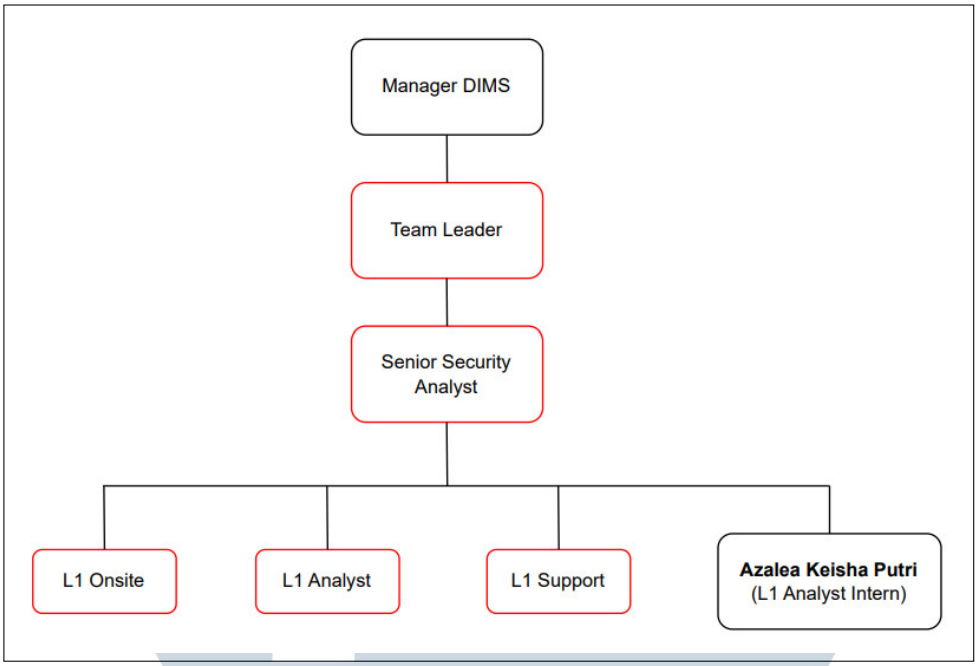
BAB 3
PELAKSANAAN KERJA MAGANG

3.1 Kedudukan dan Koordinasi

Dalam susunan organisasi PT. Defender Nusa Semesta, posisi yang ditempati adalah L1 Security Analyst Intern, yang berada pada level awal dalam unit operasional keamanan informasi. Sebagai bagian dari tim Security Analyst, jabatan ini berperan penting sebagai lini terdepan dalam proses deteksi, pemantauan, serta analisis terhadap potensi insiden keamanan siber yang terjadi pada sistem. Tugas utama dari posisi ini meliputi pelaksanaan analisis tingkat pertama (first-level analysis), melakukan eskalasi insiden sesuai dengan prosedur yang berlaku, serta menyusun laporan awal sebagai dasar bagi tim berikutnya dalam menindaklanjuti ancaman yang telah teridentifikasi.

Secara struktural, posisi ini termasuk dalam tim Level 1 (L1) bersama dengan peran operasional lainnya seperti L1 Analyst, L1 Onsite, dan L1 Support. Namun, sebagai L1 Security Analyst Intern, lingkup pekerjaan difokuskan secara khusus pada aspek keamanan informasi, yang membedakan kedudukannya dari peran L1 lainnya. Koordinasi serta pengarahan operasional diberikan langsung oleh Senior Security Analyst, yang berperan sebagai koordinator teknis sekaligus pengarah utama dalam aktivitas harian tim Security Analyst. Seluruh proses eskalasi, permintaan klarifikasi teknis, maupun penyusunan laporan awal dilaksanakan terlebih dahulu melalui Senior Security Analyst, sebelum diteruskan ke Team Leader apabila diperlukan.

Meskipun Team Leader memiliki peran sentral dalam pengambilan keputusan dan pengawasan tim secara keseluruhan, dalam kegiatan operasional sehari-hari posisi L1 Security Analyst Intern lebih sering berinteraksi serta berkoordinasi dengan Senior Security Analyst. Di atas struktur tersebut, terdapat Manager DIMS yang memegang tanggung jawab penuh atas manajemen strategis keamanan informasi digital perusahaan. Susunan posisi dan alur koordinasi ini divisualisasikan pada Gambar 3.1 untuk memperjelas kedudukan dalam tim.



Gambar 3.1. Struktur kedudukan L1 Security Analyst dalam tim SOC Defenxor

Selama periode magang, koordinasi antar tim maupun lintas divisi di PT. Defender Nusa Semesta dilakukan melalui beberapa platform utama. WhatsApp Group dimanfaatkan sebagai media komunikasi harian, baik untuk koordinasi internal dengan anggota tim maupun interaksi dengan divisi lain dalam konteks operasional. Selain itu, aplikasi Thunderbird/Outlook digunakan secara khusus untuk mengirim notifikasi hasil analisis, melakukan eskalasi insiden kepada customer, serta menjalin komunikasi dengan tim onsite maupun pihak internal SOC. Sementara itu, email resmi Defenxor tetap digunakan dalam lingkup yang lebih luas, seperti koordinasi dengan divisi lain, berkomunikasi dengan pihak CTI, serta penjadwalan dan konfirmasi pertemuan (meeting) yang memerlukan dokumentasi lebih formal. Dengan adanya pemanfaatan berbagai saluran komunikasi tersebut, peran Security Analyst dapat dijalankan secara terarah, kolaboratif, dan selaras dengan kebutuhan operasional di bidang keamanan informasi.

3.2 Tugas yang Dilakukan

Selama masa magang di PT. Defender Nusa Semesta, penempatan dilakukan di divisi *Security Operations Center* (*SOC*) dengan fokus utama pada aktivitas *monitoring* keamanan siber dan analisis insiden. Program magang sebenarnya berlangsung selama 52 minggu, terhitung sejak 03 Februari 2025 hingga 02 Februari 2026, yang terbagi menjadi dua bagian, yaitu 6 bulan pertama dan

6 bulan kedua. Program magang yang dijalani saat ini telah memasuki fase kedua dan telah berjalan selama 15 minggu, terhitung sejak 01 Juli hingga 10 Oktober 2025. Seluruh kegiatan dilaksanakan secara sistematis dan dibimbing oleh mentor, dengan pendekatan kerja yang selaras dengan prinsip-prinsip pengamanan informasi berdasarkan standar *ISO/IEC 27001*. Berikut merupakan poin-poin utama aktivitas dan pembelajaran yang dilakukan selama fase kedua program magang:

1. Monitoring ketersediaan log (log availability).

   Melakukan pemantauan berkelanjutan terhadap agen pengirim log antara lain FortiGate traffic, Elastic Windows Agent, dan Elastic System Agent untuk memastikan setiap sumber log tetap aktif dan mengalir ke platform pusat. Aktivitas ini mencakup pengecekan status *healthy/unhealthy*, mendeteksi lebih dini potensi gangguan (misalnya agen berhenti mengirim, antrean log menumpuk, atau jeda waktu pengiriman melebar), serta menindaklanjuti anomali dengan pembuatan notifikasi otomatis. Hasil pemantauan dicatat sebagai *uptime* dan *data completeness* agar tim mengetahui kelengkapan log yang tersedia untuk analisis insiden.

2. Monitoring menggunakan SIEM Elastic Stack.

   Memantau *event* yang terpicu oleh *detection rules* dan eskalasi menjadi *alert* pada Elastic SIEM. Fokus pekerjaan meliputi korelasi lintas indeks/sumber log, penelusuran pola aktivitas mencurigakan, serta triase insiden secara *real time* berdasarkan tingkat keparahan. Dalam prosesnya, saya menilai konteks (sumber, akun, host, dan *timeline*), melakukan *enrichment* sederhana (misalnya *geo-IP* atau *process ancestry* bila tersedia), dan mengklasifikasikan insiden untuk menentukan prioritas penanganan.

3. Notifikasi dan rekomendasi remediasi ke pelanggan.

   Setelah analisis korelasi pada beberapa indeks perangkat di Elastic SIEM, menyusun ringkasan insiden yang jelas mencakup *what/when/where/how*, tingkat keparahan, dampak potensial, serta bukti pendukung lalu mengirimkan notifikasi insiden kepada pelanggan. Bersamaan dengan itu, saya menyertakan rekomendasi langkah remediasi yang dapat segera dijalankan (misalnya isolasi host, pemblokiran *indicator of compromise*, penyesuaian kebijakan firewall/EDR, dan penjadwalan *patching*), lengkap dengan urutan prioritas dan verifikasi pasca-tindakan.

4. Monitoring Infrastruktur dan Endpoint Security.

   Melakukan pemantauan keamanan dan kesehatan sistem secara terintegrasi dari sisi infrastruktur dan endpoint, dengan memanfaatkan SolarWinds untuk memonitor performa infrastruktur seperti CPU utilization, penggunaan memori, kapasitas dan I/O disk, serta network throughput guna mengidentifikasi anomali dan potensi bottleneck, serta Grafana untuk memastikan log availability dari seluruh sumber log tetap konsisten dan tidak mengalami keterlambatan pengiriman ke SIEM. Dari sisi endpoint, pemantauan dilakukan menggunakan Check Point Harmony untuk memantau status agen, kepatuhan endpoint, deteksi malware, serta verifikasi proses remediasi seperti karantina file, pemblokiran proses berbahaya, dan isolasi jaringan.

5. Pelaksanaan Incident Response.

   Pelaksanaan incident response dilakukan melalui investigasi lanjutan terhadap insiden keamanan dengan menyusun attacker workflow untuk memahami kronologi dan teknik serangan, melakukan tindakan mitigasi seperti pemblokiran IP berbahaya, serta menyusun incident response report sebagai dokumentasi resmi dan bahan evaluasi. Proses penanganan insiden ini mengacu pada kerangka kerja NIST SP 800-61, yang mencakup tahapan preparation, detection and analysis, containment, eradication and recovery, serta post-incident activity guna memastikan respons insiden berjalan sistematis dan terukur, serta didukung oleh penerapan metode OODA (Observe, Orient, Decide, Act) sebagai kerangka kerja mikro untuk mempercepat pengambilan keputusan operasional dan memastikan tindakan respons yang adaptif, efektif, dan berkelanjutan.

Secara umum, program magang ini memberikan peluang untuk terlibat langsung dalam pemantauan dan penanganan keamanan siber di lingkungan perusahaan. Proses pembelajaran ditempuh melalui pendekatan berbasis insiden dengan dukungan beragam perangkat keamanan, serta berlandaskan kerangka kerja ISO/IEC 27001. Pengalaman tersebut menjadi modal penting untuk memperdalam pemahaman tentang keamanan informasi sekaligus meningkatkan kemampuan dalam mendeteksi, menganalisis, dan mendokumentasikan insiden secara sistematis. Rangkuman aktivitas magang yang telah dilaksanakan disajikan pada Tabel 3.1.

Tabel 3.1. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
1	Operasional night shift, mempelajari tools, dashboard, cara membuat daily report, dan mengenal environment customer.
2	Operasional mid shift, migrasi notifikasi email ke Thunderbird/Outlook, setting Thunderbird, dan deteksi aktivitas login VPN mencurigakan dari luar Indonesia.
3	Operasional early shift, mempelajari cara menyusun daily report pagi dan melakukan monitoring observability.
4	Operasional night shift, deteksi aktivitas mencurigakan pada perangkat Checkpoint, dengan aktivitas scanning SMB di jaringan internal customer.
5	Operasional mid shift, deteksi credential leak melalui SOCRadar, dengan banyak kredensial milik customer terdeteksi di hacker forum.
Lanjut pada halaman berikutnya	

Tabel 3.2. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
6	Operasional early shift, deteksi impersonating domain yang menyerupai domain internal customer, dilanjutkan dengan remediasi dan takedown.
7	Operasional night shift, deteksi banyak agent mati/unhealthy, eskalasi ke internal dan eksternal untuk penanganan lebih lanjut.
8	Operasional mid shift, deteksi aktivitas audit log Windows oleh user Domain Admin dengan perubahan user pada sistem.
9	Operasional early shift, deteksi indikasi email phishing melalui perangkat FortiMail, dikategorikan sebagai virus/malware.
10	Operasional night shift, deteksi multiple failed login attempts pada host internal dalam waktu dekat, menunjukkan potensi serangan brute force.
Lanjut pada halaman berikutnya	

Tabel 3.3. Pekerjaan yang dilakukan tiap minggu selama pelaksanaan kerja magang

Minggu Ke -	Pekerjaan yang dilakukan
11	Operasional mid shift, deteksi insecure policy set via Set-ExecutionPolicy dengan script PowerShell yang mengubah policy eksekusi dan bypass.
12	Operasional early shift, deteksi file tidak terenkripsi di shared drive, menunjukkan potensi keberadaan password atau informasi sensitif.
13	Operasional night shift, troubleshooting dengan SSA untuk mengatasi masalah Thunderbird yang tidak bisa mengirim notifikasi.
14	Operasional night shift, meeting dengan SSA untuk centralized filter rules terkait email yang perlu diarsipkan dan dibackup pada laptop SOC.
15	Operasional mid shift, pembuatan laporan security advisory berdasarkan CVE terbaru yang ditemukan.

3.3 Uraian Pelaksanaan Magang

Kegiatan magang mencakup berbagai aktivitas operasional di bidang *Security Operations Center* (SOC), meliputi penerapan dan integrasi *tools* keamanan, pemantauan ketersediaan log, serta penanganan kendala pada agen monitoring. Selain itu, dilakukan pula monitoring performa sistem, analisis melalui platform *Security Information and Event Management* (SIEM), penanganan insiden, serta proses dokumentasi dan notifikasi untuk mendukung efektivitas operasional SOC. Untuk lebih detailnya, uraian pelaksanaan magang dijelaskan sebagai berikut.

3.3.1 Penerapan Tools dan Integrasi Sistem

Pada tahap penerapan *tools* dan integrasi sistem, kegiatan difokuskan pada pengelolaan serta penyelarasan berbagai komponen pendukung keamanan untuk memastikan sistem berjalan optimal dan terintegrasi dengan baik. Aktivitas ini mencakup instalasi *Endpoint Detection and Response* (EDR) CrowdStrike, integrasi koneksi *Virtual Private Network* (VPN) dengan MobaXterm, penerapan sertifikat keamanan pada browser untuk akses sistem internal, serta penggunaan Bitwarden

sebagai *password manager*. Untuk lebih detailnya, penerapan *tools* dan integrasi sistem dijelaskan sebagai berikut.

A. Instalasi EDR CrowdStrike

Sebelum memasuki fase operasional di divisi *Security Operations Center* (SOC), dilakukan pemasangan *endpoint security* CrowdStrike Falcon pada perangkat kerja, termasuk laptop pribadi yang digunakan untuk mengakses sistem internal perusahaan. CrowdStrike Falcon merupakan solusi *Endpoint Detection and Response* (EDR) sekaligus *Next-Generation Antivirus* (NGAV) yang berfungsi memberikan perlindungan berbasis perilaku terhadap ancaman seperti *malware*, *ransomware*, dan *phishing-driven payload*. Instalasi sensor CrowdStrike dilakukan berdasarkan panduan dari tim SOC, di mana supervisor menyediakan berkas instalasi sensor berformat *WindowsSensor.exe* beserta *Customer ID Checksum* (CID) unik yang berfungsi untuk mengaitkan *endpoint* dengan konsol CrowdStrike Falcon milik perusahaan. Proses instalasi dilakukan dengan menjalankan *Command Prompt* sebagai *Administrator* dan mengetik perintah *WindowsSensor.exe /install /quiet CID= ¡Customer_ID_Checksum¿*, di mana kode CID tersebut merupakan identitas khusus milik perusahaan. Setelah instalasi selesai, sistem secara otomatis menambahkan layanan bernama *csfalconservice* ke dalam *Windows Service Manager*. Untuk memastikan sensor telah aktif dan berjalan dengan benar, digunakan perintah *sc query csfalconservice*. Jika hasilnya menampilkan status *STATE : 4 RUNNING* sebagaimana terlihat pada Gambar 3.2, dimana sensor CrowdStrike telah berjalan secara normal di latar belakang.

```
C:\Users\Lenovo>sc query csfalconservice

SERVICE_NAME: csfalconservice
        TYPE               : 10  WIN32_OWN_PROCESS
        STATE              : 4  RUNNING
                                (STOPPABLE, NOT_PAUSABLE, ACCEPTS_PRESHUTDOWN)
        WIN32_EXIT_CODE    : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        CHECKPOINT         : 0x0
        WAIT_HINT          : 0x0
```

Gambar 3.2. Visualisasi Crowdstrike

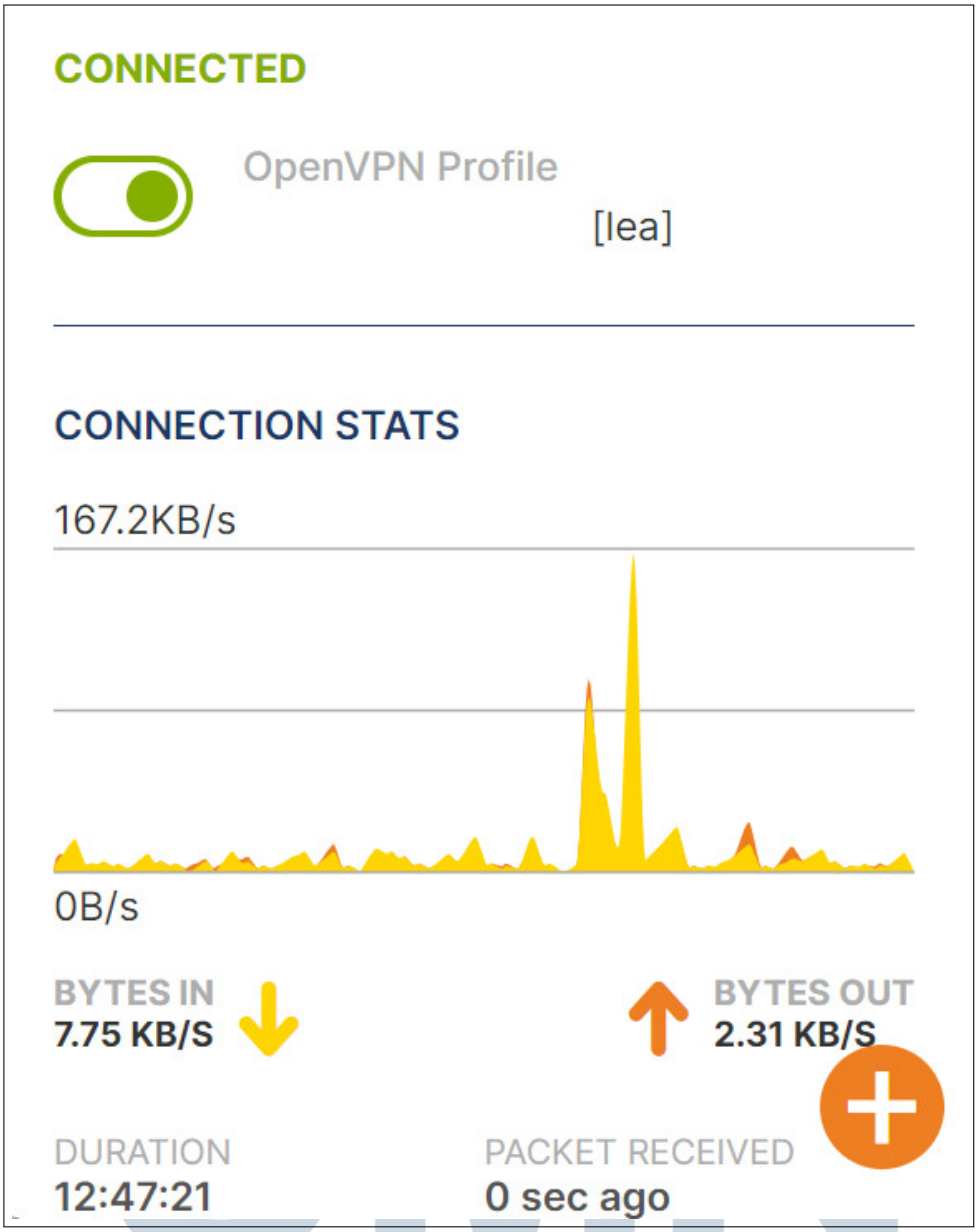
Sensor CrowdStrike mengirimkan data aktivitas endpoint secara *real-time* ke *console* pusat untuk dianalisis, sehingga ancaman dapat dideteksi melalui kombinasi *Indicators of Attack* (IOA) dan *Indicators of Compromise* (IOC). Sistem ini juga dapat melakukan respons otomatis seperti karantina *file*, isolasi jaringan,

pembatasan perangkat eksternal, serta penerapan kebijakan *firewall* dan *device control*. CrowdStrike dipilih karena bersifat *cloud-native*, memiliki tingkat deteksi tinggi, mendukung analisis *machine learning*, dan dapat dikelola secara terpusat tanpa membebani performa perangkat.

B. Integrasi VPN dan Mobaxterm

Dalam operasional SOC, VPN digunakan sebagai jakur aman agar host *security analyst* dapat terhubung ke jaringan internal kantor tanpa terekspos langsung ke internet. Dengan OpenVPN sebagai solusi *open-source*, seluruh trafik dienkripsi dan diautentikasi (mis. sertifikat + kredensial/MFA), sehingga akses ke aset seperti SIEM, server log, dan perangkat jaringan berjalan *private* dan terkontrol. Kebijakan seperti *least-privilege, split-tunneling* (bila diperlukan), serta pencatatan akses membantu meminimalkan risiko kebocoran data dan memastikan aktivitas dapat diaudit.

Setelah jalur aman terbentuk lewat VPN, MobaXterm dipakai sebagai alat kerja harian untuk mengelola dan mengakses sistem internal. MobaXterm menyediakan terminal terpadu dengan SSH/SFTP, manajemen sesi, *port forwarding (tunnel)*, hingga X11 *forwarding*, sehingga analis bisa *login* ke server, menyalin log, menjalankan perintah pemantauan, serta melakukan otomasi rutin dari satu *interface*. Berikut merupakan contoh ketika vpn berhasil terkoneksi pada gambar 3.3.

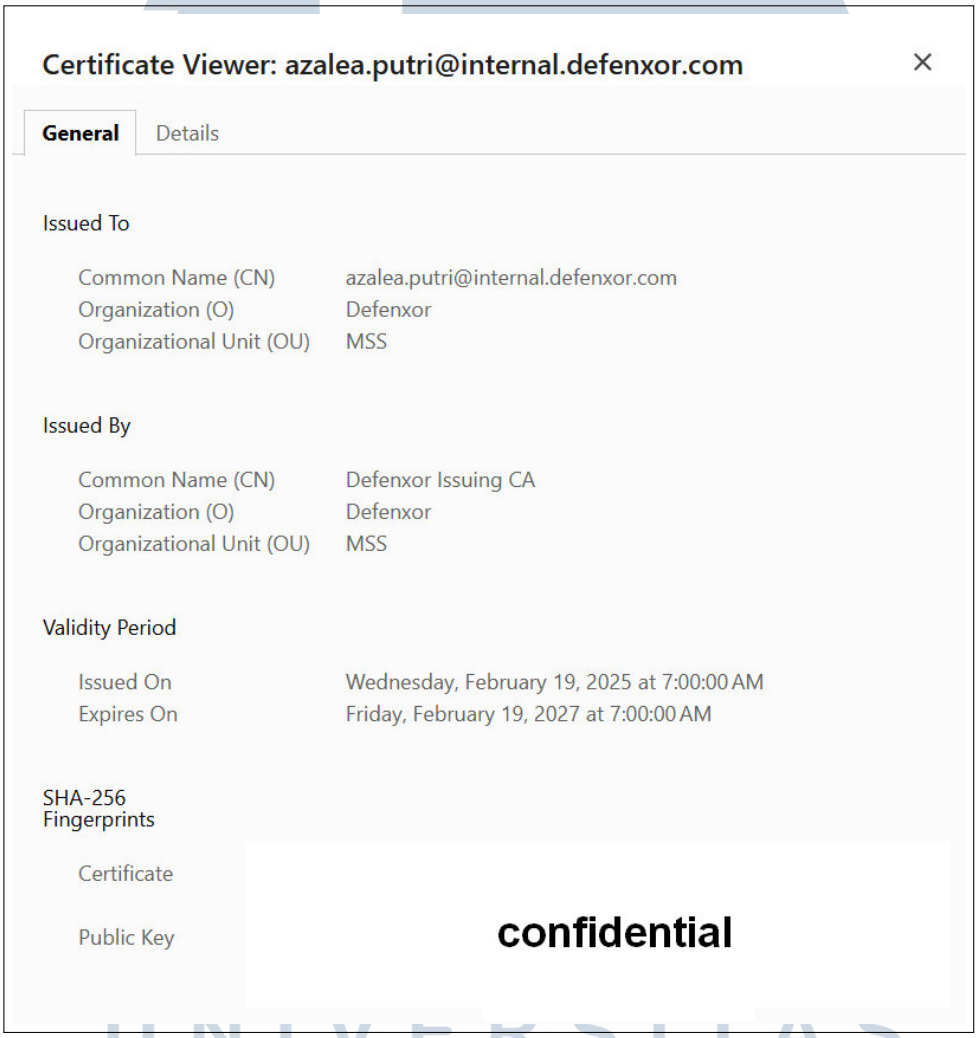


Gambar 3.3. Visualisasi OpenVPN

C. Penerapan Sertifikat Keamanan pada Browser untuk Akses Sistem Internal

Untuk dapat mengakses sistem internal perusahaan, browser terlebih dahulu harus dilengkapi dengan sertifikat keamanan sebagai bentuk autentikasi dan enkripsi koneksi. Proses ini dilakukan melalui menu *Settings* → *Security* → *Manage Certificates* dengan cara mengimpor beberapa berkas sertifikat yang disediakan oleh *supervisor*, yaitu azalea.putri@internal.defenxor.com.crt, ca-chain.crt, defenxor-ca.crt, dan root-ca.crt. Pada tahap ini, pengguna dapat diminta

memasukkan *password* untuk verifikasi tambahan sesuai kebijakan keamanan perusahaan. Setelah seluruh sertifikat berhasil diimpor dan terdaftar pada browser, sistem akan mengenali perangkat sebagai klien yang sah sehingga akses ke portal internal perusahaan dapat dilakukan secara aman melalui koneksi terenkripsi dan terverifikasi. Berikut merupakan visualisasi ketika sertifikat telah terpasang pada browser pada gambar 3.4.

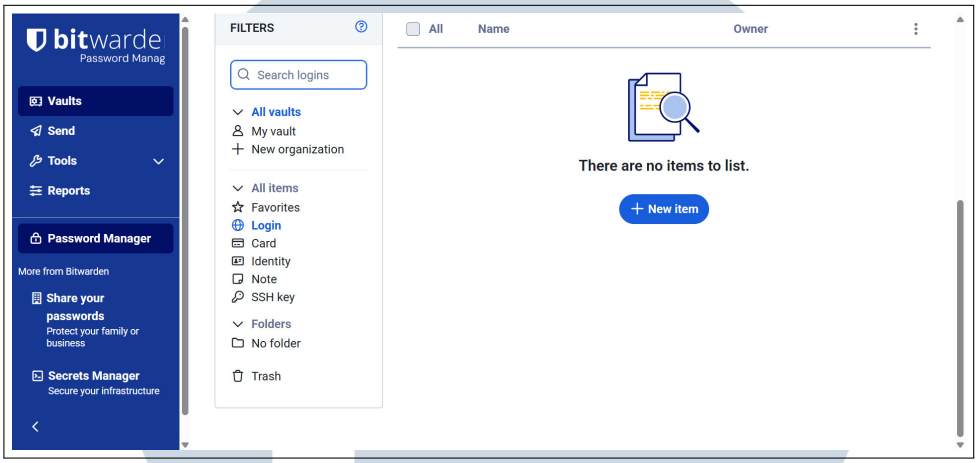


Gambar 3.4. Visualisasi Certificate Browser

D. Penggunaan Bitwarden untuk Password Manager

Untuk pengelolaan kredensial, digunakan Bitwarden sebagai pengelola kata sandi yang lebih aman dibanding fitur penyimpanan otomatis di browser. Prosesnya diawali dengan menambahkan ekstensi Bitwarden di Chrome, kemudian membuat akun pribadi untuk menyimpan dan mengelola seluruh kata sandi maupun

kredensial login. Setiap kali melakukan autentikasi atau mengakses sistem internal, Bitwarden secara otomatis mengisi data login yang tersimpan dengan aman. Berikut merupakan visualisasi Bitwarden pada gambar 3.5.

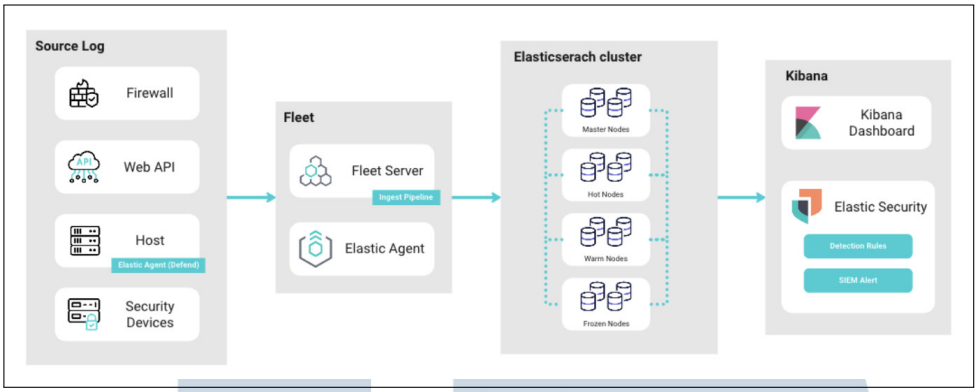


Gambar 3.5. Visualisasi Password Manager Bitwarden

Penggunaan Bitwarden dipilih karena bersifat *open-source*, mendukung enkripsi ujung-ke-ujung (*E2EE*) dengan arsitektur *zero-knowledge*, serta memiliki fitur *multi-factor authentication* (MFA), *shared vault*, dan pemantauan pelanggaran (*data breach monitoring*). Dengan penerapan ini, risiko *credential leak* dapat diminimalkan dan kontrol keamanan terhadap akses sistem menjadi lebih terpusat serta terjamin.

3.3.2 Infrastructure Elastic

Sebelum membahas lebih jauh mengenai monitoring agent dan mekanisme alert pada *Elastic*, terlebih dahulu ditampilkan *infrastruktur Elastic* untuk memahami gambaran besar cara kerja SIEM Elastic secara menyeluruh. Tujuan dari infrastruktur ini adalah memastikan seluruh log dari berbagai perangkat keamanan dapat dikumpulkan, dirapikan, disimpan secara efisien, lalu dianalisis secara konsisten untuk keperluan monitoring maupun investigasi insiden. Alur tersebut divisualisasikan pada Gambar 3.6 sebagai berikut.



Gambar 3.6. Visualisasi Infrastructure Pada Elastic

Pada tahap awal, seluruh log berasal dari *source log* seperti firewall, web API, host/server, atau perangkat keamanan lainnya. Masing-masing perangkat ini menghasilkan log dengan format, struktur field, dan level detail yang berbeda-beda misalnya firewall menggunakan istilah *src*, sementara web API memakai *client_ip*, dan endpoint security menggunakan *source.address*. Perbedaan inilah yang membuat log mentah tidak bisa dianalisis langsung. Karena itu, log dikumpulkan oleh *Elastic Agent*, lalu dikirim ke *Fleet Server* untuk melalui proses normalisasi. Di dalam Fleet, log diproses oleh *ingest pipeline* yang mengekstraksi, mem-parsing, dan menyamakan nama field menggunakan *ECS (Elastic Common Schema)*, sehingga field seperti *source.ip*, *destination.ip*, *event.action*, *user.name*, dan lainnya menjadi seragam meskipun berasal dari vendor yang berbeda. Normalisasi ini penting agar SIEM dapat menerapkan *detection rules* secara konsisten. Selain itu, Fleet juga menerapkan *policy*, yaitu konfigurasi terpusat yang mengatur integrasi apa saja yang aktif di setiap agent, seperti modul firewall, system logs, sysmon, API monitoring, atau agent defense. Dengan *policy* ini, administrator dapat mengontrol data apa yang dikumpulkan, berapa frekuensinya, dan bagaimana pipeline pemrosesan diterapkan secara terstandarisasi ke seluruh agent di lingkungan tersebut.

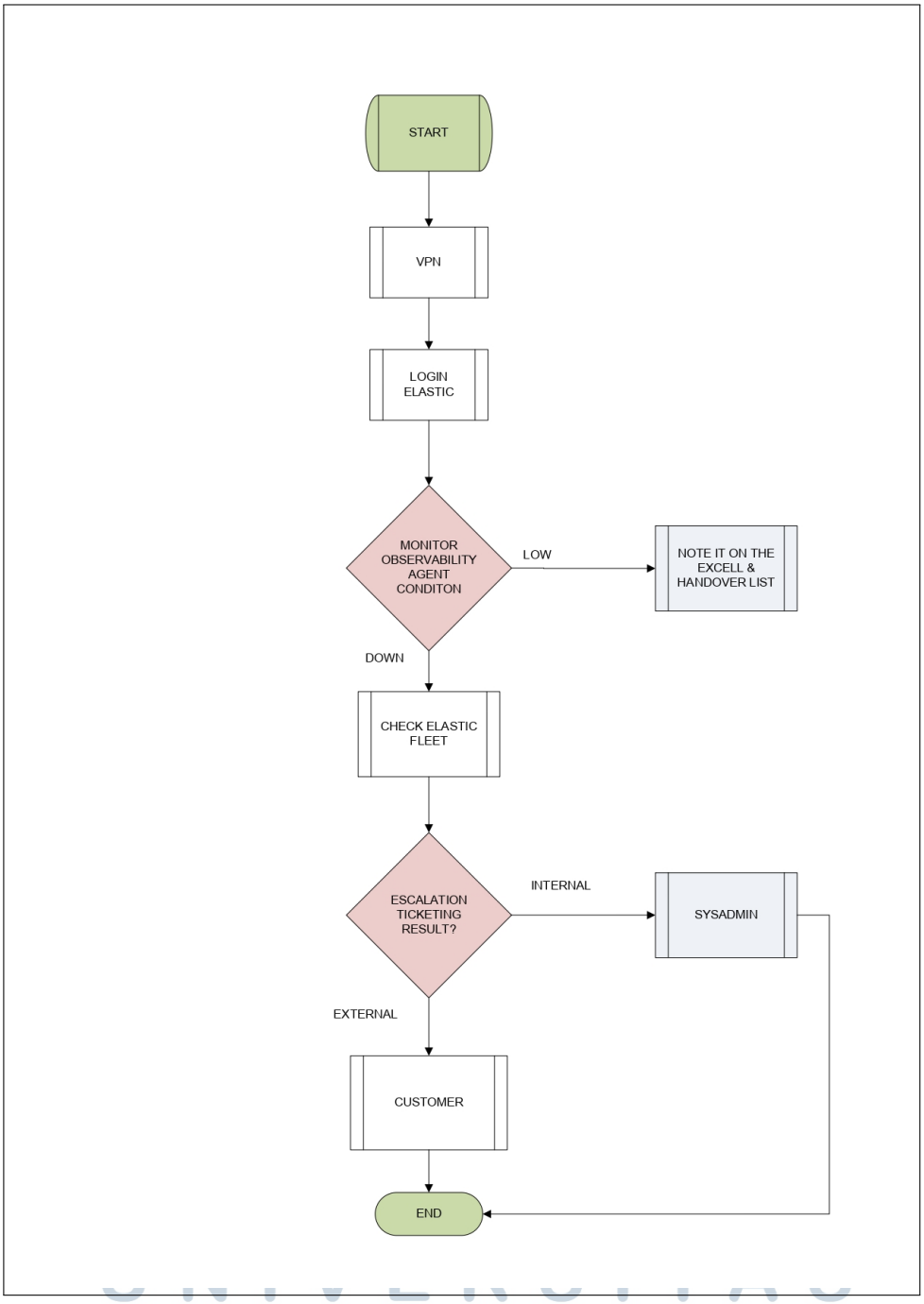
Setelah proses normalisasi selesai, data dikirimkan ke *Elasticsearch Cluster* sebagai tempat penyimpanan dan analisis. Cluster menggunakan arsitektur *tiered storage* yang membagi data berdasarkan umur dan kebutuhan performa. *Hot nodes* menyimpan data terbaru yang paling sering dicari atau dianalisis, biasanya log dalam rentang 7–14 hari. *Warm nodes* menyimpan data yang lebih lama namun masih dibutuhkan untuk investigasi, misalnya data antara 1–3 bulan. Sementara itu, *frozen nodes* (atau *cold tier* di versi tertentu) digunakan untuk menyimpan data jangka panjang misalnya antara 3 bulan hingga 1 tahun—dengan biaya

penyimpanan yang jauh lebih murah. Data di *frozen* tetap dapat dicari, tetapi performanya lebih lambat karena mengandalkan kompresi tinggi atau *searchable snapshot*. Dengan arsitektur ini, organisasi dapat menyimpan log dalam jumlah besar tanpa membebani performa dan biaya operasional cluster.

Tahap akhir adalah Kibana, tempat seluruh data yang sudah terindeks ditampilkan untuk analisis. Pada dashboard Kibana, analis dapat melihat tren log, grafik serangan, dan visualisasi lainnya. Bagian *Elastic Security* kemudian memanfaatkan log yang sudah dinormalisasi untuk menjalankan *detection rules*, membuat alert otomatis, dan menyediakan *timeline* investigasi yang terstruktur. Karena seluruh log sudah melalui proses normalisasi ECS dan disimpan dalam arsitektur cluster yang efisien, Kibana dapat menampilkan informasi dengan akurat dan cepat, sehingga analis dapat memahami konteks serangan, menelusuri sumber ancaman, dan mengambil tindakan secara efektif.

3.3.3 Log Availability Monitoring

Secara garis besar pada proses monitoring *log availability*, langkah-langkah dilakukan secara sistematis untuk memastikan seluruh *agent observability* berjalan normal dan mampu mengirimkan log ke sistem *Elastic*. Kegiatan ini dimulai dengan menghubungkan jaringan menggunakan *VPN*, kemudian melakukan *login* ke *Elastic* untuk memantau kondisi *agent observability*. Jika kondisi agent terdeteksi *low* (tidak optimal), maka analis mencatatnya pada daftar *Excel* dan *handover list* sebagai bahan laporan dan tindak lanjut. Namun, apabila status agent *down* atau tidak aktif, maka dilakukan pengecekan lebih lanjut melalui *Elastic Fleet* untuk mengetahui penyebab gangguan. Berdasarkan hasil pemeriksaan tersebut, analis menentukan arah eskalasi apakah bersifat *internal* atau *external*. Jika penyebabnya berasal dari sisi internal, maka tiket eskalasi diteruskan kepada tim *sysadmin* untuk penanganan. Sebaliknya, apabila masalah berasal dari sisi eksternal atau perangkat milik *customer*, maka eskalasi dikirimkan kepada pihak *customer*. Setelah seluruh langkah penanganan dilakukan, proses monitoring dinyatakan selesai. Berikut merupakan *flowchart* dari *log availability* pada Gambar 3.7.

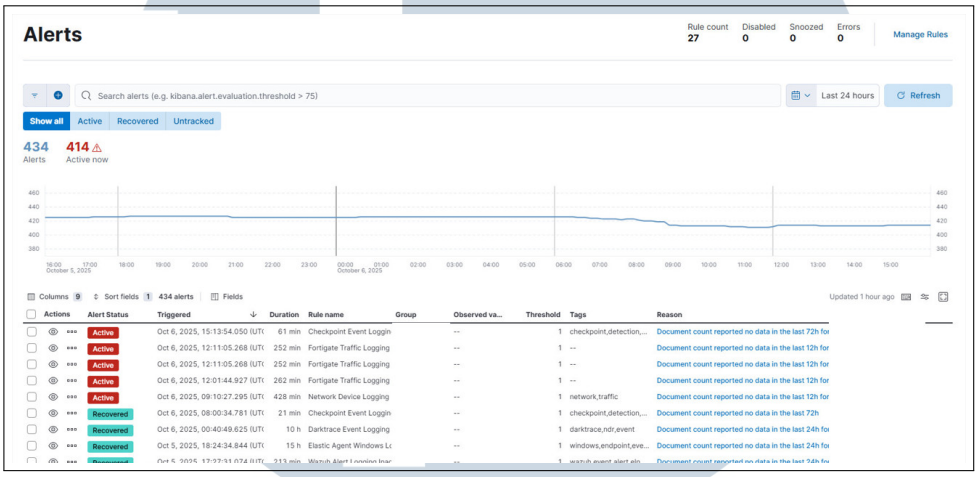


Gambar 3.7. Flowchart Monitoring Log Availability Elastic

A. Pengecekan Observability Agent pada Elastic

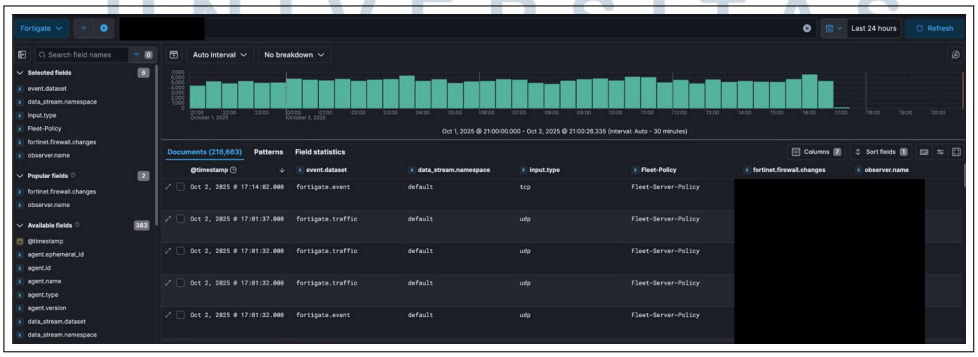
Pada tahap ini, kegiatan pemantauan dilakukan melalui fitur *Log Availability Monitoring* yang disediakan oleh Elastic Observability. Fitur ini berfungsi untuk memantau ketersediaan (*availability*) agent secara *real-time* dan menampilkan

informasi apabila terdapat agent yang mengalami gangguan atau berhenti mengirimkan log. Dashboard *Observability Alert* menampilkan daftar agent yang tidak aktif disertai dengan detail seperti waktu *triggered*, durasi downtime, *rule name*, *group*, *observed value*, *threshold*, *tags*, serta *reason* yang menjelaskan penyebab terjadinya anomali. Berikut merupakan visualisasi dari observability pada gambar 3.8.



Gambar 3.8. Visualisasi Observability

Apabila ditemukan agent yang mati atau menyentuh ambang batas (*threshold*) tertentu, dilakukan analisis lebih lanjut dengan membuka tampilan detail melalui ikon "mata" (*View in App*). Pada tahap ini, rentang waktu (*time range*) biasanya diperluas hingga tujuh hari terakhir untuk memastikan apakah log benar-benar berhenti pada tanggal tersebut atau hanya mengalami penurunan aktivitas sementara. Langkah ini juga membantu membedakan apakah kondisi tersebut merupakan aktivitas normal seperti *scheduled task* atau indikasi gangguan abnormal yang memerlukan eskalasi lebih lanjut. Berikut merupakan visualisasi dari log mati pada gambar 3.9.

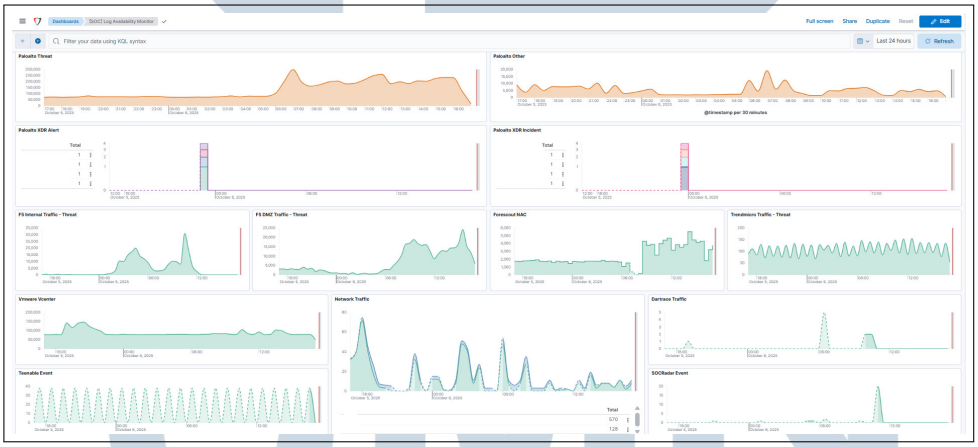


Gambar 3.9. Visualisasi ketika log down

B. Pengecekan pada *Dashboard Log Availability Monitor* Lanjutan

Pengecekan lanjutan dilakukan pada *Dashboard Log Availability Monitor* untuk setiap regional beserta masing-masing *index* dan *agent* yang digunakan, seperti *Forescout NAC*, *Trend Micro Traffic – Threat*, *Darktrace Traffic*, dan komponen keamanan lainnya. Tahap ini bertujuan untuk memastikan bahwa seluruh aliran log dari berbagai sumber tetap aktif dan tidak mengalami gangguan pengiriman data.

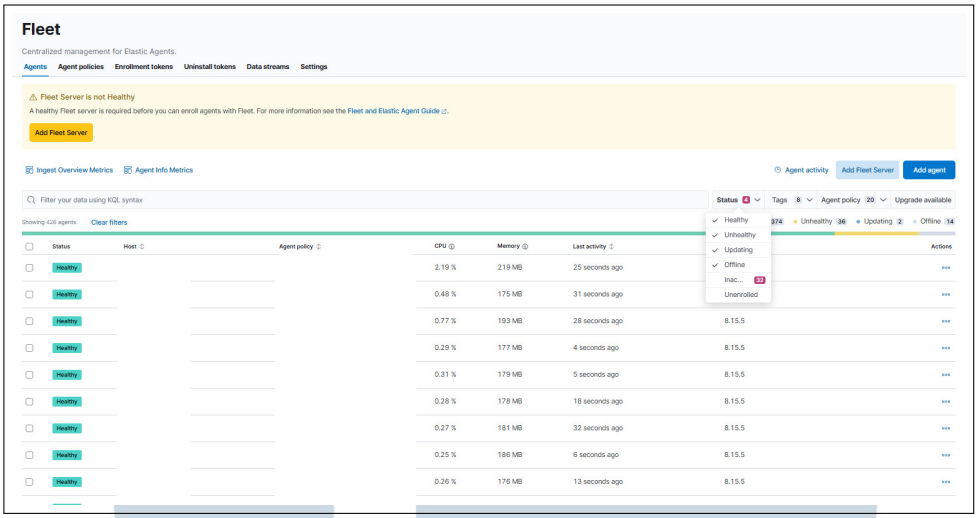
Proses pemantauan dilakukan secara manual dengan mengatur rentang waktu (*time range*) pada tampilan dashboard ke *Last 24 Hours*. Dari hasil visualisasi grafik, diperhatikan apakah terdapat penurunan mendadak atau penghentian aktivitas log yang dapat mengindikasikan adanya masalah pada agent tertentu. Melalui fitur *hover* pada grafik, analis dapat melihat jumlah catatan log yang masuk (*count of records*) serta waktu terakhir log diterima oleh sistem. Berikut merupakan contoh dari *dahsboard* grafik *log availabilit*y pada gambar 3.10.



Gambar 3.10. Visualisasi Dashboard Log Availability

C. Pengecekan Agent Health pada Fleet

Untuk agent yang terdeteksi mati pada *Log Availability Monitoring*, dilakukan pengecekan lanjutan melalui menu *Management* → *Fleet* pada Elastic. Melalui fitur ini, analis dapat memasukkan nama agent yang sebelumnya teridentifikasi tidak aktif untuk melihat kondisi aktual dari host tersebut, apakah tergolong *healthy* atau *unhealthy*. Tampilan *Fleet Management* juga memudahkan proses identifikasi dengan menampilkan status agent secara visual, sebagaimana ditunjukkan pada Gambar 3.11.



Gambar 3.11. Visualisasi Agent Health pada Fleet

Selain status utama, *Fleet* menyediakan informasi tambahan yang cukup detail mengenai setiap agent, seperti kategori status (*healthy*, *unhealthy*, *updating*, *offline*, *inactive*, dan *unenrolled*), serta informasi teknis lain seperti *agent policy*, penggunaan *CPU* dan *memory*, waktu *last activity*, versi agent yang digunakan, hingga daftar *actions* yang dapat dilakukan terhadap agent tersebut. Dengan adanya fitur ini, tim SOC dapat melakukan evaluasi cepat terhadap kondisi endpoint dan menentukan langkah tindak lanjut apabila ditemukan agent yang bermasalah atau tidak lagi berfungsi sebagaimana mestinya.

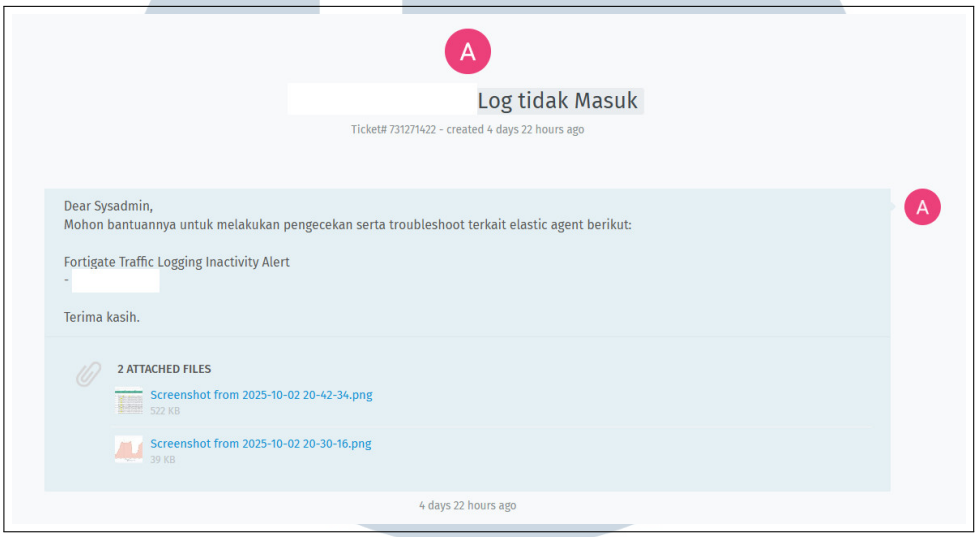
3.3.4 Eskalasi Terkait Masalah Agent

Pada bagian eskalasi terkait masalah *agent*, kegiatan difokuskan pada penanganan kendala teknis yang muncul pada perangkat atau sistem monitoring. Proses ini mencakup eskalasi internal melalui sistem *ticketing* untuk koordinasi antar tim, serta eskalasi eksternal kepada pihak *customer* apabila permasalahan terjadi pada aset yang berada di lingkungan mereka. Untuk lebih detailnya, proses eskalasi terkait masalah *agent* dijelaskan sebagai berikut.

A. Proses Eskalasi Melalui Ticketing (Internal)

Setelah dilakukan pengecekan terhadap *observability agent* dan ditemukan adanya agent yang mati atau tidak mengirim log, langkah pertama yang dilakukan adalah melakukan proses eskalasi melalui sistem internal yang disebut *Ticketing System*. Pada tahap ini, *Security Analyst* membuat tiket baru dengan judul

"Log Tidak Masuk" dan melampirkan bukti tangkapan layar sebagai *evidence* pendukung. Tiket tersebut kemudian dikirimkan kepada tim *System Administrator* (sysadmin) untuk dilakukan penanganan lebih lanjut. Setelah tiket berhasil dibuat, tautan tiket dibagikan ke grup WhatsApp *Operations* agar dapat segera ditindaklanjuti. Berikut merupakan contoh ketika melakukan *ticketing* pada gambar 3.12.



Gambar 3.12. Visualisasi Ticketing Agent

Selain itu, seluruh daftar agent yang telah dilakukan pengecekan baik yang telah diekalasi ke sysadmin maupun yang tidak (karena tergolong *low log* atau aktivitas normal) dicatat dalam daftar *handover* untuk diserahkan kepada *shift* berikutnya. Dengan cara ini, kegiatan pemantauan dapat terus berlanjut secara konsisten meskipun terjadi pergantian personel antar *shift*.

B. Proses Eskalasi ke Customer (Eksternal)

Dalam beberapa kasus, hasil *troubleshooting* menunjukkan bahwa *agent* mati disebabkan oleh permasalahan internal yang dapat diperbaiki langsung oleh sysadmin. Namun, apabila penyebab gangguan berasal dari sisi eksternal atau berkaitan dengan lingkungan milik *customer* di mana tim sysadmin tidak memiliki visibilitas penuh untuk melakukan perbaikan, maka eskalasi dilakukan ke pihak *customer*. Pada situasi seperti ini, sysadmin akan menyarankan *security analyst* untuk melakukan komunikasi langsung kepada *customer* menggunakan platform resmi seperti Thunderbird atau Outlook. Langkah ini bertujuan agar pihak *customer* dapat melakukan pengecekan pada sistem mereka dan memberikan tindak lanjut

yang diperlukan guna memastikan agent kembali aktif serta proses pengiriman log berjalan normal.

3.3.5 Monitoring Performa Sistem

Kegiatan monitoring performa sistem dilakukan untuk memastikan seluruh komponen infrastruktur dan perangkat keamanan berfungsi secara optimal. Aktivitas ini mencakup pengecekan kondisi sistem melalui *Grafana* untuk pemantauan performa server, *SolarWinds* untuk pemantauan jaringan, serta *Check Point Harmony* untuk memonitor status dan kondisi *endpoint* secara menyeluruh. Untuk lebih detailnya, kegiatan monitoring performa sistem dijelaskan sebagai berikut.

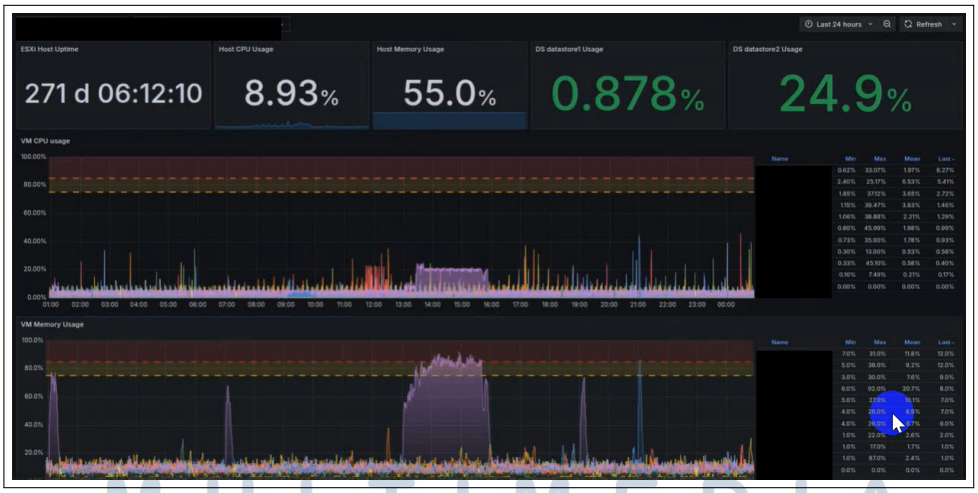
A. Pengecekan pada Grafana

Pengecekan performa sistem melalui Grafana dilakukan secara rutin setiap hari, biasanya pada awal shift malam sekitar pukul 00.00. Tujuan utama dari kegiatan ini adalah untuk memastikan seluruh komponen sistem keamanan berjalan dengan stabil dan tidak mengalami kelebihan beban sumber daya. Grafana digunakan sebagai alat pemantauan visual untuk melihat penggunaan *CPU*, *memory*, serta status layanan dari berbagai komponen penting dalam infrastruktur keamanan.

Pemantauan dilakukan melalui beberapa dashboard, di antaranya *Grafana Summary*, *Grafana HW7*, dan *Grafana HW6*. Pada *Grafana HW7*, fokus pengecekan meliputi performa *Vulnerability Management* seperti *Tenable-Scanner* dan *Tenable-Proxy*, serta modul *Security Information and Event Management* (SIEM), *Endpoint Server Security* dan *Application Security Testing*. Nilai performa diukur menggunakan formula seperti *CPU_last* dan *Memory_MAX* untuk mendeteksi lonjakan penggunaan sumber daya secara cepat. Berikut merupakan visualisasi dari grafana HW7 pada gambar 3.13.

Gambar 3.13. Visualisasi Grafana HW7

Sementara itu, *Grafana HW6* digunakan untuk memantau modul *Network Detection & Response* serta *Security Orchestration, Automation, and Response* (SOAR). Parameter yang diperhatikan mencakup status perangkat (*Device Healthy*), layanan aktif (*Service Up and Running*), serta tingkat pemakaian sumber daya. Nilai performa diukur menggunakan formula seperti CPU: {cpu_last} – Memory: {memory_max}– Storage Usage: {memory_last}. Berikut merupakan visualisasi dari grafana HW6 pada gambar 3.14.



Gambar 3.14. Visualisasi Grafana HW6

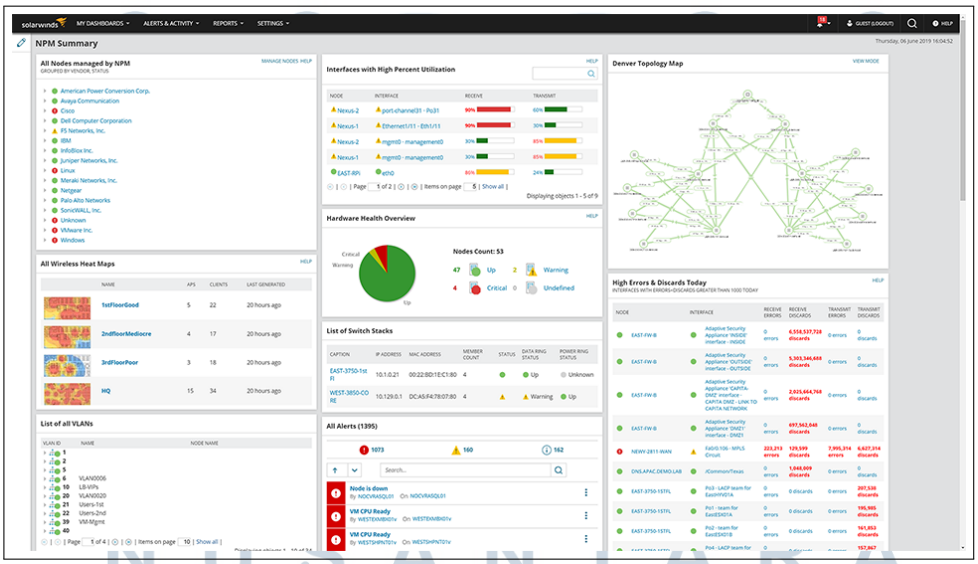
Pengecekan performa melalui Grafana ini bertujuan untuk memastikan stabilitas sistem secara keseluruhan, mendeteksi lebih dini potensi anomali pada penggunaan sumber daya, serta memberikan dasar analisis bagi tim *Security Operations Center* (SOC) dalam menjaga ketersediaan dan keandalan layanan keamanan informasi perusahaan.

B. Pengecekan pada SolarWinds

Pengecekan pada SolarWinds dilakukan secara rutin setiap dini hari sekitar pukul 00.00 sebagai bagian dari aktivitas *monitoring* performa sistem. Tujuan utama dari pengecekan ini adalah untuk memastikan seluruh perangkat jaringan dan sistem keamanan perusahaan berjalan dengan optimal, serta tidak mengalami kelebihan beban sumber daya yang dapat memengaruhi stabilitas layanan.

Melalui dashboard SolarWinds, dilakukan pemantauan terhadap penggunaan *CPU* dan *memory* yang ditampilkan dalam tabel *Average CPU Load & Memory Statistic*. Pengecekan ini mencakup perangkat-perangkat penting seperti *Firewall*, *VPN Gateway*, *Web Application Firewall*, *Network Access Control* (NAC), serta *Security Information and Event Management* (SIEM). Data dari masing-masing perangkat digunakan untuk menilai kinerja sistem secara menyeluruh dan mengidentifikasi potensi anomali sejak dini.

Selain itu, dashboard SolarWinds juga menampilkan berbagai komponen pendukung seperti *Hardware Health Overview*, *Top 10 Nodes by Average CPU Load*, *Top 10 Interfaces by Traffic*, *Disk Volumes*, dan *CPUs by Percent Load*, yang membantu tim SOC dalam memperoleh gambaran menyeluruh mengenai kondisi infrastruktur. Contoh tampilan dashboard simulasi dari SolarWinds ditunjukkan pada Gambar 3.15.



Gambar 3.15. Visualisasi Simulasi Solarwind

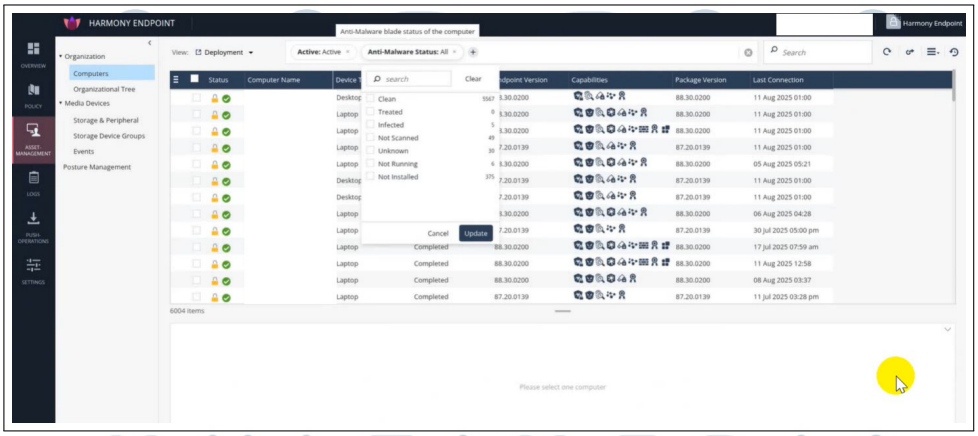
Parameter pengecekan umumnya diukur menggunakan formula: CPU: {cpu_usage19} — Memory: {memory_available57} — Storage Usage

memory_usage43}. Dengan pemantauan ini, tim SOC dapat memastikan performa perangkat jaringan dan keamanan tetap dalam kondisi sehat, serta segera melakukan eskalasi apabila ditemukan penurunan performa yang signifikan.

C. Pengecekan Kondisi Endpoint Melalui CheckPoint Harmony

Pengecekan kondisi endpoint melalui Check Point Harmony dilakukan secara rutin setiap pukul 00.00 sebagai bagian dari aktivitas pemantauan kesehatan sistem keamanan endpoint di lingkungan perusahaan. Tujuan utama dari pengecekan ini adalah untuk memastikan seluruh perangkat endpoint berada dalam kondisi aman, memiliki perlindungan aktif terhadap ancaman *malware*, serta menjalankan *agent security* secara optimal tanpa gangguan.

Proses pemantauan dilakukan melalui menu *Asset Management* $\rightarrow$ *Anti-Malware Status* pada dashboard Check Point Harmony. Bagian ini menampilkan status perlindungan *anti-malware* yang dikategorikan ke dalam empat klasifikasi, yaitu *Clean*, *Infected*, *Not Scanned*, dan *Not Updated*. Dengan adanya klasifikasi ini, tim *Security Operations Center* (SOC) dapat dengan mudah mengidentifikasi perangkat yang memerlukan perhatian khusus, seperti endpoint yang belum diperbarui atau terdeteksi terinfeksi. Contoh tampilan dashboard Check Point Harmony dapat dilihat pada Gambar 3.16.



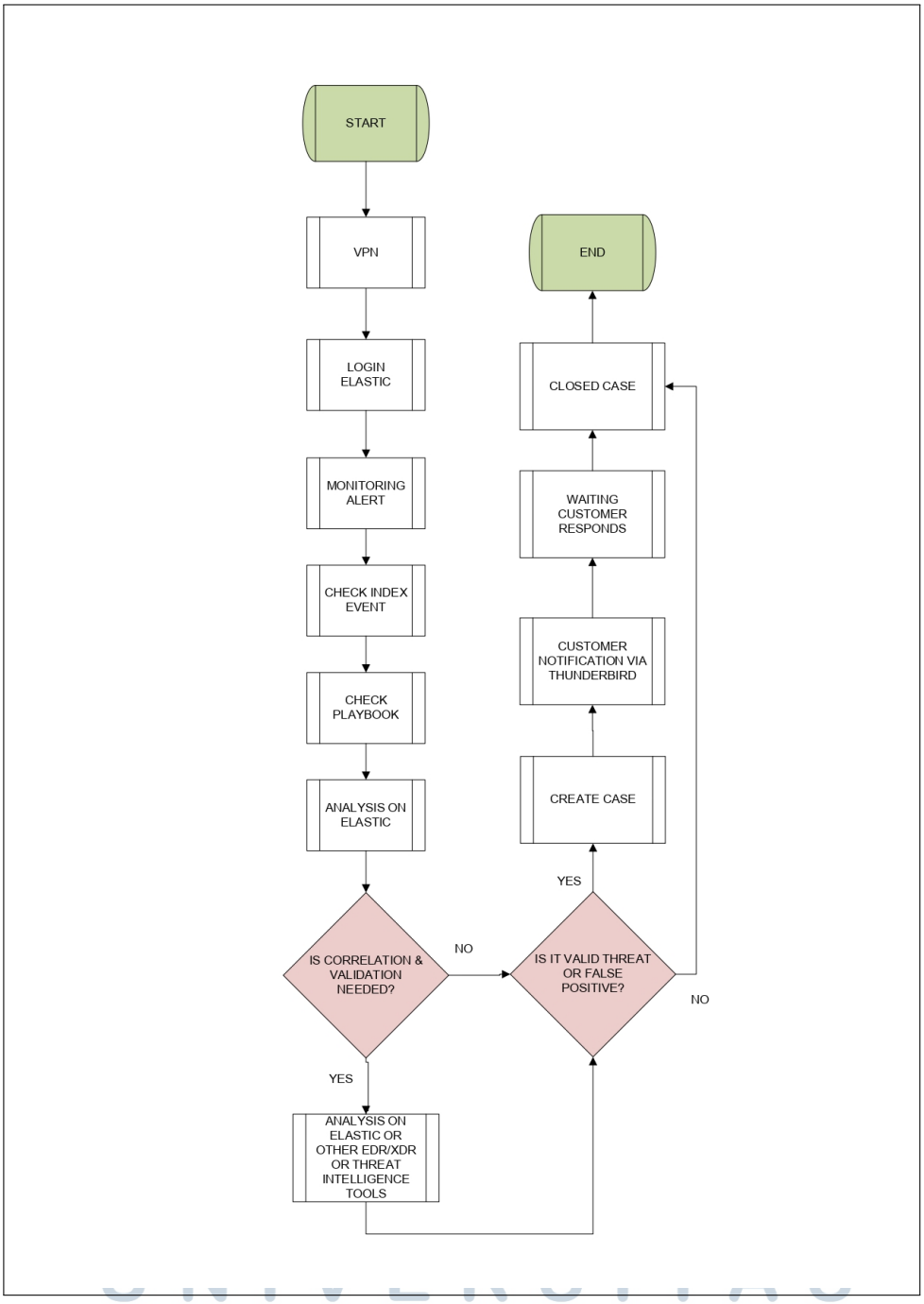
Gambar 3.16. Visualisasi Checkpoint Harmony

Selain fitur status *anti-malware*, dashboard Check Point Harmony juga menyediakan informasi tambahan seperti jumlah *active endpoint* untuk mengetahui berapa banyak perangkat yang sedang aktif, daftar *devices with operational issues* untuk mendeteksi masalah fungsional pada endpoint, *deployment status* untuk

memantau status pemasangan agent, serta *outdated capabilities* yang menunjukkan modul keamanan yang sudah memerlukan pembaruan.

3.3.6 Monitoring SIEM dan Penanganan Insiden

Pada proses kerja *Elastic SIEM* dalam penanganan *alert* dan notifikasi kepada *customer*, langkah-langkah dilakukan secara berurutan untuk memastikan setiap potensi ancaman dapat diverifikasi dan ditangani dengan tepat. Kegiatan dimulai dengan menghubungkan jaringan melalui *VPN* dan melakukan *login* ke *Elastic* untuk memantau *alert* yang muncul. Setelah *alert* terdeteksi, analis memeriksa *index event* dan meninjau *playbook* sebagai panduan prosedural. Tahap selanjutnya adalah melakukan analisis pada *Elastic* untuk menentukan apakah diperlukan korelasi dan validasi lebih lanjut menggunakan alat lain seperti *EDR*, *XDR*, atau *threat intelligence tools*. Jika hasil analisis menunjukkan bahwa ancaman bersifat valid (*valid threat*), maka analis membuat *case* dan mengirimkan notifikasi kepada *customer* melalui *Thunderbird*. Setelah itu, sistem menunggu respons dari *customer* sebelum kasus dinyatakan selesai dan ditutup. Sebaliknya, jika hasil analisis menunjukkan *false positive*, maka proses tidak dilanjutkan dan kembali ke tahap pemantauan *alert*. Berikut merupakan *flowchart* dari *Elastic SIEM* dalam proses notifikasi *customer* pada Gambar 3.17.

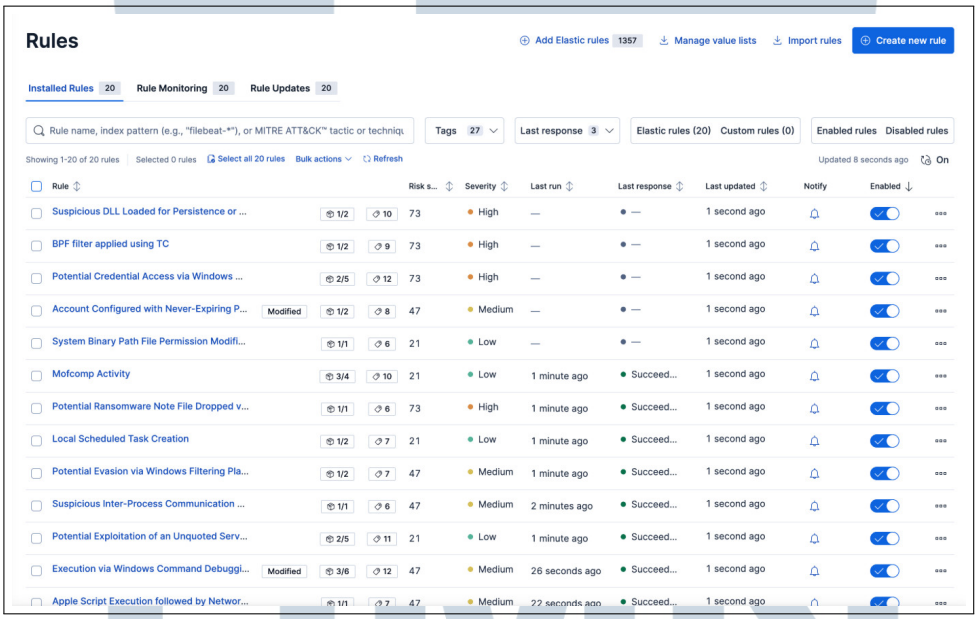


Gambar 3.17. Flowchart Monitoring Log SIEM dan Notifikasi Customer

A. Pemantauan Alert pada SIEM Elastic

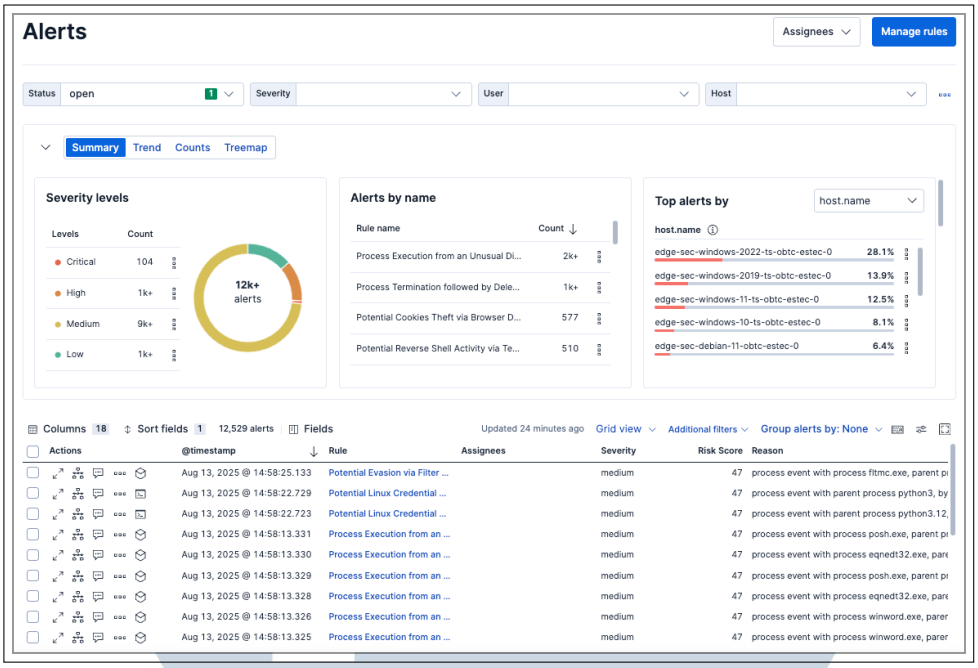
Pada sistem *SIEM Elastic*, fitur *alert* berfungsi sebagai komponen utama dalam mendeteksi aktivitas mencurigakan berdasarkan log yang dikumpulkan dari berbagai sumber, seperti *endpoint*, server, *firewall*, dan perangkat jaringan

lainnya. Prosesnya diawali dengan pengiriman data mentah oleh *agent* atau integrasi (misalnya *Elastic Agent* atau *Beats*) ke Elasticsearch. Data tersebut kemudian melalui tahap pemrosesan dan dinormalisasi menggunakan *Elastic Common Schema (ECS)* agar setiap *event* memiliki struktur yang seragam, seperti *host.name*, *user.name*, dan *event.action*. Setelah data tersimpan di indeks, proses deteksi dilakukan melalui *detection rules*, yaitu seperangkat aturan yang dirancang untuk mengenali pola atau perilaku tertentu yang mengindikasikan adanya potensi ancaman. Setiap *rule* berjalan secara terjadwal dalam interval waktu tertentu, dengan jendela pencarian (*look-back window*) untuk memastikan tidak ada *event* yang terlewat. Berikut merupakan simulasi dari rules Elastic pada gambar 3.18.



Gambar 3.18. Visualisasi Simulasi Rules pada Elastic

Apabila suatu *rule* menemukan log yang sesuai dengan kondisi yang telah ditentukan, sistem akan menghasilkan *alert* yang berisi informasi penting seperti nama *rule*, waktu kejadian, nama *host* dan pengguna, tingkat keparahan (*severity*), skor risiko (*risk score*), serta alasan deteksi (*reason*). *Alert-alert* ini kemudian dikelompokkan berdasarkan tingkat keparahan yang divisualisasikan dalam bentuk grafik pada *dashboard*, sehingga memudahkan analis dalam menentukan prioritas penanganan. Selain itu, Elastic juga dapat melakukan *enrichment* terhadap *alert* dengan menambahkan konteks tambahan seperti *threat intelligence* atau *risk scoring* untuk memberikan gambaran lebih jelas terkait tingkat risiko dari *host* atau pengguna yang terlibat. Berikut merupakan visualisasi dari simulasi Elastic alert pada gambar 3.19.



Gambar 3.19. Visualisasi Simulasi Alert pada Elastic

Dalam operasional *Security Operations Center* (SOC), sistem SIEM *Elastic* berperan dalam mendeteksi potensi ancaman keamanan melalui mekanisme *alerting* berbasis *detection rules* yang telah dikonfigurasi sebelumnya. Setiap *alert* yang terbentuk merupakan hasil pemicu (*trigger*) dari *rule* tertentu berdasarkan pola, frekuensi, dan karakteristik *event* yang terdeteksi pada log sumber. *Alert* tersebut kemudian dianalisis dan diklasifikasikan ke dalam kategorisasi tingkat keparahan insiden (*severity*) yang terdiri dari *Low, Medium, High,* hingga *Critical*, dengan mempertimbangkan indikator teknis, potensi dampak terhadap sistem, serta kemungkinan eskalasi menjadi insiden keamanan. Klasifikasi ini menjadi dasar bagi SOC dalam menentukan prioritas penanganan dan respons insiden yang tepat dan terarah.

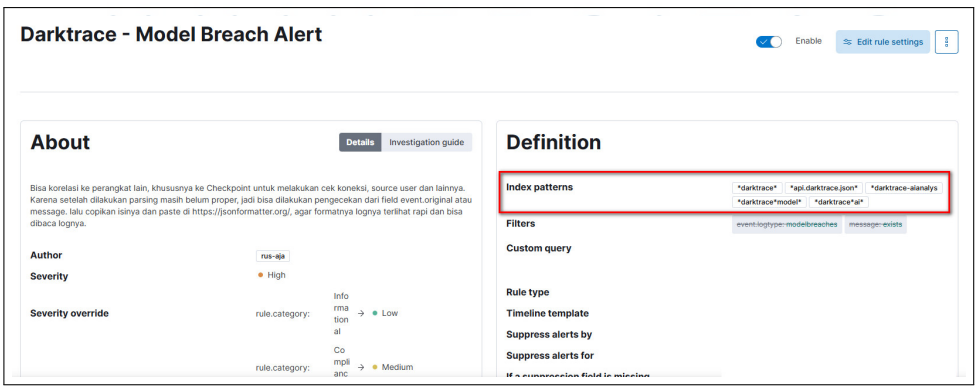
Selanjutnya, setiap kategori *severity* memiliki *Service Level Agreement* (SLA) penanganan yang telah ditetapkan di dalam *incident response playbook* dan disesuaikan dengan ketentuan masing-masing *customer*. SLA ini berfungsi sebagai acuan waktu maksimal bagi tim SOC untuk melakukan analisis awal, eskalasi, serta tindakan mitigasi terhadap *alert* yang terdeteksi. Sebagai contoh, *alert* dengan tingkat *Critical* umumnya memiliki SLA respons awal maksimal 30 menit karena berpotensi menimbulkan dampak besar terhadap layanan atau keamanan data. *Severity High* ditangani dengan SLA sekitar 2 jam, *Medium* dengan SLA 4 jam, dan *Low* dengan SLA hingga 6 jam. Penerapan SLA berbasis *severity* ini

memastikan bahwa sumber daya SOC difokuskan pada ancaman dengan risiko tertinggi, sekaligus menjaga konsistensi, akuntabilitas, dan kepatuhan terhadap standar layanan keamanan yang telah disepakati dengan *customer*.

B. Penentuan Index sebagai Dasar Analisis Log pada Elastic

Sebelum melakukan analisis lebih lanjut pada Elastic, langkah awal yang penting adalah menentukan *index* tempat munculnya *alert*. Identifikasi *index* diperlukan agar proses analisis dapat difokuskan pada sumber data yang relevan dan efisien. Dengan mengetahui *index* asal *alert*, analis dapat menelusuri log secara lebih tepat untuk menemukan penyebab, pola, maupun konteks aktivitas yang memicu peringatan tersebut.

Sebagai contoh, untuk alert brute-force biasa terdeteksi pada index windows ataupun elastic agent, adapun contoh pada *alert Possible Brute-Force Activity* yang terdeteksi oleh NDR Darktrace, SOC mengidentifikasi adanya percobaan login berulang dalam waktu singkat yang mengarah pada indikasi brute-force, yaitu upaya menebak kredensial secara terus-menerus. Untuk kasus *Possible Brute-Force Activity* ini, proses analisis pada Elastic diarahkan ke sejumlah *index pattern* yang berkaitan langsung dengan log Darktrace. Pengecekan dilakukan pada beberapa pola *index* seperti *darktrace*, *api.darktrace.json*, *darktrace-aianalys*, *darktrace*model, dan *darktrace*ai. Melalui *index* tersebut, SOC dapat menelusuri detail event mulai dari informasi koneksi antar host, pola autentikasi berulang, hingga konteks analisis lanjutan yang diberikan oleh Darktrace. Pemeriksaan pada *index* yang tepat menjadi langkah awal yang penting untuk memastikan bahwa aktivitas yang teridentifikasi benar-benar mengarah pada indikasi brute-force atau hanya merupakan aktivitas internal yang masih bersifat valid. Berikut merupakan contoh *index pattern elastic* pada gambar 3.20.



Gambar 3.20. Visualisasi Index Pattern pada Elastic

C. Pengecekan Concern dan Referensi Playbook

Setelah mengetahui *index* dari suatu *alert*, langkah selanjutnya sebelum melakukan investigasi lebih mendalam adalah melakukan pengecekan pada *playbook* terhadap *index* tersebut untuk memahami *concern* atau perhatian khusus dari sisi *customer*. Tahap ini penting karena setiap *customer* umumnya memiliki kebijakan, konfigurasi, dan batasan keamanan yang berbeda-beda. Melalui pengecekan ini, analis dapat mengetahui konteks *alert* secara lebih akurat, apakah terdapat prosedur khusus, atau apakah aktivitas yang terdeteksi memang bersifat mencurigakan atau hanya merupakan bagian dari proses yang legitimate.

Informasi yang diperiksa pada tahap ini mencakup *concern* untuk setiap *index* atau *event*, daftar *whitelist IP*, catatan pengecualian atau *concern* terkait *alert* tertentu, serta data referensi seperti daftar *server* dan *hostname* di tiap regional. Seluruh informasi tersebut umumnya terdokumentasi dalam *playbook* internal yang menjadi acuan standar investigasi bagi tim *Security Operations Center* (SOC). Pada kasus *brute force*, yang tergolong kedalam kategori *web attack*, dilakukan pengecekan tambahan pada Check Point sesuai panduan yang tercantum dalam *playbook*. Contoh tampilan *playbook* yang digunakan dalam proses ini ditunjukkan pada Gambar 3.21.

No.	APPL	TITLE/CASE	ROOT CAUSE	CONCERN	DOCUMENTATION
1		ATT&CK T1037: Logon Scripts (UserInitMprLogonScript)	Internal	PSMInitSession.exe, yang merupakan perilaku normal dalam lingkungan PAM. Meskipun terdeteksi sebagai teknik ATT&CK T1037 (Logon Scripts), tidak ditemukan indikasi aktivitas malicious, namun tetap perlu dipastikan bahwa isi skrip UsrLogon.cmd dan executable yang dijalankan tidak dimodifikasi di luar prosedur resmi.	
2		Trendmicro Visionone, Source Command Execution via Bash	Internal	Aktivitas ini adalah pengecekan lokasi mount partisi /dev/sda7 oleh sistem menggunakan perintah findmnt, sebagai bagian dari proses manajemen kernel crash dump (kdump). findmnt -k -n -r -o TARGET,SOURCE --source /dev/sda7	
3		ATT&CK T1489: Stop Windows Service ATT&CK T1035: Service Execution	Internal	Aktivitas ini merupakan interaksi manual oleh Administrator untuk menjalankan dan menghentikan service FileZilla secara langsung melalui GUI dan command bawaan Windows, tanpa adanya indikasi aktivitas mencurigakan atau teknik eksekusi berbahaya.	
4		Checkpoint untuk Web Attack	Internal	- Deteksi ini berasal dari perangkat Checkpoint terkait adanya web attack, web scanning, log audit, dan lainnya. - Kasus DNS Query to Malicious Site biasanya perlu dinotifikasi apabila aktivitas tersebut belum berhasil dicegah oleh sistem keamanan. Namun, kalau ada aktivitas dengan action detect yang terlihat mengarah ke server internal tertentu, langkah pertama yang dilakukan adalah mengecek apakah koneksi dari server tersebut menuju lingkungan pusat sudah diblok oleh perangkat keamanan di jalur berikutnya. Jika pada lapsan selanjutnya aktivitas itu sudah terprevent, maka insiden tersebut tidak perlu dinaikkan sebagai notifikasi.	

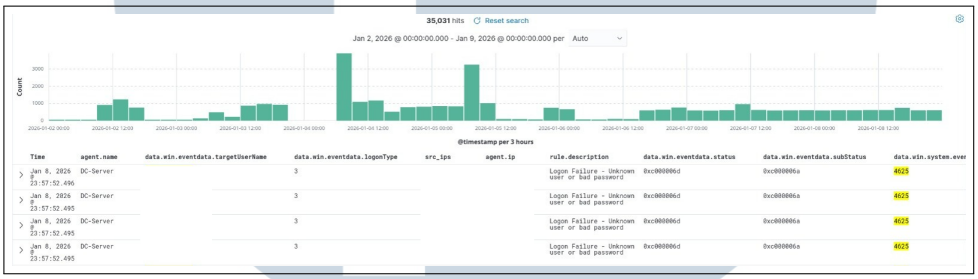
Gambar 3.21. Visualisasi Simulasi Playbook

D. Analisis Detail Alert Melalui Kibana

Untuk mengimplementasikan klasifikasi *severity* dan *Service Level Agreement* (SLA) tersebut secara operasional, *Security Operations Center* (SOC) melakukan analisis detail terhadap setiap *alert* melalui *Kibana* sebagai antarmuka utama *SIEM Elastic*. Pada tahap *monitoring* dan *alert handling*, setiap *alert* yang dihasilkan oleh *SIEM Elastic* tidak langsung diperlakukan sebagai insiden, melainkan terlebih dahulu melalui proses analisis berdasarkan tingkat

keparahan (*severity*) yang telah ditetapkan dalam *playbook* SOC. Analisis ini dilakukan dengan meninjau *index* log terkait, melakukan korelasi *event*, serta memahami konteks aktivitas untuk menentukan apakah *alert* bersifat *false positive*, *suspicious activity*, atau benar-benar merupakan insiden keamanan. Pendekatan ini memungkinkan SOC untuk menerapkan *response* yang proporsional dan sesuai dengan tingkat risiko yang ditimbulkan.

Pada tingkat *Low severity*, SOC mendeteksi aktivitas *logon failure* pada sistem Windows yang dikategorikan sebagai *Unauthorized Access* dengan risiko rendah. Berikut ditampilkan contoh aktivitas *brute force* dengan kategori *low severity* pada Gambar 3.22.

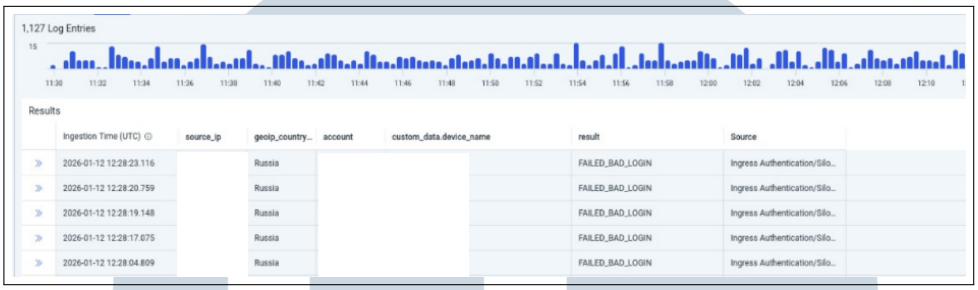


Gambar 3.22. Brute-force Low Severity

Berdasarkan hasil analisis di Kibana, aktivitas ini tercatat lebih dari 100 kali percobaan dalam rentang waktu tertentu dan teridentifikasi melalui Event ID 4625 (*An account failed to log on*). Selain itu aktivitas tersebut menggunakan *Logon Type 3* yang menunjukkan percobaan autentikasi melalui jaringan (*network logon*). Status code yang muncul adalah 0xC000006D dengan substatus 0xC000006A, yang secara teknis mengindikasikan bahwa *username* valid namun *password* yang digunakan salah. Pola ini menunjukkan kesalahan autentikasi tanpa indikasi eksploitasi lanjutan atau keberhasilan login. *Root cause* pada tahap ini dinilai sebagai kesalahan kredensial atau aktivitas tidak disengaja, seperti kesalahan input kata sandi atau kegagalan proses otomatis. Berdasarkan *playbook* SOC, meskipun jumlah *hit* relatif tinggi, karena tidak terdapat indikasi eskalasi atau dampak lanjutan, *alert* ini tetap dikategorikan sebagai *low severity*. Respons SOC pada tahap ini terbatas pada verifikasi aktivitas dan permintaan konfirmasi legitimasi kepada pihak terkait, disertai rekomendasi preventif seperti pengecekan aktivitas login dan penguatan konfigurasi keamanan.

Selanjutnya, pada tingkat *Medium severity*, SOC mendeteksi adanya eskalasi ancaman berupa aktivitas gagal login VPN dari satu alamat IP eksternal yang mencoba melakukan autentikasi ke banyak akun berbeda. Pola

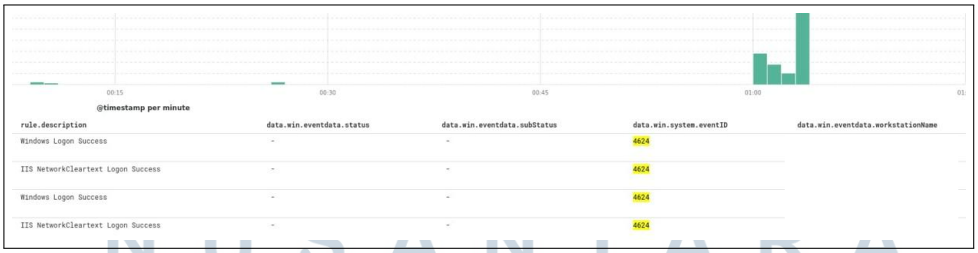
ini mengindikasikan serangan *credential spraying*, yang menunjukkan tingkat kompleksitas lebih tinggi dibandingkan kesalahan login biasa. Berikut ditampilkan contoh aktivitas *brute force* dengan kategori *Medium severity* pada Gambar 3.23.



Gambar 3.23. Brute-force Medium Severity

Analisis log di Kibana memperlihatkan percobaan autentikasi berulang dengan pola seragam terhadap akun yang berbeda dalam rentang waktu singkat. *Root cause* pada level ini mengarah pada upaya *brute force* terdistribusi terhadap layanan VPN yang berfungsi sebagai gerbang akses ke jaringan internal. Dampak potensial dinilai lebih signifikan karena keberhasilan serangan dapat membuka akses jarak jauh ke sistem internal. Oleh karena itu, SOC melakukan respons aktif sesuai *playbook*, antara lain melakukan pemblokiran IP sumber, memberikan notifikasi kepada *customer*, serta merekomendasikan penguatan kontrol keamanan seperti *hardening* layanan VPN, rotasi kredensial akun terkait, dan evaluasi kebijakan autentikasi.

Ancaman kemudian meningkat ke tingkat *High severity* ketika SOC menemukan aktivitas *logon failure* secara masif ke beberapa host internal dari satu sumber IP internal, yang dikorelasikan dengan adanya *login success* melalui Event ID 4624 (*An account was successfully logged on*). Berikut ditampilkan contoh aktivitas *brute force* dengan kategori *High severity* pada Gambar 3.24.

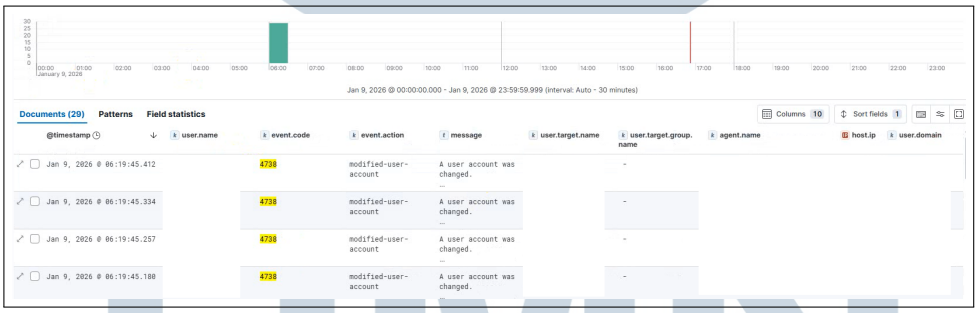


Gambar 3.24. Brute-force High Severity

Analisis di Kibana menunjukkan bahwa meskipun masih didominasi oleh Event ID 4625 dengan *Logon Type 3* dan status kesalahan yang sama,

cakupan aktivitas meluas ke banyak host dalam rentang waktu hingga 12 jam serta melibatkan akun yang berpotensi memiliki fungsi layanan atau hak administratif. Keberadaan Event ID 4624 setelah serangkaian kegagalan autentikasi menjadi indikator penting bahwa setidaknya satu percobaan berhasil, sehingga meningkatkan risiko terhadap integritas sistem. *Root cause* pada tahap ini dinilai sebagai *brute force internal* atau penyalahgunaan kredensial yang telah terkompromi, dengan indikasi awal adanya *lateral movement*. Pada level ini, SOC tidak hanya melakukan notifikasi, tetapi juga melaksanakan analisis dampak yang lebih menyeluruh, meminta klarifikasi legitimasi aktivitas, serta merekomendasikan tindakan mitigasi ketat seperti isolasi host terdampak, reset kredensial, dan pembatasan akses jaringan internal.

Insiden dikategorikan sebagai *Critical* atau *Security Incident* ketika SOC mendeteksi aktivitas perubahan akun pada *Domain Controller* melalui Event ID 4738 (*User Account Changed*) yang dilakukan oleh akun dengan hak istimewa tinggi. Berikut ditampilkan contoh aktivitas *brute force* dengan kategori *Critical severity* pada Gambar 3.25.



Gambar 3.25. Brute-force Critical Severity

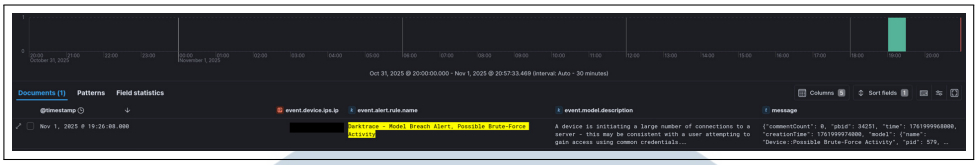
Secara operasional, aktivitas ini tidak selalu bersifat berbahaya karena dapat merupakan bagian dari proses administratif yang sah. Oleh karena itu, *alert* pada tahap awal diperlakukan sebagai *potential suspicious activity* dan memerlukan proses validasi. Namun, apabila hasil verifikasi menunjukkan bahwa perubahan tersebut tidak sesuai dengan prosedur *change management* atau tidak dilakukan oleh pihak yang berwenang, maka kejadian tersebut ditetapkan sebagai insiden keamanan kritikal. Kondisi ini mengindikasikan kemungkinan *privilege escalation* atau kompromi akun administratif yang berpotensi memberikan kendali penuh terhadap domain, sehingga memerlukan respons insiden segera sesuai SLA *critical* dan pembuatan *ticket* ke *team incident response* untuk penanganan lebih lanjut. Untuk memperjelas perbedaan karakteristik ancaman, *root cause*, dampak,

serta respons SOC pada setiap tingkat *severity*, rangkuman analisis severity alert disajikan dalam Tabel 3.4 berikut.

Tabel 3.4. Resume Severity Alert

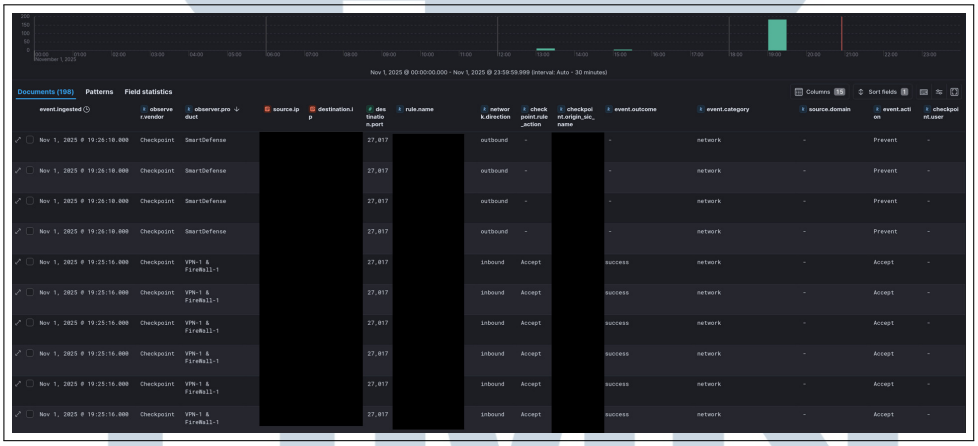
Severity	Deteksi Teknis	Root Cause	Risiko/Dampak	Respons SOC
Low	Event ID 4625 (Logon Failed), Logon Type 3 (Login via network), status 0xC000006A (Password salah)	Kesalahan kredensial/aktivitas tidak disengaja	Dampak rendah, tidak ada eskalasi	Verifikasi, konfirmasi legitimasi, rekomendasi preventif
Medium	Gagal login VPN multi-akun dari 1 IP	Credential spraying/brute force eksternal	Potensi akses ke jaringan internal	Blok IP, notifikasi customer, hardening VPN
High	4625 (Logon Failed) masif + 4624 (Logon Success) success, multi-host	Kredensial terkompromi, lateral movement	Ancaman integritas sistem internal	Isolasi host, reset kredensial, pembatasan akses
Critical	Event ID 4738 (User Account Changed) pada DC oleh privileged account	Privilege escalation/admin compromise	Kendali penuh domain	Incident response penuh, SLA critical, eskalasi IR team

Selain alert yang dihasilkan langsung oleh SIEM Elastic, SOC juga menerima alert dari sistem deteksi lain yang terintegrasi, seperti Network Detection and Response (NDR) Darktrace, yang kemudian dianalisis dan divalidasi melalui Kibana dengan subject "Possible Brute-Force Activity". Ketika dilakukan analisis lebih lanjut, Darktrace mendeteksi adanya percobaan autentikasi berulang dan aktivitas *scanning* terhadap layanan SMB dan WinRM dalam waktu singkat dari satu host internal di lingkungan server produksi. Aktivitas tersebut berasal dari alamat IP internal (10.x.x.x) dengan *hostname* server konsol yang melakukan koneksi ke beberapa server tujuan pada segmen yang sama melalui port layanan tertentu (8080 sebagai port contoh), dengan frekuensi hingga 198 *hits* yang menyerupai pola *brute force* terhadap layanan internal. Deteksi pada darktrace dapat dilihat pada gambar 3.26 berikut.



Gambar 3.26. Visualisasi Analisis Darktrace pada Kibana Elastic

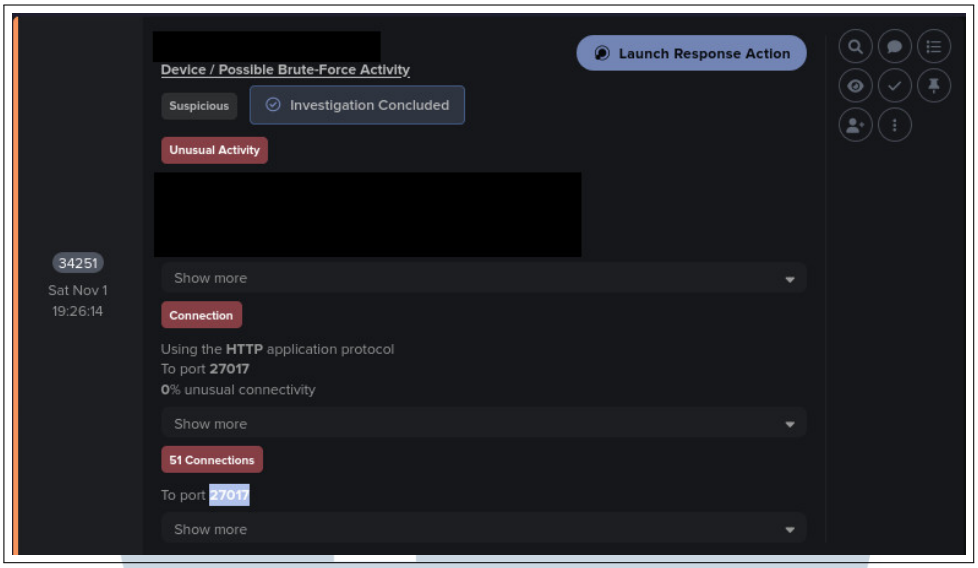
Penelusuran lanjutan pada *index* Check Point di Kibana menunjukkan bahwa aktivitas jaringan yang sama juga tercatat pada perangkat *firewall*, dengan *action* berupa *Accept*, *Prevent*, dan *Drop* sesuai kebijakan keamanan yang berlaku. Sebagian koneksi diizinkan sementara sebagian lainnya diblokir oleh mekanisme proteksi. Tidak ditemukan indikasi aktivitas berbahaya tambahan pada *endpoint agent* (Elastic Agent), sehingga fokus analisis tertuju pada pola koneksi dan autentikasi berulang. Korelasi antara deteksi Darktrace dan log Check Point ini berfungsi sebagai *supporting evidence* yang memperkuat validitas *alert*. Deteksi pada checkpoint dapat dilihat pada gambar 3.27.



Gambar 3.27. Visualisasi Analisis Checkpoint pada Kibana Elastic

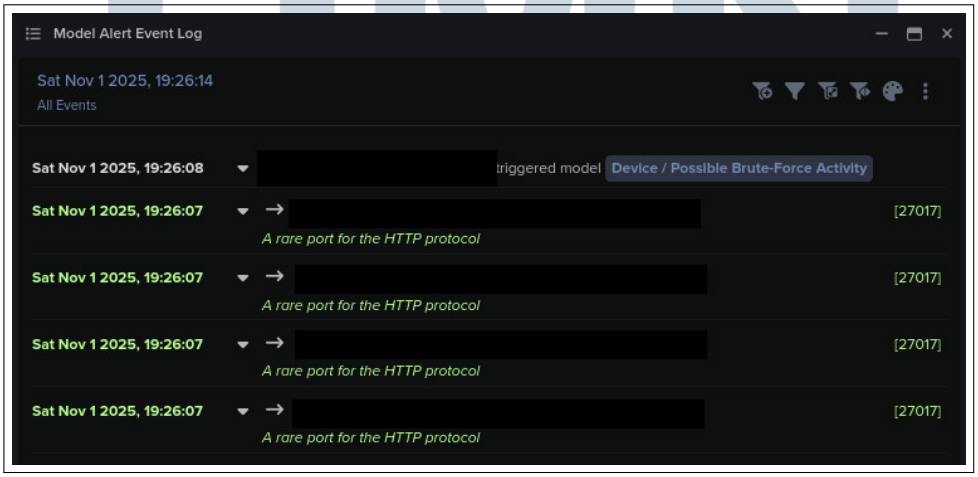
E. Korelasi Log untuk Deep Analysis

Setelah dilakukan analisis pada *Elastic* melalui *index checkpoint* dan *darktrace* untuk kasus *Device Possible Brute-Force Activity*, dilakukan korelasi terhadap log pada *Console Darktrace* sebagai validitas dan *evidence* tambahan untuk memastikan kebenaran aktivitas yang terdeteksi. Berikut merupakan visualisasi alert Dakrtrace pada gambar 3.28.



Gambar 3.28. Visualisasi Korelasi Log pada Darktrace

Darktrace menampilkan sebuah *model alert* dengan judul yang sama, yang dipicu oleh salah satu host internal ketika melakukan koneksi berulang ke salah satu server aplikasi pada jaringan perusahaan. Dari panel *Model Alert Event Log*, terlihat bahwa host tersebut melakukan banyak koneksi dalam waktu yang sangat berdekatan menuju alamat IP internal sebagai *destination IP* dengan port tujuan 8080 (port contoh), yang dikategorikan oleh sistem sebagai *a rare port for the HTTP protocol*. Koneksi ini tercatat menggunakan aplikasi HTTP dengan total sekitar 51 koneksi ke port yang sama dalam satu rangkaian kejadian. Berikut merupakan visualisasi lanjutan dari Darktrace pada gambar 3.29.



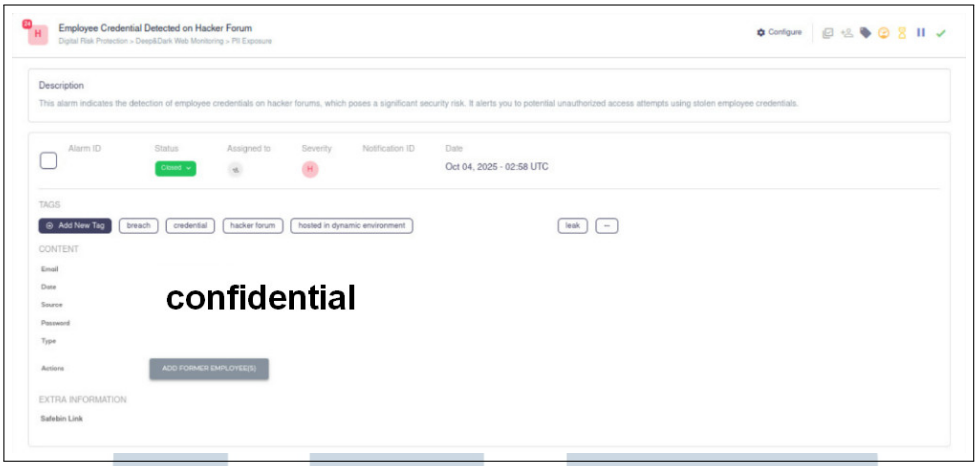
Gambar 3.29. Visualisasi Korelasi Log pada Darktrace

Hasil analisis lebih lanjut pada *Console Darktrace* menunjukkan bahwa aktivitas tersebut diklasifikasikan sebagai *Unusual Activity* dengan beberapa

matching metrics seperti *Internal Active Connections*, *Internal Connections to Closed Ports*, dan *Internal Connections*. Selain itu, *alert* juga mencatat bahwa host tersebut menggunakan kredensial akun layanan internal (*service account*) ketika membangun koneksi ke server tujuan. Meskipun pada bagian *connection* ditampilkan bahwa konektivitas HTTP ke port 8080 (port contoh) memiliki persentase *unusual connectivity* yang rendah, kombinasi antara frekuensi koneksi yang tinggi, penggunaan port yang tidak umum untuk HTTP, serta pola koneksi yang terarah ke satu host dan port yang sama dalam interval waktu singkat membuat sistem *Darktrace* mengategorikan kejadian ini sebagai indikasi *possible brute-force* atau setidaknya *service probing* terhadap layanan tertentu.

Dari sisi risiko, pola koneksi seperti ini berpotensi mengarah pada percobaan akses tidak sah apabila proses yang berjalan bukan merupakan aktivitas terotorisasi. Jika percobaan koneksi berulang tersebut merupakan bagian dari upaya untuk menguji kredensial atau kestabilan layanan, maka terdapat kemungkinan bahwa pihak yang tidak berwenang dapat menemukan celah untuk mendapatkan akses ke sistem internal. Selain itu, lonjakan jumlah koneksi dalam waktu singkat juga berpotensi menambah beban pada server tujuan, yang dapat memengaruhi performa maupun ketersediaan layanan. Oleh karena itu, korelasi antara log *checkpoint* dan *Darktrace* ini penting untuk memastikan apakah aktivitas tersebut merupakan proses rutin yang sah, seperti proses koleksi atau sinkronisasi data, atau justru aktivitas tidak sah yang perlu ditindaklanjuti dengan langkah pengamanan lanjutan, seperti peninjauan konfigurasi layanan pada port 8080 (port contoh), pembatasan akses dari host terkait, serta evaluasi ulang terhadap penggunaan akun layanan internal di lingkungan produksi.

Disamping itu, untuk kasus yang berbeda seperti "Employee Credential Detected on Hacker Forum", perlu dilakukan korelasi lebih lanjut menggunakan *Console SOCRadar*. Platform *SOCRadar* dilengkapi dengan *threat intelligence* yang berfungsi untuk mendeteksi *impersonating domain* maupun kebocoran kredensial (*credential leak*). Melalui korelasi yang dilakukan pada *SOCRadar*, ditemukan informasi yang lebih rinci pada sumber *Safebin*, mencakup data seperti *URL*, *username*, dan *password* yang terindikasi bocor. Informasi ini kemudian digunakan untuk melakukan verifikasi terhadap keaslian kredensial tersebut, guna menentukan apakah data yang muncul merupakan *false positive* atau benar-benar valid. Contoh visualisasi dari *dashboard SOCRadar* ditampilkan pada Gambar 3.30.



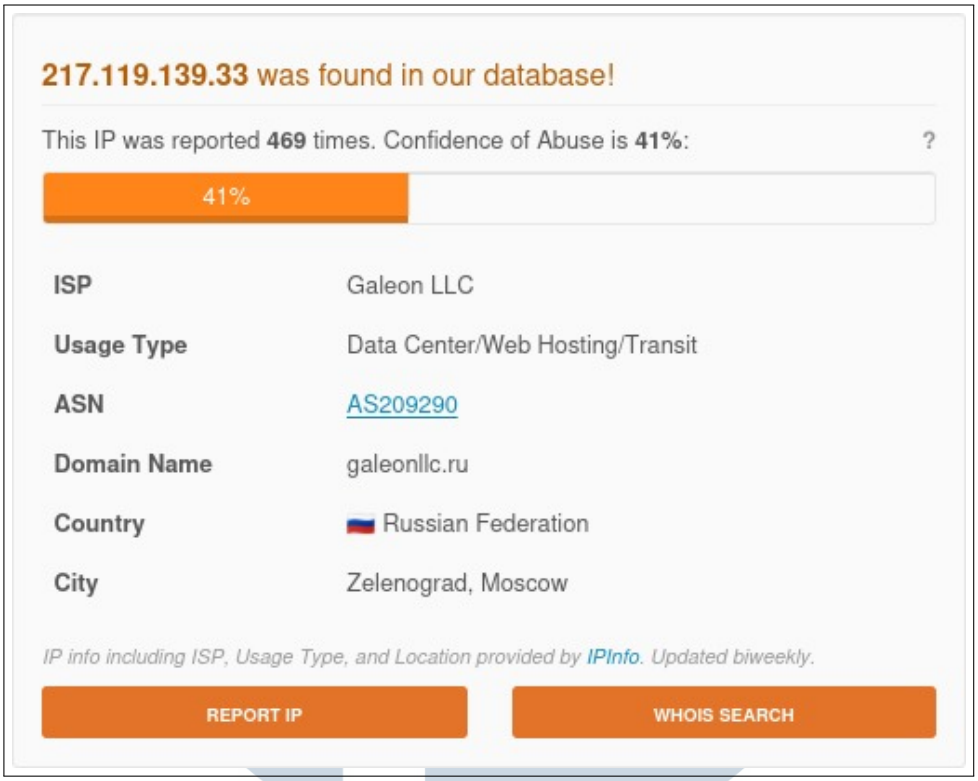
Gambar 3.30. Visualisasi Korelasi Log pada SocRadar

Setelah dilakukan pengecekan lebih lanjut terhadap *URL*, *username*, dan *password* tersebut, ditemukan bahwa kredensial yang dimaksud ternyata valid dan dapat digunakan untuk melakukan *login* ke akun terkait. Temuan ini menunjukkan adanya kebocoran akun yang nyata, sehingga perlu dilakukan langkah *remediasi* lebih lanjut dengan berkoordinasi bersama pihak *customer* untuk melakukan pergantian kredensial dan peningkatan keamanan. Korelasi log seperti ini bersifat *primary* dalam beberapa kasus, karena memberikan bukti langsung mengenai validitas insiden dan menjadi dasar penting dalam proses analisis serta penanganan ancaman keamanan siber.

F. Validasi Alert Melalui Threat Intelligence Tools

Dalam beberapa kasus, proses validasi tambahan melalui *Threat Intelligence Tools* diperlukan untuk memperkuat *evidence* dan memastikan tingkat keabsahan suatu *alert*. Langkah ini membantu analis dalam menentukan apakah aktivitas atau entitas yang terdeteksi benar-benar berpotensi berbahaya atau masih tergolong aktivitas normal.

Sebagai contoh, pada *alert* "Suspicious Authentication - Possible Brute Force Attack on VPN Service", dilakukan pengecekan terhadap alamat IP yang terdeteksi menggunakan beberapa *threat intelligence platform* seperti AbuseIPDB, ThreatBook, atau LiveIPMap. Tujuan pengecekan ini adalah untuk mengetahui apakah IP tersebut tercatat sebagai sumber aktivitas berbahaya (*abuse*), seperti *brute-force attack*, *spam*, atau *malware distribution*, atau justru merupakan alamat IP yang bersih (*clean*). Berikut merupakan gambar dari pengecekan *abuse ip* pada 3.31.



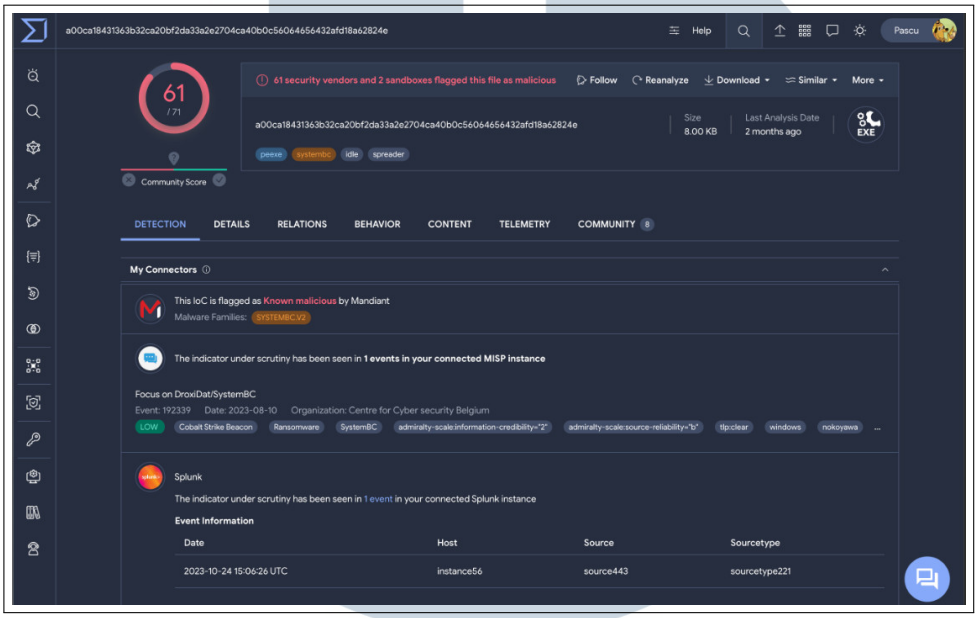
Gambar 3.31. Visualisasi Abuse IP

Untuk *alert* lain seperti "FortiMail – Virus Signature Match Found", validasi dilakukan menggunakan MXToolbox guna memeriksa reputasi dan kesehatan server email (*email health check*) serta memastikan apakah domain pengirim termasuk dalam daftar *blacklist*. Hasil ini membantu menentukan apakah pesan email yang diterima mengandung potensi ancaman atau hanya merupakan *false positive*. Berikut merupakan gambar dari pengecekan *email health* pada 3.32.

Gambar 3.32. Visualisasi Email Health

Sementara itu, untuk *alert* "QRadar – Bluecoat, Query to Suspicious Domain", dilakukan analisis lebih lanjut terhadap *hash*, *URL*, maupun *domain* yang terdeteksi. Pemeriksaan ini bertujuan untuk mengidentifikasi apakah domain tersebut berindikasi terkait dengan aktivitas *malware*, *phishing*, *command and*

control (C2), atau sebenarnya mengarah ke layanan normal seperti *Cloudflare* atau *content delivery network* (CDN). Validasi ini biasanya dilakukan melalui platform seperti VirusTotal, Kaspersky OpenTIP, dan berbagai *threat intelligence feeds* lainnya. Berikut merupakan gambar dari pengecekan *domain abuse* pada 3.33.



Gambar 3.33. Visualisasi Domain and Hash

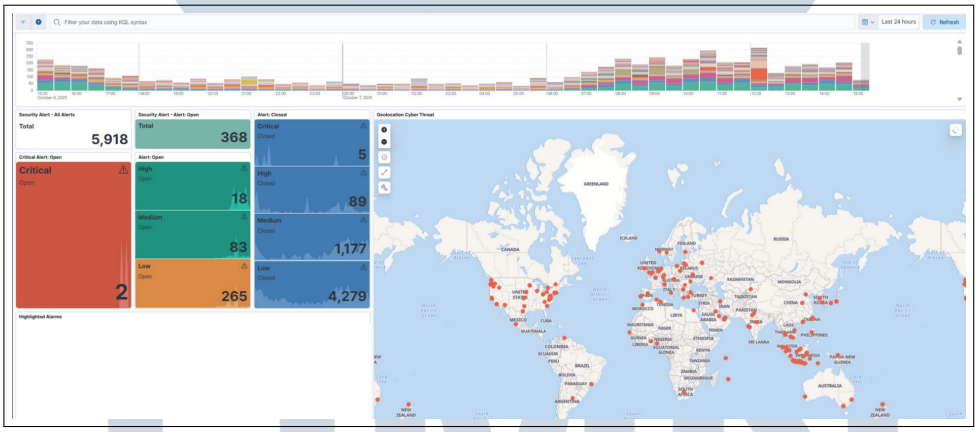
Dengan adanya proses validasi melalui *Threat Intelligence Tools*, tim *Security Operations Center* (SOC) dapat meningkatkan akurasi dalam klasifikasi *alert*, memperkuat bukti investigasi, serta memastikan bahwa setiap tindakan respons dilakukan berdasarkan data intelijen yang kredibel.

3.3.7 Dokumentasi dan Notifikasi

Kegiatan dokumentasi dan notifikasi dilakukan untuk menjaga kontinuitas informasi serta memastikan setiap aktivitas operasional tercatat dengan baik. Proses ini meliputi *handover* pada setiap pergantian shift, pembuatan dan penutupan *case* pada platform *Elastic*, serta penyampaian notifikasi kepada *customer* melalui *Thunderbird*. Selain itu, dilakukan pula penyusunan *daily report* dan *monthly report* sebagai bentuk pelaporan rutin aktivitas monitoring dan penanganan insiden. Untuk lebih detailnya, kegiatan dokumentasi dan notifikasi dijelaskan sebagai berikut.

A. *Handover* Untuk Setiap Pergantian *Shift*

Setiap pergantian *shift* dalam kegiatan monitoring di *Security Operations Center* (SOC) diawali dengan proses *handover* atau serah terima catatan operasional. Tujuan utama dari *handover notes* ini adalah untuk memastikan kesinambungan monitoring dan meminimalkan potensi kelalaian akibat pergantian personel. Pada setiap awal *shift*, *analyst* dari *shift* sebelumnya akan menyusun dan mengirimkan *handover notes* ke grup operasional berisi daftar *alert* yang muncul selama periode sebelumnya, termasuk jumlah *alert* yang masih berstatus *open* maupun yang telah *closed* dari berbagai tingkat *severity* (low, medium, high, dan critical). Selain itu, dicantumkan pula ringkasan informasi penting dari masing-masing *alert* untuk memberikan konteks kepada tim penerima *shift*. Daftar *alert* ini dapat dilihat secara langsung melalui *Alert Dashboard*, seperti yang divisualisasikan pada Gambar 3.34.



Gambar 3.34. Visualisasi Alert Dashboard

Selain laporan *alert*, *handover notes* juga mencakup hasil pengecekan *log availability* yang meliputi status *low log*, *log mati* yang telah dieskalasi, maupun perubahan *hostname* pada agent. Jika terdapat catatan baru, baik berupa *concern* dari sisi *customer* maupun *concern internal SOC*, informasi tersebut juga ditambahkan agar menjadi acuan bagi tim berikutnya dalam melakukan pemantauan lanjutan. Dengan adanya proses *handover* yang terdokumentasi dengan baik, komunikasi antar *shift* dapat berjalan lancar dan meminimalisir informasi yang terlewat.

B. *Create Case* Pada Elastic

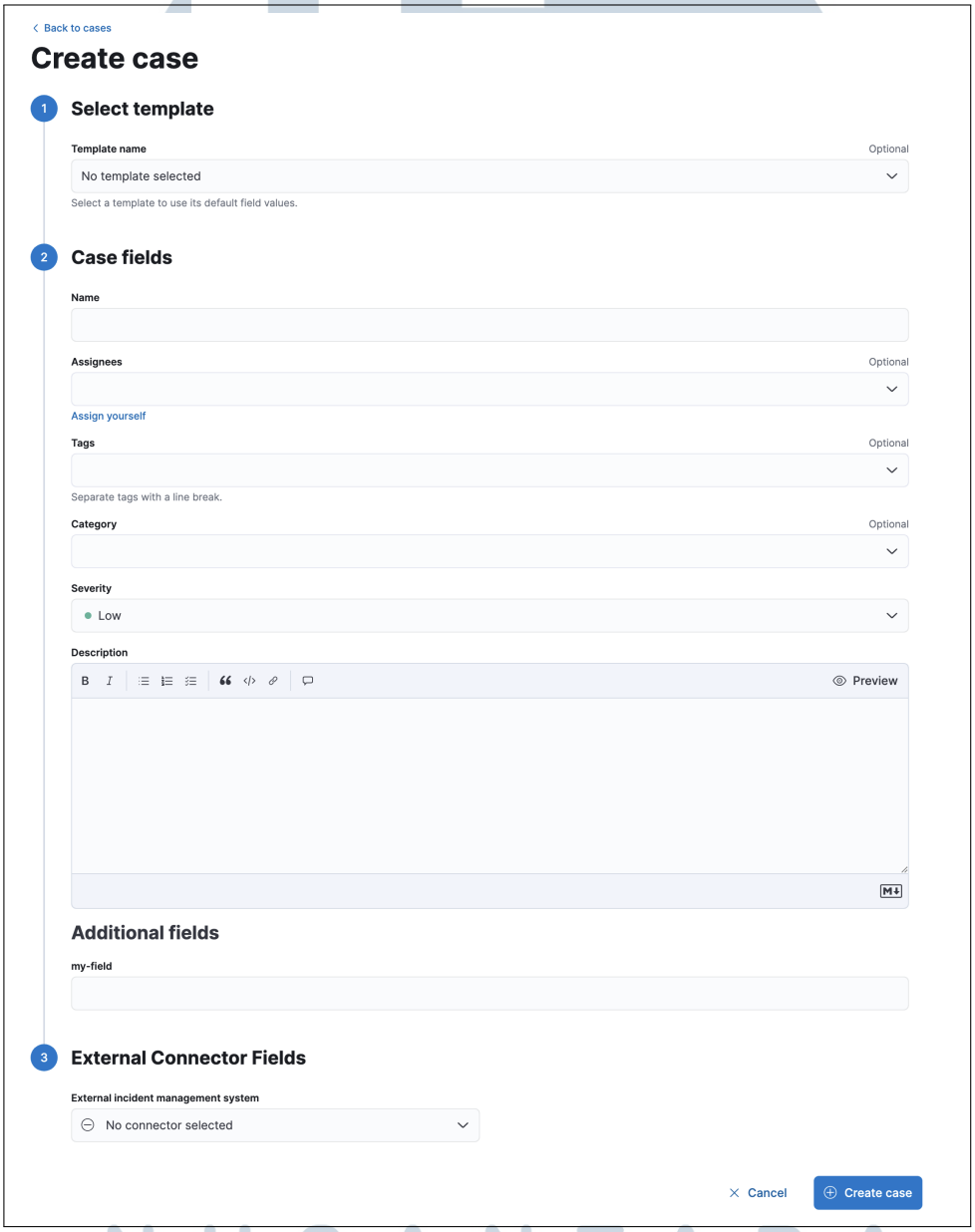
Setelah melalui tahapan investigasi lanjutan, korelasi log, dan validasi menggunakan *threat intelligence tools*, langkah berikutnya adalah menentukan status akhir dari *alert*. Hasil analisis dapat mengarah pada dua kemungkinan utama: *alert* dikategorikan sebagai *false positive* apabila terbukti merupakan aktivitas normal atau sudah termasuk dalam daftar *whitelist*, atau sebaliknya, dinyatakan sebagai *valid threat* apabila aktivitas tersebut benar-benar bersifat berbahaya (*malicious*) atau mencurigakan (*suspicious*). Dalam kasus tertentu, apabila aktivitas belum dapat dipastikan sifatnya, *security analyst* akan mengirimkan notifikasi kepada pihak *customer* untuk meminta konfirmasi apakah aktivitas tersebut bersifat sah (*legitimate*) atau tidak.

Apabila *alert* dinyatakan sebagai *valid threat*, maka *security analyst* akan membuat *case* baru di Elastic sebagai bagian dari proses dokumentasi dan tindak lanjut insiden. Proses ini dilakukan melalui fitur *Create Case* pada Elastic, di mana analis mengisi beberapa komponen penting seperti:

1. *Subject*, berisi judul atau ringkasan singkat insiden.
2. *Name*, yang berisi *case ID* dan *regional* terkait.
3. *Assignees*, diisi dengan nama *security analyst* yang membuat *case*.
4. *Tags*, untuk menandai jenis kasus seperti *internal*, *external*, atau *incident*.
5. *Category*, yang disesuaikan dengan taksonomi *MITRE ATT&CK* guna mengklasifikasikan jenis serangan atau teknik yang digunakan.
6. *Severity*, untuk menentukan tingkat keparahan insiden (*low, medium, high, critical*).
7. *Description*, yang berisi penjelasan rinci mengenai *alert* yang ditemukan, dampak potensial, hasil analisis, serta rekomendasi atau langkah remediasi yang perlu disampaikan kepada *customer*.

Untuk *alert* "Possible Brute-Force Activity", pembuatan *manual case* dilakukan dengan mengisi *template* yang telah ditentukan. *Case name* disesuaikan dengan judul *subject* pengiriman email ke *customer*, sementara bagian *assignee* diisi dengan nama *security analyst* yang menangani insiden tersebut. *Tags* dikategorikan sebagai *internal* karena aktivitas berasal dari IP internal, dengan

klasifikasi MITRE *Credential Access – T1110* pada *category* dan tingkat *severity* ditetapkan pada level *high*. Proses pembuatan *case* ini memastikan setiap ancaman yang terdeteksi terdokumentasi secara sistematis dan transparan, sehingga mempermudah tindak lanjut, pelaporan, serta keseluruhan tahapan *incident response*. Contoh tampilan pembuatan *case* dapat dilihat pada Gambar 3.35.

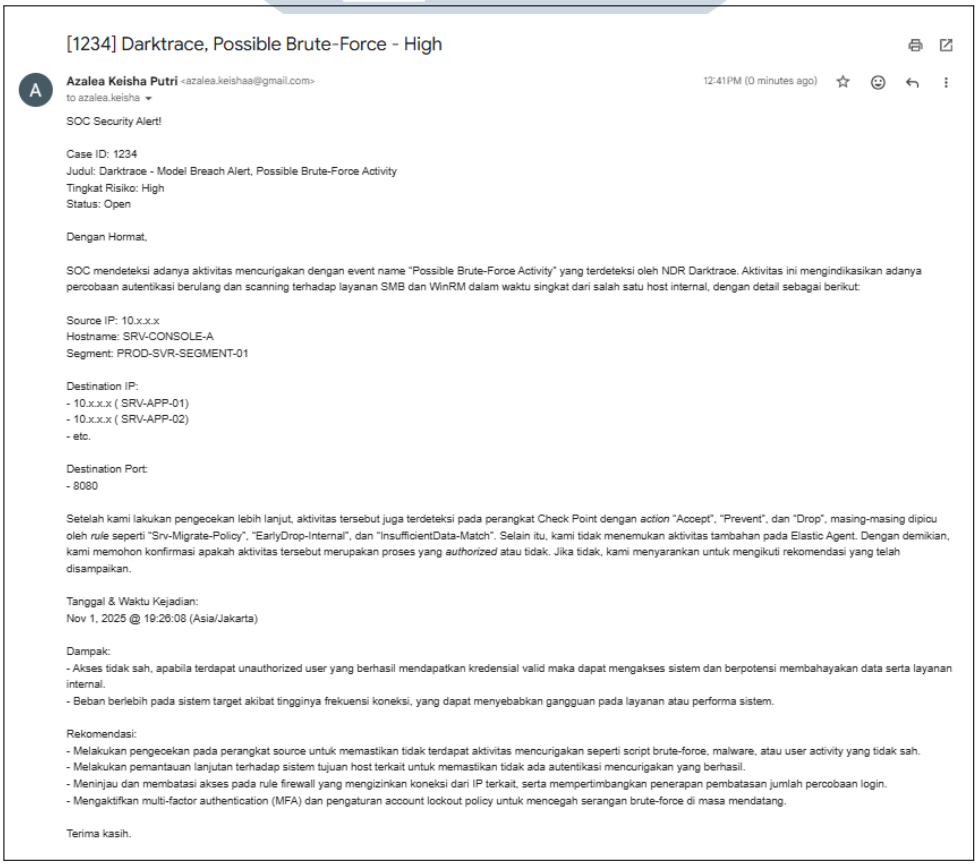


Gambar 3.35. Visualisasi Create Case pada Elastic

C. Notifikasi ke *Customer* Melalui *Thunderbird*

Setelah proses *create case* pada Elastic dilakukan, langkah selanjutnya adalah mengirimkan notifikasi resmi kepada *customer* melalui platform surel seperti Thunderbird atau Outlook. Hal ini diperlukan karena pembuatan *case* di Elastic hanya berfungsi sebagai pencatatan internal dan hanya dapat diakses oleh tim *Security Operations Center* (SOC) baik yang bertugas *onsite* maupun *offsite*, sedangkan pihak *customer* tidak memiliki akses langsung terhadap sistem tersebut.

Proses notifikasi dilakukan dengan menggunakan draf yang telah disusun pada tahap *create case*, kemudian diadaptasi menjadi format komunikasi resmi untuk *customer*. Dalam pesan tersebut, *security analyst* mencantumkan informasi utama seperti judul dan deskripsi *case*, hasil temuan, tingkat *severity*, serta rekomendasi tindak lanjut atau langkah *remediation* yang perlu dilakukan. Untuk memperkuat kredibilitas laporan, disertakan pula lampiran berupa *evidence* seperti tangkapan layar, log terkait, atau hasil validasi dari *threat intelligence tools*. Berikut merupakan contoh notifikasi ke customer pada gambar 3.36.



[1234] Darktrace, Possible Brute-Force - High

Azalea Keisha Putri <azalea.keishaa@gmail.com> 12:41 PM (0 minutes ago)
to azalea.keisha

SOC Security Alert!

Case ID: 1234
Judul: Darktrace - Model Breach Alert, Possible Brute-Force Activity
Tingkat Risiko: High
Status: Open

Dengan Hormat,

SOC mendeteksi adanya aktivitas mencurigakan dengan event name "Possible Brute-Force Activity" yang terdeteksi oleh NDR Darktrace. Aktivitas ini mengindikasikan adanya percobaan autentikasi berulang dan scanning terhadap layanan SMB dan WinRM dalam waktu singkat dari salah satu host internal, dengan detail sebagai berikut:

Source IP: 10.x.x.x
Hostname: SRV-CONSOLE-A
Segment: PROD-SVR-SEGMENT-01

Destination IP:
- 10.x.x.x (SRV-APP-01)
- 10.x.x.x (SRV-APP-02)
- etc.

Destination Port:
- 8080

Setelah kami lakukan pengecekan lebih lanjut, aktivitas tersebut juga terdeteksi pada perangkat Check Point dengan *action* "Acoept", "Prevent", dan "Drop", masing-masing dipicu oleh *rule* seperti "Srv-Migrate-Policy", "EarlyDrop-Internal", dan "InsufficientData-Match". Selain itu, kami tidak menemukan aktivitas tambahan pada Elastic Agent. Dengan demikian, kami memohon konfirmasi apakah aktivitas tersebut merupakan proses yang *authorized* atau tidak. Jika tidak, kami menyarankan untuk mengikuti rekomendasi yang telah disampaikan.

Tanggal & Waktu Kejadian:
Nov 1, 2025 @ 19:26:08 (Asia/Jakarta)

Dampak:
- Akses tidak sah, apabila terdapat unauthorized user yang berhasil mendapatkan kredensial valid maka dapat mengakses sistem dan berpotensi membahayakan data serta layanan internal.
- Beban berlebih pada sistem target akibat tingginya frekuensi koneksi, yang dapat menyebabkan gangguan pada layanan atau performa sistem.

Rekomendasi:
- Melakukan pengecekan pada perangkat source untuk memastikan tidak terdapat aktivitas mencurigakan seperti script brute-force, malware, atau user activity yang tidak sah.
- Melakukan pemantauan lanjutan terhadap sistem tujuan host terkait untuk memastikan tidak ada autentikasi mencurigakan yang berhasil.
- Meninjau dan membatasi akses pada rule firewall yang mengizinkan koneksi dari IP terkait, serta mempertimbangkan penerapan pembatasan jumlah percobaan login.
- Mengaktifkan multi-factor authentication (MFA) dan pengaturan account lockout policy untuk mencegah serangan brute-force di masa mendatang.

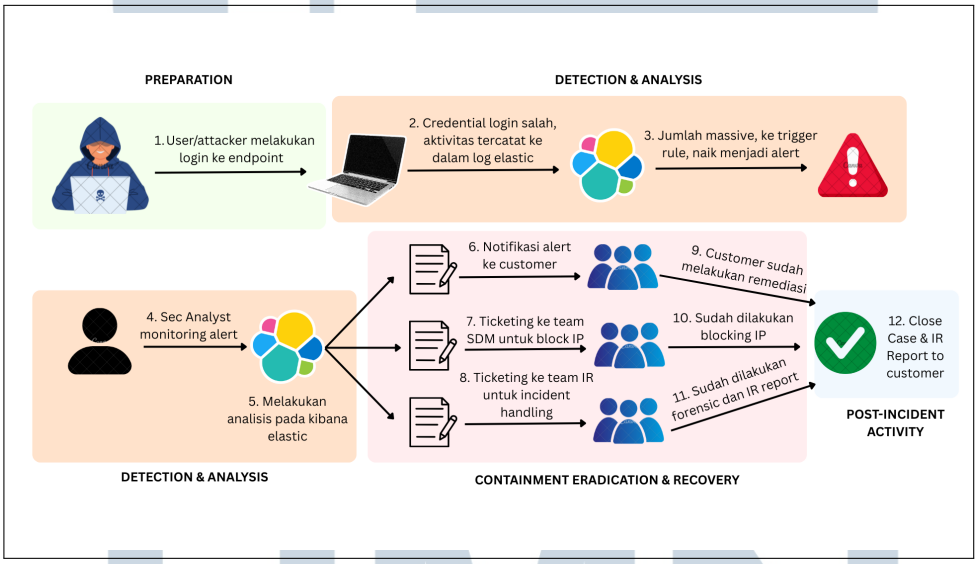
Terima kasih.

Gambar 3.36. Simulasi Notifikasi ke Customer via Thunderbird

Email notifikasi ini dikirimkan kepada kontak *customer* pada regional yang terdampak oleh insiden keamanan, dan selalu ditembuskan (*cc*) kepada tim SOC internal Defenxor untuk keperluan dokumentasi serta koordinasi tindak lanjut.

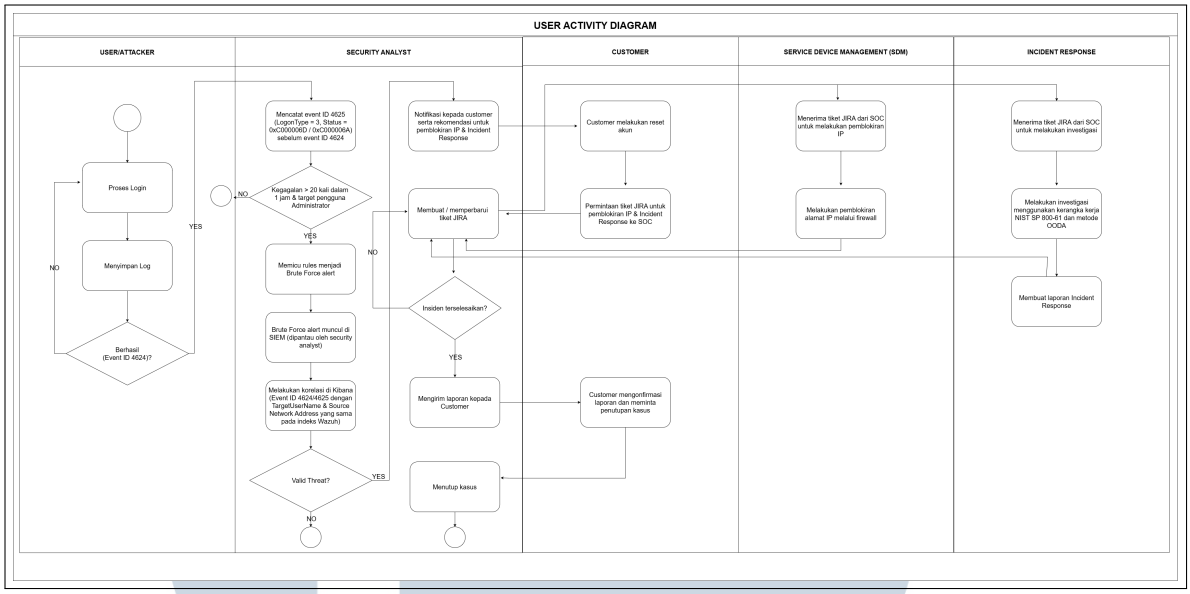
D. Incident Response

Ketika terjadi sebuah insiden keamanan, *company* biasanya menjalankan proses *incident response* untuk memastikan insiden tersebut dapat ditangani secara cepat dan terstruktur. Berikut merupakan contoh visualisasi *workflow incident response* untuk *case Brute Force wazuh windows* pada gambar 3.37.



Gambar 3.37. Work Flow Incident Response Brute Force

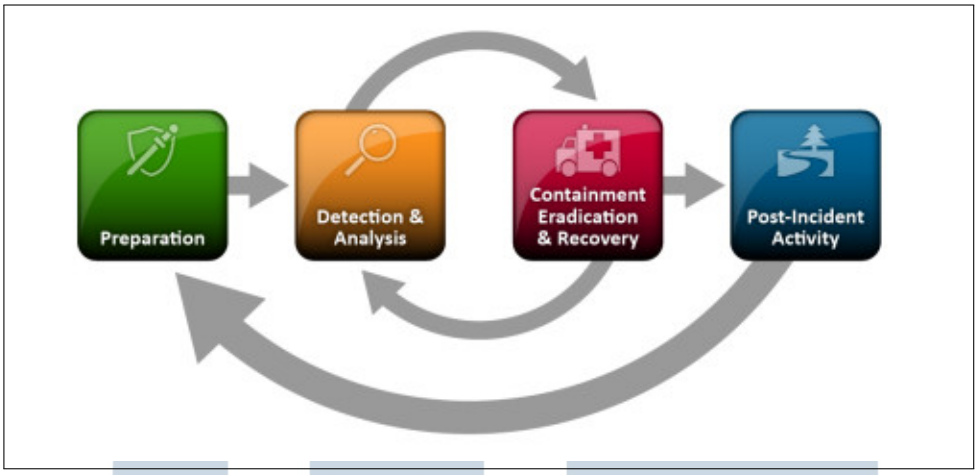
Gambar 3.33 menunjukkan alur umum *incident response* saat terjadi percobaan *brute force* pada *endpoint* Windows yang dipantau melalui *Wazuh Elastic*. Upaya *brute force* dimulai dari *user/attacker* yang melakukan percobaan *login* berulang hingga memicu banyak autentikasi gagal. Aktivitas tersebut terekam di *Elastic* dan dieskalasi menjadi *security alert* oleh *SIEM*. *Alert* kemudian dianalisis oleh *Security Analyst*, yang selanjutnya mengoordinasikan respons dengan mengirim notifikasi ke *customer*, membuat *ticket* ke tim SDM untuk pemblokiran IP, serta meneruskan kasus ke tim *Incident Response* untuk investigasi dan forensik. Setelah *customer* melakukan remediasi akun, tim SDM memblokir IP, dan tim *IR* menyelesaikan investigasi serta laporannya, insiden ditutup dan laporan akhir disampaikan ke *customer*. Detail variabel teknis serta alur yang lebih spesifik dari setiap aktor dan keputusan dapat dilihat pada gambar *user activity diagram* 3.38 berikut.



Gambar 3.38. User Activity Diagram

Pada gambar tersebut digambarkan *detail* alur aktivitas antar pihak ketika insiden *brute force* terjadi, dimulai dari *user/attacker* yang berulang kali mencoba masuk hingga menghasilkan *event log* 4625 dan 4624. Aktivitas ini kemudian terdeteksi dan dianalisis oleh *security analyst* melalui korelasi log di *Kibana*. Hasil analisis tersebut diteruskan kepada *customer*, tim SDM, dan tim *Incident Response* melalui tiket untuk melakukan remediasi akun, pemblokiran IP, serta investigasi forensik. Setelah seluruh langkah penanganan selesai dan laporan insiden disusun, kasus kemudian dinyatakan tertutup.

Dalam hal ini, *company* menerapkan kerangka kerja *NIST SP 800-61* sebagai kerangka makro yang menggambarkan siklus *incident response* melalui empat tahap utama: *Preparation*, *Detection & Analysis*, *Containment, Eradication & Recovery*, serta *Post-Incident Activity*. Berikut merupakan *cycle model* dari *incident response NIST SP 800-61* pada Gambar 3.39.



Gambar 3.39. Cycle Model Incident Response NIST SP 800-61

1. Preparation

Tahap *Preparation* mencakup seluruh aktivitas persiapan yang memungkinkan siklus *OODA* berjalan efektif ketika insiden terjadi. Pada fase ini, organisasi menyusun kebijakan keamanan, mendefinisikan alur *attack flow*, serta mengonfigurasi sistem sehingga *Windows Security Log* dapat terkumpul secara konsisten ke SIEM (Elastic), termasuk memastikan event penting seperti 4624 dan 4625 tercatat dengan lengkap beserta field seperti *TargetUserName*, *Source Network Address*, *LogonType*, dan *SubStatus*. Di saat yang sama, tim menyusun *playbook* investigasi, misalnya skenario serangan *brute force* terhadap akun *privilege* seperti Administrator, yang di dalamnya sudah mengatur langkah *observe* (filter event 4625), *orient* (mengenali pola banyak kegagalan sebagai indikasi *brute force*), hingga kriteria kapan insiden harus dinaikkan menjadi *security incident* prioritas tinggi. Dengan demikian, ketika insiden nyata muncul, analis tidak memulai dari nol, tetapi mengikuti prosedur yang sudah distandardisasi dan selaras dengan kerangka *NIST SP 800-61*.

2. Detection & Analysis

Pada tahap *Detection & Analysis*, aktivitas *observe* dan *orient* dari metode *OODA* terimplementasi secara konkret melalui proses pengumpulan dan analisis log. Tim mengandalkan *alert* dari SIEM untuk mengidentifikasi indikasi serangan, misalnya dengan memfokuskan pencarian pada Event ID 4625 (*failed logon*) di *Windows Security Log* yang telah terintegrasi ke Elastic. Dalam kasus ini, analis menerapkan filter *targetUserName* "Administrator", *Source Network Address*

dari salah satu IP internal, serta rentang waktu 18 November 2025, sehingga terdeteksi sekitar 1.061 event logon gagal ke salah satu server di segmen produksi dengan *logonType* 3 (*network logon*) dan status 0xC000006D/0xC000006A yang menunjukkan akun valid namun kata sandi salah. Informasi tersebut kemudian diorientasikan sebagai pola serangan *brute force* karena banyak percobaan kata sandi ke satu akun *privilege* dari satu IP internal melalui akses jaringan. Mengacu pada *playbook*, analis memandang volume besar Event 4625 sebagai *noise* dan mengarahkan fokus ke pencarian Event 4624 (*successful logon*) yang muncul setelah rangkaian kegagalan tersebut dengan korelasi berbasis kombinasi *TargetUserName* dan *Source Network Address*, baik melalui PowerShell *Get-WinEvent* di sisi host maupun agregasi di SIEM. Ketika korelasi tidak menemukan Event 4624 yang relevan, insiden belum dikategorikan sebagai kompromi akun, tetapi tetap dinilai berisiko tinggi dan pada tahap ini dilakukan klasifikasi sebagai *Security Incident – Brute-Force Network Logon* ke akun Administrator dengan prioritas tinggi sebagai bagian dari proses *decide* dalam *Detection & Analysis*.

3. Containment, Eradication & Recovery

Tahap *Containment, Eradication & Recovery* kemudian mengimplementasikan langkah *decide* dan *act* dari siklus *OODA* dalam bentuk tindakan teknis yang terukur. Setelah insiden ditetapkan sebagai serangan *brute force* prioritas tinggi, tim memutuskan strategi *containment* seperti memblokir sementara IP internal sumber aktivitas melalui pembuatan tiket ke tim pengelola perangkat keamanan untuk menerapkan aturan pada firewall atau ACL, serta mengganti kata sandi akun Administrator di server sasaran dan membatasi penggunaan akun tersebut untuk akses jaringan. Secara paralel, dilakukan *eradication* dan verifikasi di sisi host sumber, meliputi identifikasi host, pemeriksaan proses aktif, *scheduled task*, dan skrip mencurigakan, serta pemindaian AV/EDR untuk memastikan tidak ada *tool* atau malware *brute force* yang berjalan; jika ditemukan akun lain yang dicurigai terkompromi, akun tersebut dapat dinonaktifkan sementara hingga proses forensik dan *credential reset* selesai. Pada fase *recovery*, sistem dikembalikan ke kondisi operasi normal disertai penguatan kontrol, seperti penerapan *Fine-Grained Password Policy* untuk akun kritikal, pemanfaatan fitur *smart lockout*, pengamanan akses *remote* (misalnya RDP hanya melalui VPN atau RD Gateway dengan *Network Level Authentication* diaktifkan), serta aktivasi *Multi-Factor Authentication* untuk seluruh akses

administrasi. Seluruh tindakan ini diikuti pemantauan ulang pola Event 4625/4624 di SIEM untuk memastikan serangan berhenti dan tidak berpindah ke akun atau sumber lain, sekaligus menjadi jembatan menuju aktivitas evaluasi pada fase *Post-Incident Activity*.

4. Post-Incident Activity

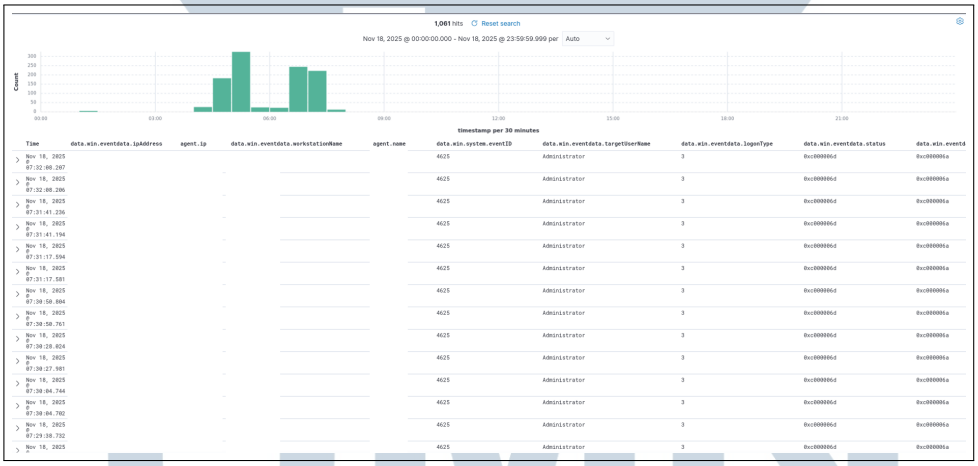
Pada tahap *post-incident activity*, seluruh temuan dan tindakan selama penanganan insiden didokumentasikan secara sistematis sebagai bagian dari proses evaluasi dan peningkatan berkelanjutan. Implementasi tahap ini dilakukan melalui penyusunan *incident response report* dalam bentuk slide PowerPoint yang memuat ringkasan kronologi insiden, sumber dan jenis ancaman, hasil analisis teknis, serta langkah *containment*, *eradication*, dan *recovery* yang telah dilakukan. Selain itu, disusun pula matriks dampak dalam format Excel yang berisi daftar aset atau sistem terdampak, akun yang terpengaruh, tingkat keparahan, estimasi durasi gangguan, serta klasifikasi risiko pasca-insiden. Kedua dokumen tersebut direview secara internal dan disampaikan kepada customer melalui email sebagai laporan formal, sekaligus menjadi bahan diskusi *lesson learned* dan perencanaan penguatan kontrol keamanan.

Setelah seluruh tahapan respons insiden dinyatakan selesai, SOC juga melakukan monitoring lanjutan terhadap sistem kritikal, khususnya Domain Controller, akun dengan hak istimewa, serta pola perubahan akun pengguna, untuk memastikan tidak terdapat aktivitas mencurigakan lanjutan atau kejadian berulang. Hasil monitoring pasca-insiden ini digunakan untuk memvalidasi efektivitas tindakan mitigasi yang telah diterapkan. Berdasarkan evaluasi tersebut, SOC menyusun rekomendasi jangka panjang yang mencakup penguatan kontrol akses administratif, penerapan prinsip *least privilege*, peningkatan mekanisme *logging* dan *alerting* untuk aktivitas akun kritikal, serta penyesuaian (*tuning*) rule deteksi agar insiden serupa dapat teridentifikasi lebih cepat di masa mendatang.

E. Penerapan Metode *Observe, Orient, Decide, Act* (OODA) pada Proses *Incident Response*

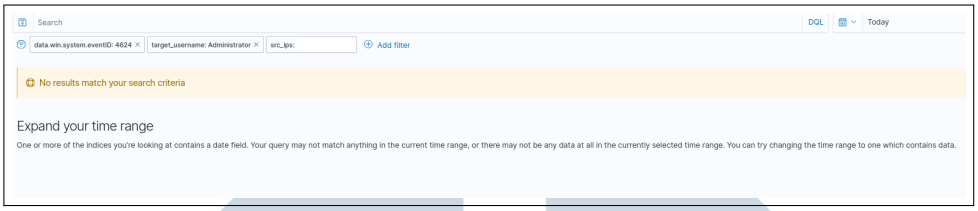
Dalam praktik operasional sehari-hari, *security analyst* turut menjalankan sebagian proses *incident response* sebagai *first layer* dalam alur penanganan insiden. *Security analyst* bertugas melakukan triase awal terhadap setiap *alert* yang

masuk, memberikan notifikasi *event* kepada *customer*, serta melakukan tindakan teknis cepat seperti *blocking malicious IP*, karantina malware di endpoint, atau memutus koneksi mencurigakan sebagai upaya *containment* awal. Langkah-langkah ini merupakan implementasi langsung dari fase *Detection & Analysis* serta *Containment* dalam kerangka *NIST SP 800-61*. Apabila insiden membutuhkan penanganan lebih mendalam, misalnya *root cause analysis*, *forensic investigation*, atau pemulihan sistem berskala luas, *security analyst* akan membuat tiket insiden melalui platform seperti *Jira* untuk diteruskan kepada tim *Incident Response* yang lebih spesialis. Dalam konteks ini, baik *security analyst* pada *first layer* maupun tim *Incident Response* di tingkat berikutnya menggunakan pendekatan *OODA loop* (*Observe, Orient, Decide, Act*) sebagai kerangka mikro untuk cara berpikir dan mengambil keputusan di setiap tahap dalam menganalisis dan menangani insiden. Berikut merupakan contoh alert Brute Force wazuh windows pada gambar 3.40 dan 3.41.



Gambar 3.40. Case Wazuh Windows, Logon failure - Unknown user or bad password.

Pada hasil pencarian log di *Elastic* pada *index wazuh* untuk kasus *brute force* ini, terlihat lebih dari seribu *hits* berupa deretan event 4625 yang terjadi dalam rentang waktu singkat pada tanggal yang sama. Seluruh event tersebut menunjukkan percobaan *logon* ke akun *Administrator* dengan *logonType = 3* (*network logon*) dan status 0xC000006D / 0xC000006A yang mengindikasikan kredensial salah, sehingga membentuk pola banyaknya kegagalan *login* berulang dari sumber yang sama yang konsisten dengan aktivitas serangan *brute force*.



Gambar 3.41. Case Wazuh Windows, Logon failure - Unknown user or bad password.

Setelah dilakukan korelasi menggunakan *event ID 4624* dengan filter *target username* Administrator dan sumber IP yang sama, tidak ditemukan log keberhasilan *login* yang sesuai dengan rangkaian kegagalan 4625 tersebut dan muncul pesan *No results match your search criteria*. Temuan ini menunjukkan bahwa meskipun terjadi percobaan *brute force* dalam jumlah besar, tidak ada percobaan yang berhasil masuk ke akun Administrator sehingga insiden ini tidak menyebabkan kompromi kredensial, tetapi aktivitas ini tetap dikategorikan berisiko tinggi dan perlu diwaspadai.

Untuk menjabarkan insiden ini secara lebih terstruktur, digunakan kerangka mikro metode OODA (Observe, Orient, Decide, Action) yang memetakan langkah-langkah pengambilan keputusan pada setiap fase penanganan insiden. Berikut merupakan hasil penerapan metode OODA dalam menguraikan tiap tahap *cycle model* incident response yang ditampilkan pada Gambar 3.42.

Gambar 3.42. OODA Method pada Incident Response

1. Observe

Pada tahap *Observe*, tim mengumpulkan *Windows Security Log* yang telah terintegrasi ke SIEM (Elastic) dan memfokuskan pencarian pada Event ID 4625 (*failed logon*). Dengan filter *targetUserName* "Administrator", *Source Network Address* dari salah satu IP internal, serta rentang waktu 18 November 2025, ditemukan sekitar 1.061 event gagal logon menuju salah satu server pada segmen produksi dengan *logonType* 3 (*network logon*) dan kombinasi status 0xC000006D/0xC000006A (akun valid, kata sandi salah). Pencarian lanjutan terhadap Event ID 4624 (*successful logon*) dengan filter akun dan IP yang sama pada rentang waktu serupa tidak menghasilkan temuan sehingga belum ada indikasi logon sukses dari sumber tersebut.

2. Orient

Pada tahap *Orient*, pola ini diinterpretasikan sebagai indikasi kuat serangan *brute force* terhadap akun *privilege* Administrator. Karakteristiknya adalah banyak percobaan kata sandi ke satu akun dari satu IP internal melalui akses jaringan (*Logon Type 3*), dan seluruh kegagalan bertipe *bad password*. Mengacu pada *playbook* investigasi, banyaknya Event 4625 dianggap sebagai *noise* atau deretan kegagalan biasa, sehingga fokus utama diarahkan pada pencarian Event 4624 yang muncul setelah rangkaian gagal tersebut. Event 4624 inilah yang berpotensi menjadi titik awal atau *patient zero* jika ada upaya login yang akhirnya berhasil. Korelasi dilakukan dengan menggabungkan *TargetUserName* dan *Source Network Address* untuk mencari pola "IP yang sama, sebelumnya menghasilkan banyak gagal (4625), kemudian muncul minimal satu sukses (4624)", baik di sisi host (menggunakan PowerShell *Get-WinEvent* dengan *parsing* XML untuk field seperti *TargetUserName*, *IpAddress*, *LogonType*, *SubStatus*) maupun di level SIEM melalui agregasi berdasarkan IP dan user. Pada kasus ini, pencarian tidak menemukan Event 4624 yang relevan sehingga belum ada bukti kompromi, namun risiko tetap dinilai tinggi karena sasaran merupakan akun dengan hak istimewa.

3. Decide

Berdasarkan hal tersebut, pada tahap *Decide* insiden diklasifikasikan sebagai *Security Incident – Brute-Force Network Logon* terhadap akun Administrator dengan prioritas tinggi. Rekomendasi utama meliputi penguatan akun

Administrator (penggantian kata sandi dengan kompleksitas tinggi, pembatasan penggunaan *built-in* Administrator untuk akses jaringan, serta penerapan kebijakan *account lockout* yang seimbang), pengamanan akses *remote* seperti RDP (tidak diekspos langsung ke internet, dipaksa melalui VPN atau RD Gateway dengan *Network Level Authentication* diaktifkan), serta peningkatan kemampuan deteksi di SIEM untuk pola *brute force*, *password spray*, dan kombinasi "gagal tinggi + satu sukses" pada akun sensitif. Untuk jangka panjang, lingkungan disarankan menerapkan *Fine-Grained Password Policy* bagi akun-akun kritikal, memanfaatkan fitur *smart lockout* bila tersedia, dan mengaktifkan *Multi-Factor Authentication* (MFA) untuk seluruh akses administrasi sehingga serangan berbasis kata sandi saja tidak lagi memadai.

4. Act

Pada tahap *Act*, tim melakukan *containment* dengan memblokir sementara IP internal yang menjadi sumber aktivitas. Secara operasional, tindakan ini umumnya dilakukan melalui pembuatan *ticket* ke tim pengelola perangkat keamanan (SDM) untuk melakukan pemblokiran IP pada firewall/ACL, atau melalui tiket ke tim *Incident Response* untuk berkoordinasi dengan pihak *customer* dalam penerapan aturan blokir di sisi jaringan mereka. Selain itu, kata sandi akun Administrator pada server yang menjadi sasaran turut diganti dan akses jaringan menggunakan akun tersebut dibatasi. Host sumber kemudian diidentifikasi dan dilakukan pemeriksaan *host-based* (proses aktif, *scheduled task*, skrip, serta pemindaian AV/EDR) untuk memastikan tidak ada *tool* atau malware *brute force* yang berjalan. Jika pada insiden lain ditemukan akun yang diduga terkompromi, akun tersebut dapat segera dinonaktifkan (misalnya melalui *Disable-ADAccount*) hingga proses forensik dan *credential reset* selesai. Seluruh temuan dan tindakan terdokumentasi dalam laporan insiden, dan dilakukan pemantauan ulang terhadap pola Event 4625/4624 di SIEM untuk memastikan serangan berhenti dan tidak berpindah ke akun atau sumber lain, sekaligus menjadi masukan pada fase *Post-Incident Activity* dalam siklus *NIST SP 800-61*.

F. Close Case Pada Elastic

Setelah *customer* memberikan konfirmasi bahwa insiden telah diremediasi atau aktivitas yang terdeteksi merupakan tindakan sah (*legitimate*), *security analyst*

akan menutup *case* pada Elastic berdasarkan *case ID* terkait. Saat melakukan *close case*, ditambahkan deskripsi singkat mengenai alasan penutupan, seperti hasil remediasi atau validasi aktivitas oleh *customer*. Proses ini menandakan bahwa investigasi telah selesai dan seluruh tindakan yang diperlukan telah dilakukan.

G. Daily Report

Daily report dibagi menjadi tiga bagian berdasarkan pembagian *shift*, yaitu *daily pagi*, *daily siang*, dan *daily malam*. Seluruh laporan dibuat dalam format Excel dengan menjalankan *script* otomatis melalui Google Colab agar data dapat direkap secara cepat dan konsisten setiap harinya.

Pada *daily pagi*, dilakukan rekapitulasi *Daily WAF Trend Attack* serta *Daily High and Critical Severity*, yang kemudian dikirimkan kepada *customer* melalui *SharePoint* setiap pukul 07.00 pagi. Untuk *daily siang*, laporan berisi rekap *Daily Fortigate Web Filter* yang menggambarkan aktivitas penyaringan web dan dikirimkan ke *customer* melalui Thunderbird atau Outlook setiap pukul 17.00 sore. Sementara itu, *daily malam* berfokus pada *Daily Checklist*, yang mencakup pengecekan performa sistem di Grafana, SolarWinds, Check Point Harmony, serta *Daily Checklist Health* yang melampirkan status kesehatan koneksi VPN yang dikirimkan melalui Thunderbird atau Outlook setiap pukul 05.00 pagi.

Ketiga *daily report* ini bertujuan untuk memastikan seluruh aktivitas monitoring dan status sistem keamanan tercatat secara rutin, menjaga transparansi antar *shift*, serta memberikan pembaruan berkala kepada *customer* mengenai kondisi keamanan dan performa infrastruktur secara menyeluruh.

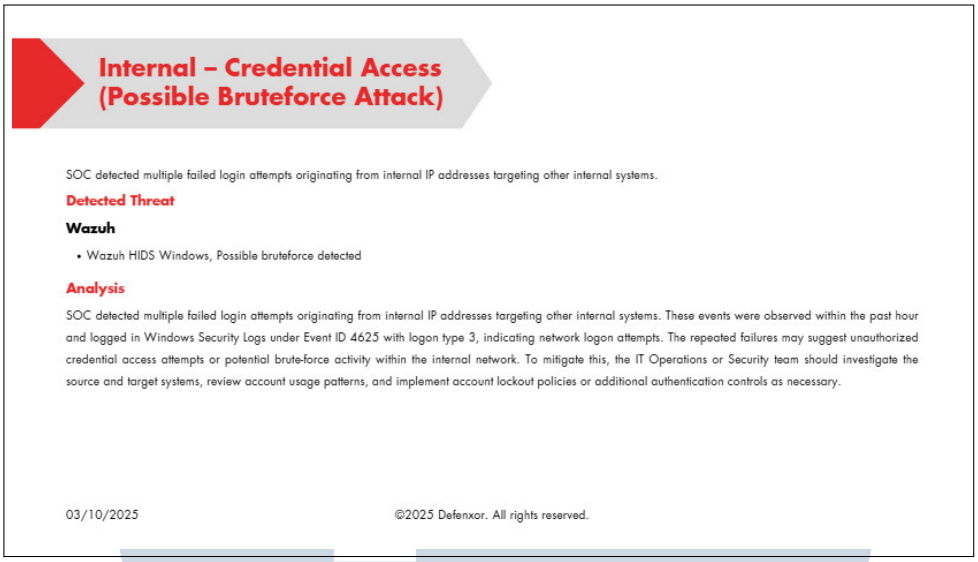
H. Monthly Report

Monthly report disusun pada akhir setiap bulan sebagai *output* dari ringkasan laporan keamanan (*security report*) selama periode berjalan. Kegiatan ini mencakup penarikan data satu bulan terakhir dengan keluaran lembar kerja Excel (misalnya dari FortiGate Web Attack, Palo Alto Threat, Cloud Armor, dan sumber relevan lainnya), serta penyusunan presentasi *PowerPoint* yang memuat *executive summary*. *Executive summary* tersebut pada umumnya menampilkan metrik inti, antara lain total *raw events* bulanan, jumlah *identified threats, valid threats* yang dibedakan menjadi eksternal dan internal, *notified cases*, serta *security incidents*.

Laporan juga menyajikan *Threat Classification Overview* yang

mengklasifikasikan aktivitas ancaman berdasarkan kerangka taktik MITRE ATT&CK. Beberapa taktik yang lazim terdeteksi meliputi: (1) *Credential Access*, yakni upaya penyerang untuk menangkap atau mencuri kredensial pengguna (username, password, token, atau hash) guna memperoleh akses tidak sah; (2) Discovery, yaitu kegiatan pengumpulan informasi mengenai lingkungan target meliputi akun pengguna, konfigurasi jaringan, layanan, dan perangkat terhubung sebagai dasar perencanaan langkah lanjutan; (3) Defense Evasion, yaitu upaya menyamarkan aktivitas berbahaya atau memodifikasi konfigurasi sistem agar luput dari deteksi alat keamanan; (4) Execution, yakni eksekusi perintah, skrip, atau berkas berbahaya pada sistem target untuk memulai tindakan seperti instalasi malware atau manipulasi sistem; (5) Impact, yaitu tindakan yang mengganggu, memanipulasi, atau merusak sistem, layanan, atau data sehingga menimbulkan downtime, kehilangan data, atau kerugian finansial; (6) Initial Access, yakni perolehan akses awal ke sistem, misalnya melalui phishing, eksploitasi layanan yang menghadap internet, atau pemanfaatan kerentanan yang telah diketahui; dan (7) Lateral Movement, yaitu perpindahan dari satu sistem ke sistem lain di dalam jaringan untuk memperluas cakupan akses atau meningkatkan privilege, sering kali dengan memanfaatkan kredensial curian atau alat bawaan sistem.

Untuk setiap kategori, dipilih setidaknya satu kasus yang dinotifikasi pada bulan tersebut dan dilengkapi evidence sebagai bukti serta dokumentasi teknis. Sebagai contoh, ancaman "Wazuh HIDS Windows Possible brute force detected" dapat diklasifikasikan sebagai kategori internal - credential access. Setiap kasus disajikan secara sistematis mencakup analisis peristiwa, penilaian dampak terhadap layanan maupun data, rekomendasi remediasi yang diberikan kepada pelanggan, serta catatan pendukung yang relevan. Berikut merupakan contoh presentasi monthly report yang disampaikan kepada *customer* ditunjukkan pada Gambar 3.43 dan 3.44.



Internal – Credential Access (Possible Bruteforce Attack)

SOC detected multiple failed login attempts originating from internal IP addresses targeting other internal systems.

Detected Threat

Wazuh

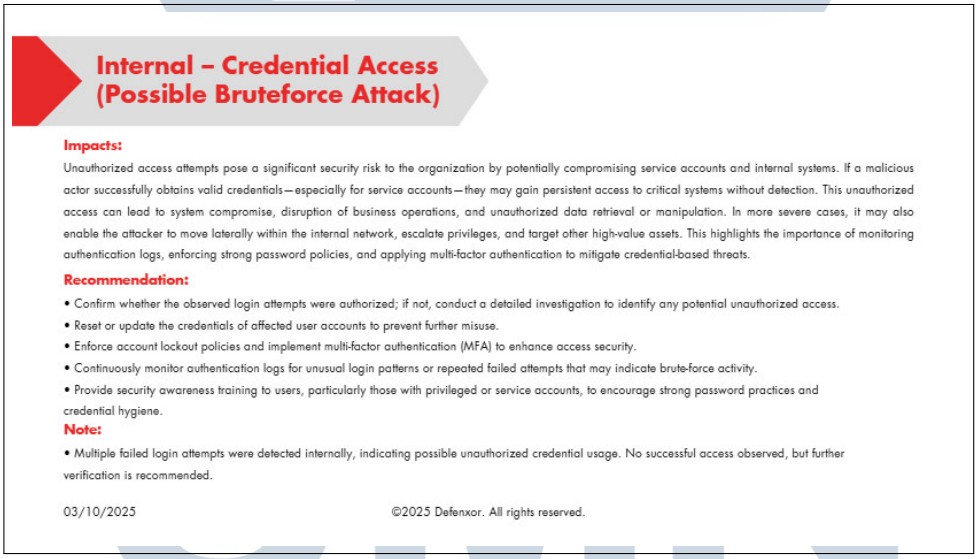
- Wazuh HIDS Windows, Possible bruteforce detected

Analysis

SOC detected multiple failed login attempts originating from internal IP addresses targeting other internal systems. These events were observed within the past hour and logged in Windows Security Logs under Event ID 4625 with logon type 3, indicating network logon attempts. The repeated failures may suggest unauthorized credential access attempts or potential brute-force activity within the internal network. To mitigate this, the IT Operations or Security team should investigate the source and target systems, review account usage patterns, and implement account lockout policies or additional authentication controls as necessary.

03/10/2025 ©2025 Defenxor. All rights reserved.

Gambar 3.43. Contoh Presentasi Monthly Report



Internal – Credential Access (Possible Bruteforce Attack)

Impacts:

Unauthorized access attempts pose a significant security risk to the organization by potentially compromising service accounts and internal systems. If a malicious actor successfully obtains valid credentials—especially for service accounts—they may gain persistent access to critical systems without detection. This unauthorized access can lead to system compromise, disruption of business operations, and unauthorized data retrieval or manipulation. In more severe cases, it may also enable the attacker to move laterally within the internal network, escalate privileges, and target other high-value assets. This highlights the importance of monitoring authentication logs, enforcing strong password policies, and applying multi-factor authentication to mitigate credential-based threats.

Recommendation:

- Confirm whether the observed login attempts were authorized; if not, conduct a detailed investigation to identify any potential unauthorized access.
- Reset or update the credentials of affected user accounts to prevent further misuse.
- Enforce account lockout policies and implement multi-factor authentication (MFA) to enhance access security.
- Continuously monitor authentication logs for unusual login patterns or repeated failed attempts that may indicate brute-force activity.
- Provide security awareness training to users, particularly those with privileged or service accounts, to encourage strong password practices and credential hygiene.

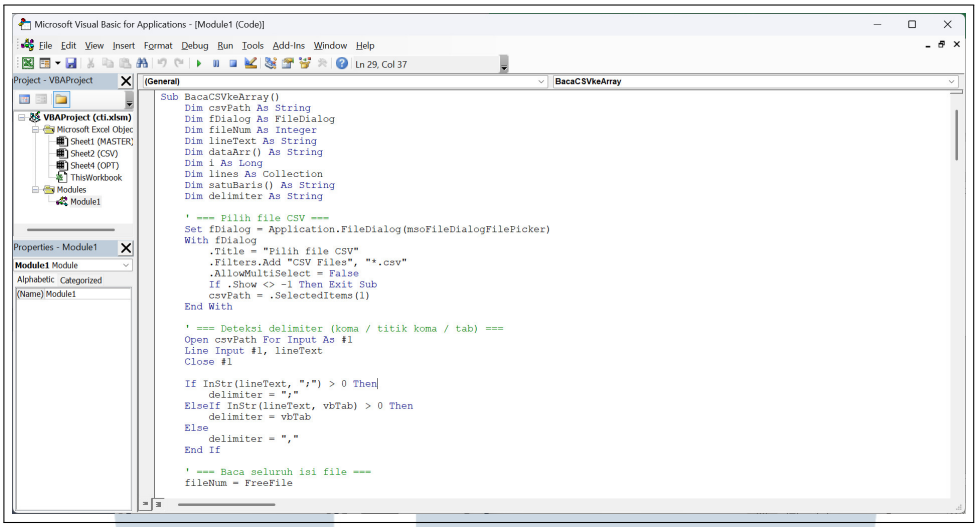
Note:

- Multiple failed login attempts were detected internally, indicating possible unauthorized credential usage. No successful access observed, but further verification is recommended.

03/10/2025 ©2025 Defenxor. All rights reserved.

Gambar 3.44. Contoh Presentasi Monthly Report

Selain contoh dalam bentuk presentasi PowerPoint, *ouput* laporan juga mencakup data dalam format *Excel*. Berkas tersebut dihasilkan melalui eksekusi script macro VBA di *Excel* untuk memastikan proses yang efisien serta konsistensi format laporan. Berikut merupakan visualisasi script yang diimplementasi VBA pada gambar 3.45.



Gambar 3.45. Contoh Script VBA untuk Monthly Report

Salah satu *file Excel* yang diekspor dari Elastic adalah SIEM Alarm yang memuat keseluruhan log pada bulan pelaporan. Ekstraksi dilakukan melalui *dashboard Raw Event* sehingga terbentuk tabel dengan kolom *Title, Src_IPs, Dst_IPs, Risk_Class, Status, Tag*, dan *Category*, di mana *Category* dikelompokkan berdasarkan taktik MITRE ATT&CK. Pada periode ini tercatat total 3.553 *raw event*; sebanyak 3.294 di antaranya terverifikasi sebagai ancaman valid, sementara 259 diklasifikasikan sebagai *false positive*. Visualisasi ringkas dari SIEM Alarm disajikan pada Gambar 3.46.

title	src_ips	dst_ips	risk_class	status	tag	category
ATT&CK T1023: Suspicious desktop.ini Action	x.x.x.x	x.x.x.x	High	Open	False Positive	Persistence
ATT&CK T1023: Suspicious desktop.ini Action	x.x.x.x	x.x.x.x	High	Open	False Positive	Persistence
ATT&CK T1023: Suspicious desktop.ini Action	x.x.x.x	x.x.x.x	High	Open	False Positive	Persistence
ATT&CK T1027 S0139: Executable in ADS	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	Medium	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	Medium	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	Medium	Open	False Positive	Defense Evasion
ATT&CK T1036: File Created with System Process Name	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1038: Windows Registry Persistence COM Search Order Hijacking	x.x.x.x	x.x.x.x	High	Open	False Positive	Persistence
ATT&CK T1038: Windows Registry Persistence COM Search Order Hijacking	x.x.x.x	x.x.x.x	High	Open	False Positive	Persistence
ATT&CK T1117: BlueMashroom DLL Load	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
ATT&CK T1117: BlueMashroom DLL Load	x.x.x.x	x.x.x.x	High	Open	False Positive	Defense Evasion
						Credential

Gambar 3.46. Contoh Raw Evenet Pada SIEM Alarm

Selanjutnya, data tersebut diolah ke dalam *pivot table* untuk memperoleh jumlah (*count of Title*) pada setiap kategori, sehingga memberikan gambaran distribusi ancaman per kategori taktik MITRE ATT&CK serta membantu prioritisasi tindakan mitigasi. Hasilnya ditunjukkan pada Gambar 3.47.

category	COUNT of title
Active Scanning	94
Antivirus	1062
Exploit Public-Facing Application	1
Malware Infection	8
Network Service Scanning	13
Valid Accounts	3
wazuh alert level 5	104
wazuh alert level 9	77
Web Scanning and Attack	1
Grand Total	**1363**

Gambar 3.47. Contoh Pivot Pada SIEM Alarm

Selain itu juga terdapat laporan Recap Event yang digunakan untuk merangkum seluruh notifikasi alert secara terstruktur. Setiap alert yang dinotifikasi terlebih dahulu dicatat sebagai entri awal, kemudian seluruh entri tersebut dimasukkan ke dalam action tracking untuk dipantau secara sistematis, meliputi status penanganan, penanggung jawab, linimasa, rekomendasi, serta evidensi. Berdasarkan keluaran dari action tracking tersebut, disusunlah Recap Event sebagai rekapitulasi yang lebih rinci dan menjadi bahan utama dalam penyusunan resume laporan bulanan. Recap Event memuat kolom informasi seperti Case ID, Category, Threat/Issue, Affected IP, Hostname, Network, Risk, Case Status (On-Progress/Closed), Business Unit, Source/Destination IP, Threat Source, dan Detection, sehingga setiap kasus dapat ditelusuri secara komprehensif. Selain lembar all cases, Recap Event juga dilengkapi sheet terpisah per regional yang menyajikan agregasi notifikasi per wilayah untuk mendukung prioritisasi serta evaluasi penanganan. Visualisasi Recap Event ditampilkan pada Gambar 3.48.

No	Case ID	Category	Threat/Issue	Affected IP	Hostname	Network	Risk	Case Status (On-Progress/ Closed)	Bussiness Unit	Source/Destination IP	Threat Source	Detection
1	1020250001	Execution	KSC, Disinfection impossible (VHO:Trojan.Win32.Exnet.gen)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
2	1020250002	Execution	KSC, Disinfection impossible (Trojan.Multi.Accesstr.aum)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
3	1020250003	Execution	KSC, Disinfection impossible (VHO:Trojan.Win32.Exnet.gen)	x.x.x.x	X	X	Low	closed	C		Internal	KSC
4	1020250004	Execution	KSC, Disinfection impossible (Virus.Win32.Pioneer.cz)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
5	1020250005	Execution	KSC, Cannot be deleted (HEUR:Trojan.Win32.Generic)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
6	1020250006	Execution	KSC, Disinfection impossible (Virus.Win32.Renamer.j)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
7	1020250007	Execution	KSC, Disinfection impossible (HEUR:Trojan.Win32.Generic)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
8	1020250008	Initial Access	NIDS, SUSPICIOUS SMTP EXE - EXE SMTP Attachment	x.x.x.x	X	X	Low	closed	B	x.x.x.x	External	Suricata
9	1020250009	Execution	KSC, Disinfection impossible (HEUR:Trojan.Win32.Generic)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
10	1020250010	Execution	KSC, The object cannot be deleted (PDM:Trojan.Win32.Generic)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
11	1020250011	Execution	KSC, Cannot be deleted (avp.exe, HEUR:Trojan.Win32.Generic)	x.x.x.x	X	X	Medium	closed	A		Internal	KSC
12	1020250012	Execution	KSC, Cannot be deleted (UDS:Trojan.Win32.Inject.aqfhl)	x.x.x.x	X	X	Medium	closed	A		Internal	KSC
13	1020250013	Credential Access	Wazuh HIDS Windows, Possible bruteforce detected	x.x.x.x	X	X	Low	closed	A	x.x.x.x	Internal	Wazuh
14	1020250014	Execution	KSC, Disinfection impossible (HEUR:Trojan-Dropper.MC.Convagent.gen)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
15	1020250015	Credential Access	Wazuh HIDS Windows, Possible bruteforce detected	x.x.x.x	X	X	Low	closed	A	x.x.x.x	Internal	Wazuh
16	1020250016	Execution	KSC, Disinfection impossible (Trojan.Multi.Accesstr.aum)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
17	1020250017	Impact	Cloudarmor, Possible DoS Attack Detected	x.x.x.x	X	X	Medium	closed	A	x.x.x.x	External	Cloud Armor
18	1020250018	Impact	Cloudarmor, Possible DoS Attack Detected	x.x.x.x	X	X	Medium	closed	A	x.x.x.x	External	Cloud Armor
19	1020250019	Impact	Cloudarmor, Possible DoS Attack Detected	x.x.x.x	X	X	Medium	closed	A	x.x.x.x	External	Cloud Armor
20	1020250020	Execution	KSC, Disinfection impossible (Trojan.Multi.Accesstr.aum)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
21	1020250021	Execution	KSC, Disinfection impossible (HEUR:IM-Worm.Win32.Chydo.gen)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
22	1020250022	Execution	KSC, Disinfection impossible (HEUR:Trojan.Win32.Generic)	x.x.x.x	X	X	Low	closed	A		Internal	KSC
23	1020250023	Initial Access	Cloud Armor, Unblocked PHP Injection Attack Detected	x.x.x.x	X	X	Low	closed	A	x.x.x.x	External	Cloud Armor
24	1020250024	Credential Access	Wazuh HIDS Windows, Possible bruteforce detected	x.x.x.x	X	X	Low	closed	A	x.x.x.x	Internal	Wazuh
25	1020250025	Execution	KSC, Disinfection impossible (Trojan.Multi.Accesstr.aum)	x.x.x.x	X	X	Low	closed	A		Internal	KSC

Gambar 3.48. Contoh Laporan Recap Event

Untuk setiap perangkat keamanan umumnya disusun menjadi laporan Excel terpisah yang disesuaikan dengan kebutuhan serta *request customer*. Tujuan utamanya adalah memberi visibilitas ringkas atas pola serangan per perangkat, mengidentifikasi sumber/target serta jenis serangan yang paling dominan, dan menilai efektivitas kontrol (*blocked vs unblocked*) agar mitigasi serta *tunning* kebijakan bisa diprioritaskan. Umumnya laporan per perangkat ini mencakup seluruh *raw event* pada bulan tersebut, sebagai contoh laporan untuk *FortiGate Web Attack* pada bulan Oktober mencatat total 23.488 log. Visualisasi dapat dilihat pada gambar 3.49.

@timestamp: Descending	Source IP	Source Country	Target IP	Event Name	Severity	Action	Count
Oct 4, 2025 @ 17:16:55.995	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	14
Oct 4, 2025 @ 17:16:55.995	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	11
Oct 4, 2025 @ 17:16:55.995	x.x.x.x	Singapore	x.x.x.x	Apache.Struts.2.OGNL.Script.Injection	critical	dropped	5
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	9
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	7
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Apache.Struts.2.OGNL.Script.Injection	critical	dropped	4
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Weaver.E-Cology.Beanshell.Remote.Code.Execution	high	dropped	2
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	HTTP.URI.Java.Code.Injection	critical	dropped	1
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Oracle.Forms.And.Reports.Remote.Code.Execution	high	dropped	1
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	PHP.CGI.Argument.Injection	high	dropped	1
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	Web.Server.Password.File.Access	high	dropped	1
Oct 4, 2025 @ 08:56:22.597	x.x.x.x	Singapore	x.x.x.x	WordPress.Creative.Contact.Form.Arbitrary.File.Upload	critical	dropped	1
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	10
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	9
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	Apache.Struts.2.OGNL.Script.Injection	critical	dropped	3
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	Web.Server.Password.File.Access	high	dropped	2
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	HTTP.GET.Request.Path.Traversal	medium	dropped	1
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	HTTP.URI.Java.Code.Injection	critical	dropped	1
Oct 4, 2025 @ 17:16:54.499	x.x.x.x	Singapore	x.x.x.x	PHP.CGI.Argument.Injection	high	dropped	1
Oct 10, 2025 @ 10:57:46.681	x.x.x.x	Cambodia	x.x.x.x	AndroxGh0st.Malware	high	dropped	25
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	8
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	8
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	Apache.Struts.2.OGNL.Script.Injection	critical	dropped	3
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	Web.Server.Password.File.Access	high	dropped	2
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	PHP.CGI.Argument.Injection	high	dropped	1
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	Weaver.E-Cology.Beanshell.Remote.Code.Execution	high	dropped	1
Oct 4, 2025 @ 17:14:01.662	x.x.x.x	Singapore	x.x.x.x	anji-plus.AJ-Report.swagger-ui.Authentication.Bypass	critical	dropped	1
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	9
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	Weaver.E-Cology.Beanshell.Remote.Code.Execution	high	dropped	4
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	Web.Server.Password.File.Access	high	dropped	4
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	2
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	HTTP.GET.Request.Path.Traversal	medium	dropped	1
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	HTTP.URI.Java.Code.Injection	critical	dropped	1
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	PHP.CGI.Argument.Injection	high	dropped	1
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	WordPress.Creative.Contact.Form.Arbitrary.File.Upload	critical	dropped	1
Oct 4, 2025 @ 17:20:21.841	x.x.x.x	Singapore	x.x.x.x	anji-plus.AJ-Report.swagger-ui.Authentication.Bypass	critical	dropped	1
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	Remote.CMD.Shell	critical	dropped	8
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	5
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	Web.Server.Password.File.Access	high	dropped	4
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	Apache.Struts.2.OGNL.Script.Injection	critical	dropped	3
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	PHP.CGI.Argument.Injection	high	dropped	1
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	Weaver.E-Cology.Beanshell.Remote.Code.Execution	high	dropped	1
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	WordPress.Creative.Contact.Form.Arbitrary.File.Upload	critical	dropped	1
Oct 4, 2025 @ 18:42:35.046	x.x.x.x	Singapore	x.x.x.x	anji-plus.AJ-Report.swagger-ui.Authentication.Bypass	critical	dropped	1
Oct 4, 2025 @ 08:56:14.563	x.x.x.x	Singapore	x.x.x.x	Bash.Function.Definitions.Remote.Code.Execution	critical	dropped	9

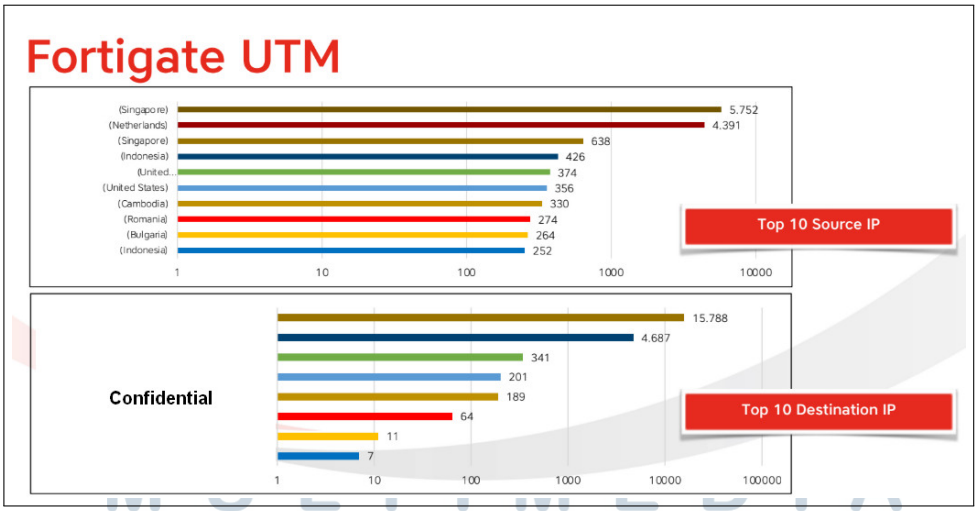
Gambar 3.49. Contoh Laporan Fortigate Web Attack

Data tersebut kemudian diolah melalui pivot untuk mengekstraksi Top 10 (*source IP, destination IP, blocked attack*, dan *unblocked attack*) serta menyusun ringkasan yang memuat jumlah (*count*) berdasarkan *action* dan *severity*. Berikut merupakan contoh dari laporan *FortiGate Web Attack* untuk *log raw event* dan salah satu contoh *pivot* dari *Top 10 Source IP* pada gambar 3.50.

Top 10 Source IP	Source Country	Total	IP (Country)
x.x.x.x	Singapore	5,752	x.x.x.x (Singapore)
x.x.x.x	Netherlands	4,391	x.x.x.x (Netherlands)
x.x.x.x	Singapore	638	x.x.x.x (Singapore)
x.x.x.x	Indonesia	426	x.x.x.x (Indonesia)
x.x.x.x	United States	374	x.x.x.x (United States)
x.x.x.x	United States	356	x.x.x.x (United States)
x.x.x.x	Cambodia	330	x.x.x.x (Cambodia)
x.x.x.x	Romania	274	x.x.x.x (Romania)
x.x.x.x	Bulgaria	264	x.x.x.x (Bulgaria)
x.x.x.x	Indonesia	252	x.x.x.x (Indonesia)

Gambar 3.50. Contoh Pivot Top 10 Source IP

Kemudian, hasil pivot tersebut dikonversi ke dalam bentuk bar chart untuk memudahkan visualisasi, sehingga skor IP dengan nilai tertinggi serta jumlahnya dapat terlihat secara jelas, khususnya untuk top 10 source IP maupun destination IP. Berikut contoh dari visualisasi bar chart pada gambar 3.51.



Gambar 3.51. Contoh Bar Chart Top 10 Source IP

3.4 Kendala dan Solusi yang Ditemukan

3.4.1 Kendala

Selama pelaksanaan kegiatan magang, terdapat berbagai kendala yang muncul dalam proses pekerjaan, baik bersifat ringan maupun cukup kompleks. Beberapa di antaranya memerlukan penyesuaian teknis maupun koordinasi lintas tim untuk diselesaikan. Berikut merupakan beberapa kendala yang dihadapi selama pelaksanaan magang di PT Defender Nusa Semesta.

1. Waktu yang tersedia untuk menyelesaikan *daily report* relatif terbatas, sehingga diperlukan pengelolaan tugas yang *automated* agar dapat membantu meringankan pekerjaan *security analyst*.
2. Terdapat banyak *alert* dari SOCRadar pada Elastic yang tidak ter-*parsing* dengan baik, sehingga data terlihat tidak terstruktur dan menimbulkan *alert bombing*.
3. Aplikasi Thunderbird dan Outlook sering mengalami kegagalan dalam proses pengiriman email akibat kapasitas *disk memory* yang penuh.

3.4.2 Solusi

Berbagai kendala yang muncul selama pelaksanaan magang berhasil diatasi melalui sejumlah langkah perbaikan. Berikut merupakan beberapa solusi yang diterapkan untuk menangani permasalahan yang terjadi selama kegiatan magang di PT Defender Nusa Semesta.

1. Dibuat *script* Python di Google Colab untuk mengotomatisasi pembuatan *daily report*, sehingga hasilnya konsisten, mengurangi kesalahan manual, dan lebih efisien waktu.
2. Dikembangkan *script* Python di Google Colab yang merangkum kredensial penting ke dalam bentuk tabel untuk mempermudah uji kebocoran kredensial pada situs, serta diberikan akses ke dashboard SOCRadar agar tim dapat langsung melihat *approved* dan *potential alarms* serta melakukan *takedown* atau *close* alarm.

3. Diterapkan *centralized rules* pada laptop SOC yang disediakan khusus untuk *backup email*, termasuk konfigurasi *auto-archive* dan *auto-delete* berdasarkan filter rules untuk mengelola lalu lintas *email* masuk.