa jenis XSS, kemudian mengevaluasi waktu respons Elasticsearch dalam satu arsitektur pemantauan yang terintegrasi. Kebutuhan tersebut menjadi dasar pengujian sistem pemantauan keamanan jaringan berbasis Suricata dan Elastic Stack pada Instansi ABC.

Penelitian ini bertujuan menguji kemampuan sistem pemantauan keamanan jaringan yang terdiri atas IDS Suricata dan SIEM Elastic Stack dalam mendeteksi serta membantu analisis serangan SQL injection dan XSS. Pengujian dilakukan untuk membandingkan tingkat keberhasilan deteksi antara penerapan aturan bawaan Suricata dan kombinasi aturan bawaan dengan aturan kustom, sekaligus mengukur waktu respons Elasticsearch dalam menyediakan data hasil deteksi. Kontribusi penelitian ini adalah menyediakan bukti empiris mengenai pengaruh penambahan aturan kustom terhadap kemampuan deteksi SQL injection, DOM-Based XSS, Reflected XSS, dan Stored XSS, serta menunjukkan pemanfaatan Elastic Stack untuk mengelola dan memvisualisasikan log Suricata sehingga dapat mendukung kegiatan pemantauan dan analisis oleh tim SOC.

STUDI LITERATUR

Anomali lalu lintas dan serangan siber dapat dideteksi menggunakan IDS. IDS merupakan sistem keamanan yang dirancang untuk mendeteksi aktivitas berbahaya atau serangan pada sistem maupun jaringan. Untuk membangun IDS, diperlukan perangkat lunak Suricata yang dipasang pada server IDS Suricata (Al Hilmi & Khujaemah, 2022; Wahyat & Kudadiri, 2024; Waleed et al., 2022).

Anomali lalu lintas dan serangan siber yang terdeteksi oleh IDS Suricata akan disimpan dalam bentuk log. Namun, jumlah log aktivitas yang dicatat pada IDS Suricata dapat sangat banyak sehingga sulit dibaca oleh tim SOC. SIEM diperlukan sebagai sistem untuk mengelola log dari IDS Suricata. Dalam membangun SIEM, diperlukan perangkat lunak Elasticsearch, Filebeat, dan Kibana dari Elastic Stack. Elastic Stack merupakan produk sumber terbuka yang dikembangkan oleh Elastic BV dan digunakan untuk mengumpulkan, mengindeks, serta menganalisis data (Nova et al., 2022; Sijabat & Evo, 2023; Vazão et al., 2023; Yudhianto, 2023).

Suricata

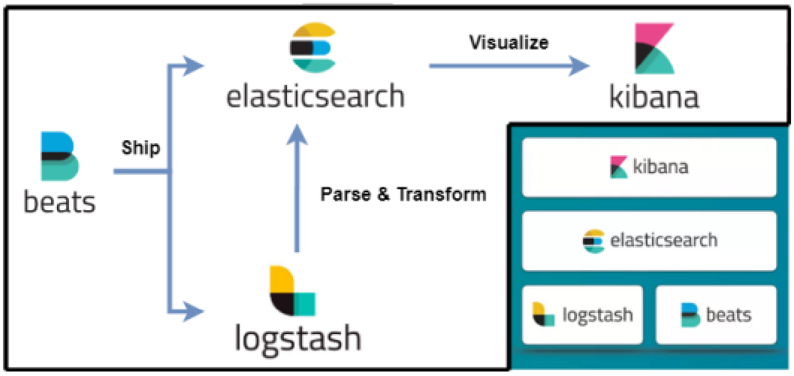
Suricata merupakan perangkat lunak untuk deteksi dan pencegahan intrusi atau Intrusion Detection and Prevention System (IDPS), yang merupakan generasi lanjut dari IDS/IPS. Suricata adalah perangkat lunak sumber terbuka yang dimiliki oleh komunitas nirlaba Open Information Security Foundation (OISF). Seperti Snort, Suricata merupakan Network-based Intrusion Detection Prevention System (NIDPS). Suricata menggunakan metode berbasis signature dan berbasis anomali untuk mendeteksi serangan siber. Metode berbasis signature bekerja dengan mencocokkan lalu lintas jaringan terhadap basis data signature yang memuat metode serangan dan intrusi umum yang digunakan oleh penyerang. Metode berbasis anomali bekerja dengan membandingkan pola serangan umum dengan pola yang sedang dipantau (Díaz-Verdejo et al., 2022; Raharjo & Salman, 2023; Waleed et al., 2022).

Security Information and Event Management (SIEM)

Security Information and Event Management (SIEM) merupakan sistem pemantauan yang bertanggung jawab untuk mengumpulkan, menganalisis, dan melaporkan data keamanan secara terpusat guna mendeteksi ancaman atau insiden serangan. SIEM memiliki kemampuan analitik keamanan secara real time serta analisis riwayat peristiwa dengan menghubungkan berbagai log. Selain itu, SIEM dapat bertukar informasi ancaman atau serangan siber dengan menyediakan koneksi ke platform lain dan menyajikan visualisasi analisis jaringan bagi penggunanya (Setiawan & Sulistyo, 2023; Vazão et al., 2023; Yudhianto, 2023).

Elastic Stack

Elastic Stack, atau kombinasi aplikasi sumber terbuka Elasticsearch, Logstash, dan Kibana, merupakan perangkat yang digunakan untuk mengumpulkan dan memvisualisasikan log. Kombinasi tersebut kemudian disebut Elastic Stack setelah komponen Beats ditambahkan ke dalam stack. Gambar 1 memperlihatkan komponen utama Elastic Stack, yaitu Elasticsearch, Kibana, Logstash, dan Beats. Elasticsearch menyediakan fitur penyimpanan, pencarian, dan analitik. Kibana merupakan komponen yang menyediakan visualisasi dan antarmuka yang ramah pengguna bagi Elastic Stack. Logstash dan Beats merupakan komponen dengan tujuan serupa, yaitu membantu mengumpulkan data yang akan digunakan dalam Elastic Stack (Sijabat & Evo, 2023; Vazão et al., 2023; Zain et al., 2023).



Gambar 1. Arsitektur dan komponen Elastic Stack

Elasticsearch

Elasticsearch merupakan inti dari Elastic Stack yang berperan penting sebagai mesin pencarian dan analitik. Elasticsearch adalah produk sumber terbuka yang dikembangkan menggunakan bahasa pemrograman Java. Dalam Elasticsearch terdapat proses yang disebut data ingestion, yaitu penerimaan data secara real time maupun secara batch. Pada proses ini, data mentah diproses melalui tahapan parsing, normalisasi, dan pengayaan dengan data tambahan sebelum menjalani proses pengindeksan. Pengindeksan merupakan kunci keberhasilan Elasticsearch dalam melakukan pencarian data secara cepat. Selain itu, Elasticsearch bersifat scalable, yang berarti dapat disesuaikan dengan kebutuhan sistem (Elastic, n.d.; Vazão et al., 2023).

Kibana

Kibana merupakan perangkat sumber terbuka untuk visualisasi dan pengelolaan data pada Elasticsearch. Kibana menyediakan berbagai jenis visualisasi data, termasuk histogram, deret waktu, bagan, grafik, peta, tabel, dan lain-lain. Bagian utama Kibana terdiri atas empat bagian, yaitu Management, Discover, Visualize, dan Canvas. Management merupakan bagian untuk melakukan konfigurasi Kibana dan menetapkan index pattern agar index pattern tersebut dapat mengakses data Elasticsearch yang akan dieksplorasi (Vazão et al., 2023).

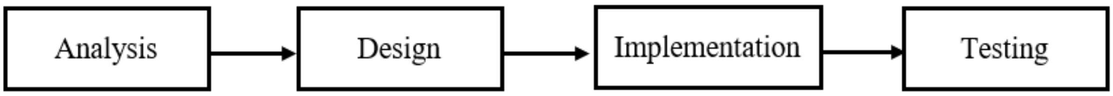
Discover memungkinkan pengguna menelusuri dan menganalisis entri data mentah secara interaktif serta mencari dan memfilternya dengan mudah berdasarkan waktu atau properti dokumen menggunakan bahasa kueri yang disebut Kibana Querying Language (KBL). Kibana menyediakan fitur Dashboard yang berfungsi menggabungkan data yang telah divisualisasikan ke dalam satu halaman dasbor (Sijabat & Evo, 2023; Vazão et al., 2023).

Beats

Beats merupakan kumpulan pengirim log khusus untuk Elastic Stack yang bertindak sebagai agen dan dipasang pada server serta host di lingkungan sistem untuk mengumpulkan log atau metrik. Beats adalah pengirim data yang ringan dan bersifat sumber terbuka. Beats serupa dengan Logstash yang berfungsi sebagai pengumpul dan pengirim data. Perbedaannya, Logstash merupakan komponen sisi server, sedangkan Beats merupakan komponen sisi klien. Beats dapat mengirim data ke Logstash atau langsung ke Elasticsearch. Beats memiliki berbagai jenis yang masing-masing dirancang untuk mengumpulkan data dari sumber tertentu. Beats yang dikembangkan oleh tim Elastic meliputi Filebeat, Metricbeat, Packetbeat, Winlogbeat, Auditbeat, Heartbeat, dan Functionbeat (Sijabat & Evo, 2023; Vazão et al., 2023).

METODOLOGI PENELITIAN

Penelitian ini bertujuan menerapkan Suricata sebagai IDS yang berfungsi mendeteksi anomali lalu lintas dan serangan siber. Selain itu, penelitian ini menerapkan Elastic Stack sebagai SIEM yang berfungsi mengelola log dari IDS Suricata. Penelitian ini terdiri atas empat tahap, yaitu analisis, perancangan, implementasi, dan pengujian. Tahapan penelitian dapat dilihat pada Gambar 2 (Setiawan & Sulistyo, 2023; Zain et al., 2023).



Gambar 2. Tahapan penelitian

Analisis

Analisis dilakukan melalui konsultasi dengan tim SOC untuk mengidentifikasi permasalahan dan kebutuhan yang mendukung penelitian. Tim SOC mengalami kesulitan dalam mendeteksi dan menganalisis anomali lalu lintas serta serangan siber pada sistem jaringan di Instansi ABC. Oleh karena itu, diperlukan penerapan pemantauan keamanan jaringan yang terdiri atas sistem IDS Suricata dan SIEM di Instansi ABC. Terdapat perangkat keras dan perangkat lunak yang diperlukan untuk menerapkan sistem pemantauan keamanan jaringan tersebut (Yudhianto, 2023).

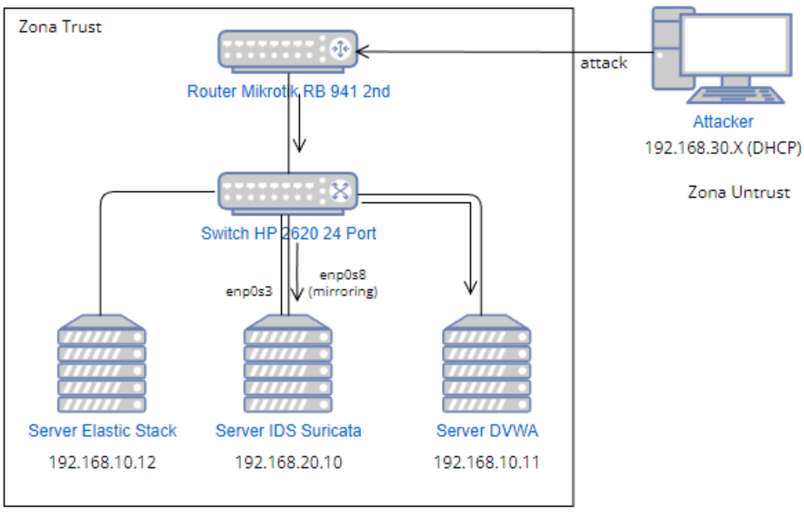
Perangkat keras terdiri atas tiga unit komputer pribadi Asus ROG G20CI, router Mikrotik RB 941-2ND, dan switch HP 1410 24-port. Perangkat lunak meliputi sistem operasi Ubuntu 18.04, sistem operasi Windows 10 Pro 64-bit, VirtualBox, Suricata, Elasticsearch, Filebeat, Kibana, Nginx, Apache2, MySQL Server, dan Damn Vulnerable Web Application (DVWA).

Perancangan

Perancangan topologi jaringan harus dilakukan sebelum memasuki tahap implementasi. Hal ini bertujuan memastikan setiap perangkat dapat terhubung dengan baik. Dengan adanya topologi, setiap perangkat dapat berfungsi secara optimal dan sesuai dengan yang diharapkan. Topologi jaringan dalam penelitian ini dapat dilihat pada Gambar 3.

Berdasarkan Gambar 3, penyerang melakukan serangan dengan memasukkan payload ke situs web DVWA. Situs web DVWA berjalan pada server DVWA. Ketika penyerang melakukan serangan, terjadi aliran data yang melewati router Mikrotik RB 941 2nd dan switch HP 2620, kemudian masuk ke server DVWA. Serangan yang dilakukan penyerang dapat menghasilkan lalu lintas jaringan.

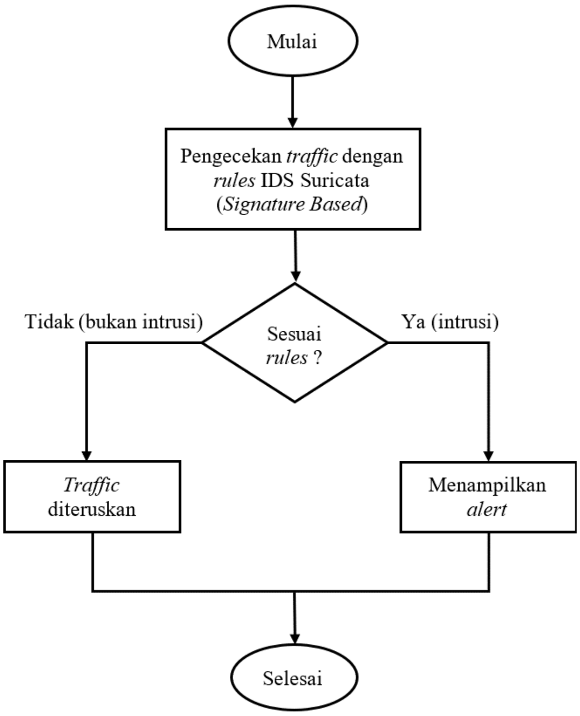
Gambar 3 menunjukkan tiga server virtual, yaitu server Elastic Stack, server IDS Suricata, dan server DVWA yang terhubung ke switch HP. Server Elastic Stack dan server DVWA merupakan mesin virtual yang berada pada sebuah komputer yang terhubung ke port 11 pada switch. Server IDS Suricata terhubung ke port 2 pada switch yang digunakan untuk port mirroring. Port mirroring digunakan untuk mengirimkan salinan lalu lintas pada port 11 agar dapat dipantau.



Gambar 3. Topologi jaringan

Terdapat dua port NIC virtual pada server IDS Suricata, yaitu enp0s3 dan enp0s8. Port enp0s8 pada server IDS Suricata menerima lalu lintas hasil mirroring dari port 2 pada switch HP. Lalu lintas yang diterima berupa lalu lintas masuk maupun keluar dan disimpan dalam bentuk log. Selanjutnya, port enp0s3 pada server IDS

Suricata meneruskan log lalu lintas tersebut ke server Elastic Stack sebagai SIEM. Berdasarkan rancangan topologi jaringan, jenis IDS yang digunakan adalah Network-based Intrusion Detection System (NIDS) (Waleed et al., 2022).



Gambar 4. Diagram alir IDS Suricata

Gambar 4 merupakan diagram alir yang menjelaskan cara kerja IDS Suricata. Berdasarkan metode deteksi berbasis signature yang digunakan, langkah pertama adalah memeriksa lalu lintas terhadap aturan bawaan yang disediakan oleh Suricata dan aturan kustom yang dibuat oleh penulis. Apabila lalu lintas tidak sesuai dengan aturan, lalu lintas tersebut dinyatakan tidak berbahaya atau bukan merupakan intrusi, kemudian diteruskan. Apabila lalu lintas sesuai dengan aturan, lalu lintas tersebut dinyatakan sebagai intrusi dan Suricata menampilkan peringatan pada log (Díaz-Verdejo et al., 2022; Raditya & Sidabutar, 2022).

Implementasi

Dalam mengimplementasikan sistem pemantauan keamanan jaringan yang terdiri atas IDS dan SIEM, diperlukan kegiatan instalasi dan konfigurasi pada server Elastic Stack, router Mikrotik RB 941-2ND, server IDS Suricata, dan server DVWA.

Pengujian

Tahap pengujian dilakukan untuk mengetahui keberhasilan dan kekurangan sistem yang telah dibuat. Pengujian menggunakan dua skenario. Pengujian pertama dilakukan dengan hanya menerapkan aturan bawaan Suricata, sedangkan pengujian kedua dilakukan dengan menerapkan aturan bawaan Suricata dan menambahkan aturan kustom yang dibuat oleh penulis. Pengujian mencakup sejumlah percobaan serangan oleh penyerang terhadap situs web DVWA melalui penyisipan payload SQL injection dan XSS. Pengujian bertujuan mengetahui keberhasilan sistem IDS Suricata dalam mendeteksi berbagai serangan SQL injection dan Cross-Site Scripting (XSS). Selain itu, pengujian juga bertujuan mengukur kinerja waktu respons Elasticsearch pada Elastic Stack dalam menyediakan data yang berasal dari IDS Suricata (Anugrah et al., 2022; Sijabat & Evo, 2023).

HASIL

Hasil Pengujian Menggunakan Serangan SQL Injection

Pengujian serangan SQL injection menggunakan dua jenis serangan, yaitu SQL injection dan blind SQL injection. Kedua jenis serangan tersebut diuji menggunakan dua skenario. Skenario pertama menerapkan aturan bawaan Suricata, sedangkan skenario kedua menerapkan aturan bawaan Suricata dengan penambahan aturan kustom yang dibuat oleh penulis.

Pada kedua skenario, serangan dilakukan secara manual dengan memasukkan payload SQL injection ke situs web DVWA. Situs web DVWA memiliki kerentanan dan digunakan sebagai sasaran serangan SQL injection dalam pengujian ini (Anugrah et al., 2022).

Sebanyak 50 payload SQL injection diuji pada skenario pertama. Hasilnya menunjukkan bahwa sistem IDS dengan aturan bawaan Suricata dapat mendeteksi 16 dari 50 payload SQL injection, sehingga menghasilkan tingkat keberhasilan deteksi sebesar 32%. Demikian pula pada blind SQL injection, sistem IDS dengan aturan bawaan Suricata dapat mendeteksi 16 dari 50 payload, sehingga menghasilkan tingkat keberhasilan deteksi sebesar 32%.

Pada skenario kedua, percobaan serangan dilakukan terhadap 50 payload SQL injection. Hasilnya menunjukkan bahwa sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 dari 50 payload SQL injection, sehingga mencapai tingkat keberhasilan deteksi sebesar 100%. Demikian pula pada blind SQL injection, sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 dari 50 payload, sehingga mencapai tingkat keberhasilan deteksi sebesar 100%.

Berdasarkan dua skenario yang telah dilakukan, dapat diketahui bahwa dari 50 payload SQL injection, sistem IDS dengan aturan bawaan Suricata hanya dapat mendeteksi 16 payload atau memiliki tingkat keberhasilan deteksi sebesar 32%. Sementara itu, sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 payload, yang berarti mampu mendeteksi seluruh payload yang ada. Hal ini terjadi karena aturan kustom yang ditambahkan memuat signature yang menyatakan konten dalam bentuk AND, OR, LIKE, ORDER BY, WHERE, HAVING, dan SLEEP. Dengan demikian, penerapan aturan bawaan Suricata dengan penambahan aturan kustom dapat mendeteksi seluruh payload (Anugrah et al., 2022; Díaz-Verdejo et al., 2022).

Hasil Pengujian Menggunakan Serangan Cross-Site Scripting

Pengujian dilakukan untuk mengetahui keberhasilan sistem IDS dalam mendeteksi berbagai serangan Cross-Site Scripting (XSS). Pengujian serangan XSS menggunakan tiga jenis serangan, yaitu DOM-Based XSS, Reflected XSS, dan Stored XSS. Ketiga jenis serangan tersebut diuji menggunakan dua skenario. Skenario pertama dilakukan dengan menerapkan aturan bawaan Suricata, sedangkan skenario kedua dilakukan dengan menerapkan aturan bawaan Suricata disertai penambahan aturan kustom yang dibuat oleh penulis. Pada kedua skenario, serangan dilakukan secara manual dengan memasukkan payload XSS ke situs web DVWA. Situs web DVWA memiliki kerentanan dan digunakan sebagai objek serangan XSS dalam pengujian ini (Díaz-Verdejo et al., 2022; Sijabat & Evo, 2023).

Pada skenario pertama, percobaan serangan dilakukan terhadap 50 payload XSS. Hasil percobaan menunjukkan bahwa sistem IDS dengan aturan bawaan Suricata dapat mendeteksi 16 dari 50 payload DOM-Based XSS atau memiliki tingkat keberhasilan deteksi sebesar 32%. Demikian pula pada Reflected XSS, sistem IDS dengan aturan bawaan Suricata dapat mendeteksi 16 dari 50 payload atau memiliki tingkat keberhasilan deteksi sebesar 32%. Namun, pada Stored XSS, sistem IDS dengan aturan bawaan Suricata tidak dapat mendeteksi satu pun dari 50 payload sehingga tingkat keberhasilan deteksinya sebesar 0%.

Pada skenario kedua, percobaan serangan dilakukan terhadap 50 payload XSS. Hasil percobaan menunjukkan bahwa sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 dari 50 payload DOM-Based XSS atau memiliki tingkat keberhasilan deteksi sebesar 100%. Pada Reflected XSS, sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 dari 50 payload atau memiliki tingkat keberhasilan deteksi sebesar 100%. Demikian pula pada Stored XSS, sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi 50 dari 50 payload atau memiliki tingkat keberhasilan deteksi sebesar 100%.

Berdasarkan dua skenario yang telah dilakukan, hasil menunjukkan bahwa sistem IDS dengan aturan bawaan Suricata hanya dapat mendeteksi 16 dari 50 payload DOM-Based XSS dan Reflected XSS, dengan tingkat keberhasilan deteksi sebesar 32%. Namun, pada Stored XSS, sistem IDS dengan aturan bawaan Suricata tidak dapat mendeteksi satu pun dari 50 payload sehingga tingkat keberhasilan deteksinya sebesar 0%. Hal ini terjadi karena ketika penyerang memasukkan payload XSS pada halaman Stored XSS, halaman tersebut melakukan encoding terhadap payload XSS yang dimasukkan. Oleh karena itu, dengan hanya menerapkan aturan bawaan, Suricata tidak dapat mendeteksi seluruh payload yang dimasukkan pada halaman Stored XSS (Díaz-Verdejo et al., 2022).

Sementara itu, sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom dapat mendeteksi serangan DOM-Based XSS, Reflected XSS, dan Stored XSS pada seluruh 50 payload, yang berarti dapat mendeteksi semua payload dengan tingkat keberhasilan deteksi sebesar 100%. Hal ini terjadi karena aturan kustom yang ditambahkan memuat signature yang menyatakan konten dalam bentuk Image, Audio, dan Video yang telah melalui encoding. Dengan demikian, penerapan aturan bawaan Suricata dengan penambahan aturan kustom dapat mendeteksi seluruh payload. Berdasarkan kedua skenario tersebut, dapat disimpulkan bahwa sistem IDS dengan aturan bawaan Suricata dan penambahan aturan kustom lebih efektif dalam mendeteksi serangan XSS dibandingkan dengan sistem IDS yang hanya menerapkan aturan bawaan Suricata (Díaz-Verdejo et al., 2022; Raditya & Sidabutar, 2022).

Hasil Pengujian Elastic Stack

Pengujian dilakukan menggunakan dua jenis serangan, yaitu SQL injection dan Cross-Site Scripting (XSS). Pengujian juga dilakukan dengan menerapkan aturan bawaan Suricata yang disertai penambahan aturan kustom yang dibuat oleh penulis.

Pengujian bertujuan mengukur kinerja waktu respons Elasticsearch pada Elastic Stack dalam menyediakan data yang berasal dari IDS Suricata. Kinerja waktu respons Elasticsearch dapat dinyatakan normal apabila rata-rata ΔT tidak melebihi ambang batas waktu respons Elasticsearch dalam menyediakan data, yaitu 1 detik berdasarkan referensi Elastic.co. Pengukuran kinerja waktu respons Elasticsearch dilakukan dengan menghitung rata-rata ΔT atau rata-rata selisih waktu antara event.created (T1) dan timestamp (T0), dengan keterangan sebagai berikut (Elastic, n.d.; Vazão et al., 2023).
T1: event.created adalah waktu ketika Elasticsearch pertama kali memproses data
T0: timestamp adalah waktu yang diambil dari peristiwa asli
N: jumlah payload yang dimasukkan
ΔT: selisih waktu antara event.created (T1) dan timestamp (T0)
Rata-rata ΔT: rata-rata selisih waktu antara event.created (T1) dan timestamp (T0)

$$\Delta T = T_1 - T_0$$

$$\text{Rata-rata } \Delta T = \frac{\sum \Delta T}{N}$$

Untuk jenis serangan SQL injection, sebanyak 50 payload SQL injection diuji. Hasil pengujian menunjukkan bahwa rata-rata ΔT sebesar 640,54 milidetik atau 0,640 detik, sebagaimana ditunjukkan pada Tabel I. Hasil tersebut menunjukkan bahwa rata-rata ΔT tidak melebihi 1 detik atau tidak melampaui ambang batas waktu respons Elasticsearch dalam menyediakan data.

Tabel 1
Hasil Rata-rata Delta T pada Pengujian Menggunakan Serangan SQL Injection

No.	Payload	T1 (event.created)	T0 (timestamp)	ΔT (ms)
1	% ' or '0' = '0	Jun 23, 2023 @ 22:30:11.358	Jun 23, 2023 @ 22:30:10.705	653
2	%' or 0=0 union select null, version() #	Jun 23, 2023 @ 22:33:35.918	Jun 23, 2023 @ 22:33:35.382	536
3	ORDER BY 1--	Jun 23, 2023 @ 22:35:15.212	Jun 23, 2023 @ 22:35:14.588	624
4	'% or 0=0 union select null, version() #	Jun 23, 2023 @ 22:36:31.433	Jun 23, 2023 @ 22:36:30.470	963
5	'LIKE'	Jun 23, 2023 @ 22:38:35.806	Jun 23, 2023 @ 22:38:35.221	585
6	UNION ALL SELECT 1--	Jun 23, 2023 @ 22:40:31.156	Jun 23, 2023 @ 22:40:30.663	493
7	" OR 1 -- -	Jun 23, 2023 @ 22:42:04.431	Jun 23, 2023 @ 22:42:03.436	995
8	' OR 'x'='x	Jun 23, 2023 @ 22:43:28.681	Jun 23, 2023 @ 22:43:27.844	837
..	..	…	…	…
50	%’ UNION SELECT null, concat(user,0x0a, password) FROM users#	Jun 24, 2023 @ 00:00:30.620	Jun 24, 2023 @ 00:00:30.251	369
ΔT (ms)	ΔT (ms)	ΔT (ms)	ΔT (ms)	640,54
(s)	(s)	(s)	(s)	0,64054

Tabel I juga menunjukkan bahwa nilai ΔT atau selisih waktu untuk setiap payload SQL injection berbeda. Hal ini terjadi karena setiap payload SQL injection memiliki fungsi yang berbeda. Selain itu, perbedaan selisih waktu juga dapat disebabkan oleh tingginya penggunaan sumber daya perangkat.

Untuk jenis serangan XSS, sebanyak 50 payload XSS diuji. Hasil pengujian menunjukkan bahwa rata-rata ΔT sebesar 666,2 milidetik atau 0,666 detik, sebagaimana ditunjukkan pada Tabel II.

Tabel 2 menunjukkan bahwa nilai ΔT atau selisih waktu untuk setiap payload XSS juga berbeda. Hal ini terjadi karena setiap payload XSS memiliki fungsi yang berbeda. Selain itu, perbedaan selisih waktu juga dapat disebabkan oleh tingginya penggunaan sumber daya perangkat.

Tabel 2
Hasil Rata-rata Delta T pada Pengujian Menggunakan Serangan XSS

No.	Payload	T1 (event.created)	T0 (timestamp)	ΔT (ms)
1	<script>alert('Hacked')</script>	Jun 24, 2023 @ 00:51:25.625	Jun 23, 2023 @ 00:51:25.000	625
2	<script/src=data:,alert()>	Jun 24, 2023 @ 00:52:53.728	Jun 24, 2023 @ 00:52:53.222	506
3	<img/src/onerror=prompt(8)>	Jun 24, 2023 @ 05:42:06.740	Jun 24, 2023 @ 05:42:06.104	636
4	<image/src/onerror=prompt(8)>	Jun 24, 2023 @ 05:43:12.850	Jun 24, 2023 @ 05:43:12.089	761
5	<image src/onerror=prompt(8)>	Jun 24, 2023 @ 05:44:29.946	Jun 24, 2023 @ 05:44:29.690	256
6	</script</script><img src=q onerror=promp(8)>	Jun 24, 2023 @ 05:45:45.041	Jun 24, 2023 @ 05:45:43.804	1237
7	<img src=1 href=1 onerror="javascript:alert(1)"></img >	Jun 24, 2023 @ 05:46:58.149	Jun 24, 2023 @ 05:46:57.590	559
8	<audio src=1 href=1 onerror="javascript:alert(1)"></aud io>	Jun 24, 2023 @ 05:47:59.240	Jun 24, 2023 @ 05:47:58.988	252
..	..	…	…	…
50	<script>alert(document.cookie)</sc ript>	Jun 24, 2023 @ 08:37:06.457	Jun 24, 2023 @ 08:37:06.152	305
ΔT (ms)	ΔT (ms)	ΔT (ms)	ΔT (ms)	666,2
(s)	(s)	(s)	(s)	0,6662

PEMBAHASAN

Pengujian yang dilakukan menggunakan dua jenis serangan, yaitu SQL injection dan Cross-Site Scripting (XSS), menghasilkan rata-rata ΔT yang tidak melebihi 1 detik atau tidak melampaui ambang batas waktu respons Elasticsearch dalam menyediakan data. Berdasarkan hasil tersebut, dapat disimpulkan bahwa hasil pengukuran kinerja waktu respons Elasticsearch tidak melebihi 1 detik berdasarkan referensi Elastic.co, sehingga SIEM berbasis Elastic Stack dapat berjalan secara normal (Elastic, n.d.; Vazão et al., 2023).

Perbandingan kedua konfigurasi Suricata menunjukkan bahwa aturan bawaan dapat mendeteksi 16 dari 50 payload SQL injection serta 16 dari 50 payload DOM-Based XSS atau Reflected XSS, sedangkan Stored XSS tidak terdeteksi. Kombinasi aturan bawaan dan aturan kustom memungkinkan sistem mendeteksi seluruh 50 payload pada setiap kategori serangan yang diuji. Hasil ini menegaskan bahwa signature kustom diperlukan untuk menangani pola serangan yang belum tercakup dalam kumpulan aturan bawaan.

Temuan tersebut konsisten dengan karakteristik IDS berbasis signature yang sangat bergantung pada kelengkapan pola serangan di dalam kumpulan aturan. Aturan bawaan efektif untuk mengenali pola umum yang telah tersedia, tetapi hasil deteksi sebesar 32% pada SQL injection, DOM-Based XSS, dan Reflected XSS, serta 0% pada Stored XSS menunjukkan adanya celah deteksi ketika bentuk payload tidak secara eksplisit terwakili. Penambahan aturan kustom yang memuat pola AND, OR, LIKE, ORDER BY, WHERE, HAVING, dan SLEEP untuk SQL injection serta pola elemen Image, Audio, dan Video yang telah melalui encoding untuk XSS memperluas cakupan pencocokan signature. Hasil ini sejalan dengan penelitian terdahulu yang menegaskan pentingnya penyesuaian aturan terhadap karakteristik lalu lintas dan pola serangan pada lingkungan yang dilindungi (Anugrah et al., 2022; Díaz-Verdejo et al., 2022; Raditya & Sidabutar, 2022). Dengan demikian, efektivitas Suricata tidak hanya ditentukan oleh perangkat lunaknya, tetapi juga oleh proses pemeliharaan dan pengembangan aturan secara berkelanjutan.

Dari sisi pengelolaan peristiwa keamanan, integrasi Filebeat, Elasticsearch, dan Kibana membuat peringatan Suricata tidak berhenti sebagai log mentah, tetapi dapat dikumpulkan, diindeks, ditelusuri, dan divisualisasikan untuk mendukung analisis tim SOC. Rata-rata waktu respons sebesar 0,640 detik untuk SQL injection dan 0,666 detik untuk XSS menunjukkan bahwa data pengujian tersedia dalam waktu kurang dari satu detik pada konfigurasi dan beban perangkat yang digunakan. Hasil ini mendukung pemanfaatan Elastic Stack sebagai SIEM sebagaimana diuraikan dalam penelitian sebelumnya (Setiawan & Sulistyo, 2023; Vazão et al., 2023; Yudhianto, 2023). Namun, hasil tersebut perlu ditafsirkan sesuai ruang lingkup penelitian karena pengujian hanya menggunakan dua kelompok serangan, masing-masing 50 payload, pada lingkungan DVWA dan sumber daya perangkat tertentu. Perubahan volume lalu lintas, jumlah sumber log, kompleksitas kueri, atau penggunaan sumber daya dapat memengaruhi waktu respons. Oleh karena itu, sebelum diterapkan pada lingkungan operasional yang lebih besar, sistem perlu diuji kembali menggunakan lalu lintas nyata, variasi serangan yang lebih luas, dan skenario beban bertahap.

KESIMPULAN

Penelitian ini menyimpulkan bahwa integrasi IDS Suricata dan SIEM berbasis Elastic Stack dapat digunakan sebagai sistem pemantauan keamanan jaringan untuk membantu tim Security Operation Center (SOC) mendeteksi, mengelola, dan menganalisis anomali lalu lintas serta serangan siber. Pengujian terhadap 50 payload SQL injection menunjukkan bahwa aturan bawaan Suricata hanya mendeteksi 16 payload atau 32%, sedangkan kombinasi aturan bawaan dan aturan kustom mendeteksi seluruh 50 payload atau 100%. Pola yang sama terlihat pada DOM-Based XSS dan Reflected XSS, yaitu tingkat deteksi meningkat dari 32% menjadi 100% setelah aturan kustom diterapkan. Pada Stored XSS, aturan bawaan tidak mendeteksi satu pun dari 50 payload, sedangkan penambahan aturan kustom menghasilkan tingkat deteksi 100%. Temuan tersebut menunjukkan bahwa aturan bawaan belum mencakup seluruh variasi payload yang diuji dan bahwa pengembangan signature kustom sesuai karakteristik serangan merupakan faktor penting dalam meningkatkan kemampuan deteksi Suricata. Dari sisi pengelolaan data keamanan, Filebeat berhasil meneruskan log Suricata ke Elasticsearch, sedangkan Kibana menyediakan fasilitas penelusuran dan visualisasi yang mendukung proses analisis. Pengukuran kinerja Elasticsearch menghasilkan rata-rata selisih waktu 640,54 milidetik atau 0,640 detik untuk SQL injection dan 666,2 milidetik atau 0,666 detik untuk XSS; kedua nilai tersebut berada di bawah satu detik pada lingkungan pengujian yang digunakan. Kontribusi penelitian ini terletak pada pembuktian terukur mengenai peningkatan efektivitas deteksi melalui kombinasi aturan bawaan dan aturan kustom, sekaligus menunjukkan keterpaduan proses deteksi, pengiriman log, pengindeksan, dan visualisasi dalam satu sistem pemantauan. Meskipun demikian, hasil penelitian masih terbatas pada dua jenis serangan, jumlah payload tertentu, aplikasi DVWA, dan konfigurasi perangkat yang digunakan, sehingga hasilnya tidak langsung mewakili seluruh kondisi jaringan operasional. Sebagai pengembangan selanjutnya, pengujian disarankan mencakup jenis serangan lain, lalu lintas nyata dengan volume yang lebih tinggi, beberapa sumber log, evaluasi penggunaan CPU, memori, dan penyimpanan, serta pengujian false positive dan false negative. Aturan kustom juga perlu ditinjau dan diperbarui secara berkala agar tetap relevan terhadap perubahan pola serangan dan tidak menimbulkan beban pemrosesan yang berlebihan.

REFERENSI

Al Hilmi, M. A., & Khujaemah, E. (2022). Network security monitoring with intrusion detection system. *Jurnal Teknik Informatika (JUTIF), 3*(2), 249–253. https://doi.org/10.20884/1.jutif.2022.3.2.117

Anugrah, F. T., Ikhwan, S., & Gusti A. G., J. (2022). Implementasi intrusion prevention system (IPS) menggunakan Suricata untuk serangan SQL injection [Implementation of a Suricata-based intrusion prevention system for SQL injection attacks]. *Techné: Jurnal Ilmiah Elektroteknika, 21*(2), 199–210. https://doi.org/10.31358/techne.v21i2.320

Badan Siber dan Sandi Negara. (2022). *Laporan tahunan monitoring keamanan siber 2021* [Annual cybersecurity monitoring report 2021]. https://www.bssn.go.id/laporan-tahunan-monitoring-keamanan-siber-tahun-2021/

Díaz-Verdejo, J., Muñoz-Calle, J., Estepa Alonso, A., Estepa Alonso, R., & Madinabeitia, G. (2022). On the detection capabilities of signature-based intrusion detection systems in the context of web attacks. *Applied Sciences, 12*(2), 852. https://doi.org/10.3390/app12020852

Elastic. (n.d.). Near real-time search. In *Elasticsearch guide*. Retrieved July 23, 2026, from https://www.elastic.co/guide/en/elasticsearch/reference/current/near-real-time.html

Nova, F., Pratama, M. D., & Prayama, D. (2022). Wazuh sebagai log event management dan deteksi celah keamanan pada server dari serangan DoS [Wazuh as log event management and server vulnerability detection against DoS attacks]. *JITSI: Jurnal Ilmiah Teknologi Sistem Informasi, 3*(1), 1–7.

https://doi.org/10.62527/jitsi.3.1.59

Raditya, F., & Sidabutar, J. (2022). Analisis rules intrusion detection prevention system (IDPS) Suricata untuk mendeteksi dan menangkal aktivitas crypto mining pada jaringan [Analysis of Suricata IDPS rules for detecting and preventing crypto-mining activity on networks]. *Jurnal Edukasi dan Penelitian Informatika (JEPIN), 8*(2), 348–355. https://doi.org/10.26418/jp.v8i2.56194

Raharjo, D. H. K., & Salman, M. (2023). Analyzing Suricata alert detection performance issues based on active indicator of compromise rules. *Jurnal Teknik Informatika (JUTIF), 4*(3), 601–610. https://doi.org/10.52436/1.jutif.2023.4.3.1013

Setiawan, H., & Sulistyo, W. (2023). SIEM (security information event management) model for malware attack detection using Suricata and Evebox. *International Journal of Engineering Technology and Natural Sciences, 5*(2), 138–147. https://doi.org/10.46923/ijets.v5i2.241

Sijabat, D. R., & Evo, S. (2023). Design of security information and event management (SIEM) to detect incidents on websites. *J-INTECH, 11*(1), 10–17. https://doi.org/10.32664/j-intech.v11i1.860

Vazão, A. P., Santos, L., Costa, R. L. de C., & Rabadão, C. (2023). Implementing and evaluating a GDPR-compliant open-source SIEM solution. *Journal of Information Security and Applications, 75*, 103509. https://doi.org/10.1016/j.jisa.2023.103509

Wahyat, W., & Kudadiri, P. (2024). Implementation of intrusion detection system with Suricata on Ubuntu 22.04 LTS. *Jurnal Teknologi Elekterika, 21*(1), 50–53. https://jurnal.poliupg.ac.id/index.php/JTE/article/view/5069

Waleed, A., Jamali, A. F., & Masood, A. (2022). Which open-source IDS? Snort, Suricata or Zeek. *Computer Networks, 213*, 109116. https://doi.org/10.1016/j.comnet.2022.109116

Yudhianto, I. (2023). Simple, fast, and accurate cybercrime detection on e-government with Elastic Stack SIEM. *Jurnal Edukasi dan Penelitian Informatika (JEPIN), 9*(2), 263–276. https://doi.org/10.26418/jp.v9i2.64213

Zain, A. R., Oktivasari, P., Soelaiman, N. F., & Umam, F. W. (2023). Implementasi intrusion detection system (IDS) Suricata dan management log ELK Stack untuk pendeteksian kegiatan mining [Implementation of Suricata IDS and ELK Stack log management for mining-activity detection]. *Jurnal Poli-Teknologi, 22*(1), 23–29. https://doi.org/10.32722/pt.v22i1.4974