Faktor Exacta

Vol. 16, No. 1, March 2023, pp. 50~56
eISSN: 2502-339X, pISSN: 1979-276X, DOI: 10.30998/faktorexacta.v16i1.16534

Monitoring dan Evaluasi Keamanan Jaringan dengan Pendekatan *Security Information and Security Management* (SIEM)

Muhamad Ramli[1], Benfano Soewito[2]
[1,2]Megister Teknik Informatika, Universitas Bina Nusantara, Indonesia

Article Info

Article history:

Received Feb 16, 2022
Revised Feb 21, 2023
Accepted Feb 28, 2023

Keywords:

Log
Monitoring
Network Security
Open Source
SIEM

ABSTRACT

Every system produces independent logs. This makes monitoring logs difficult if not done centrally. The research objective is to monitor and evaluate network security using open source-based Security Information and Event Management (SIEM). The research methods include literature studies, SIEM review, observation at the Data and Information System Center (PDSI), simulation of Open Source SIEM implementation by combining devices in real and GNS3 simulation networks, SIEM deployment using Docker, and the final stage of SIEM application evaluation. The implemented SIEM is able to fulfill 84% of the initial requirements. SIEM integrated with Pfsense firewall and Suricata-Intrusion Prevention System (IPS). Monitoring and evaluation features such as detection and alerting, analysis and investigation, compliance and audit, integration and interoperability, monitoring and reporting, support, and maintenance are important parts of SIEM.

Corresponding Author:

Muhamad Ramli,
Magister Teknik Informatika,
Universitas Bina Nusantara,
Jl. Raya Kebon Jeruk No.27, Kec. Kebon Jeruk, Kota Jakarta Barat.
Email: muhamad.ramli001@binus.ac.id

1. PENDAHULUAN

Dalam upaya penerapan keamanan biasanya dilakukan menggunakan perangkat seperti *firewall, antivirus, filtering spam,* sistem intrusi dan lainnya yang bekerja secara terpisah satu dengan lainnya. Oleh karena itu, dalam monitoring setiap sistem tersebut memiliki log masing - masing. Masalah mendasar dalam pengelolaan log adalah menyeimbangkan jumlah sumber daya pengelolaan log dengan pasokan data log yang berkelanjutan. Faktor yang mempersulit dalam pembuatan dan pengelolaan log diantaranya besarnya log, konten yang tidak konsisten, format dan *timestamp* antar sumber dan volume data log yang semakin besar. Selain itu harus memastikan administrator keamanan, sistem dan jaringan secara teratur melakukan analisis data log secara efektif. Pengelolaan log juga melindungi kerahasiaan, integritas dan ketersediaan dari log [1]. Untuk menangani data, meningkatkan keamanan serta mengelola log secara terpusat dan analisis keamanan informasi dan pengelolaan event dibutuhkan sebuah sistem tersendiri.

Sistem ini dikenal dengan istilah SIEM *(Security Information and Event Management)*. Istilah SIEM diciptakan oleh Mark Nicolett dan Amrit Williams dari Gartner pada Tahun 2005 [2]. SIEM merupakan kombinasi dari *Security Information Management* (SIM) dan *Security Event Management* (SEM). SIM berfokus pada pengendalian internal, yaitu memantau perilaku pihak yang berwenang dan akses ke sumber daya internal, dan menyediakan kepatuhan pengelolaan, biasanya menyediakan penyimpanan dalam jangka panjang yang ditujukan untuk analisis dan pelaporan data log sedangkan SEM berhubungan dengan pemantauan nyata perilaku internal dan eksternal, korelasi peristiwa, tanggap darurat untuk insiden keamanan dan notifikasi, yang lebih berfokus pada keamanan itu sendiri [3].

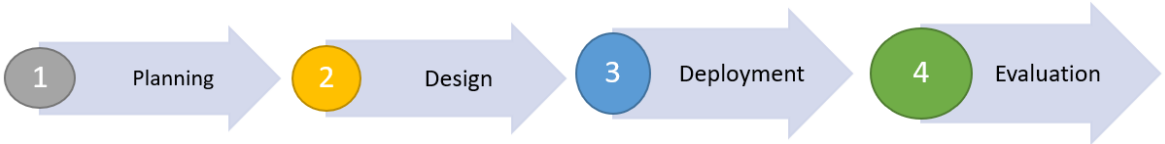
Saat ini banyak vendor yang memberikan solusi implementasi SIEM secara komersial [4]. Namun implementasi memiliki harga yang sangat mahal. Selain itu, untuk implementasi diperlukan kepakaran pada bidang keamanan informasi. Beberapa solusi pendekatan *open source* SIEM dibuat diantaranya OSSIM, ELK Stack, Gray log, DSIEM, Wazuh. Beberapa penelitian telah membandingkan implementasi SIEM *open source* [5] dan melakukan *profiling tools* SIEM dan mesin korelasi untuk analisis keamanan [6]. Perbandingan memberikan pendekatan mendasar pada SIEM *open source* yaitu dengan membandingkan OSSIM Alient Vault, ELK Stack, Splunk free, dan Grayslog. Kemudian penelitian [7] membandingkan software OSSIM, Prelude, dan iView pada lingkungan *smart grid*. Hasilnya OSSIM alient Vault banyak yang dipilih. Saat ini juga berkembang wazuh sebuah *open source security platfom* [8] sebagai alternatif SIEM.

Penelitian [9] mengajukan beberapa usulan perancangan *serverless* SIEM. Pendekatan yang diusulkan memiliki tiga tantangan diantaranya: pertama, bagaimana menambah retensi untuk *event* keamanan pada SIEM saat ini. Tantangan kedua yaitu bagaimana menyediakan aturan atau mesin korelasi secara elastis dan berbiaya rendah di lingkungan *cloud*. Tantangan ketiga, bagaimana membuat SIEM *cloud* yang *elastic*, layanan murah seperti *object storage* atau *Function as a Service* (FaaS). Selain itu, penelitian [10] mengusulkan metode evaluasi pemilihan SIEM sebelum implementasi pada lingkungan perusahaaan. Usulan yang diberikan berupa saran strategi pra-instalasi, cara untuk mengevaluasi komponen fungsionalitas yang harus ada pada SIEM dalam hal persyaratan teknis dan organisasi. Pendekatan pemilihan SIEM dibagi menjadi lima komponen yaitu [10] yaitu: 1. *Platform*, menggambarkan kebutuhan teknis yang ada pada platform. 2. *Operation*, kumpulan dari kebutuhan untuk mengelola solusi. 3. *Integration*, bagian dari kelompok kebutuhan untuk mengintegrasikan solusi SIEM kedalam sistem informasi perusahaan. 4. *Advanced Features*, pada bagian ini menggambarkan fitur lebih lanjut yang mempertimbangkan persyaratan yang harus dimiliki. *4. Licensing and support*, pada bagian ini menggambarkan daftar dan dukungan dari layanan yang dibutuhkan.

Berdasarkan penelitian yang sudah diuraikan sebelumnya maka penelitian ini akan mengadopsi pendekatan solusi *deployment* SIEM dikombinasikan dengan *Next Generation Firewall* (NGFirewall) sebagai bagian tantangan pada penelitian [9] dan skema evaluasi dari penelitian [10] dalam implementasi SIEM sehingga tujuan penelitian ini untuk memonitoring dan evaluasi SIEM *open source* pada lingkungan institusi/perusahaan dapat tercapai. Pengukuran pada kebutuhan evaluasi persyaratan SIEM dilakukan dengan skala ordinal yaitu tidak baik(1), Kurang Baik (2), Cukup (3), Baik (4), Baik Sekali (5). Simulasi implementasi dari SIEM ini dilakukan dengan mengadopsi topologi dari sebuah jaringan Pusat Data dan Sistem Informasi (PDSI) sebuah institusi pendidikan.

2. METODE

Dalam mencapai tujuan penelitian yang diharapkan, dirancang langkah penelitian yang akan berkontribusi pada saat monitoring dan evaluasi penerapan SIEM berbasis *open source*. Langkah penelitian ini terdiri dari empat tahap yaitu: tahap perencanaan, tahap desain, tahap *deployment* dan terakhir tahap evaluasi sebagaimana ditunjukkan pada Gambar 1.



Gambar 1. Tahapan Penelitian

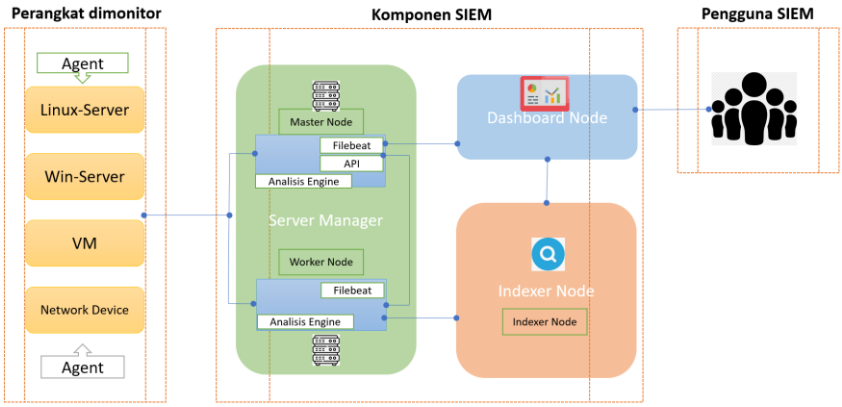
2.1 Perencanaan

Pada tahap pertama yaitu perencanaan, tahap ini dilakukan studi literatur penelitian – penelitian yang berkaitan dengan keamanan, log, SIEM. Studi pendahuluan ini dilakukan observasi pada obyek penelitian disebuah Pusat Data dan Sistem Informasi (PDSI). Observasi dilakukan untuk melihat penerapan keamanan informasi dan permasalahannya. Salah satu permasalahan belum adanya monitoring keamanan jaringan. Hasil dari studi literatur dan observasi menjadi bahan identifikasi permasalahan dalam penelitian. Dalam tahap ini juga dilakukan kajian terhadap aplikasi SIEM yang dirujuk di penelitian [5] seperti ELK *Stack*, OSSIM Alient vault dan Grayslog. Kemudian melakukan kajian dengan Wazuh *open source security platform* serta kemungkinan untuk diimplementasikan pada lingkungan jaringan PDSI.

2.2 Perancangan

Pada tahap ini juga dilakukan rancangan arsitektur SIEM yang kemudian diimplementasikan dengan perangkat lunak *open source*. Perancangan sistem SIEM berbasis *container* sehingga tersedia minimal tiga *container* untuk server manager, *indexer* dan *dashboard*. SIEM dijalankan dengan menggunakan *container docker*. Perancangan lingkungan simulasi implementasi SIEM digunakan lingkungan *virtual machine* (VM)

dengan menggunakan GNS3 untuk simulator jaringan, GNS3 VM sebagai tempat menjalankan *template* GNS3 dan VMWare *Player*.

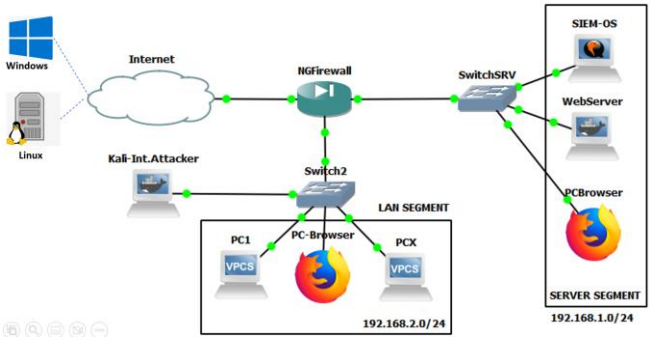


Gambar 2. Arsitekur SIEM

Arsitektur SIEM ditunjukkan pada Gambar 2. Arsitektur SIEM terdiri dari beberapa komponen yaitu termasuk agen yang dipasang pada setiap host yang akan diamati. Agen ini mengumpulkan data keamanan dan melaporkan aktivitas log terkait, sementara sumber monitoring log berasal dari firewall dan sistem operasi atau sistem lainnya. Server manager berfungsi sebagai pusat kontrol dan menerima data dari setiap agent untuk memantau dan mengevaluasi tingkat keamanan jaringan. Filebeat digunakan untuk mengirimkan data peringatan dan peristiwa ke *indexer node* menggunakan enkripsi TLS. *Indexer node* bertanggung jawab untuk mengindeks dan memproses data dari agen, dan terdiri dari Elasticsearch/Opensearch, Logstash, dan Kibana yang membantu dalam memantau dan mengelola log sistem dan aplikasi serta memberikan visibilitas dan analisis log secara *real-time.*

2.3 Deployment

Arsitektur *deployment* diperlihatkan pada Gambar 3. Topologi yang dibuat didalam GNS3. Terdapat segmen jaringan yaitu Segmen LAN, segmen Server dan Interent. Pada bagian *server* ditempatkan *Server* SIEM OS dengan sistem operasi Ubuntu, selanjutnya terdapat web *server* menggunakan docker dan PC browser linux Tiny-core yang merupakan host yang dimonitor. *Server* SIEM-OS ditempatkan berada dibelakang *Next Generation Firewall* (NGFirewall) yang didalamnya dikonfigurasi dengan Pfsense dan Suricata IPS. *Firewall* ini juga termasuk *device* yang dimonitor. Ada sistem operasi linux (*server*) dan windows yang dimonitor berada di luar dari lingkungan GNS3. Kedua sistem ini menggunakan perangkat sebenarnya bukan merupakan virtual berada dalam jaringan lokal PDSI.



Gambar 3. Arsitektur *Deployment* SIEM

2.4 Evaluasi

Tahap ini akan mengevaluasi implementasi SIEM dan menganalisis data yang ditampilkan pada SIEM dengan tujuan mengevaluasi kinerja sistem. Setelah semua tahap selesai, langkah terakhir adalah menarik kesimpulan dari penelitian ini.

3. HASIL DAN PEMBAHASAN

3.1. Hasil Analisis Kebutuhan Persyaratan SIEM

Pada komponen *Platform* terdapat 19 persyaratan. Kompnen *Operation* terdiri 3 persyaratan. Komponen *Integration* terdiri 5 persyaratan. Komponen *Advanced Features* terdiri 4 persyaratan. Komponen

Licensing and support terdiri 5 persyaratan. Setiap persyaratan akan dinilai dengan skala ordinal (1-5) dan dan kemudian diberi bobot *mandatory* (1) atau *nice to have* (0.5). Perhitunganya setiap nilai persyaratan dikali bobot kemudian dijumlahkan. Hasil dari analisis kebutuhan ini telah didiskusikan dengan pengguna SIEM dan membutuhkan persyaratan dan dapat dipenuhi dari fitur Wazuh SIEM ditunjukkan pada Tabel 1.

Tabel 1. Hasil Perhitungan Skor SIEM

Fitur SIEM	Jumlah Fitur	Skor
Platform	19	83,5
Operation	3	9,5
Integration	5	11
Advanced Features	4	8
Licensing and support	5	11,5

Total skor yang diperolah yaitu 123,5. Total nilai minimum dengan kriteria cukup akan memiliki nilai 88,5 (60%) dari total nilai terbaik 147,5 (100%). Oleh karena itu, pada penilaian ini sudah diatas nilai minimal dan mendapatkan nilai 84% dari kebutuhan maksimal fitur SIEM.

3.2. Instalasi, Konfigurasi dan Uji Coba Pada Lingkungan SIEM

Lingkungan yang digunakan dalam simulasi dan teknis ini terdiri perangkat keras yaitu IntelNuc Core I7-10710U, RAM 64GB, 1TB disk. Perangkat lunak yang digunakan sebagai *host* adalah sistem operasi Windows 11. Diatas *host* diinstall GNS3 dan VMware Player, didalam VMware terdapat *guest host* GNS3 VM. *Software* yang dijalankan diatas GNS3VM yaitu Pfsense bertindak sebagai *firewall* ditambah dengan suricata sebagai *network intrution detection system*. *Software* SIEM adalah Wazuh diinstallkan sebagai *container* diatas *server* linux Ubuntu. Simulasi penyerang dilakukan dengan menggunakan Kali Linux. Terdapat dua host (linux server dan windows) yang bertindak di luar dari jaringan simulasi virtual.

Tabel 2. Bash *Script* Menjalankan *Container* SIEM

Bash Script

```
#!/bin/bash
#set memory
sysctl -w vm.max_map_count=262144
#install curl
apt install curl -y
#install docker
curl -sSL https://get.docker.com/ | sh
#jalankan docker
systemctl start docker
#download docker-compose
curl -L "https://github.com/docker/compose/releases/download/v2.12.2/docker-compose-$(uname -s)-
$(uname -m)" -o /usr/local/bin/docker-compose
#tambahkan grant eksekusi
chmod +x /usr/local/bin/docker-compose
# test docker compose
docker-compose --version
#instalasi wazuh
git clone https://github.com/wazuh/wazuh-docker.git -b v4.3.10
cd single-node
#generate sertifikat
docker-compose -f generate-indexer-certs.yml run --rm generator
#jalankan docker-compose
docker-compose up -d
```

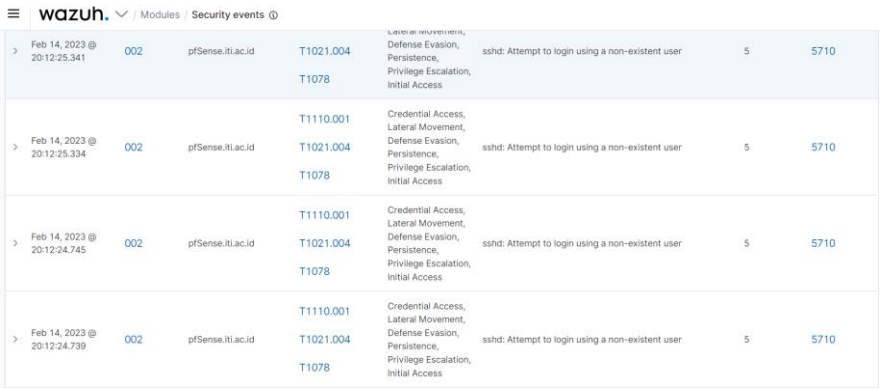
Tahap pertama adalah melakukan konfigurasi terhadap sistem sesuai dengan Gambar 3. Konfigurasi *firewall* pfsense dan menambahkan suricata karena menjadi bagian penting sebagai gerbang lalu lintas jaringan dari jaringan virtual ke jaringan nyata. Proses berikutnya dengan menambahkan *template* GNS3 untuk perangkat lainnya yaitu server ubuntu, *server* web. *Server* ubuntu dijadikan sebagai *host* dari SIEM sedangkan *server* web merupakan *host* yang dimonitor. Instalasi dan konfigurasi SIEM dijalankan pada Tabel 2 pada *fresh system*. Oleh karena itu, proses instalasi diawali dengan *setting* kebutuhan memori, dan menginstall *software* pendukung seperti curl, docker dan docker-compose.

Tahap kedua Integrasi dengan sistem yang berada didalam jaringan atau diluar jaringan dilakukan dengan menginstallkan agen pada setiap *host*. Agen diinstal pada beberapa jenis sistem operasi yaitu perangkat *firewall* (Pfsense/Freebsd), web *server* (Linux Ubuntu), Windows 11. Contoh perintah yang digunakan untuk install agen pada Linux berbasis Ubuntu/Debian ditunjukkan pada Tabel 3. Agen harus dijalankan untuk mulai melakukan mengkolekasi log yang ada pada setiap *host*, yang kemudian dikirim ke server pusat.

Tabel 3. Script Install dan Menjalankan Agent

Bash Script
#/bin/bash #unduh paket agen curl -so wazuh-agent-4.3.10.deb https://packages.wazuh.com/4.x/apt/pool/main/w/wazuh-agent/wazuh-agent_4.3.10-1_amd64.deb && sudo WAZUH_MANAGER='192.168.1.2' WAZUH_AGENT_GROUP='server-linux' dpkg -i ./wazuh-agent-4.3.10.deb #jalankan agen systemctl daemon-reload systemctl enable wazuh-agent systemctl start wazuh-agent

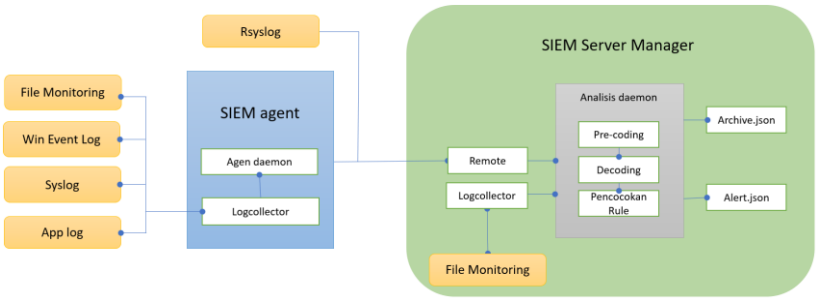
Pada skenario pengujian akan difokuskan serangan pada *firewall* (pfsense) dan server Linux, serangan akan dilancarkan dari *host* kali linux dimulai dari *scanning* port, *brute force attack* dan *unauthorized login*, *brute force attack failed attempt*, *privilege escalation*, tambah dan hapus *user*, sampai install program (*backdoor*). *Scanning* menggunakan *tool* nmap, dan *brute force attack* menggunakan hydra. Hasil serangan ini terekam dan ditampilkan dalam *dashboard* seperti ditunjukkan pada Gambar 4. Jenis aktifitas yang terekam di log *host* akan dievaluasi oleh *server manager* dan dicocokan dengan aturan yang diterapkan.



Gambar 4. Contoh *Alert* dari Serangan.

3.3. Analisis Bagaimana *Alert* Dihasilkan

Dalam memudahkan memahami bagaimana *alert* dihasilkan dibuat sebuah began ditunjukkan pada Gambar 5. Diambil sebuah contoh log yang berada pada syslog yaitu pada login ssh yang gagal sebagai berikut *:"Feb 14 06:38:15 pfSense sshd[15105]: error: PAM: Authentication error for admin from 192.168.200.67",*



Gambar 5. Arsitektur Agen dan *Alert* Dihasilkan

Agen SIEM akan mengambil log oleh log *collector*, kemudian di *triger* oleh agen daemon untuk dikirimkan kepada *server* pusat. Di *server* pusat log dianalisis dengan analisis daemon. Analisis dilakukan tiga tahap yaitu *pre-coding*, *decoding* dan pencocokan *rules*. Pada fase analisis *pre-decoding*, analisis log mengekstrak informasi seperti dilakukan Syslog yaitu *timestamp*, *hostname* dan nama program dari *header*. Contoh informasi hasil ekstraksi pre-*decoding* : *Timestamp : Feb 14 06:38:15, Hostname: pfSense, Program_name: sshd*.

Selanjutnya pada fase *decoding*, mesin analisis mencari *decoder* yang cocok dengan log. *Decoder* yang cocok kemudian mengektrak field yang didefinisikan dari log tersebut. Contoh Informasi hasil ekstraksi fase *decoder* yaitu *dstuser : admin, srcip: 192.168.200.67*. Fase selanjutnya adalah pencocokan dengan rule.

Informasi log yang diekstraksi dibandingkan dengan kumpulan aturan untuk mencari kecocokan. Dari contoh sebelumnya *rule* yang cocok ditunjukkan pada Tabel 4. Setelah aturan dicocokan, *server manager* akan membuat sebuah *alert*. *Alert* akan disimpan di /var/ossec/logs/alerts/alerts.(json|log) dan *event* di /var/ossec/logs/archives/archives.(json|log).

Tabel 4. Pencocokan *Rule*

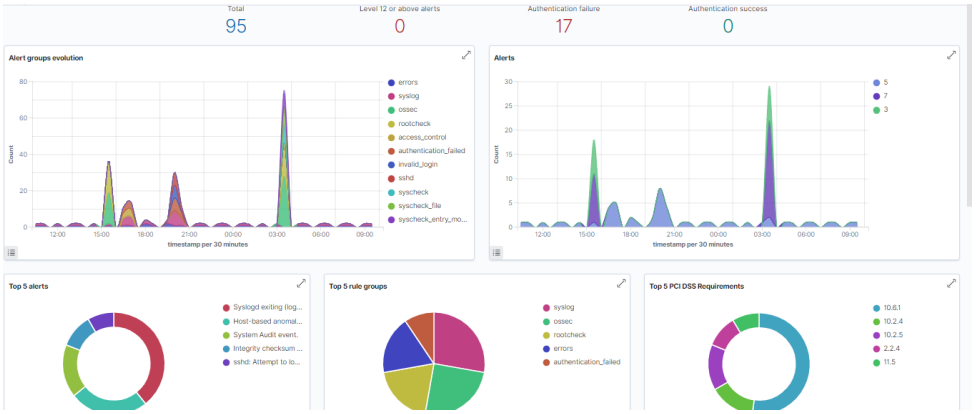
JSON Script

```
<rule id="5760" level="5">
  <if_sid>5700,5716</if_sid>
  <match>Failed password|Failed keyboard|authentication error</match>
  <description>sshd: authentication failed.</description>
  <mitre>
    <id>T1110.001</id>
    <id>T1021.004</id>
  </mitre>
<group>authentication_failed,gdpr_IV_35.7.d,gdpr_IV_32.2,gpg13_7.1,
hipaa_164.312.b,nist_800_53_AU.14,nist_800_53_AC.7,pci_dss_10.2.4,
pci_dss_10.2.5,tsc_CC6.1,tsc_CC6.8,tsc_CC7.2,tsc_CC7.3,</group>
</rule>
```

3.4. Analisis Hasil *Dashboard* SIEM

Halaman ini menyediakan *dashboard* yang berguna untuk memastikan kepatuhan terhadap beberapa peraturan seperti PCI DSS, GDPR, HIPAA, dan NIST 800-53. *Dashboard* ini juga menyediakan antarmuka yang memudahkan navigasi melalui kerangka MITRE ATT&CK. Agen akan membaca log dari sistem operasi dan aplikasi, dan meneruskan informasi tersebut ke *server manager* untuk dianalisis lebih lanjut. Analisis ini berbasis penyimpanan (*storage*) dan *rule*, dimana *rule* akan membantu mengidentifikasi kesalahan aplikasi atau sistem, kesalahan konfigurasi, percobaan atau aktivitas jahat yang berhasil, pelanggaran kebijakan, serta masalah keamanan dan operasional lainnya. Contoh *dashboard* ditunjukkaan pada Gambar 6.

SIEM juga memainkan peran penting dengan memantau file sistem dan mengidentifikasi perubahan konten, akses pengguna, dan atribut file yang perlu diperhatikan. SIEM juga secara *real-time* mengidentifikasi pengguna dan aplikasi yang digunakan untuk membuat atau memodifikasi file. Fungsi ini terdapat pada modul *integrity monitoring*.



Gambar 6. *Dashboard*

Pada *dashboard* diketahui tentang inventaris asset. Agen akan menarik data inventaris perangkat lunak dan mengirimkannya ke *server*. Informasi tersebut akan dikorelasikan dengan basis data CVE (*Common Vulnerabilities and Exposure*) yang terus diperbarui. Dengan begitu, dapat diidentifikasi tingkat kerentanan perangkat lunak yang digunakan. Penilaian kerentanan akan dilakukan secara otomatis untuk menemukan titik lemah pada aset yang dianggap penting dan mengambil tindakan korektif sebelum penyerang mengeksploitasinya. Agen juga memantau pengaturan konfigurasi sistem dan aplikasi untuk memastikan semuanya sesuai dengan kebijakan keamanan, standar, dan panduan hardening. Selain itu, agen juga melakukan pemindaian berkala untuk mendeteksi aplikasi yang rentan, belum di-patch, atau dikonfigurasi secara tidak aman. Konfigurasi ini dapat disesuaikan agar selaras dengan kebijakan yang berlaku. Peringatan dan rekomendasi juga akan diberikan untuk konfigurasi, referensi, dan pemetaan yang lebih baik dengan kepatuhan terhadap peraturan.

Dari hasil analisis fitur *dashboard* yang tersedia memudahkan petugas keamanan dalam memantau berbagai perangkat yang dimonitor, log disatukan dalam sebuah pusat penyimpanan, dilakukan audit dan

kendali atas berbagai macam informasi log. *Dashboard* dapat dikorelasikan dengan *opensearch* dibuat *custom dashboard*. Dengan penerapan SIEM dapat diketahui kondisi keamanan terkini dari host yang dimonitor. Eskalasi tindakan dapat dilakukan, *dashboard* mempermudah dalam melakukan pekerjaan – pekerjaan perbaikan keamanan.

4. PENUTUP

Implementasi monitoring dan evaluasi keamanan jaringan dengan SIEM *open source* dapat diimplementasikan dengan menggunakan Wazuh dan melindungi *host* SIEM dengan Pfsense dan Suricata. SIEM mampu berfungsi sesuai dengan kebutuhan yang didefinisikan awal. Hasil evaluasi menunjukkan bahwa 84% kebutuhan terpenuhi. Fitur-fitur penting pada SIEM mencakup deteksi dan *alert*, analisis dan investigasi, kepatuhan dan audit, integrasi dan interoperabilitas, monitoring dan pelaporan, serta dukungan dan perawatan tersedia dan berjalan. Perusahaan / Organisasi lain dapat mengimplementasi SIEM *open source* dalam rangka melakukan pembenahan tata kelola keamanan informasi. Selain itu, dibutuhkan dukungan pimpinan tertinggi dari suatu organisasi dalam melakukan implementasi, monitoring dan evaluasi SIEM. Penelitian ini dapat dikembangkan dengan standar keamanan informasi yang berlaku di Indonesia seperti indeks KAMI dan menyatukan menjadi sebuah sistem terpadu.

DAFTAR PUSTAKA

[1] I. Anastasov e D. Davcev, “SIEM Implementation for Global and Distributed Environment,” em *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, Hammamet, 2014.

[2] J. W. W. Qingrong, X. Z. S. Zhu, K. K. E. Guo e C. L. M. Lu, “Light SIEM for Semiconductor Industry,” em *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM),*, Singapore, 2017.

[3] K. Dekten, T. Rix, C. Keiner, B. Hellmann e L. Renners, “SIEM Approach for a Higher Level of IT Security in Enterprise Networks,” em *2015 The 8th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Warsaw, 2015.

[4] P. Shoard, A. Davies e M. Schneider, “Magic Quadrant for Security Information and Event Management,” Gartner, 10 October 2022. [Online]. Available: https://www.gartner.com/doc/reprints?id=1-2BDGWSVV&ct=221011&st=sb. [Acesso em 2 December 2022].

[5] A. Vazão, L. Santos, M. B. Piedade e C. Rabadão, “SIEM Open Source Solutions: a Comparative Study,” em *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, 2019.

[6] S. S. Sekharan e . K. Kandasamy, “Profiling SIEM Tools and Correlation Engines for Security Analytics,” em *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Chennai, 2017.

[7] R. Leszczyna e M. R. Wróbel, “Evaluation of Open Source SIEM for Situation Awareness Platform in the Smart Grid Environment,” em *2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, Singapore, 2017.

[8] Wazuh, “Wazuh Quickstart,” Wazuh Inc., 16 November 2022. [Online]. Available: https://documentation.wazuh.com/current/quickstart.html. [Acesso em 2 December 2022].

[9] A. Serckumecka, I. Medeiros e A. Bessani, “Low-cost Serverless SIEM in the Cloud,” em *2019 38th Symposium on Reliable Distributed Systems (SRDS)*, Lyon, 2019.

[10] H. Mokalled, R. Catelli, . V. Casola, D. Debertol, E. Meda e R. Zunino, “The applicability of a SIEM solution: Requirements and Evaluation,” em *2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, Napoly, 2019.

[11] K.-O. Dekten, M. Jahnke, C. Kleiner e M. Rohde, “Combining Network Access Control (NAC) and SIEM Functionality based on Open Source,” em *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Bucharest, 2017.