Journal of Digital Business and Technology Innovation (DBESTI)

https://journal.nurulfikri.ac.id/index.php/DBESTI
P-ISSN: 3047-1028 E-ISSN: 3032-775X

PENERAPAN ELK STACK PADA APLIKASI WEB UNTUK ANALISIS DATA SERANGAN KEAMANAN SIBER

Muhammad Dzaky Zahran[1], Henry Saptono[2]

[1,2]Teknik Informatika, Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Jakarta Selatan, DKI Jakarta, Indonesia 12640
zahran1607@gmail.com, henry@nurulfikri.ac.id

Abstract

This research aims to develop a cybersecurity attack data analysis web using the ELK Stack, which includes Elasticsearch, Logstash, and Kibana, and is supported by Suricata as an Intrusion Detection System (IDS). The system is designed to detect and visualize cyberattacks such as SQL Injection, Port Scanning, and UDP Flood through dashboards accessible to users. Logs generated by Suricata are read by Filebeat, processed by Logstash, stored in Elasticsearch, and visualized using Kibana. The dashboard is embedded in a local web server. System testing was conducted using White Box Testing and User Acceptance Testing (UAT), showing that the system successfully detects attacks in real-time and presents the data in informative and interactive visual formats. Evaluation results indicate that the system achieved an overall detection success rate of 90%, with limitations mainly due to the custom rules used. These findings demonstrate the system's effectiveness in aiding cybersecurity analysis, particularly in small to medium network environments.

Keywords: *Cybersecurity, ELK Stack, Log Visualization, Port Scanning, SQL Injection, Suricata, UDP Flood*

Abstrak

Penelitian ini bertujuan untuk mengembangkan web analisis data serangan keamanan siber menggunakan ELK Stack, yang terdiri dari Elasticsearch, Logstash, dan Kibana, serta didukung oleh Suricata sebagai *Intrusion Detection System* (IDS). Sistem ini dirancang untuk mendeteksi dan memvisualisasikan serangan siber seperti *SQL Injection*, *Port Scanning*, dan *UDP Flood* dalam bentuk *dashboard* yang mudah diakses oleh pengguna. Log yang dihasilkan oleh Suricata dibaca oleh Filebeat dan diteruskan ke Logstash untuk diproses sebelum disimpan di Elasticsearch dan divisualisasikan di Kibana. *Dashboard* ini kemudian diintegrasikan ke dalam web lokal. Pengujian sistem dilakukan melalui metode *White Box Testing* dan *User Acceptance Testing* (UAT), yang menunjukkan bahwa sistem berhasil mendeteksi serangan secara *real-time* dan menyajikan data dalam bentuk visualisasi yang informatif dan interaktif. Hasil evaluasi menunjukkan bahwa sistem ini memiliki tingkat keberhasilan deteksi dan klasifikasi sebesar 90% secara keseluruhan, dengan kekurangan terletak pada keterbatasan *rule custom* yang digunakan. Dengan hasil ini, sistem telah terbukti cukup efektif dalam membantu proses analisis keamanan siber pada lingkungan jaringan kecil hingga menengah.

Kata kunci: ELK Stack, Keamanan Siber, *Port Scanning*, *SQL Injection*, Suricata, *UDP Flood*, Visualisasi Log

1. PENDAHULUAN

Di era digital saat ini, ancaman terhadap keamanan siber semakin meningkat seiring dengan bertambahnya ketergantungan pada sistem elektronik. Serangan siber seperti *deface*, *port scanning*, *SQL injection*, hingga *UDP flooding* menjadi semakin umum terjadi dan berpotensi menyebabkan kerugian serius pada organisasi atau instansi yang menjadi targetnya. Menurut laporan tahunan Pusopskamsinas Badan Siber dan Sandi Negara (BSSN), domain pemerintah Indonesia (.go.id) merupakan salah satu target utama serangan *defacement*, diikuti oleh domain lain seperti .ac.id dan .co.id [1]. Kondisi ini menunjukkan urgensi penerapan sistem keamanan siber yang efektif untuk mencegah dan mendeteksi berbagai potensi serangan.

Pemerintah Indonesia sendiri telah mengatur tata kelola keamanan sistem elektronik melalui Peraturan Menteri Komunikasi dan Informatika Republik Indonesia Nomor 4 Tahun 2016 tentang Sistem Manajemen Pengamanan Informasi. Dalam regulasi tersebut ditegaskan pentingnya penerapan prinsip keamanan informasi yang mencakup kerahasiaan (*confidentiality*), integritas (*integrity*), dan ketersediaan (*availability*) informasi [2].

Diterima 14 Juli 2025, Direvisi 24 November 2026, Diterima untuk publikasi 24 November 2026

Tujuan dari serangan siber adalah memberikan sebuah dampak pada korban di dunia nyata dengan melakukan pemusnahan integritas (*loss integrity*), ketersediaan (*availability*), kerahasiaan (*confidentiality*), dan pemusnahan fisik (*physical destruction*) [3]. Dalam konteks teknis, keamanan siber mencakup berbagai aspek perlindungan terhadap sistem informasi dan jaringan dari ancaman siber yang terus berkembang. Hal ini melibatkan penerapan teknologi keamanan, kebijakan, dan standar industri untuk memastikan data dan infrastruktur digital tetap aman dari akses yang tidak sah, manipulasi, atau penghancuran oleh pihak yang berniat jahat. Dengan meningkatnya ketergantungan pada teknologi digital, serangan siber tidak hanya mengancam individu atau organisasi, tetapi juga dapat berdampak pada keamanan nasional serta stabilitas ekonomi. Oleh karena itu, perlu ada upaya yang harus dilakukan untuk meningkatkan keamanan secara komprehensif [4].

Salah satu implementasi penting dari prinsip tersebut adalah penerapan sistem yang mampu memonitor, menganalisis, dan mendeteksi aktivitas yang mencurigakan melalui log sistem. Beberapa penelitian terdahulu menyatakan bahwa analisis log sistem dapat menjadi metode yang efektif dalam mendeteksi dan merespons berbagai jenis serangan siber. Dengan memanfaatkan log yang dihasilkan oleh sistem keamanan seperti IDS, *firewall*, atau server, administrator dapat melakukan identifikasi dini terhadap pola serangan dan mengambil tindakan mitigasi yang tepat [5]. Selain itu, visualisasi log melalui *dashboard* yang terintegrasi juga dapat meningkatkan efisiensi pemantauan dan mempercepat pengambilan keputusan [6].

Log berisi informasi tentang pesan-pesan dari server, operasi *kernel*, dan segala *service* dan operasi yang berjalan pada sistem. Log memainkan peran penting dalam administrasi sistem karena mereka berisi tiap-tiap detail penting tentang hal-hal yang dilakukan oleh pengguna sistem atau operasi yang dilakukan oleh *file-file* berbahaya yang berada dalam sistem. Dengan menganalisis log-log ini, kita bisa mengidentifikasi cara yang digunakan oleh penyerang untuk mendapatkan akses ke dalam sistem, beserta hal-hal yang dilakukannya setelah memasuki sistem [7].

Berdasarkan permasalahan tersebut, penelitian ini bertujuan untuk merancang dan mengimplementasikan web analisis data serangan keamanan siber berbasis ELK Stack (Elasticsearch, Logstash, dan Kibana) yang terintegrasi dengan IDS Suricata. Dengan mengintegrasikan ELK Stack, sistem IDS berbasis web dapat meningkatkan efektivitas deteksi dan respons terhadap ancaman siber, serta memungkinkan pemantauan *real-time* terhadap anomali dalam jaringan [8]. Selain dari pemrosesan data deteksi, ELK Stack juga menyediakan berbagai opsi untuk visualisasi data terutama dari sumber lain [9]. Dalam penelitian ini, sumber tersebut adalah IDS Suricata.

Penggunaan ELK Stack dalam implementasi sistem ini memiliki beberapa kelebihan, seperti skalabilitasnya yang memungkinkan pengembangan lebih lanjut, otomasinya yang dapat menyimpan dan mengindeks data secara otomatis, dan juga antarmuka yang ramah pengguna [10]. Dengan ini, sistem akan mampu mendeteksi berbagai jenis serangan yang disimulasikan, seperti *SQL Injection*, *UDP Flooding*, dan *Port Scanning*, serta menyajikan hasilnya dalam bentuk visualisasi *dashboard* yang interaktif dan mudah diakses.

2. METODE PENELITIAN

2.1 Metode Pengumpulan Data, Instrumen Penelitian, dan Metode Pengujian

Dalam penelitian ini, data dikumpulkan menggunakan metode eksperimen yang berasal dari data log yang merekam aktivitas dalam sistem jaringan dan server, khususnya data log yang dihasilkan oleh Suricata. Data yang digunakan berasal dari data log Suricata, serta data yang sudah diproses oleh Filebeat, Logstash, dan Elasticsearch, serta yang divisualisasikan oleh Kibana. Jenis data yang dikumpulkan dalam penelitian ini berupa log keamanan yang berasal dari Suricata, yang mencakup berbagai informasi hasil inspeksi lalu lintas jaringan. Informasi yang terdapat dalam data log ini mencakup beberapa elemen penting, seperti alamat IP sumber serangan, waktu kejadian, jenis serangan yang terdeteksi, metode akses yang digunakan (*HTTP request*, *SQL query*, atau eksploitasi protokol tertentu), serta status respons dari sistem.

Penelitian ini menggunakan dua metode pengujian untuk mengevaluasi kinerja dan efektivitas Web Analisis Data Serangan Keamanan Siber menggunakan ELK Stack.

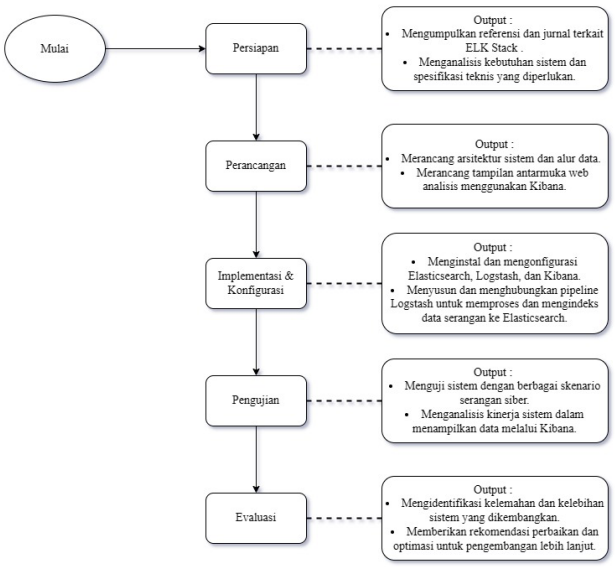
1. *White box testing* digunakan untuk menguji bagian internal dari sistem, terutama alur logika pemrosesan data dari log serangan siber yang masuk ke sistem. Pengujian ini mencakup validasi terhadap *rule alert*, *pipeline* log, dan struktur indeks yang digunakan dalam Elasticsearch. Dalam konteks ELK Stack, *white box testing* dilakukan dengan memverifikasi bahwa setiap *rule* mendeteksi pola serangan secara benar, *pipeline* memproses log dengan *field* yang tepat, dan *output* data berhasil disimpan di *index* sesuai format yang telah dirancang. Dengan pendekatan ini, *developer* dapat memastikan setiap bagian dari proses deteksi dan *parsing* log berjalan sesuai logika sistem yang diharapkan.
2. *User Acceptance Testing* dilakukan untuk memastikan bahwa sistem yang telah dibangun dapat digunakan secara nyaman dan sesuai kebutuhan pengguna akhir. Dalam konteks ini, UAT dilakukan oleh pengguna (sebagai admin jaringan atau analis keamanan) dengan cara mengakses *dashboard* yang telah dibuat, memeriksa tampilan visualisasi serangan, dan menguji fungsi *time picker* serta keakuratan informasi yang ditampilkan. Pengujian ini bertujuan untuk menilai apakah *dashboard* mudah dipahami, cepat merespons,

dan menyajikan informasi keamanan yang relevan dan berguna bagi pengambilan keputusan.

2.2 Tahapan Penelitian

Tahapan penelitian terdiri dari 5 tahap (Gambar 1):

1. Tahap awal penelitian adalah tahap Persiapan, yaitu mengumpulkan referensi dan jurnal yang berkaitan dengan ELK Stack. Referensi dan jurnal tersebut akan menjadi acuan dalam penelitian analisis keamanan siber. Persiapan untuk sistem juga dilakukan dengan analisis kebutuhan dan spesifikasi teknis yang diperlukan untuk implementasi ELK Stack.
2. Setelah kebutuhan dan spesifikasi teknis sistem telah ditetapkan, tahap selanjutnya adalah tahap Perancangan. Pada tahap ini dilakukan penyusunan arsitektur sistem dan alur data yang akan diproses oleh ELK Stack. Perancangan ini mencakup bagaimana data serangan dikumpulkan, diproses, serta merancang bagaimana data ditampilkan dalam bentuk visualisasi menggunakan Kibana dapat dipahami.
3. Pada tahap Implementasi dan Konfigurasi, setelah menyiapkan perangkat keras, dilakukan instalasi dan konfigurasi Elasticsearch, Logstash, Kibana, Filebeat, Suricata, beserta perangkat lunak lain yang diperlukan. Dilakukan juga beberapa konfigurasi khusus seperti konfigurasi Filebeat agar membaca log dari Suricata, konfigurasi *database* untuk simulasi *SQL Injection*, dan lainnya.
4. Setelah implementasi dan konfigurasi, dilakukan tahap Pengujian dengan menyimulasikan skenario serangan siber yang sudah ditentukan pada batasan masalah. Pengujian ini dilakukan untuk melihat bagaimana sistem mendeteksi, menangani, dan mencatat serangan tersebut dalam Elasticsearch, serta bagaimana informasi tersebut divisualisasikan melalui Kibana. Dari pengujian ini dilakukan analisis terhadap kinerja sistem dalam memproses serta menampilkan data dengan kecepatan dan akurasi yang optimal.
5. Tahap yang terakhir dalam penelitian ini adalah mengevaluasi sistem berdasarkan hasil pengujian yang telah dilakukan. Identifikasi kelemahan dan kelebihan sistem menjadi fokus utama untuk memahami efektivitas ELK Stack dalam mendeteksi dan menganalisis serangan siber. Dari hasil evaluasi ini, diberikan rekomendasi perbaikan serta optimasi agar sistem dapat dikembangkan lebih lanjut dan lebih efektif dalam penerapannya pada lingkungan yang lebih luas.



Gambar 1. Tahapan Penelitian

3. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem

Analisis ini bertujuan untuk menentukan spesifikasi minimum yang dibutuhkan untuk menjalankan sistem *monitoring* dan deteksi serangan menggunakan ELK Stack. Dengan spesifikasi yang tepat, sistem dapat bekerja secara *real-time* dalam menerima, memproses, dan memvisualisasikan data log dari berbagai sumber serangan (Tabel 1 dan Tabel 2).

Tabel 1. Spesifikasi Perangkat Keras

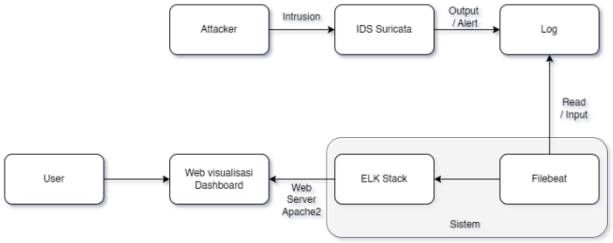
Nama	Spesifikasi	Fungsi
CPU	Ryzen 3 5300U	Memproses seluruh perintah dan operasi logika dari sistem, termasuk analisis data log oleh ELK Stack
RAM	20 GB	Menyimpan data sementara yang sedang diproses. ELK membutuhkan RAM yang cukup untuk proses *indexing* dan visualisasi log *real-time*
Storage	512 GB	Menyimpan log data dari Filebeat, Suricata, serta *index* Elasticsearch yang dapat mencapai ukuran besar tergantung intensitas trafik
Network Interface Card	VirtualBox Host-Only Adapter	Dibutuhkan untuk menangkap lalu lintas jaringan. Suricata akan memonitor lalu lintas dari *interface* ini

Tabel 2. Spesifikasi Perangkat Lunak

Nama	Versi	Fungsi
Sistem Operasi	Ubuntu 20.04	Menjadi dasar untuk komponen-komponen perangkat lunak utama, mengelola sumber daya sistem, dan menjalankan aplikasi
Virtualisasi	Oracle VM VirtualBox 7.0.18	Menjalankan mesin virtual (Ubuntu) tanpa perlu perangkat fisik terpisah
Webserver localhost	Apache2 2.4.41	Digunakan sebagai *platform hosting dashboard* visualisasi berbasis web yang di-*embed* dari Kibana
Elasticsearch	Versi 7.17.28	Mesin pencarian dan penyimpanan utama dalam ELK Stack. Menyimpan dan mengindeks semua data log untuk dianalisis

Nama	Versi	Fungsi
Logstash	Versi 7.17.28	Memproses data log sebelum dikirim ke Elasticsearch
Kibana	Versi 7.17.28	Menyediakan antarmuka visualisasi data dari Elasticsearch, serta memungkinkan pembuatan *dashboard monitoring* serangan
Filebeat	Versi 7.17.28	Agen log ringan yang membaca *file* log `eve.json` milik Suricata dan meneruskannya ke komponen selanjutnya
Suricata	7.0.10	*Intrusion Detection System* (IDS) yang menganalisis lalu lintas jaringan secara *real-time* dan menghasilkan log `eve.json` yang berisi informasi detail mengenai serangan
DVWA (*vulnerable app*)	1.10 *development version*	Aplikasi web rentan untuk simulasi serangan *SQL Injection*
Windows (*Attacker*)	Windows 11	Digunakan sebagai perangkat penyerang untuk menyimulasikan serangan

Perancangan sistem mencakup perencanaan alur kerja pendeteksian dan visualisasi serangan siber menggunakan ELK Stack yang terintegrasi dengan Suricata sebagai sistem deteksi intrusi. Filebeat digunakan untuk membaca log dari Suricata, yang selanjutnya dikirim ke Logstash guna diformat sebelum disimpan di Elasticsearch. Hasil analisis data kemudian divisualisasikan menggunakan Kibana dalam bentuk grafik serangan dan informasi penting lainnya (Gambar 2).



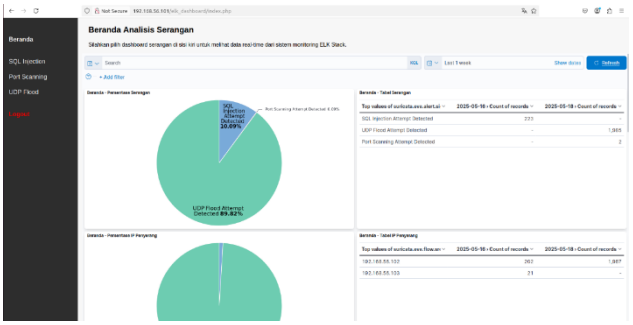
Gambar 2. Perancangan Arsitektur Sistem

Arsitektur sistem menggambarkan bagaimana log dari aktivitas mencurigakan yang terdeteksi oleh Suricata dikirim melalui Filebeat ke Logstash, lalu diteruskan ke Elasticsearch untuk diolah dan disimpan. Seluruh proses ini terintegrasi sehingga log serangan dapat ditampilkan secara *real-time* di Kibana. Diagram arsitektur sistem menunjukkan keterhubungan antarkomponen serta alur data dari sumber hingga visualisasi akhir.

3.2 Hasil Visualisasi

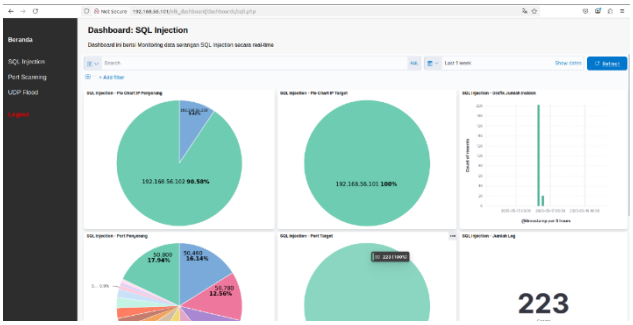
Pada masing-masing *dashboard*, *user* bisa mendapatkan informasi terkait alamat IP penyerang dan alamat IP target, *port* penyerang dan *port* target dalam bentuk *pie chart*, yang menampilkan persentase dari tiap alamat IP sesuai dengan jumlahnya. *User* juga mendapatkan informasi grafik jumlah insiden yang rentang waktunya bisa diatur, total jumlah log deteksi serangan, dan waktu terbaru log deteksi serangan sesuai dengan rentang waktu yang sudah diatur pada *time picker* di pojok kanan atas. Pada *search bar* di bagian atas, *user* juga bisa melakukan *filtering* pada visualisasi dan mengatur *field* mana saja yang ingin ditampilkan ataupun disembunyikan.

Pada Gambar 3 ini adalah tampilan *dashboard* halaman beranda web visualisasi, yang menampilkan gabungan data dan analitik hasil deteksi dari ketiga simulasi serangan.



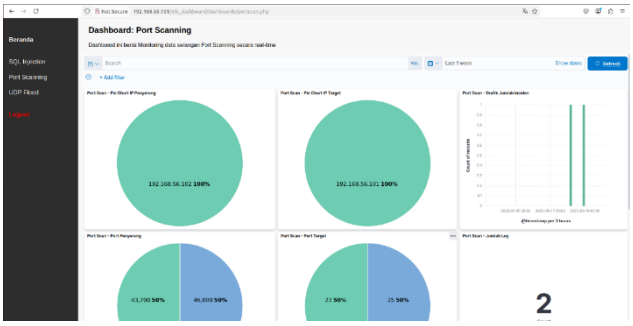
Gambar 3. Halaman Beranda Web Visualisasi

Pada Gambar 4 ini adalah tampilan *dashboard* visualisasi *SQL Injection*, yang menampilkan data dan analitik hasil deteksi simulasi serangan *SQL Injection*.



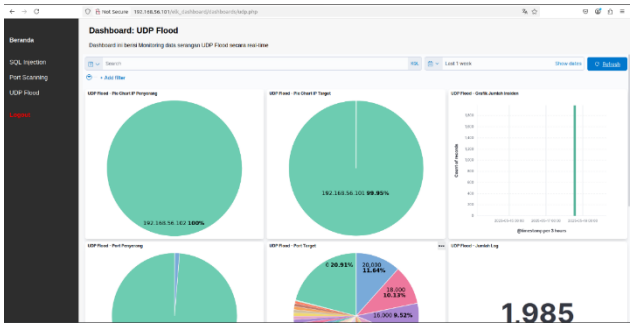
Gambar 4. Halaman Web *Dashboard SQL Injection*

Pada Gambar 5 ini adalah tampilan *dashboard Port Scanning*, yang menampilkan data dan analitik hasil deteksi simulasi serangan *Port Scanning*.



Gambar 5. Halaman Web *Dashboard Port Scanning*

Pada Gambar 6 ini adalah tampilan *dashboard UDP Flood*, yang menampilkan data dan analitik hasil deteksi simulasi serangan *UDP Flood*.



Gambar 6. Halaman Web *Dashboard UDP Flood*

3.3 Hasil Pengujian

Pada penelitian ini digunakan 2 metode penelitian, yaitu *White Box Testing* dan *User Acceptance Testing* (UAT). Kedua pendekatan ini digunakan untuk memastikan bahwa semua komponen sistem sudah berjalan baik secara internal maupun dari sudut pandang pengguna (Tabel 3 dan Tabel 4).

Tabel 3. Metode Pengujian *White Box Testing*

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Simulasi serangan *UDP Attack*.	IDS Suricata dapat mendeteksi serangan dan mencatat data serangan di log.	Berhasil
2	Simulasi serangan *SQL Injection*.	IDS Suricata dapat mendeteksi serangan dan mencatat data serangan di log.	Berhasil
3	Simulasi serangan *Port Scanning*.	IDS Suricata dapat mendeteksi serangan dan mencatat data serangan di log.	Berhasil
4	Filebeat dapat membaca log dan mengirim ke ELK Stack	Muncul data log pada bagian *Discover* yang berasal dari hasil log data serangan yang dikirim oleh Filebeat.	Berhasil
5	*Lens* dan *Metric* dapat membaca data serangan	*Lens* dan *Metric* dapat membaca data serangan yang dikirim oleh Filebeat dan memvisualisasikan data sesuai format yang sudah diatur	Berhasil
6	*Dashboard* dapat menampilkan *Lens* dan *Metric*	*Dashboard* menampilkan *Lens* dan *Metric* serta dapat *update* visualisasi secara *real-time* setiap ada serangan	Berhasil

Pada metode ini, pengujian dilakukan terhadap proses kerja sistem yang dimulai dari Filebeat yang membaca log hasil deteksi serangan oleh IDS Suricata lalu dikirim ke Logstash untuk diproses, diindeks, dan analisis oleh Elasticsearch, lalu divisualisasikan oleh Kibana. Sistem ini berhasil mendeteksi 3 simulasi serangan yang telah dijalankan, yaitu *UDP Flood* (DoS), *SQL Injection*, dan *Port Scanning* yang ditandai dengan munculnya *alert* pada log dari IDS Suricata ketika simulasi serangan dijalankan. Ketika *rule* yang dikonfigurasi pada IDS Suricata sudah *trigger*, akan muncul informasi tambahan pada log yang terbaca seperti *alert signature* dan *alert type*. *Lens* dan *Metric* pada konfigurasi *dashboard* juga sudah berhasil membaca dan menampilkan data hasil deteksi serangan dari IDS Suricata. Persentase keberhasilan sistem dalam mendeteksi dan mengklasifikasi ketiga simulasi serangan kurang lebih 90%–95% dengan sebagian kecil log serangan yang masih misidentifikasi walaupun jarang. Perlu dijelaskan juga bahwa persentase keberhasilan ini bisa berkaitan/disebabkan karena penggunaan *rule custom* untuk simulasi serangan, dengan potensi persentase keberhasilan deteksi dan klasifikasi serangan yang lebih tinggi jika menggunakan *rule* yang lain ataupun *rule* bawaan dari IDS Suricata. Web visualisasi pada sistem ini juga berhasil untuk beroperasi menampilkan *dashboard* visualisasi Kibana pada *browser* dan berjalan secara *real-time* menampilkan *update* visualisasi ketika ada data serangan baru yang masuk.

Tabel 4. Metode Pengujian *User Acceptance Testing*

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Pengguna mengakses web visualisasi	Pengguna dapat mengakses web beranda visualisasi dan bisa memilih web *dashboard* yang ingin dilihat	Berhasil
2	Pengguna mengakses web *dashboard* melalui *sidebar*	Pengguna dapat mengakses web *dashboard* melalui *sidebar* dan bisa berganti dari web *dashboard* ke web *dashboard* yang lain	Berhasil
3	Pengguna mengatur rentang waktu visualisasi data serangan menggunakan *time picker*	Pengguna bisa memilih rentang waktu yang diinginkan dan web *dashboard* dapat *update* data visualisasi sesuai rentang waktu yang dipilih pengguna	Berhasil
4	Pengguna melakukan *filtering* data visualisasi	Pengguna berhasil memilih kriteria visualisasi data tertentu yang ingin divisualisasikan menggunakan *filtering*	Berhasil
5	Pengguna melihat *update* data visualisasi secara *real-time*	Pengguna bisa mendapatkan data visualisasi serangan terbaru jika rentang waktu yang dipilih diatur sampai "*now*"	Berhasil

Pada metode ini, pengujian dilakukan dari sudut pandang pengguna yang hanya berinteraksi dengan web visualisasi. Setelah pengguna mengakses web visualisasi, pengguna akan ditampilkan web beranda visualisasi yang menampilkan data serta statistik keseluruhan dari serangan yang dideteksi, dan bisa memilih *dashboard* visualisasi data serangan yang diinginkan. Pengguna bisa mengakses web visualisasi *dashboard* dari *sidebar* pada bagian kiri, dan bisa berganti dari *dashboard* ke *dashboard* yang lain sesuai keinginan pengguna. Ketika berada di web *dashboard*, pengguna bisa mengatur rentang waktu dari visualisasi data yang ingin ditampilkan menggunakan fitur *time picker* di bagian kanan atas. Pengguna juga bisa melakukan *filtering* data yang divisualisasikan, seperti menampilkan data hanya dari IP tertentu, ataupun *filtering* sesuai dari *field* yang ada pada *Index Pattern*. Pengguna juga bisa mendapatkan

update data visualisasi secara *real-time*, jika rentang waktu yang sedang dipilih oleh pengguna mencapai "*now*" atau saat ini.

4. KESIMPULAN

Rancangan Web Analisis Data Serangan Keamanan Siber Menggunakan ELK Stack telah berhasil dibangun sesuai dengan perancangan sistem yang dirumuskan sebelumnya. Sistem terdiri dari beberapa komponen utama yaitu Suricata sebagai IDS, Filebeat sebagai pembaca log, Logstash untuk pemrosesan data, Elasticsearch sebagai penyimpanan, dan Kibana untuk visualisasi. Seluruh data visualisasi diakses melalui halaman web yang telah di-*embed* dan dilengkapi fitur autentikasi pengguna. Sistem ini terbukti mampu menampilkan data serangan secara *real-time* dalam bentuk *dashboard* yang informatif dan mudah dipahami. Efektivitas dari sistem yang dibangun dalam mendeteksi dan menampilkan data serangan menunjukkan hasil yang memuaskan, dengan tingkat keberhasilan sistem dalam mendeteksi dan mengklasifikasi ketiga simulasi serangan mencapai sekitar 90%–95%. Hasil ini didasarkan pada kemampuan sistem untuk merespons sebagian besar simulasi serangan dengan menghasilkan log yang sesuai dan menampilkannya dalam *dashboard*. Meskipun terdapat beberapa serangan yang tidak terdeteksi secara akurat karena keterbatasan *rule custom* yang digunakan pada Suricata, sistem telah membuktikan bahwa pendekatan menggunakan ELK Stack efektif dalam memproses dan memvisualisasikan data serangan. Dengan pengembangan lebih lanjut, seperti menambahkan *rule* bawaan Suricata, tingkat deteksi dapat ditingkatkan dan menjadikan sistem ini lebih andal.

Ucapan Terima Kasih

Penulis mengucapkan terima kasih kepada seluruh pihak yang membantu dalam menyelesaikan penelitian ini, khususnya pihak dosen pembimbing dan mahasiswa Sekolah Tinggi Teknologi Terpadu Nurul Fikri.

DAFTAR PUSTAKA

[1] A. Admi and A. Hakim Nur Maulana, "Penerapan Elastic Stack sebagai Tools Alternatif Pemantauan Traffic Jaringan dan Host pada Instansi Pemerintah untuk Memperkuat Keamanan dan Ketahanan Siber Indonesia," Jun. 2020.

[2] M. Rijal Kamal and M. Andri Setiawan, "Deteksi Anomali dengan Security Information and Event Management (SIEM) Splunk pada Jaringan UII," Aug. 2021.

[3] B. Hasan, "Implementasi Log Event Managmenet Web Application Firewall menggunakan Elasticsearch Logstash dan Kibana," Aug. 2021.

[4] N. Kristianti and R. Kurniasi, "Peraturan dan Regulasi Keamanan Siber di Era Digital," May 2024. [Online]. Available: https://ejournal.iahntp.ac.id/index.php/satya-dhamat

[5] M. N. Rafli, E. Naf'an, and E. P. W. Mandala, "Analisis dan Peningkatan Performa Log Analysis Pada Server Dengan Elk Stack," *JSTIE (Jurnal Sarjana Teknik Informatika) (E-Journal)*, vol. 12, no. 1, p. 1, Mar. 2024, doi: 10.12928/jstie.v12i1.28235.

[6] D. Hariyadi and Fazlurrahman, "Analisis Serangan Web Defacement pada Situs Web Pemerintah Menggunakan ELK Stack," 2019. [Online]. Available: www.zone-h.org

[7] J. B. Babu, S. Prasad, and G. S. Prasad, "Detecting and Analyzing the Malicious Linux Events using Filebeat and ELK Stack," Apr. 2019, [Online]. Available: www.ijeat.org

[8] T. S. Riera, J. R. B. Higuera, J. B. Higuera, J. J. M. Herraiz, and J. A. S. Montalvo, "Prevention and Fighting against Web Attacks through Anomaly Detection Technology. A Systematic Review," *Sustainability (Switzerland)*, vol. 12, no. 12, Jun. 2020, doi: 10.3390/su12124945.

[9] V. Kosheliuk and Y. Tulashvili, "Implementing Honeypots for Detecting Cyber Threats with AWS using the ELK," *International Journal of Computing*, pp. 618–624, Dec. 2024, doi: 10.47839/ijc.23.4.3761.

[10] W. Sholihah, S. Pripambudi, and A. Mardiyono, "Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack)," *JTIM: Jurnal Teknologi Informasi dan Multimedia*, vol. 2, no. 1, pp. 12–20, May 2020, doi: 10.35746/jtim.v2i1.79.