PERANCANGAN NEXT-GENERATION FIREWALL (NGFW) TERINTEGRASI DENGAN SIEM

SKRIPSI

CORNELIUS YULI ROSDIANTO

2207421059

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER POLITEKNIK NEGERI JAKARTA
2026

PERANCANGAN NEXT-GENERATION FIREWALL (NGFW) TERINTEGRASI DENGAN SIEM

SKRIPSI

Dibuat untuk Melengkapi Syarat-Syarat yang Diperlukan untuk Memperoleh Diploma Empat Politeknik

CORNELIUS YULI ROSDIANTO

2207421059

PROGRAM STUDI TEKNIK MULTIMEDIA DAN JARINGAN
JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER POLITEKNIK NEGERI JAKARTA
2026

SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertanda tangan di bawah ini:

Nama : Cornelius Yuli Rosdianto

NIM : 2207421059

Jurusan/Program Studi : Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

Judul Skripsi : Perancangan Next-Generation Firewall (NGFW) Terintegrasi Dengan SIEM

Menyatakan dengan sebenarnya bahwa skripsi ini benar-benar merupakan hasil karya saya sendiri, bebas dari peniruan terhadap karya dari orang lain. Kutipan pendapat dan tulisan orang lain ditunjuk sesuai dengan cara-cara penulisan karya ilmiah yang berlaku.

Apabila di kemudian hari terbukti atau dapat dibuktikan bahwa dalam skripsi ini terkandung ciri-ciri plagiat dan bentuk-bentuk peniruan lain yang dianggap melanggar peraturan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Depok, 9 Juni 2026

Yang membuat pernyataan

Cornelius Yuli Rosdianto

NIM. 2207421059

LEMBAR PENGESAHAN

Skripsi ini diajukan oleh

Nama	: Cornelius Yuli Rosdianto
NIM	: 2207421059
Jurusan/Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia
Judul Skripsi	: Perancangan Next-Generation Firewall (NGFW) Terintegrasi Dengan SIEM

Telah diuji oleh tim penguji dalam Sidang Skripsi pada hari Senin., Tanggal 22, Bulan ..Juni....., Tahun 2026 dan dinyatakan **LULUS.**

Disahkan oleh

Pembimbing I : Iik Muhamad Malik Matin, S.Kom, M.T. (.....)

Penguji I : Ayu Rosyida Zain, S.ST, M.T (.....)

Penguji II : Fachroni Arbi Murad, S.Kom., M.Kom (.....)

Penguji III : Chandra Wirawan, S.Kom.,M.Kom. (.....)

Mengetahui:
Jurusan Teknik Informatika dan Komputer
Ketua

Dr. Anita Hidayati, S.Kom., M.Kom.
NIP. 197908032003122003

KATA PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa atas segala rahmat, karunia, dan penyertaan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Perancangan Next-Generation Firewall (NGFW) Terintegrasi Dengan SIEM" dengan baik. Dalam proses penyusunan skripsi ini, penulis memperoleh banyak bantuan, dukungan, arahan, serta masukan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan terima kasih kepada:

- Tuhan Yang Maha Esa atas kesempatan dan kemudahan yang diberikan sehingga penulisan skripsi ini dapat diselesaikan tepat waktu.
- Bapak Iik Muhamad Malik Matin, S.Kom, M.T., selaku Dosen Pembimbing, yang telah memberikan arahan dan masukan selama proses penyusunan skripsi.
- Kedua orang tua dan keluarga yang senantiasa memberikan doa, dukungan, serta motivasi kepada penulis.
- Teman-teman penulis terkhusus TMJ dan rekan-rekan selama kegiatan magang yang telah memberikan dukungan dan inspirasi selama proses penyusunan skripsi.

Penulis menyadari bahwa skripsi ini masih memiliki berbagai keterbatasan dan kekurangan. Oleh karena itu, kritik serta saran sangat diterima. Semoga skripsi ini dapat memberikan manfaat dan menambah wawasan bagi pembaca

Depok, 09 Juni 2026

Cornelius Yuli Rosdianto

SURAT PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Politeknik Negeri Jakarta, saya bertanda tangan dibawah ini :

Nama	: Cornelius Yuli Rosdianto
NIM	: 2207421059
Jurusan/Program Studi	: Teknik Informatika dan Komputer / Teknik Multimedia dan Jaringan

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Politeknik Negeri Jakarta Hak Bebas Royalti Non-Eksklusif atas karya ilmiah saya yang berjudul :

Perancangan Next-Generation Firewall (NGFW) Terintegrasi Dengan SIEM

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksklusif ini Politeknik Negeri Jakarta Berhak menyimpan, mengalihmediakan/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyatan ini saya buat dengan sebenarnya.

Depok, ...9...Juni....2026

Yang menyatakan

(Cornelius Yuli Rosdianto)

NIM. 2207421059

PERANCANGAN NEXT-GENERATION FIREWALL (NGFW) TERINTEGRASI DENGAN SIEM

ABSTRAK

Peningkatan jumlah serangan siber menuntut organisasi untuk menerapkan sistem keamanan yang mampu mendeteksi, mencegah, dan memantau aktivitas berbahaya secara efektif. Namun, solusi keamanan komersial seperti Next-Generation Firewall (NGFW) dan Security Information and Event Management (SIEM) umumnya memerlukan biaya implementasi dan lisensi yang tinggi. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem keamanan berbasis open source menggunakan pfSense sebagai NGFW yang diintegrasikan dengan Suricata IPS, Zenarmor, dan Wazuh SIEM. Implementasi dilakukan pada lingkungan cloud Microsoft Azure menggunakan Virtual Machine. Suricata digunakan untuk mendeteksi dan memblokir serangan berbasis signature, sedangkan Zenarmor digunakan untuk menerapkan Application Control dan Web Control. Seluruh log keamanan diintegrasikan ke dalam Wazuh SIEM. Hasil penelitian menunjukkan bahwa sistem berhasil diimplementasikan dan diintegrasikan dengan baik. Pada pengujian SQL Injection dan Port Scanning, Suricata IPS mampu mendeteksi dan memblokir aktivitas serangan sebelum mencapai Web Server. Zenarmor juga berhasil memblokir akses menuju domain berbahaya dan website yang dibatasi berdasarkan kebijakan keamanan. Hasil pengujian Quality of Service (QoS) menunjukkan bahwa implementasi NGFW menyebabkan penurunan throughput dari 919,7 Mbps menjadi 879,3 Mbps atau sebesar 4,4% pada kondisi normal. Sementara itu nilai delay meningkat dari 1,533 ms menjadi 1,648 ms, jitter meningkat dari 0,017 ms menjadi 0,021 ms, sedangkan packet loss tetap berada pada 0%. Hasil tersebut menunjukkan bahwa implementasi sistem keamanan memberikan dampak terhadap kualitas layanan jaringan dan namun tetap berada pada kategori Sangat Bagus berdasarkan standar TIPHON.

Kata Kunci: Next-Generation Firewall, pfSense, Zenarmor, Wazuh, SIEM

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

1.1 Latar Belakang Masalah

Perkembangan teknologi informasi dan komunikasi di Indonesia telah mengalami peningkatan dalam beberapa dekade terakhir. Hal ini mengubah cara masyarakat dalam berinteraksi, belajar, dan menjalankan pekerjaannya sehari-hari. Berdasarkan data BPS pada hasil pendataan Survei Sosial Ekonomi Nasional (Susenas) 2024, jumlah pengguna internet terus meningkat secara konsisten setiap tahunnya. Pada tahun 2023, tercatat 69,21 persen penduduk Indonesia telah mengakses Internet dan meningkat pada 2024 menjadi 72,28 persen. Peningkatan ini mengindikasikan bahwa penggunaan internet tidak hanya sekadar fasilitas tambahan, melainkan telah menjadi kebutuhan primer yang menopang kehidupan sehari-hari.

Peningkatan dan transformasi ini tidak hanya membawa kemajuan dan kemudahan bagi banyak orang, namun juga membawa risiko dan ancaman baru dengan ikut berkembangnya serangan siber. Pada tahun 2024, Microsoft melaporkan bahwa pelanggan mereka menerima 600 juta serangan setiap harinya. Serangannya mulai dari ransomware, phising, hingga DDoS yang meningkat setiap tahunnya. Peningkatan serupa dilaporkan oleh Check Point pada laporan Q3 2024 yang mencatat jumlah serangan pada perusahaan meningkat 75% dari periode 2023 dan meningkat 15% dari kuartal sebelumnya.

Penggunaan firewall merupakan salah satu mekanisme untuk mengatasi banyaknya serangan dan melindungi aset organisasi. Firewall tradisional menggunakan teknik *packet filtering* yang bekerja pada Layer 3 (*Network*) dan Layer 4 (*Transport*) dari model OSI, di mana pemblokiran trafik dilakukan berdasarkan alamat IP sumber, alamat IP tujuan, dan port yang digunakan. Tujuannya adalah untuk memblokir trafik menuju alamat IP yang sudah diketahui *malicious* ataupun untuk membatasi akses dari internet menuju internal. Meskipun efektif, teknik ini memiliki keterbatasan dalam mendeteksi serangan modern ataupun kompleks. Penyerang sering kali menyisipkan *payload* berbahaya menggunakan port umum misalnya port 80 atau 443 yang biasanya digunakan organisasi untuk keperluan bisnis sehingga

diizinkan oleh firewall. Akibatnya serangan tersebut dapat masuk tanpa terdeteksi ataupun terblokir. Firewall tradisional tidak memiliki visibilitas terhadap isi paket, sehingga tidak mampu mengenali *malware* atau *payload* yang tersembunyi di balik lalu lintas jaringan yang tampak normal.

Untuk mengatasi kekurangan itu, banyak organisasi menggunakan firewall yang lebih modern atau NGFW. NGFW memiliki kemampuan *Deep Packet Inspection* (DPI) yang memungkinkan inspeksi paket hingga ke Layer 7 (*Application*). Selain mendeteksi pola serangan melalui *Intrusion Prevention System* (IPS), NGFW juga mampu mengidentifikasi aplikasi yang digunakan pengguna melalui fitur *Application Control* serta mengendalikan akses ke situs web berdasarkan kategori melalui fitur *Web Control.* Hal tersebut semakin penting karena serangan siber modern tidak berhenti pada tahap eksploitasi awal. Setelah berhasil memperoleh akses ke sistem, penyerang umumnya melakukan aktivitas lanjutan seperti mengunduh *malware*, memasang *backdoor*, melakukan komunikasi dengan *Command and Control* (C2) server, ataupun memanfaatkan aplikasi *remote access* untuk mempertahankan akses ke sistem korban. Selain firewall, organisasi juga memerlukan visibilitas mengenai apa yang terjadi di dalam jaringan untuk melakukan respons insiden yang efektif. Untuk itu NGFW perlu diintegrasikan dengan SIEM. *Security Information and Event Management* (SIEM) mampu menyatukan dan mengolah data log dari berbagai infrastruktur TI. Log dari perangkat keamanan diintegrasikan ke dalam satu dashboard untuk dilakukan analisis lanjutan oleh tim keamanan organisasi.

Namun, beberapa penelitian terkait masih berfokus pada fungsi deteksi dan monitoring serangan. Matin et al. (2025) berhasil membangun sistem deteksi dan monitoring menggunakan tools *open-source*, namun implementasi yang dilakukan masih terbatas pada proses deteksi dan pemantauan ancaman tanpa mekanisme pencegahan secara langsung. Hal serupa ditemukan pada penelitian Hafizh (2024) yang mengintegrasikan Snort dengan SIEM untuk melakukan monitoring keamanan jaringan, namun sistem yang dibangun hanya berfungsi mendeteksi serangan yang terjadi. Berbeda dengan kedua penelitian tersebut, Wiguna (2024) mengimplementasikan mekanisme *blacklist* IP setelah serangan terdeteksi. Meskipun mampu melakukan pemblokiran, arsitektur yang digunakan

menyebabkan adanya jeda waktu sehingga paket atau payload berbahaya masih dapat mencapai server target sebelum proses *blacklist* IP berhasil dilakukan. Sementara itu, Sulaiman, Seta, dan Falih (2021) berhasil mengembangkan sistem yang mampu merespons serangan secara otomatis, namun pengujiannya masih terbatas pada perangkat printer dan belum terintegrasi dengan SIEM sehingga aktivitas keamanan yang terjadi belum dapat dipantau secara terpusat.

Sebagai pembeda dari penelitian sebelumnya, penelitian ini mengimplementasikan sistem keamanan jaringan berbasis *open-source* menggunakan pfSense yang dikonfigurasi sebagai *Next-Generation Firewall* (NGFW) melalui integrasi Suricata dan Zenarmor, serta dihubungkan dengan platform SIEM Wazuh. Sistem yang dibangun tidak hanya digunakan untuk mendeteksi aktivitas serangan, tetapi juga merespon serangan melalui fitur *Intrusion Prevention System* (IPS) yang bekerja secara inline sehingga paket berbahaya dapat diblokir sebelum mencapai server tujuan. Selain itu, Zenarmor digunakan untuk menerapkan fitur *Application Control* dan *Web Control* sehingga akses aplikasi maupun situs web dapat dikendalikan sesuai kebijakan yang ditetapkan. Log dan aktivitas keamanan dari perangkat yang digunakan juga dikirim ke Wazuh untuk memudahkan proses monitoring dan analisis keamanan jaringan secara terpusat.

1.2 Perumusan Masalah

Melalui latar belakang tersebut, maka rumusan masalahnya adalah:

1. Bagaimana merancang NGFW berbasis Suricata dan Zenarmor yang berjalan pada *virtual machine* di *cloud*.
2. Bagaimana mengintegrasikan NGFW dengan SIEM sehingga serangan ataupun aktivitas jaringan yang ada dapat dipantau.
3. Bagaimana kemampuan NGFW dalam mendeteksi ataupun memblokir trafik *malicious*.

1.3 Batasan Masalah

Penelitian ini berfokus pada:

1. SIEM yang digunakan terbatas pada Wazuh SIEM
2. Integrasi pada SIEM Wazuh terbatas pada DVWA sebagai Web Server dan firewall pfSense serta komponen didalamnya.

3. Implementasi dilakukan pada lingkungan *cloud* menggunakan Virtual Machine Microsoft Azure.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan dari penelitian ini adalah sebagai berikut:

1. Merancang sistem solusi keamanan menggunakan Suricata dan Zenarmor pada lingkungan *Cloud.*
2. Mengintegrasi sistem NGFW yang telah dibangun dengan SIEM
3. Melakukan evaluasi kemampuan NGFW dalam mendeteksi dan memblokir trafik malicious ataupun tidak sesuai dengan aturan

1.4.2 Manfaat

Adapun manfaat dari penelitian ini adalah sebagai berikut.

1. Manfaat Bagi Penulis: Memperdalam pemahaman teknis mengenai implementasi perangkat keamanan, khususnya dalam mengonfigurasi dan integrasi antara NGFW dan SIEM.
2. Manfaat Bagi Akademis: Menambah literatur di bidang Keamanan Jaringan, khususnya sebagai referensi mengenai efektivitas penggunaan perangkat lunak *open-source*.
3. Manfaat Bagi Masyarakat Umum: Memberikan rekomendasi solusi keamanan siber NGFW.

1.5 Sistematika Penulisan

Adapun sistematika penulisan dari laporan ini adalah sebagai berikut

A. BAB I PENDAHULUAN

   Bab I ini menjelaskan latar belakang penelitian, perumusan masalah, batasan, tujuan, dan manfaat penelitian, serta sistematika penulisan.

B. BAB II TINJAUAN PUSTAKA

   Bab II berisikan penjelasan mengenai landasan teori atau kajian ilmu yang berhubungan dengan berbagai pokok pikiran topik penyusunan skripsi ini yang relevan dari sumber yang valid.

C. BAB III METODE PENELITIAN

Bab III berisi penjelasan mengenai rancangan penelitian, tahapan penelitian, dan objek penelitian.

D. BAB IV HASIL DAN PEMBAHASAN

Bab IV berisi proses perancangan sistem, topologi, serta penyajian data hasil pengujian beserta analisis atau evaluasi terhadap performa sistem keamanan.

E. BAB V PENUTUP

Bab V berisikan mengungkapkan kesimpulan secara singkat yang menjawab rumusan masalah berdasarkan hasil penelitian, serta menyajikan saran dari hasil temuan untuk pengembangan penelitian selanjutnya

BAB II
TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 Firewall

Firewall merupakan sistem keamanan yang dirancang untuk melindungi jaringan dan melakukan filter terhadap trafik yang masuk ataupun keluar. Firewall dapat mengizinkan ataupun memblokir trafik tersebut berdasarkan aturan tertentu yang telah ditetapkan (Patil & Mohurle 2017). Firewall menjadi komponen yang sangat penting dalam keamanan jaringan dan diletakan pada batas jaringan atau network perimeter. Hal ini dikarenakan firewall memisahkan 2 segmen yaitu jaringan internal yang dianggap aman (*trusted network*) dan jaringan yang dianggap tidak aman (*untrusted network*) misalnya internet.

Seiring dengan perkembangan teknologi dan kompleksitas ancaman siber, terjadi perkembangan dan evolusi firewall. Secara umum, berdasarkan teknologi penyaringan dan lapisan operasi pada model referensi OSI (*Open Systems Interconnection*), firewall dapat diklasifikasikan ke dalam beberapa generasi, di mana yang paling menonjol dalam arsitektur jaringan saat ini adalah Firewall Tradisional dan *Next-Generation Firewall* (NGFW).

2.1.1.1 Firewall Tradisional

Firewall tradisional merupakan generasi awal dari teknologi firewall yang berfokus pada penyaringan trafik berdasarkan rule yang sederhana. Firewall tradisional beroperasi pada Layer 3 (*Network Layer*) dan Layer 4 (*Transport Layer*) dari model OSI. Generasi pertama Firewall dikenal dengan *packet filtering* firewall. Firewall didesain untuk melindungi jaringan dengan memblokir trafik tertentu berdasarkan protokol, alamat IP, serta nomor port yang telah didefinisikan (Liang & Kim 2022).

Namun firewall tidak memiliki memori (stateless), hal ini menyebabkan setiap paket akan diterima ataupun ditolak berdasarkan *rule* yang sudah dibuat sebelumnya.

Untuk mengatasi permasalahan sebelumnya, firewall generasi selanjutnya memiliki kemampuan untuk menerima atau menolak suatu paket berdasarkan sesi koneksi yang sedang aktif ataupun riwayat paket sebelumnya. Firewall jenis ini menyimpan tabel status (*state table*) yang melacak informasi sesi komunikasi TCP atau UDP, seperti proses *handshake*, alamat IP, dan *port* yang digunakan. Dengan demikian, firewall dapat memastikan bahwa paket yang masuk dari luar atau internet benar-benar merupakan balasan dari permintaan yang diinisiasi oleh komputer di dalam jaringan internal.

Firewall tradisional bekerja dengan sangat efisien dalam memblokir trafik namun kurang baik secara fungsional. Firewall masih memiliki keterbatasan dalam mendeteksi ancaman yang beroperasi pada layer aplikasi. Hal ini disebabkan karena inspeksi hanya terbatas pada *header* paket dan tidak mencakup pengecekan terhadap isi (*payload*) dari paket tersebut

2.1.1.2 Next-Generation Firewall

Next-Generation Firewall (NGFW) merupakan pengembangan dari firewall tradisional dengan memberikan perlindungan pada layer lain yang tidak dimiliki pada generasi sebelumnya. NGFW mengintegrasikan berbagai fitur keamanan lanjutan dalam satu platform.

Deep-Packet Inspection (DPI) menjadi fitur utama yang dimiliki NGFW. DPI memungkinkan firewall untuk melakukan analisis terhadap isi paket, pengecekan terhadap *payload*, serta signature aplikasi yang digunakan. Selain itu terdapat *application control* dan *web control* yang memungkinkan identifikasi trafik berdasarkan jenis aplikasi ataupun kategori website yang diakses. Dengan demikian, administrator dapat menerapkan kebijakan keamanan yang lebih spesifik, seperti membatasi akses terhadap aplikasi ataupun situs berdasarkan kebijakan yang ditetapkan. NGFW umumnya juga diintegrasi dengan *Intrusion Detection Prevention* untuk mendeteksi dan secara otomatis memblokir intrusi maupun eksploitasi kerentanan berdasarkan *signature* ancaman yang telah diketahui.

2.1.2 SIEM

Security Information and Event Management (SIEM) adalah solusi keamanan yang berfungsi untuk mengumpulkan, mengolah, dan menganalisis log serta event

keamanan yang berasal dari berbagai sumber seperti server, *endpoint*, perangkat jaringan, aplikasi, dan sistem keamanan lainnya. SIEM bekerja dengan melakukan normalisasi dan korelasi data yang berasal dari berbagai perangkat dan mencocokan dengan aturan tertentu (Bezas & Filippidou 2023). Melalui aturan atau pola tersebut, apabila ditemukan event ataupun log yang cocok, SIEM akan membuat *alert* dan mengidentifikasi aktivitas tersebut sebagai mencurigakan secara real-time.

2.1.3 *Web Control & Application Control*

Web Control merupakan mekanisme keamanan untuk memantau, mengatur, dan membatasi koneksi menuju website tertentu (Zenarmor 2024). *Web Control* memastikan akses melalui protokol HTTP ataupun HTTPS sesuai dengan kebijakan keamanan internal yang ada. *Web control* bekerja dengan mencocokan situs website yang diakses dengan kategori website di database misalnya media sosial, pornografi, perjudian dan kemudian menentukan apakah akses menuju website tersebut diperbolehkan ataupun diblokir. Database kategori website diperbarui secara berkala oleh vendor melalui layanan *cloud threat intelligence*.

Selain *Web control*, next-generation firewall umumnya dilengkapi dengan fitur *application control*. *application control* merupakan kemampuan next-generation firewall (NGFW) untuk mengidentifikasi dan mengendalikan penggunaan aplikasi terlepas dari alamat IP, port dan protokol yang digunakan (Gupta 2024). Untuk mengidentifikasi aplikasi, NGFW memanfaatkan *deep packet inspection* (DPI) dengan menganalisis karakteristik lalu lintas jaringan, seperti pola trafik, fingerprint, dan metadata lainnya. Setelah aplikasi berhasil dikenali, NGFW dapat menerapkan kebijakan keamanan berdasarkan identitas aplikasi tersebut, seperti mengizinkan layanan *instant messaging* pada WhatsApp, tetapi memblokir fitur file transfer pada aplikasi yang sama.

Meskipun sama-sama berfungsi untuk membatasi akses, *web control* dan *application control* memiliki fokus yang berbeda. *Web control* berfokus pada pengendalian akses terhadap website yang diakses melalui protokol HTTP maupun HTTPS berdasarkan kategori website. Sebaliknya, *application control* berfokus pada identifikasi dan membatasi aplikasi yang digunakan, terlepas dari protokol,

alamat IP, maupun port komunikasi yang digunakan oleh aplikasi tersebut (Zenarmor 2024).

Mekanisme perlindungan serupa dapat ditemukan pada filtering berbasis proxy server yang umumnya berfokus pada pengendalian akses berdasarkan URL, domain, atau kategori website, Namun *web control* pada next-generation firewall (NGFW) terintegrasi dengan *deep packet inspection* (DPI) dan *Application Control*. Integrasi tersebut memungkinkan NGFW tidak hanya menerapkan kebijakan keamanan berdasarkan website yang diakses, tetapi juga berdasarkan identitas aplikasi serta karakteristik lalu lintas jaringan. Selain itu, NGFW umumnya mengintegrasikan fungsi keamanan lain seperti *intrusion prevention system* (IPS) dan *threat protection* dalam satu platform.

2.1.4 IDS

Intrusion Detection Systems (IDS) adalah sistem deteksi untuk memantau dan menganalisis aktivitas internal pada host individu, termasuk lalu lintas masuk dan keluar serta aktivitas sistem lokal untuk mendeteksi dan membuat *alert* adanya aktivitas malicious (Satilmis, Akleylek & Yuce Tok 2025). IDS dikategorikan menjadi 2 jenis, yaitu *Network-based Intrusion Detection System* (NIDS) dan *Host-Based Intrusion Detection System* (HIDS).

Pada NIDS, sistem akan mendeteksi lalu lintas jaringan dan menentukan apakah terdapat aktivitas malicious atau tidak. NIDS biasanya diterapkan pada network device sehingga dapat memonitor seluruh lalu lintas yang ada pada organisasi. Berbeda dengan NIDS, pada HIDS diterapkan dengan cara dipasang pada setiap endpoint untuk memantau aktivitas masing-masing sistem operasi, integritas berkas sistem, dan akses log perangkat secara langsung. HIDS akan membuat peringatan ataupun *alert* apabila terdeteksi sebuah aktivitas tidak biasa, serangan, ataupun hal tidak wajar yang terjadi pada perangkat host berdasarkan aturan (*rule*) deteksi yang telah ditetapkan.

2.1.5 IPS

Intrusion Prevention System (IPS) adalah teknologi keamanan yang berfungsi untuk mendeteksi sekaligus mencegah ancaman keamanan jaringan dan komputer dengan kemampuan respons aktif. Berbeda dengan IDS yang bersifat pasif, IPS dapat

mendeteksi dan memberikan tindakan berupa pemblokiran terhadap aktivitas yang dianggap malicious (Ang, Huy & Janarthanan 2025). Seperti halnya IDS, IPS juga dapat diimplementasikan pada tingkat jaringan (*Network-based* IPS atau NIPS) maupun pada tingkat host (*Host-based* IPS atau HIPS) untuk memberikan perlindungan yang berlapis.

Jenis IPS yang cukup banyak diterapkan adalah *Network-based Intrusion Prevention System*. NIPS didesain untuk bekerja dengan menganalisa *trafik* jaringan, mengidentifikasi potensi serangan, serta mengambil tindakan untuk mencegah serangan tersebut sampai ke host target. Tindakan tersebut dapat berupa membuang paket (*packet drop*), memutus koneksi TCP (*connection reset*), atau memblokir lalu lintas dari alamat IP penyerang.

2.1.6 Wazuh

Wazuh adalah platform *Security Information and Event Management* (SIEM) dan *Host-based Intrusion Detection System* (HIDS) berbasis *open-source* yang digunakan untuk memantau, menganalisis, serta mendeteksi aktivitas keamanan pada host maupun infrastruktur TI secara real-time. Wazuh berfungsi sebagai solusi terpusat yang mengumpulkan data log dari berbagai *endpoint*, melakukan analisis keamanan, mengidentifikasi anomali, serta memberikan informasi mengenai potensi ancaman. Sebagai sistem keamanan modern, Wazuh dirancang dengan arsitektur modular yang terdiri dari beberapa komponen utama yaitu Wazuh Agent, Wazuh Indexer, Wazuh Server, dan Wazuh Dashboard, yang bekerja secara terintegrasi dalam pipeline deteksi ancaman.

1. Wazuh Agent merupakan komponen yang dipasang pada *endpoint* seperti server, workstation, atau perangkat virtual untuk mengumpulkan data keamanan secara langsung dari sistem. Agent melakukan berbagai fungsi penting seperti memantau file system, mendeteksi perubahan integritas berkas *(File Integrity Monitoring)*, membaca log aplikasi dan sistem operasi, mengidentifikasi proses mencurigakan, hingga mengeksekusi modul deteksi berbasis aturan. Seluruh data yang dikumpulkan oleh Agent kemudian dikirimkan ke Wazuh Server untuk diproses lebih lanjut (Wazuh, 2024)

2. Wazuh Server berperan sebagai pusat analisis dan korelasi. Server menerima data dari seluruh Agent, kemudian memprosesnya menggunakan *rule*-based engine untuk menentukan apakah suatu aktivitas tergolong normal, mencurigakan, atau berpotensi menjadi ancaman. Server juga menyediakan mekanisme *alerting*, *active response*, serta integrasi dengan sistem keamanan lainnya seperti Suricata, YARA, VirusTotal, dan berbagai *threat intelligence feed*. Komponen ini juga menyimpan konfigurasi dan kebijakan yang didistribusikan kembali ke Agent untuk mengatur perilaku pemantauan.

3. Wazuh Indexer merupakan mesin pencari teks penuh (full-text search) dan analitik. Komponen ini berperan sebagai tempat penyimpanan data yang bertugas mengindeks dan menyimpan seluruh *alert* atau peringatan keamanan yang dihasilkan oleh Wazuh Server. Dengan adanya Indexer, data keamanan dapat dicari dan dianalisis untuk kebutuhan investigasi lebih lanjut.

4. Wazuh Dashboard merupakan antarmuka berbasis web yang digunakan untuk menampilkan data dan memberikan visibilitas terhadap seluruh aktivitas yang terjadi pada sistem. Dashboard memungkinkan analis untuk memonitor *alert*, memvisualisasikan log, melakukan investigasi insiden, memantau status Agent, serta mengelola konfigurasi sistem. Dengan tampilan yang interaktif, Dashboard mempermudah proses analisis dan membantu memahami pola ancaman secara cepat dan intuitif.

2.1.7 pfSense

pfSense merupakan sistem operasi *open-source* berbasis freeBSD yang dirancang sebagai router ataupun firewall. PfSense menyediakan berbagai fungsi dan fitur jaringan seperti firewall, *load balancing*, VPN, *Network Address Translation* (NAT), dan monitoring jaringan. Dalam penggunaannya, pfsense di-install pada hardware dan dapat dikonfigurasi serta dikelola menggunakan web interface. Salah satu keunggulan pfSense adalah fleksibilitasnya dimana terdapat paket tambahan yang dapat dipasang dan dikonfigurasi untuk menambah fitur-fitur yang ada, misalnya Suricata dan Zenarmor.

2.1.8 Suricata

Suricata adalah perangkat lunak *open-source* yang berfungsi sebagai *intrusion detection system* (IDS), *intrusion prevention system* (IPS), dan pemantauan keamanan jaringan (Network Security Monitoring). Suricata dikembangkan oleh Open Information Security Foundation (OISF) dan memiliki kemampuan melakukan inspeksi paket mendalam (*deep packet inspection*) untuk mengidentifikasi ancaman, serangan siber, serta anomali jaringan secara real-time (Open Information Security Foundation (OISF), 2026)

Suricata mampu beroperasi sebagai *intrusion detection system* (IDS) maupun *intrusion prevention system* (IPS) dalam satu platform. Berbeda dengan snort, Suricata mendukung multi-threading yang memungkinkan memproses trafik jaringan secara paralel sehingga lebih optimal.

Dalam operasionalnya, Suricata bekerja dengan mencocokkan lalu lintas jaringan terhadap serangkaian aturan (*rules*) atau tanda tangan (*signature*) yang telah ditentukan. Salah satu penyedia aturan yang menjadi standar dalam penggunaan Suricata adalah *Emerging Threats* (ET). *Ruleset* ini tersedia dalam dua varian utama, yaitu ET Open yang berbasis komunitas dan dapat digunakan secara gratis, serta ET Pro yang dikelola untuk kebutuhan komersial. Meskipun ET Open dapat digunakan secara gratis, perlu dilakukan penyesuaian (*tuning*) dan validasi aturan dibandingkan dengan versi Pro, untuk mengurangi tingkat *false positive* dan memastikan akurasi deteksi yang sesuai dengan lingkungan jaringan perusahaan.

2.1.9 Zenarmor

Zernarmor merupakan solusi keamanan jaringan berbasis perangkat lunak yang dirancang untuk memperluas kemampuan firewall *open-source* seperti pfSense. Zenarmor bekerja dengan melakukan inspeksi lalu lintas jaringan hingga Layer 7 (*Application Layer*) sehingga mampu mengidentifikasi aplikasi, mengklasifikasikan website, serta mendeteksi aktivitas jaringan. Berdasarkan hasil inspeksi, Zenarmor dapat menerapkan kebijakan keamanan seperti *web control*, *application control*, serta deteksi ancaman dengan memanfaatkan layanan *threat intelligence* berbasis *cloud* yang diperbarui secara berkala (Zenarmor 2026). Zenarmor mendukung integrasi dengan firewall *open-source* dan menyediakan

berbagai kemampuan *Next-Generation Firewall* dengan biaya yang lebih rendah dibandingkan solusi NGFW komersial.

2.1.10 DVWA

Damn Vulnerable Web Application (DVWA) adalah sebuah aplikasi web berbasis PHP dan MySQL yang dirancang dengan berbagai kerentanan keamanan di dalamnya. Aplikasi ini difungsikan sebagai platform edukasi dan pelatihan bagi praktisi keamanan siber, pengembang web, serta administrator sistem untuk memahami metode eksploitasi dan mekanisme pertahanan aplikasi web. DVWA menyediakan lingkungan laboratorium yang aman dan legal bagi pengguna untuk mempraktikkan teknik penetrasi (*penetration testing*) tanpa melanggar hukum atau merusak sistem produksi.

2.1.11 URLHaus

URLHaus merupakan platform berbasis komunitas yang dikembangkan oleh Abuse.ch untuk mengumpulkan dan berbagi informasi mengenai URL malicious yang digunakan dalam penyebaran *malware*. Malicious URL adalah alamat *Uniform Resource Locator* (URL) yang digunakan oleh pelaku ancaman siber untuk mendistribusikan *malware*, mengarahkan pengguna ke situs phishing, mencuri informasi sensitif, maupun membangun komunikasi dengan server *Command and Control* (C2).

2.2 Penelitian Terkait

Pada Tabel 2.1 menunjukkan penelitian yang berkaitan dengan sistem pendeteksian intrusi yang sebelumnya telah dilakukan.

Tabel 2. 1 Penelitian terkait

No	Peneliti & Judul	Teknologi	Hasil Penelitian	Perbedaan
1	(Muhamad Malik Matin et al. 2023) Sistem Monitoring	SIEM Wazuh, Wazuh Agent HIDS,	Penelitian berhasil mengimplementasikan sistem monitoring untuk Data Center menggunakan Wazuh	Penelitian tersebut berfokus pada monitoring deteksi seragan menggunakan

	Keamanan Pada Data Center Berbasis Security Information And Event Management (Siem) Dengan Wazuh dan IDS	Suricata HIDS	dan Suricata IDS. Sistem berhasil mendeteksi serangan berbasis jaringan melalui Suricata dan serangan yang berkaitan dengan sistem ataupun autentikasi dengan wazuh HIDS.	Wazuh dan Suricata HIDS. Penelitian ini meningkatkan dengan rancangan tersebut dengan menambahkan Zenarmor dan meletakan pada host terpisah. Tujuannya adalah agar serangan yang datang akan masuk ke host lain untuk dianalisis terlebih dahulu sebelum sampai ke server. Selain itu tools keamanan dikonfigurasi untuk melakukan deteksi dan drop packet yang sudah diketahui *malicious* sehingga serangan tidak dilanjutkan ke server.
2	(Wiguna 2024) Analisis Implementasi dan Integrasi Suricata	Suricata (IDS), Telegram Bot API	Penelitian ini berhasil mengimplementasikan sistem deteksi intrusi. Suricata dikonfigurasi untuk mendeteksi serangan	Penelitian tersebut berfokus pada deteksi serangan dan otomatisasi mengirimkan

	dengan Scirius dan ntopng untuk Pendeteksian dan Pemblokiran Serangan pada Server dengan Notifikasi WhatsApp		seperti *Port Scanning* dan *DoS*, kemudian script mengirimkan pesan peringatan (*alert*) ke aplikasi Telegram. IP yang terdeteksi banyak menggenerate *alert* akan dilakukan respon blokir sehingga tidak bisa mengakses service lagi untuk mencegah serangan lanjutan.	informasi melalui Telegram. Namun pada penelitian ini, serangan yang terdeteksi pada penelitian ini adalah hasil port miroring. Artinya serangan ataupun *payload* sebenarnya tetap masuk ke server dan tools Suricata hanya melakukan analisis pada *payload* lainnya hasil mirror. Pada penelitian penulis, firewall akan diletakan diantara server dan source trafik, sehingga tools dapat memberikan action berupa drop terhadap trafik *malicious*. Selain itu firewall juga diintegrasikan dengan SIEM untuk visualisasi dan analisis lebih lanjut

3	(Sulaiman, Seta & Falih 2021) Exploitation Prevention on Network Printer with Signature Based Suricata on PfSense	pfSense, Suricata (NIPS)	Suricata yang dipasang pada pfSense berhasil memblokir eksploitasi spesifik yang menargetkan kerentanan pada *Network Printer* menggunakan deteksi berbasis *signature*.	Penelitian tersebut hanya berfokus pada perlindungan perangkat Printer. Penelitian penulis mengganti cakupan menjadi Web Server yang lebih umum menjadi target serangan. Selain itu diintegrasikan SIEM untuk visualisasi terhadap serangan yang masuk.
4	(Manzoor et al. 2024) Cybersecurity on a budget: Evaluating security andperformance of*open-source* SIEM solutions for SMEs	Wazuh SIEM, AlienVault OSSIM, Elastic Security, SIEMonster	Hasil evaluasi menunjukkan bahwa Wazuh unggul dalam efisiensi sumber daya dan kemampuan deteksi dibandingkan solusi SIEM *open-source* lainnya.	Penelitian tersebut hanya membandingkan kinerja software SIEM. Penelitian penulis mengimplementasikan pertahanan aktif melalu NGFW. Selain itu Wazuh tidak hanya memonitor, tetapi terintegrasi dengan firewall untuk validasi dan visualisasi serangan.

5	(Wiradyaksa et al. 2024) A Comparative Evaluation of Snort and Suricata for Detecting Data Exfiltration Tunnels in *Cloud* Environments	Snort, Suricata	Dalam segi performa Suricata terbukti lebih unggul dibandingkan Snort dalam menangani *throughput* jaringan tinggi dan mendeteksi teknik eksfiltrasi data melalui *tunneling* berkat arsitektur *multi-threading*	Penelitian tersebut berfokus pada perbandingan performa deteksi engine IDS (Snort vs Suricata). Berdasarkan penelitian itu penulis menggunakan Suricata tersebut dan melengkapi dengan Zenarmor. Untuk dapat melindungi dengan baik server keduanya dikonfigurasi untuk tidak hanya mendeteksi namun untuk memblokir serangan. Selain itu diintegrasikan SIEM untuk visualisasi terhadap serangan yang masuk.

BAB III
METODE PENELITIAN

3.1 Rancangan Penelitian

Penelitian ini menggunakan pendekatan kuantitatif eksperimental yang dilakukan dengan menguji sistem secara langsung dalam lingkungan *cloud*. Tujuan dari pendekatan ini adalah untuk merancang sistem *Next-Generation Firewall* yang dapat terintegrasi dengan SIEM serta mengevaluasi performa dari sistem tersebut.

Data pengujian yang terkumpul dianalisis menggunakan metode deskriptif kuantitatif. Analisis difokuskan pada fungsionalitas NGFW, tingkat akurasi deteksi ancaman, efektivitas pemblokiran, serta beban performa sistem antara kondisi jaringan normal dengan kondisi saat terjadi serangan.

3.2 Tahapan Penelitian

Alur penelitian meliputi langkah-langkah berikut:

1. Studi Literatur
   Dilakukan pengumpulan dan pemahaman landasan teori yang relevan misalnya arsitektur SIEM, konsep IDS, IPS, dan teknologi pendukung lainnya.
2. Analisis Kebutuhan
   Berdasarkan studi literatur dan penelitian terdahulu, dilakukan analisis kebutuhan berupa arsitektur ataupun perangkat yang dapat digunakan untuk menyelesaikan permasalahan awal.
3. Perancangan dan Pembuatan Sistem
   Pembangunan topologi jaringan virtual, instalasi sistem operasi dan tools yang digunakan, serta konfigurasi integrasi antar-sistem.
4. Pengujian Sistem
   Sistem yang telah dibuat dan diintegrasikan kemudian dilakukan pengujian sesuai skenario yang telah ditentukan. Pelaksanaan pengujian berupa serangan terhadap sistem target sesuai dengan skenario ancaman yang telah ditentukan.

5. Analisis dan Evaluasi

   Data log yang terekam selama tahap pengujian kemudian dianalisis. Evaluasi akan dilakukan untuk menilai apakah sistem yang sudah dibuat berhasil mendeteksi, merespon, dan menampilkan data melalui SIEM.

6. Penyusunan Laporan

   Tahap terakhir yaitu mendokumentasikan seluruh proses penelitian, mulai dari perancangan, konfigurasi, hingga hasil analisis data. Kesimpulan akan dibuat berdasarkan perbandingan antara hasil eksperimen dengan tujuan penelitian yang telah ditetapkan sebelumnya.

3.3 Objek Penelitian

Objek utama dalam penelitian ini adalah kemampuan serta performa dari sistem NGFW dan terintegrasi dengan SIEM. Sistem mencakup NGFW menggunakan pfSense yang dipasang modul tambahan berupa Suricata serta Zenarmor, dan SIEM Wazuh sebagai pusat monitoring. Penelitian ini memfokuskan pada fungsional NGFW, pengukuran tingkat akurasi deteksi ancaman, keberhasilan pemblokiran paket serangan, serta penggunaan sumber daya perangkat keras (CPU dan RAM) pada Web Server saat menghadapi skenario serangan sebelum dan sesudah firewall digunakan.

BAB IV
HASIL DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Tahap analisis kebutuhan dilakukan untuk mendefinisikan spesifikasi perangkat keras dan perangkat lunak yang akan digunakan dalam membangun sistem *Next-Generation Firewall* (NGFW) yang terintegrasi dengan SIEM.

4.1.1 Analisis Perangkat Virtual

Penelitian dilakukan pada lingkungan *Cloud* Azure berupa Virtual Machine. Detail alokasi sumber untuk masing-masing Virtual Machine (VM) ditampilkan pada Tabel 4.1.

Tabel 4. 1 Spesifikasi Virtual Machine

No	Komponen	Spesifikasi	Keterangan
1.	Firewall	- 4 GB RAM - 2 CPU - 32 GB Memori	Virtual machine yang difungsikan sebagai *Next-Generation Firewall (NGFW)* yang bertugas melakukan filtering, inspeksi trafik, serta pengamanan jaringan
2.	SIEM Server	- 4 GB RAM - 2 CPU - 64 GB Memori	Virtual machine yang digunakan sebagai sistem *Security Information and Event Management (SIEM)* untuk mengumpulkan, menyimpan, dan menganalisis log keamanan
3.	Web Server	- 2 GB RAM - 1 CPU - 32 GB Memori	Virtual machine yang digunakan sebagai target pengujian dan serangan

4.	Penyerang	- 2 GB RAM - 2 CPU	Virtual machine yang digunakan untuk mensimulasikan aktivitas serangan terhadap jaringan dan web server

Alokasi sumber daya diberikan berdasarkan kebutuhan masing-masing layanan. Firewall dan SIEM Server diberikan alokasi CPU dan memori yang lebih besar dikarenakan kedua sistem melakukan proses inspeksi paket, pemrosesan log, indexing data, dan lainnya yang membutuhkan resource komputasi lebih tinggi.

4.1.2 Analisis Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini tertera pada Tabel 4.2.

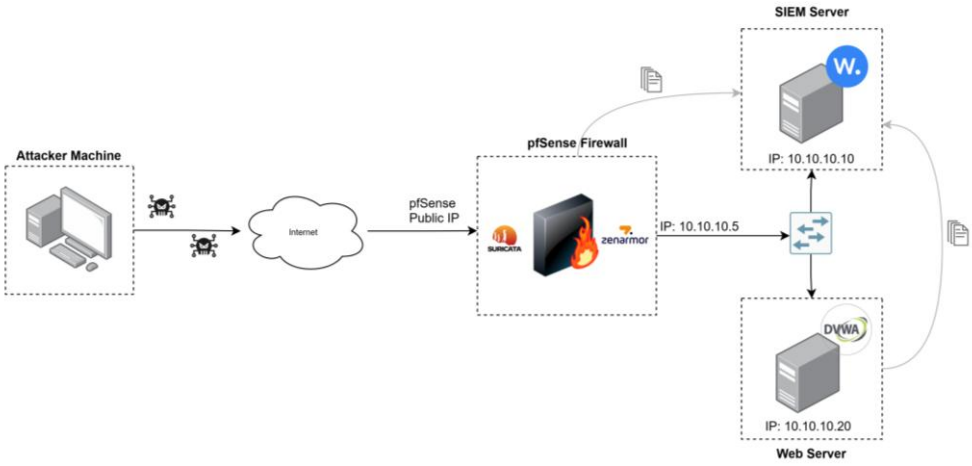
Tabel 4. 2 Kebutuhan Software

No	Komponen	Keterangan
1.	pfSense	Sistem operasi yang digunakan pada firewall
2.	Zenarmor	Modul tambahan yang dipasang pada pfSense
3.	Suricata	Modul tambahan yang dipasang pada pfSense
4.	Syslog-ng	Digunakan sebagai sistem pengiriman log (*log forwarding*) dari firewall ke server SIEM
5.	Wazuh	Digunakan sebagai platform SIEM untuk mengumpulkan, menganalisis, dan mengkorelasi log.
6.	OpenVPN	Digunakan untuk menyediakan koneksi aman (*secure remote access*) ke dalam jaringan ataupun layanan internal
7.	DVWA	*Damn Vulnerable Web Application* digunakan sebagai target pengujian keamanan

4.2 Perancangan Sistem

4.2.1 Komponen dan Arsitektur Sistem

Arsitektur sistem dirancang pada lingkungan *cloud* menggunakan Microsoft Azure dengan tujuan membangun infrastruktur sistem yang dapat menerima lalu lintas jaringan secara langsung dari internet publik. Pada rancangan ini, seluruh trafik yang menuju layanan internal akan melewati firewall pfSense sebagai gerbang utama sebelum diteruskan menuju server tujuan.



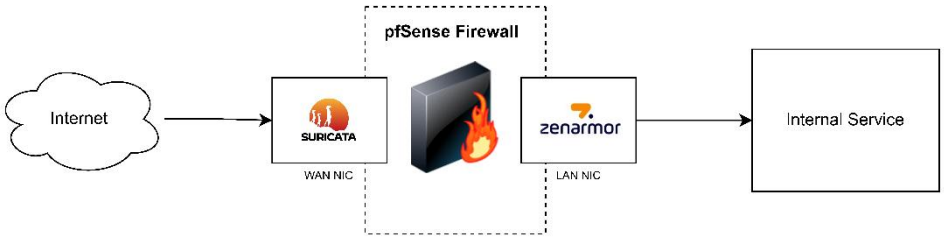
Gambar 4. 1 Diagram sistem

Komponen utama dalam rancangan ini meliputi:

1. Penyerang: Menggunakan sistem operasi Linux yang berfungsi untuk melakukan pengujian keamanan dan simulasi serangan terkontrol terhadap sistem. Sistem penyerang akan mengirimkan serangan melalui internet dan trafik akan diteruskan ke Web Server.
2. Firewall: Menggunakan sistem operasi BSD pfSense yang diintegrasikan dengan Suricata dan Zenarmor. Komponen ini menambah fungsionalitas pfSense sebagai firewall tradisional menjadi NGFW yang dapat melakukan inspeksi dan filter trafik hingga layer ke 7.
3. SIEM Server: Menggunakan Wazuh SIEM yang berfungsi untuk mengumpulkan log keamanan dari firewall dan web server, melakukan korelasi *event*, dan memvisualisasikan *alert*.

4. Web Server: Menggunakan sistem operasi Linux yang menjalankan layanan web DVWA (Damn Vulnerable Web Application) sebagai objek pengujian kerentanan.

Pada implementasinya, perangkat pfSense berperan sebagai *gateway* yang menghubungkan jaringan internet dan jaringan internal sistem (LAN). pfSense dikonfigurasi menggunakan dua Network Interface Card (NIC), yaitu *interface* WAN dan *interface* LAN. Interface LAN berfungsi sebagai jalur komunikasi untuk sistem internal seperti Wazuh Server dan Web Server, sedangkan interface WAN digunakan sebagai jalur koneksi antara jaringan internal dan internet publik.



Gambar 4. 2 NIC pfSense

Penggunaan dua *interface* tersebut juga mendukung pemisahan proses inspeksi trafik antara Suricata dan Zenarmor agar kedua *engine* dapat berjalan secara bersamaan tanpa konflik inspeksi paket. Suricata dikonfigurasi pada *interface* WAN menggunakan mode IPS *Inline* untuk melakukan inspeksi dan pemblokiran trafik berbahaya yang berasal dari internet. Sementara itu, Zenarmor dijalankan pada *interface* LAN untuk melakukan inspeksi paket, kontrol aplikasi, serta filtering trafik pada jaringan internal.

Pada segmen internal, terdapat beberapa komponen utama yaitu Wazuh Server dengan alamat IP 10.10.10.10 yang berfungsi sebagai pusat SIEM serta Web Server dengan alamat IP 10.10.10.20 yang digunakan sebagai target pengujian keamanan. Seluruh trafik menuju layanan tersebut akan melewati pfSense dan diteruskan ke masing-masing layanan.

Untuk keperluan konfigurasi dan monitoring, administrator dapat mengakses layanan internal melalui koneksi *virtual private network* (VPN) sehingga proses

pengelolaan sistem tetap dapat dilakukan secara aman meskipun infrastruktur berada pada jaringan *cloud* publik.

4.2.2 Konfigurasi Jaringan dan Alamat IP

Konfigurasi jaringan pada penelitian ini dilakukan pada lingkungan *cloud* Microsoft Azure menggunakan pfSense sebagai *gateway* utama yang menghubungkan layanan internal dengan internet publik. Setiap perangkat diberikan alamat IP internal statis untuk mempermudah proses *routing*, monitoring, dan integrasi log.

pfSense dikonfigurasi menggunakan dua network *interface*, yaitu *interface* WAN dan LAN. *Interface* WAN digunakan sebagai *gateway* menuju internet publik sekaligus menjadi titik inspeksi trafik oleh Suricata IPS. Sementara itu, *interface* LAN digunakan untuk menghubungkan layanan internal seperti Wazuh SIEM dan Web Server serta digunakan oleh Zenarmor untuk proses inspeksi dan filtering trafik internal.

Tabel 4. 3 Daftar Konfigurasi Alamat IP

No	Perangkat	*Interface*	IP Address	Keterangan
1.	Firewall	WAN	10.10.1.5	*Interface* yang terhubung ke internet publik
		LAN	10.10.10.5	Alamat IP internal dari firewall sekaligus *gateway* jaringan internal
2.	SIEM Server	LAN	10.10.10.10	Alamat IP dari SIEM Server
3.	Web Server	LAN	10.10.10.20	Alamat IP dari Web Server dimana akan dilakukan port forwarding sehingga dapat diakses dari eksternal

4.	VPN Segmen	VPN	10.20.20.0/24	Alokasi IP untuk user yang terhubung melalui VPN Alamat IP yang didapatkan secara acak (DHCP) untuk mengakses layanan internal (LAN)

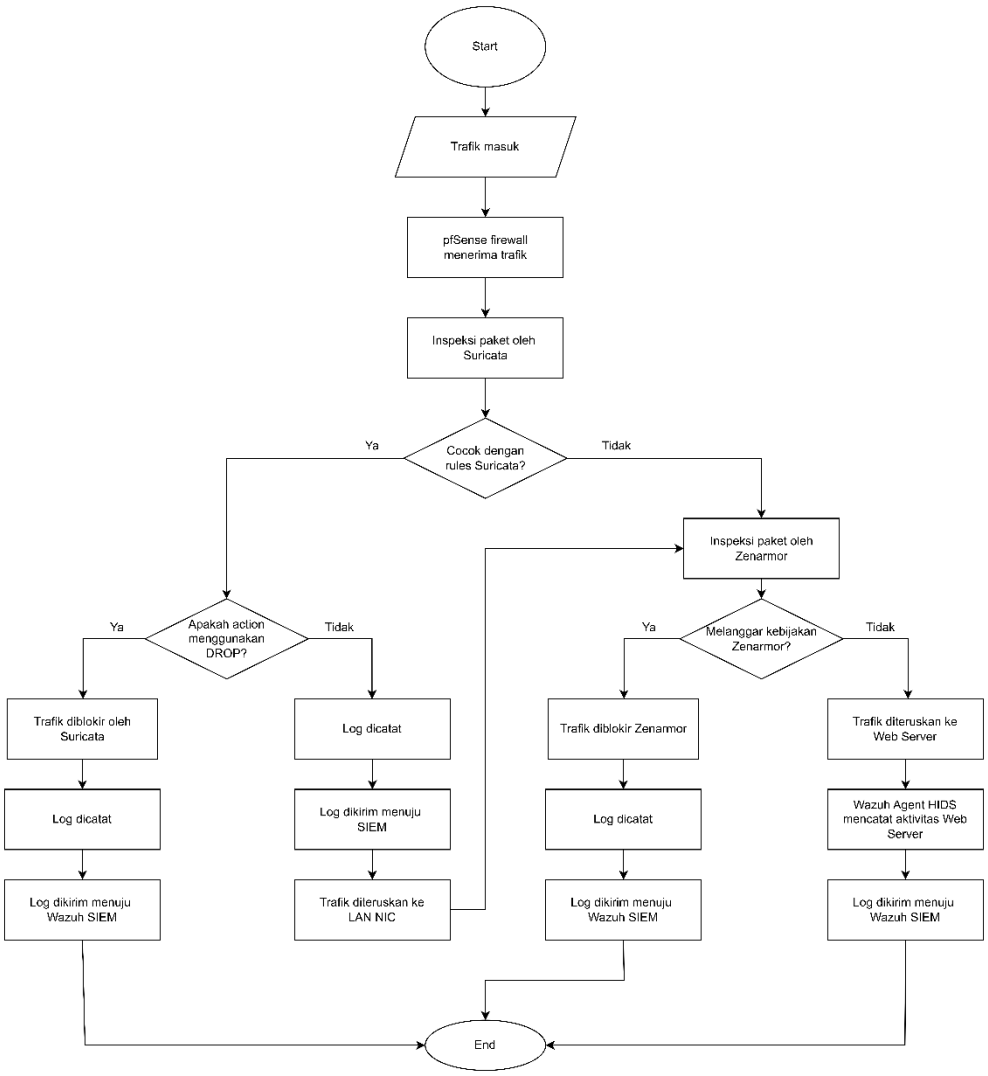
4.2.3 Network Address Translation

Agar layanan Web Server dapat diakses dari jaringan publik (internet), dilakukan konfigurasi *Network Address Translation* (NAT) menggunakan mekanisme 1:1 NAT pada firewall pfSense. Pada konfigurasi ini, pfSense berperan sebagai *gateway* utama yang menerima seluruh trafik dari jaringan internet dan meneruskannya ke alamat IP server internal sesuai dengan aturan NAT yang telah ditentukan

Pada penelitian ini, satu alamat IP publik pada *interface* WAN pfSense dilakukan *mapping* secara langsung ke alamat IP Web Server yang menjalankan aplikasi Damn Vulnerable Web Application (DVWA). Dengan mekanisme 1:1 NAT, seluruh trafik yang ditujukan ke alamat IP publik tersebut dapat diteruskan ke server internal tanpa perlu membuat aturan *port forwarding* untuk setiap port secara terpisah. Dengan konfigurasi tersebut, layanan web serta port lainnya yang berjalan pada Web Server dapat diakses melalui alamat IP publik untuk kebutuhan pengujian keamanan dan analisis serangan.

4.2.4 Flowchart Sistem

Diagram alur (*flowchart*) sistem menggambarkan bagaimana alur inspeksi, proses, atau tahapan sistem bekerja pada trafik masuk (inbound traffic) dan trafik keluar (*outbound traffic*) pada sistem *Next-Generation Firewall* (NGFW) yang dibangun menggunakan pfSense, Suricata, Zenarmor, dan Wazuh SIEM. Flowchart trafik masuk dan trafik keluar dapat dilihat pada Gambar 4.3 dan Gambar 4.4.



Gambar 4. 3 Flowchart Trafik Masuk

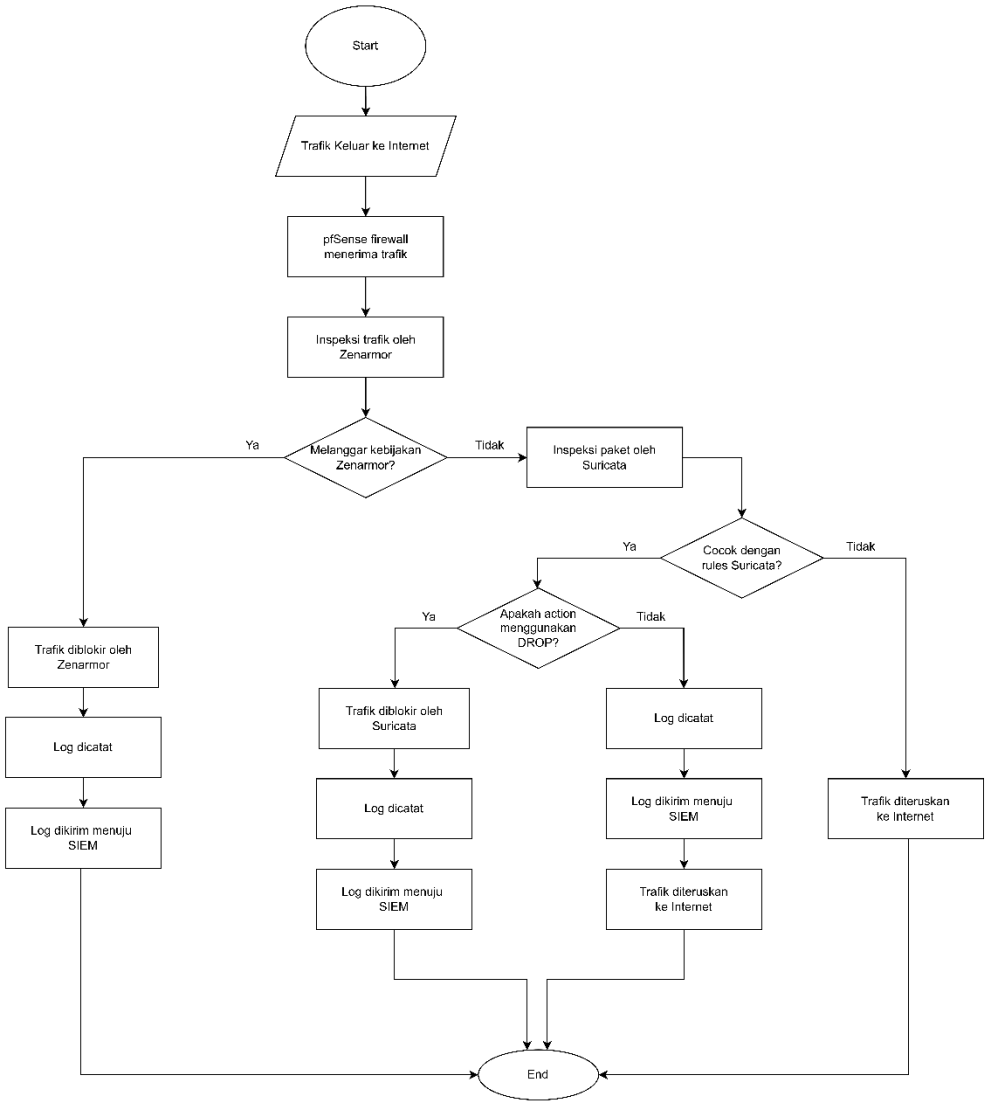
Pada trafik masuk, proses dimulai ketika pfSense firewall menerima trafik dari internet. Trafik yang diterima akan masuk melalui NIC WAN dan dilakukan inspeksi paket oleh Suricata IPS menggunakan mekanisme signature-based detection. Apabila trafik cocok dengan *rule*s Suricata, firewall akan menentukan tindakan berdasarkan *rule*s yang digunakan. Apabila *rule*s menggunakan action DROP, maka trafik akan diblokir sebelum mencapai server tujuan. Selanjutnya *alert* dicatat dan log dikirim menuju Wazuh SIEM. Apabila *rule*s menggunakan ALERT, maka *alert* tetap dicatat dan dikirim ke SIEM, namun trafik tetap diteruskan menuju proses inspeksi berikutnya.

Trafik yang lolos dari Suricata kemudian diperiksa oleh Zenarmor menggunakan mekanisme application awareness dan security policy filtering. Pada tahap ini Zenarmor melakukan evaluasi terhadap trafik berdasarkan kebijakan keamanan yang telah diterapkan. Jika trafik tidak sesuai dengan kebijakan keamanan, maka trafik akan diblokir dan aktivitas keamanan dicatat sebelum log dikirim menuju SIEM. Sebaliknya, apabila trafik dianggap aman, maka trafik diteruskan menuju Web Server.

Aktivitas pada Web Server dipantau menggunakan Wazuh Agent sebagai Host-based Intrusion Detection System (HIDS). Wazuh Agent mencatat aktivitas host dan mengirimkan log menuju Wazuh SIEM. Melalui log yang telah dikumpulkan pada SIEM, administrator dapat melakukan monitoring keamanan secara terpusat terhadap aktivitas jaringan maupun endpoint.

Pada trafik keluar, seluruh trafik berasal dari user internal maupun server akan melewati pfSense Firewall sebelum diteruskan menuju internet. Trafik yang diterima akan masuk melalui NIC LAN dan dilakukan inspeksi paket oleh Zenarmor untuk dinilai apakah trafik sesuai dengan *security policy* yang ditetapkan. Zenarmor digunakan untuk mengontrol akses aplikasi, website, dan mendeteksi adanya trafik berbahaya seperti koneksi menuju *malware* ataupun CnC.

Jika trafik melanggar kebijakan keamanan yang telah dibuat, maka Zenarmor akan memblokir trafik dan mencatat aktivitas keamanan. Kemudian log tersebut akan dikirim menuju Wazuh SIEM. Sebaliknya, apabila trafik dianggap aman, maka trafik diteruskan menuju proses inspeksi berikutnya menggunakan Suricata.



Gambar 4. 4 Flowchart Trafik Keluar

Suricata melakukan inspeksi paket outbound menggunakan *rule*s berbasis signature. Jika ditemukan kecocokan dengan *rule*s dan menggunakan aksi "*drop*", maka trafik akan diblokir sebelum diteruskan menuju internet publik. Apabila *rule*s hanya menghasilkan *alert* tanpa aksi "*drop*", maka *alert* tetap dicatat dan dikirim menuju SIEM Wazuh, namun trafik tetap diteruskan menuju internet. Seluruh aktivitas keamanan yang dihasilkan selama proses inspeksi oleh Suricata dan Zenarmor kemudian dikirim menuju Wazuh SIEM.

4.2.5 Rancangan Kebijakan Keamanan

Dalam merancang Next-Generation Firewall (NGFW), kemampuan pencegahan ancaman bergantung pada aturan (rules) dan kebijakan (policies) yang diterapkan

pada sistem. Oleh karena itu, dirancang kebijakan keamanan untuk Suricata dan Zenarmor yang disesuaikan dengan kebutuhan perlindungan jaringan internal dan Web Server.

4.2.5.1 Kebijakan Suricata

Suricata dirancang untuk beroperasi pada mode Inline (IPS) yang berfokus pada inspeksi trafik masuk dari antarmuka WAN. Kebijakan tindakan (*action*) yang akan diterapkan untuk ancaman yang terdeteksi adalah *drop*, di mana paket berbahaya akan dibuang secara diam-diam untuk mencegah serangan mencapai Web Server target tanpa memberikan respons kembali ke penyerang. Aturan yang digunakan pada Suricata bersumber dari dua jenis ruleset:

1. Emerging Threats (ET) Open Rules: Kumpulan ruleset IDS/IPS yang dikembangkan oleh komunitas keamanan dan didistribusikan secara gratis untuk digunakan pada Suricata maupun Snort. Rule yang akan digunakan spesifik pada kelemahan web dan percobaan scanning.
2. Custom Rules: Dibuat secara manual untuk melindungi aplikasi web (DVWA) dari serangan yang mungkin tidak tercakup oleh ruleset standar. Perancangan custom rules difokuskan pada deteksi scanning dan eksploitasi web.

4.2.5.1 Kebijakan Zenarmor

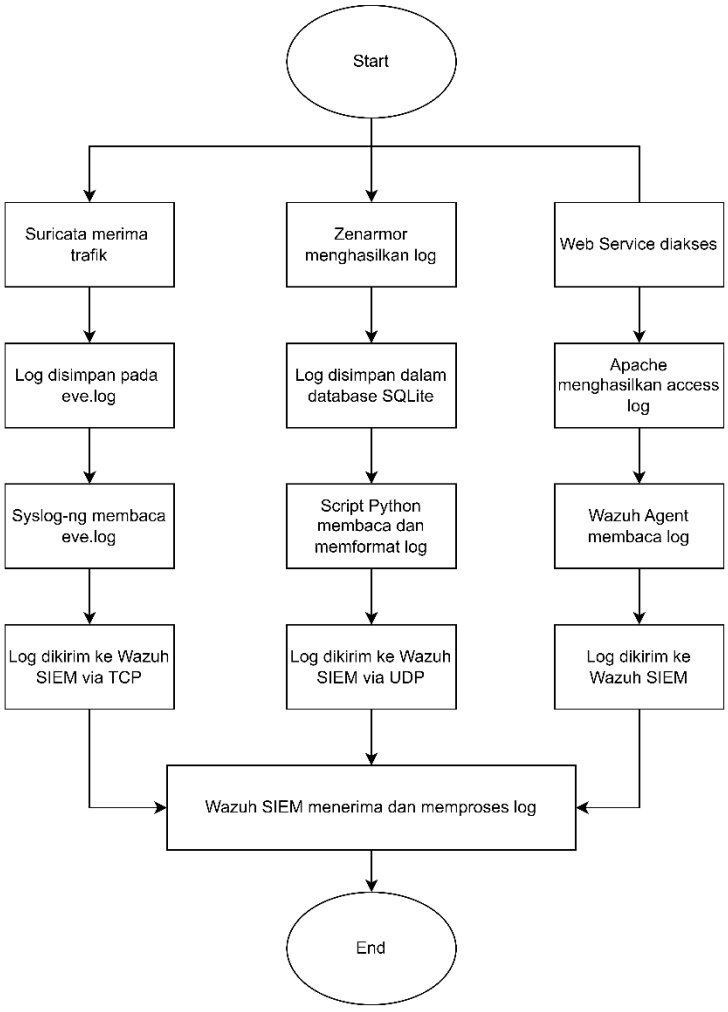
Zenarmor dirancang untuk mengontrol aktivitas dari jaringan internal menuju internet. Kebijakan tindakan yang dipilih adalah *blocked*, di mana trafik yang sesuai dengan kategori tertentu yang dilarang akan diblokir sehingga tidak diteruskan menuju internet. Zenarmor memiliki beberapa fitur untuk melakukan pemfilteran jaringan, dan pada penelitian ini fitur yang akan diaktifkan yaitu:

1. Application Control: Membatasi aplikasi spesifik yang yang berpotensi membawa malware, membuka celah bypass jaringan, atau mengurangi produktivitas. Contoh dari jenis aplikasi yang akan dibatasi yaitu proxy, gaming, ads, dan tracker.
2. Web Control: Membatasi akses menuju kategori situs yang berisiko dan tidak relevan dengan kebutuhan organisasi.

3. Threat Protection: Membatasi koneksi alamat IP ataupun domain yang berkaitan dengan malware, viruses, dan phishing.

4.2.6 Rancangan Integrasi Sistem

Diagram alur (*flowchart*) log forwarding menjelaskan proses pembuatan, pengumpulan, dan pengiriman log pada sistem *Next-Generation Firewall* (NGFW) yang menggunakan pfSense terdiri dari Suricata dan Zenarmor menuju Wazuh SIEM.



Gambar 4. 5 Flowchart Pemrosesan Log

Ketika terdapat trafik yang melewati firewall, Suricata menghasilkan *alert* dan mencatat sebagai log jika hasil inspeksi paket cocok dengan *rule*s yang ada. Log disimpan dalam file eve.json pada pfSense. Selanjutnya, layanan Syslog-ng membaca file log tersebut dan meneruskannya ke Wazuh SIEM menggunakan

protokol TCP. Penggunaan protokol TCP dipilih dikarenakan protokol UDP memiliki batas ukuran pengiriman bit yang dapat menyebabkan log Suricata terpotong selama proses forwarding log.

Selain Suricata, Zenarmor juga menghasilkan log dari aktivitas filtering. Log Zenarmor disimpan dalam database SQLite internal. Karena Zenarmor versi free tidak menyediakan fitur *native log forwarding* menuju SIEM, digunakan script Python *custom* untuk membaca database SQLite secara berkala. Data log yang berhasil kemudian akan dilakukan *parsing* dan *formatting* sebelum dikirim menuju Wazuh SIEM menggunakan protokol UDP.

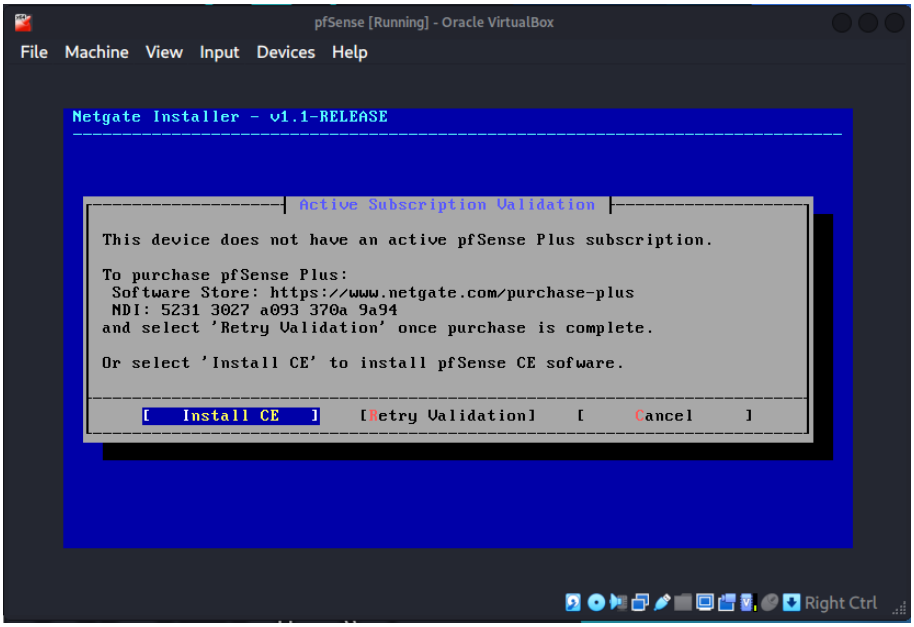
Pada sisi endpoint, aktivitas Web dicatat dalam Apache access log dan system log. Wazuh Agent yang terpasang pada Web Server akan membaca log tersebut dan mengirimkannya ke Wazuh SIEM sehingga aktivitas pada level host tetap dapat dimonitor.

Seluruh log yang diterima oleh Wazuh SIEM kemudian diproses dan dianalisis melalui dashboard monitoring. Dengan mekanisme ini, administrator dapat melakukan monitoring keamanan secara terpusat terhadap aktivitas jaringan maupun endpoint.

4.3 Implementasi Sistem

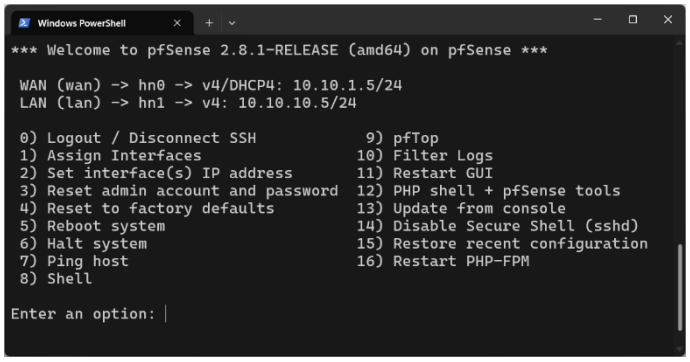
4.3.1 Instalasi pfSense

Tahap awal yang dilakukan adalah dengan menyiapkan *Virtual Machine* (VM) untuk pfSense. Instalasi dapat dilakukan melalui *deployment* sistem operasi pfSense yang disediakan pada Marketplace Azure. Namun Azure hanya menyediakan pfSense Plus yang memiliki biaya lisensi tambahan. Cara lainnya yang dilakukan pada penelitian ini adalah dengan menggunakan *custom image deployment* yaitu dengan membuat dan melakukan *deploy image* secara lokal. Setelah itu *image dimigrasikan* ke lingkungan *cloud*. Dengan pendekatan ini, penelitian dapat menggunakan pfSsense CE (*Community Edition*) yang *open source* dan gratis.



Gambar 4. 6 Proses Instalasi pfSense

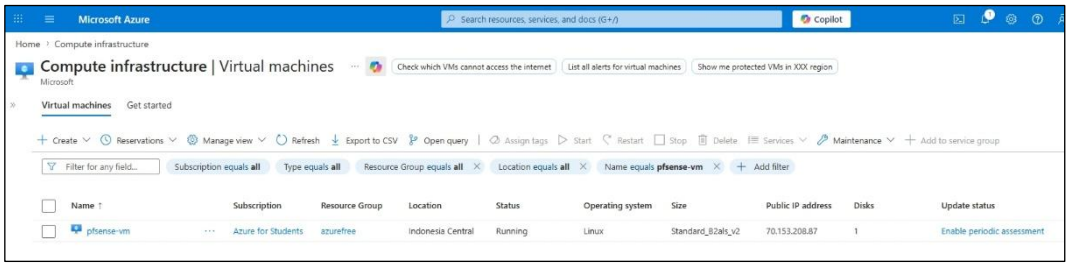
Tahap pertama dimulai dengan melakukan instalasi pfSense pada lingkungan *hypervisor* lokal. Seperti pada perancangan, *virtual machine* dikonfigurasi menggunakan dua NIC, yaitu *interface* WAN dan *interface* LAN. *Interface* WAN digunakan sebagai jalur koneksi menuju internet publik, sedangkan *interface* LAN digunakan sebagai jalur trafik untuk sistem internal seperti Wazuh SIEM dan Web Server.



Gambar 4. 7 NIC dan Konfigurasi IP pfSense

Setelah proses instalasi dasar selesai dilakukan, tahap berikutnya adalah konfigurasi dan penyesuaian pfSense seperti pengaturan alamat IP *interface*, *gateway*, serta konfigurasi dashboard pfSense. Konfigurasi ini dilakukan agar pfSense dapat berfungsi sebagai *gateway* utama sekaligus mempermudah proses administrasi sistem pada tahap implementasi berikutnya.

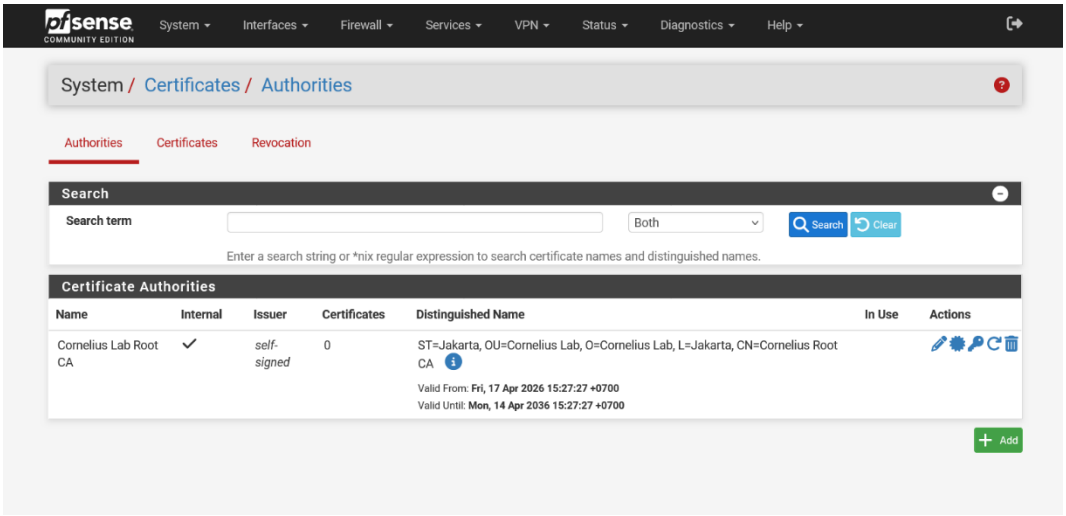
Selanjutnya, disk virtual dari VM pfSense dikonversi ke format yang kompatibel dengan Microsoft Azure. File image disk hasil konversi kemudian diunggah ke penyimpanan Azure. Image tersebut akan digunakan sebagai dasar pembuatan virtual machine baru. Dengan pendekatan tersebut, seluruh konfigurasi dasar yang sebelumnya dilakukan pada lingkungan lokal dapat tetap sama setelah proses migrasi ke Azure selesai dilakukan.



Gambar 4. 8 Deployment Virtual Machine pada Azure

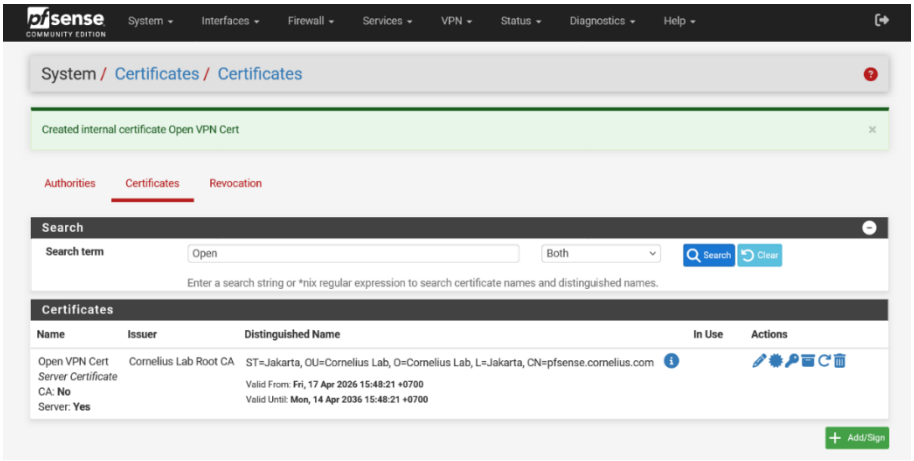
4.3.2 Kongfigurasi VPN

Sebelum melakukan instalasi dan konfigurasi lainnya, *Virtual Private Network* (VPN) diimplementasikan terlebih dahulu pada pfSense. Langkah ini dilakukan untuk membangun jalur komunikasi terenkripsi dari luar jaringan ke layanan internal. Melalui *tunnel* VPN ini, administrator dapat mengakses, mengelola, dan melakukan pengujian terhadap layanan-layanan yang berada di dalam jaringan lokal (LAN) secara aman, tanpa perlu mengekspos layanan internal tersebut secara langsung ke internet publik.



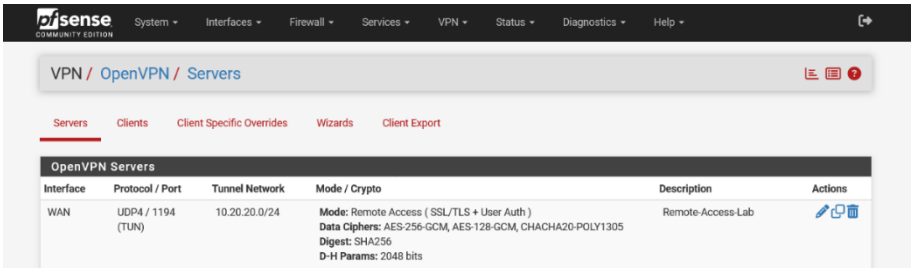
Gambar 4. 9 Konfigurasi *Certificate Authority* pada pfSense

Tahap awal melibatkan pembuatan *Certificate Authority* (CA) internal pada pfSense. CA berfungsi sebagai otoritas sertifikat internal yang digunakan untuk menerbitkan dan memvalidasi sertifikat digital pada layanan OpenVPN.



Gambar 4. 10 Konfigurasi *Certificate* VPN pada pfSense

Setelah CA berhasil dibuat, dilakukan pembuatan sertifikat server serta sertifikat pengguna (*user certificate*) yang digunakan sebagai bagian autentikasi klien.

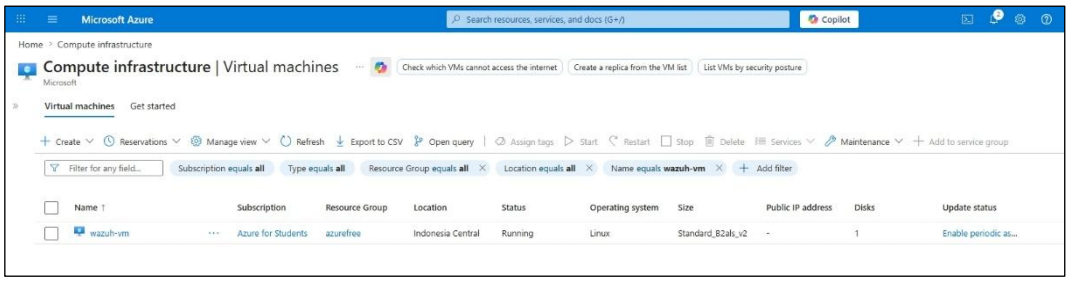


Gambar 4. 11 Konfigurasi VPN Server pada pfSense

Layanan OpenVPN Server diaktifkan pada *interface* WAN. Jaringan dikonfigurasi dengan menetapkan subnet VPN yaitu 10.20.20.0/24 sebagai Tunnel Network untuk klien, dan mendaftarkan subnet 10.10.10.0/24 pada agar klien VPN mendapatkan akses menuju jaringan LAN.

4.3.3 Instalasi Wazuh

Komponen SIEM yang digunakan yaitu Wazuh stack. Wazuh dipasang ke 1 host dengan sistem operasi Ubuntu Server 24.03. Spesifikasi yang digunakan berupa CPU 2 core dan 4 GB memori.

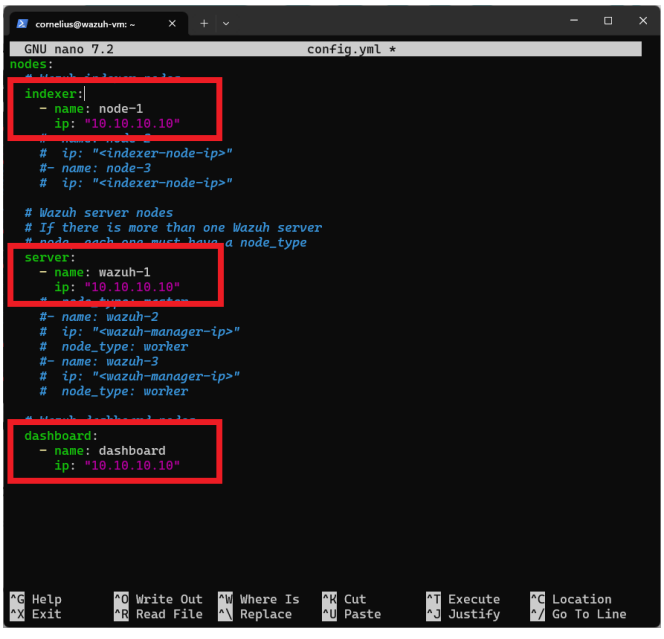


Gambar 4. 12 Deployment Virtual Machine untuk Wazuh SIEM

Konfigurasi instalasi dilakukan melalui berkas *config.yml* yang digunakan untuk menentukan parameter node dan komponen yang akan dipasang. Berkas skrip instalasi dan file konfigurasi dapat diperoleh dengan menjalankan perintah yang ditunjukkan pada Gambar 4.13.

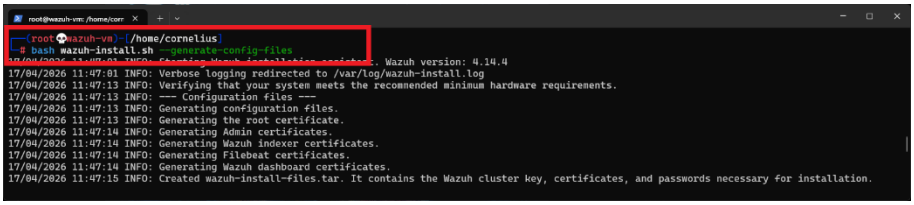
Gambar 4. 13 Instalasi *Script* Wazuh SIEM

Pada Gambar 4.14 terlihat konfigurasi yang digunakan untuk mendefinisikan komponen Wazuh yang akan dijalankan pada server. Berkas konfigurasi ini digunakan selama proses deployment seluruh layanan Wazuh Stack.



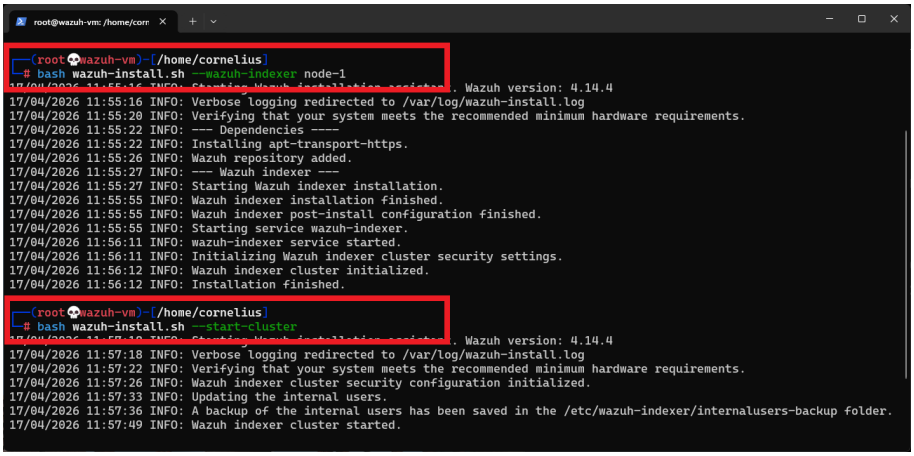
Gambar 4. 14 Konfigurasi config.yml

Setelah penyesuaian konfigurasi selesai, dilakukan proses pembuatan sertifikat dan kredensial yang dibutuhkan untuk proses Wazuh SIEM. Hasil proses tersebut ditunjukkan pada Gambar 4.15.



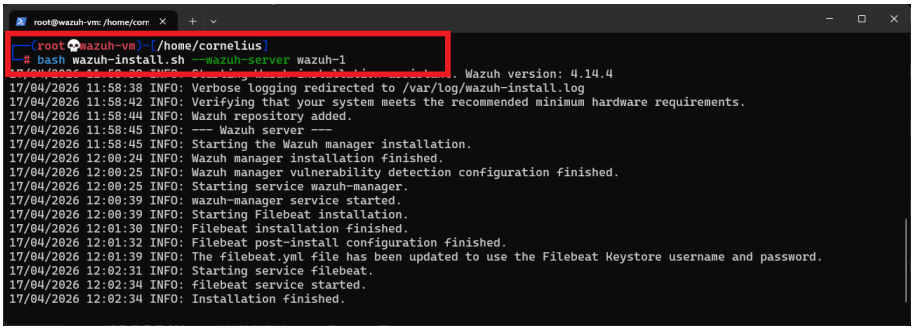
Gambar 4. 15 Proses *Generate* Komponen Instalasi Wazuh SIEM

Instalasi Wazuh Indexer dilakukan dengan menjalankan skrip instalasi menggunakan opsi `--wazuh-indexer` yang diikuti dengan nama node yang akan dipasang. Setelah Instalasi wazuh indexer selesai, script installasi dengan parameter `--start-cluster`. Tujuannya agar wazuh indexer dapat memuat informasi mengenai *certificate* dan memulai *cluster*. Gambar 4.16 menunjukkan layanan Wazuh Indexer berhasil dijalankan dan siap digunakan sebagai media penyimpanan data log.



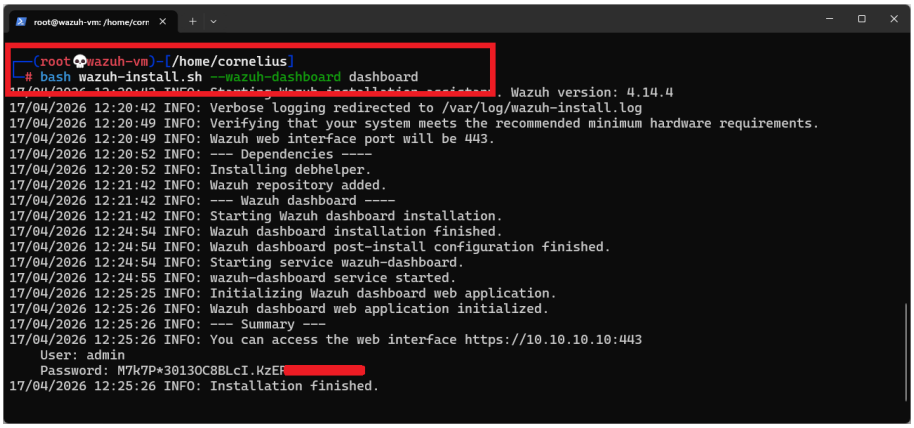
Gambar 4. 16 Instalasi Wazuh Indexer

Instalasi Wazuh Server dilakukan dengan menjalankan skrip instalasi menggunakan opsi `--wazuh-server` dan nama node yang sesuai dengan konfigurasi pada file config.yml. Gambar 4.17 menunjukkan bahwa instalasi Wazuh Server berhasil dijalankan dan siap menerima log dari perangkat yang akan diintegrasikan pada penelitian ini.



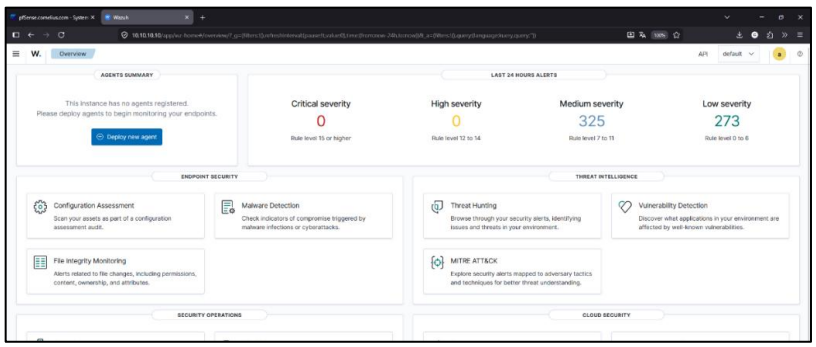
Gambar 4. 17 Instalasi Wazuh Server

Instalasi Wazuh Dashboard dilakukan dengan menjalankan skrip instalasi menggunakan opsi `--wazuh-dashboard` yang diikuti dengan nama node sesuai dengan konfigurasi yang telah dibuat



Gambar 4. 18 Instalasi Wazuh Dashboard

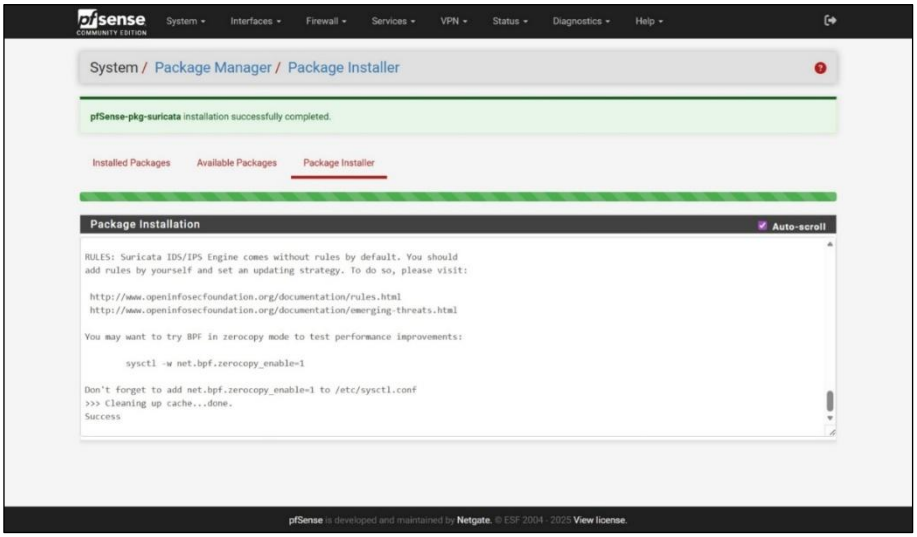
Setelah proses instalasi selesai, dashboard diakses melalui browser dan digunakan sebagai antarmuka utama untuk memantau log, melihat *alert*, serta melakukan analisis. Berdasarkan Gambar 4.19, layanan Wazuh Dashboard berhasil diakses dan terhubung dengan komponen Wazuh lainnya sehingga dapat digunakan untuk melakukan monitoring serta analisis keamanan.



Gambar 4. 19 Tampilan Wazuh Dashboard

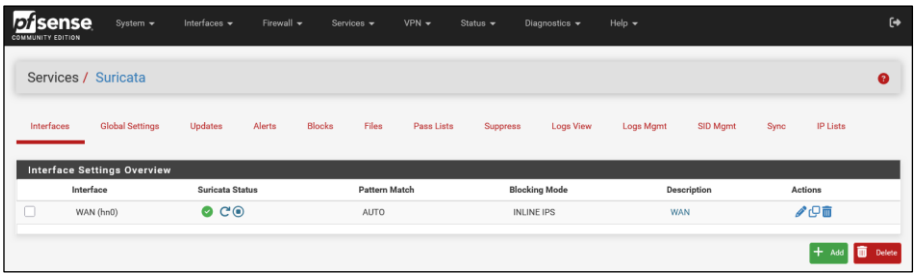
4.3.4 Installasi dan Konfigurasi Suricata

Suricata diimplementasikan sebagai komponen *Intrusion prevention system* (IPS) pada pfSense untuk mendeteksi dan memblokir aktivitas jaringan yang terindikasi sebagai serangan. Instalasi Suricata dilakukan melalui Package Manager pfSense sehingga pengelolaan konfigurasi dan perubahan *rules* dapat dilakukan pada *console* pfSense. Proses instalasi Suricata ditunjukkan pada Gambar 4.20.



Gambar 4. 20 Instalasi Modul Suricata

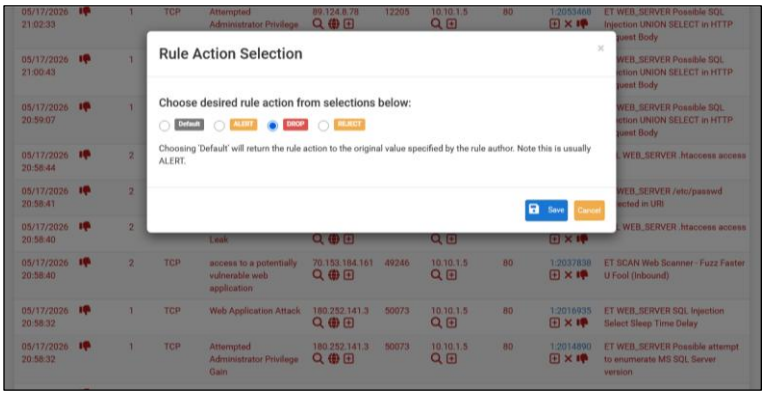
Setelah proses instalasi selesai, dilakukan pemasangan Suricata pada *interface* WAN yang menghubungkan Internet dan jaringan LAN. Penempatan Suricata pada *interface* ini bertujuan agar seluruh trafik yang berasal dari internet dapat diperiksa terlebih dahulu sebelum diteruskan menuju server internal.



Gambar 4. 21 Konfigurasi Suricata pada NIC WAN

Pada penelitian ini, Suricata dikonfigurasi menggunakan mode IPS Inline dengan dukungan Netmap. Mode ini memungkinkan Suricata melakukan inspeksi paket secara langsung (*inline*) sehingga paket yang teridentifikasi sebagai *malicious* dapat diblokir sebelum mencapai server tujuan. Dengan konfigurasi tersebut, Suricata

tidak hanya berfungsi sebagai Intrusion Detection System (IDS), tetapi juga mampu melakukan merespon serangan sebagai *Intrusion prevention system* (IPS). Konfigurasi respon IPS yang digunakan ditunjukkan pada Gambar 4.22.



Gambar 4. 22 Konfigurasi Respon Suricata

Respon Suricata yang digunakan adalah *drop* paket. Dengan respon *drop*, paket *malicious* akan dibuang secara diam-diam tanpa memberikan respons kembali ke pengirim. Pemilihan respon ini dilakukan karena penggunaan aksi lain seperti *reject* berpotensi menimbulkan konflik pada mekanisme Netmap di pfSense dan berpotensi menggunakan lebih banyak *resource*. Sementara respon *alert* merupakan repson bawaan untuk serangan yang terdeteksi dimana Suricata mencatat aktivitas trafik dalam log tanpa melakukan pemblokiran.

Suricata menggunakan *ruleset* dari *Emerging Threats* (ET) *Open Rules* sebagai *rule* utama untuk deteksi ancaman. *Rule*set ini diperbarui secara berkala untuk mendeteksi berbagai pola serangan umum seperti port scanning, SQL Injection, dan eksploitasi layanan web lainnya.

Tabel 4. 4 Rules Suricata

No.	Nama Rule	Tujuan
1	emerging-scan.rules, emerging-icmp.rules, emerging-info.rules, emerging-dos.rules, emerging-attack_response.rules, emerging-telnet.rules	Mendeteksi aktivitas reconnaissance, scanning jarigan, dan trafik suspicious lainnya

2	emerging-web_server.rules, emerging-web_client.rules, emerging-web_specific_apps.rules, emerging-exploit.rules, emerging-exploit_kit.rules, emerging-sql.rules, emerging-shellcodes.rules, emerging-hunting.rules	Mendeteksi upaya eksploitasi kerentanan sistem dan web
3	emerging-bottc.rules, emerging-bottc.portgrouped.rules, emerging-worm.rules, emerging-tor.rules,	Mendeteksi aktivitas pasca serangan, komunikasi botnet, dan Command & Control (C2).

Selain menggunakan ruleset ET Open, penelitian ini juga mengimplementasikan beberapa custom rule yang dikembangkan sesuai kebutuhan penelitian. Custom rule tersebut digunakan untuk mendeteksi pola serangan yang belum tercakup secara spesifik oleh ruleset utama.

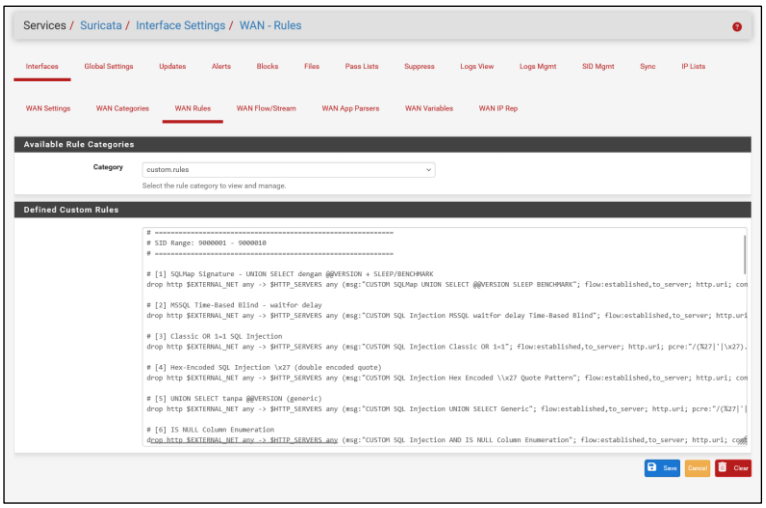
```
drop http $EXTERNAL_NET any -> $HTTP_SERVERS any (msg:"CUSTOM Credential
or Secret File Access by Filename Keyword"; flow:established,to_server;
http.uri;
pcre:"/\/(passwd|password|passwords|credentials?|secrets?|tokens?|
apikey|api[-_]key|masterkey|private[-_]key|id[-_]rsa|id[-_]dsa|id[-
_]ecdsa)[^\/]*(\?|#|$)/i"; classtype:web-application-attack;
sid:9100050; rev:1; metadata:signature_severity Critical, attack_target
Web_Server, tag Credential_Access;)
```

Aturan kustom di atas dapat dijabarkan menjadi beberapa komponen utama sebagai berikut:

- `drop`: Merupakan *rule action* yang mendefinisikan tindakan yang akan diambil apabila ada paket yang cocok dengan aturan. Pada kasus ini, Suricata akan membuang (memblokir) paket tersebut secara diam-diam.
- `http`: Mendefinisikan protokol yang diinspeksi. Pada *rule* ini, Suricata hanya akan memeriksa lalu lintas yang menggunakan protokol HTTP.
- `$EXTERNAL_NET any -> $HTTP_SERVERS any`: Merupakan alamat sumber, arah koneksi, dan alamat tujuan. Aturan ini memeriksa paket yang

berasal dari luar jaringan internal ($EXTERNAL_NET) dari *port* mana saja (*any*), yang menuju *Web Server* ($HTTP_SERVERS) pada *port* mana saja (*any*).

- `msg:"...":` Pesan atau *alert* yang akan dicatat di log apabila serangan terdeteksi.
- `flow:established,to_server`: Konfigurasi Suricata untuk hanya memeriksa paket yang merupakan bagian dari sesi TCP yang sudah terbentuk (*established*) dan paket yang mengarah ke server.
- `http.uri`: Membatasi proses pencocokan pola (*pattern matching*) hanya pada bagian *URI (Uniform Resource Identifier)* di dalam *header* HTTP.
- `pcre:"...":` Merupakan *Perl Compatible Regular Expressions* (PCRE) kata kunci yang digunakan untuk mencocokkan pola teks menggunakan expresi reguler (*regular expressions*). Pada rule ini, pcre digunakan untuk mendeteksi pola nama file yang berpotensi mengandung informasi sensitif seperti file kredensial.
- `classtype:web-application-attack`: Mengklasifikasikan jenis ancaman ini ke dalam kategori serangan aplikasi web.
- `sid:9100050; rev:1;`: *Signature ID* (SID) merupakan nomor identifikasi unik untuk *rule* ini (angka 9100050 menunjukkan ini adalah *custom rule*), sedangkan *rev* menunjukkan versi revisi dari *rule* tersebut.
- `metadata:...:` Menyediakan informasi tambahan yang akan disertakan pada log *alert*, seperti tingkat keparahan, target serangan, dan label tag.



Gambar 4. 23 Konfigurasi *Rule* Suricata

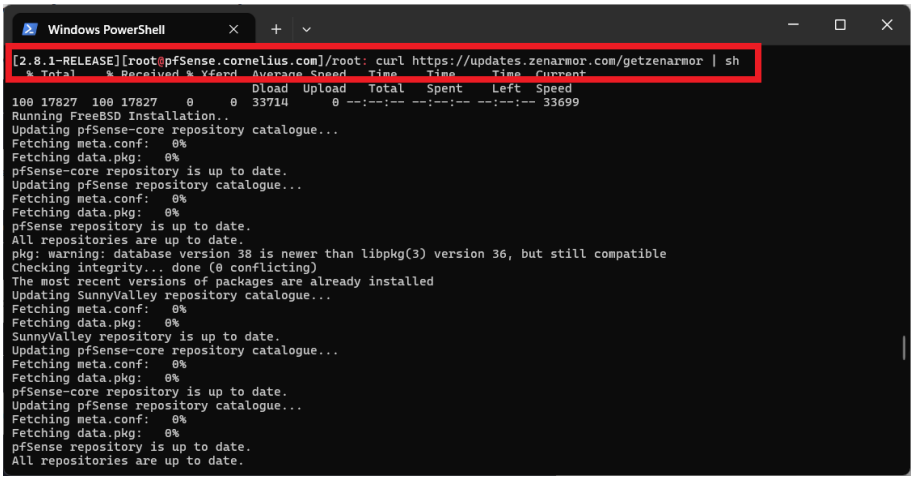
Setelah seluruh konfigurasi selesai diterapkan, Suricata mulai melakukan inspeksi terhadap seluruh trafik yang melewati antarmuka WAN sesuai ruleset dan custom rule yang telah diaktifkan. Setiap aktivitas yang terdeteksi akan menghasilkan alert, sedangkan aktivitas yang memenuhi aksi drop akan diblokir sebelum mencapai Web Server.

Seluruh alert dan log yang dihasilkan Suricata kemudian diteruskan menuju Wazuh SIEM menggunakan protokol Syslog. Integrasi ini memungkinkan proses pemantauan dan analisis dilakukan smelalui dashboard Wazuh, sekaligus mengurangi ketergantungan terhadap penyimpanan log lokal pada firewall pfSense.

4.3.5 Instalasi dan Konfigurasi Zenarmor

Zenarmor digunakan sebagai komponen *Next-Generation Firewall* (NGFW) yang menyediakan fitur *Application Control*, *Web Control*, dan *Threat Protection*. Berbeda dengan Suricata yang berfokus pada deteksi pola serangan berbasis signature, Zenarmor digunakan untuk melakukan pembatasan terhadap trafik aplikasi dan akses website yang melewati jaringan.

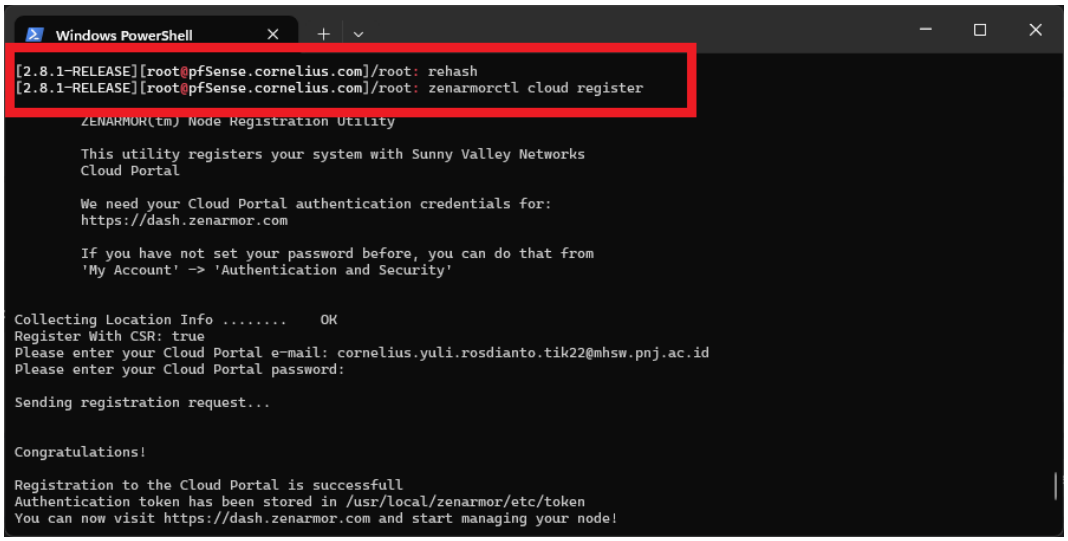
Instalasi Zenarmor dilakukan dengan mengeksekusi skrip instalasi yang disediakan secara resmi oleh pihak developer untuk sistem operasi pfSense. Metode ini digunakan karena Zenarmor tidak berjalan sebagai paket bawaan pfSense dan memerlukan komponen tambahan.



Gambar 4. 24 Instalasi Zenarmor *Engine*

Setelah proses instalasi selesai, Zenarmor *engine* perlu diregistrasikan ke layanan manajemen berbasis *cloud* milik Zenarmor yang disebut Zenconsole. Registrasi ini diperlukan agar perangkat dapat dikelola melalui dashboard Zenarmor dan memperoleh pembaruan konfigurasi maupun database keamanan secara berkala

Setelah engine Zenarmor berhasil dipasang pada pfSense, selanjutnya dilakukan registrasi perangkat ke portal manajemen *cloud* Zenarmor, yaitu Zenconsole. Registrasi ini diperlukan karena sebagian besar konfigurasi dan manajemen kebijakan keamanan dilakukan melalui dashboard tersebut.

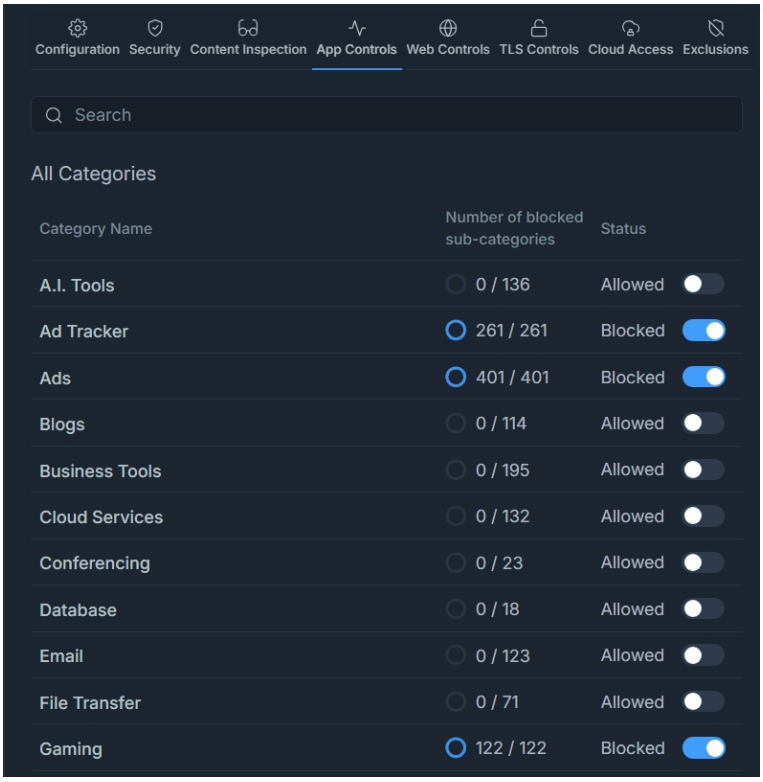


Gambar 4. 25 Registrasi Zenarmor *Engine*

Setelah *engine* berhasil terhubung dengan Zenconsole, proses konfigurasi dan manajemen dilakukan melalui portal tersebut untuk menyesuaikan *policy* keamanan serta pengaturan *filtering* sesuai dengan kebutuhan keamanan jaringan.

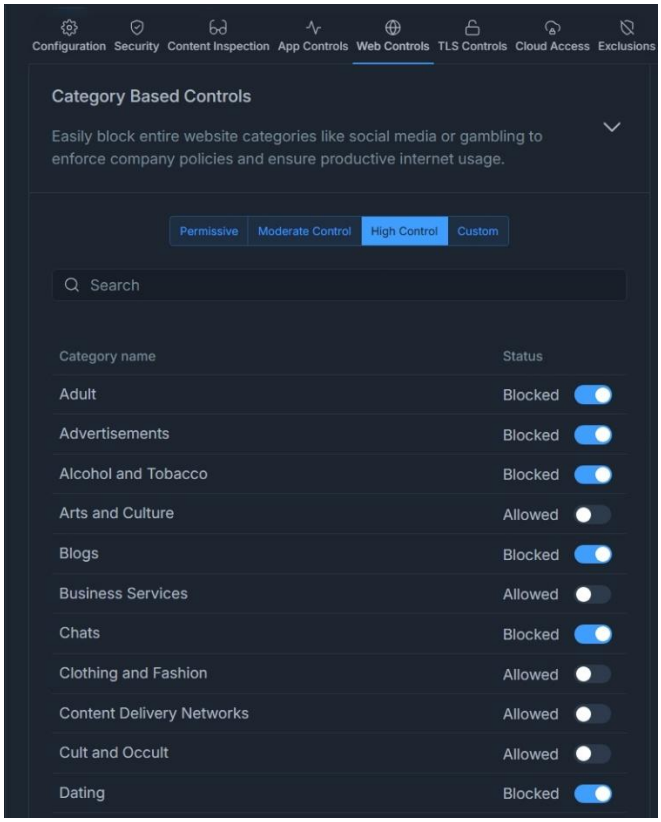
Gambar 4. 26 Tampilan Zenconsole

Pada penelitian ini, konfigurasi Zenarmor difokuskan pada beberapa fitur utama yaitu Application Control, Web Control, dan Threat Protection. Melalui fitur Application Control, administrator dapat memantau maupun membatasi akses terhadap aplikasi tertentu yang berjalan di jaringan terlepas dari alamat IP ataupun port yang digunakan. Pada penelitian ini dilakukan pembatasan berdasarkan kategori aplikasi. Kategori aplikasi yang dibatasi yaitu Ads, Ad Tracker, Gaming, Proxy, dan Social Network. Kategori Ad Tracker dan Ads dibatasi karena berpotensi membawa risiko keamanan seperti malvertising. Kategori Proxy dibatasi karena berpotensi menimbulkan celah keamanan, di mana proxy dapat digunakan untuk melewati (bypass) kebijakan filtering yang telah diterapkan. Kategori Gaming dan Social Network dibatasi karena penggunaan aplikasi tersebut berpotensi mengurangi produktivitas pengguna serta mengalihkan fokus dari kegiatan utama.



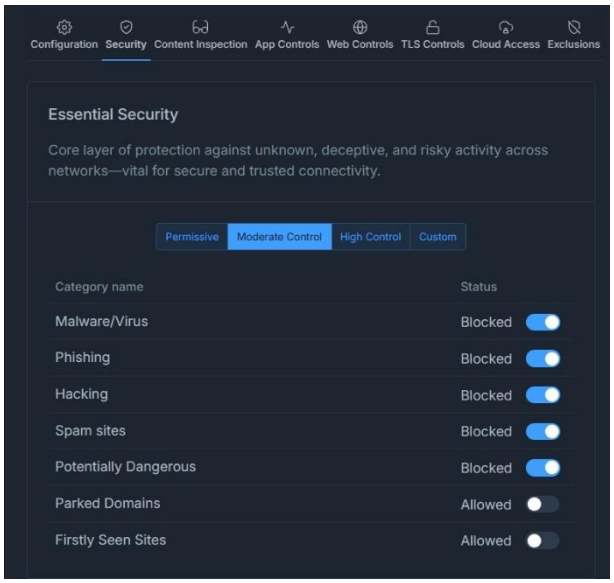
Gambar 4. 27 Konfigurasi *App Control* Zenarmor

Berbeda dengan Application Control yang pembatasannya ditentukan secara manual berdasarkan kategori aplikasi, fitur Web Control pada Zenarmor telah menyediakan mekanisme preset pembatasan kategori situs web, yaitu Restrictive, Moderate Control, High Control, dan Custom. Pada penelitian ini digunakan preset High Control karena preset ini memberikan pembatasan yang ketat terhadap kategori situs web yang berisiko bagi keamanan jaringan maupun kategori yang berpotensi mengganggu produktivitas pengguna. Kategori yang diblokir berdasarkan preset High Control yaitu Adult, Advertisement, Alcohol & Tobacco, Blogs, Chats, Dating, Forums, Gambling, Games, Hate/Violence/Illegal, Illegal Drugs, Jobs Search, Online Storage, Online Video, Pornography, Social Networks, Software Downloads, Swimsuits and Underwear, Warez, dan Weapons and Military.



Gambar 4. 28 Konfigurasi *Web Control* Zenarmor

Selain pemfilteran aplikasi dan website, fitur Threat Protection juga diaktifkan untuk mendeteksi dan memblokir koneksi yang dianggap berbahaya seperti *malware, phishing*, *hacking*, spam sites, serta domain yang terindikasi berbahaya oleh Zenarmor.

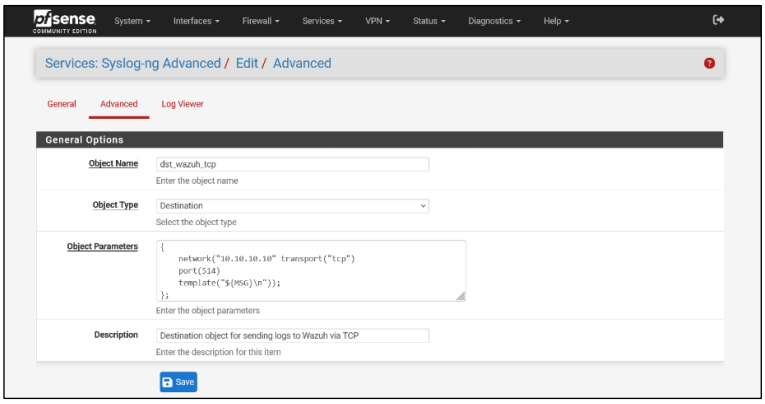


Gambar 4. 29 Konfigurasi *Threat Protection* Zenarmor

4.3.6 Integrasi SIEM

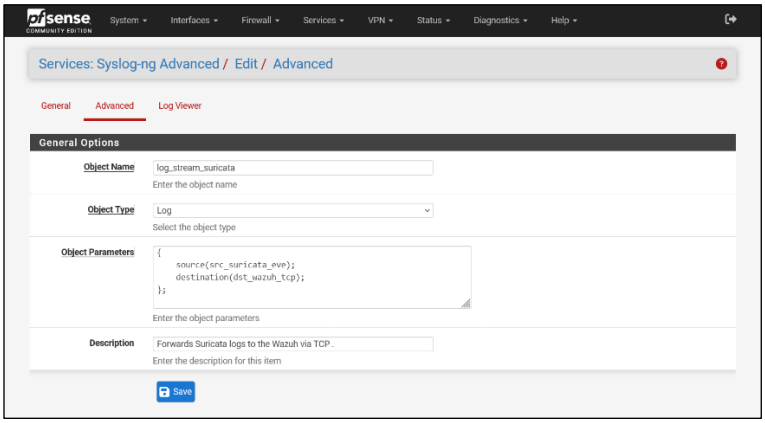
Seluruh log perangkat keamanan diintegrasikan ke dalam platform *Security Information and Event Management* (SIEM) menggunakan Wazuh. Integrasi dilakukan terhadap beberapa komponen, yaitu pfSense, Suricata, dan Zenarmor agar seluruh log keamanan dapat dikumpulkan dan dianalisis secara terpusat.

Pengiriman log Suricata menuju Wazuh dilakukan menggunakan layanan Syslog-ng pada pfSense. Penggunaan Syslog-ng dipilih karena layanan syslog bawaan pfSense hanya mendukung pengiriman log menggunakan protokol UDP, sedangkan log Suricata terutama dalam format eve.json sering memiliki ukuran data yang cukup besar.



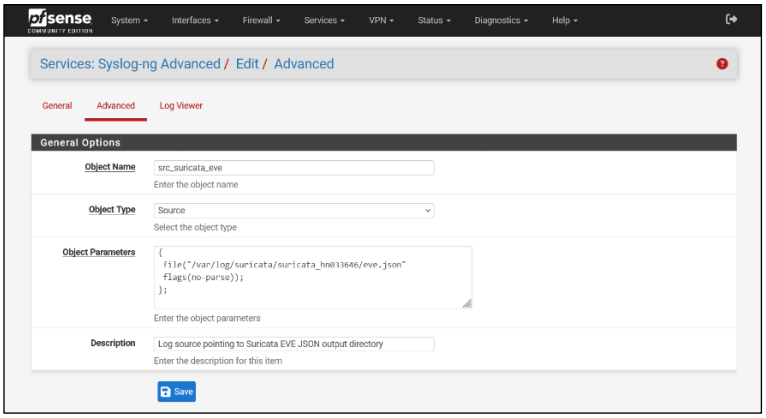
Gambar 4. 30 Konfigurasi Tujuan Pengiriman Log pada Syslog-ng

Gambar 4.30 menunjukkan konfigurasi tujuan pengiriman log yang mengarahkan seluruh log Suricata menuju server Wazuh.



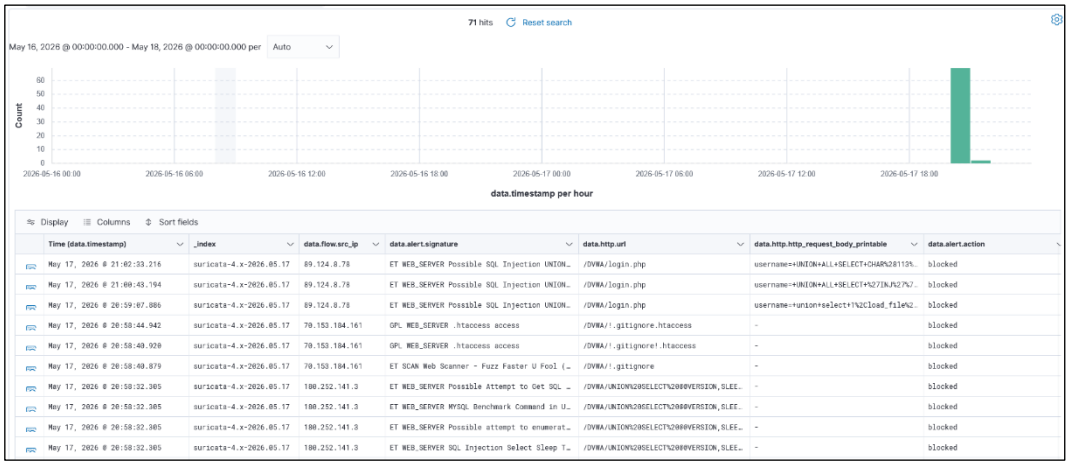
Gambar 4. 31 Konfigurasi Sumber Log Suricata pada Syslog-ng

Pada Gambar 4.31 ditunjukkan konfigurasi sumber log yang digunakan oleh Syslog-ng untuk membaca log yang dihasilkan oleh Suricata.



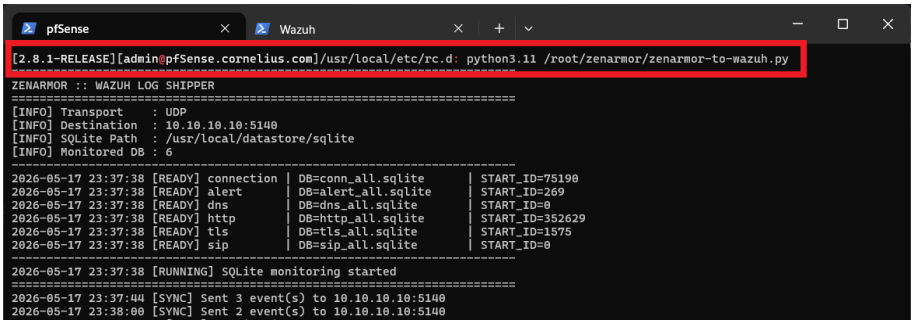
Gambar 4. 32 Konfigurasi Forwarding Log pada Syslog-ng

Setelah sumber dan tujuan log dikonfigurasi, Syslog-ng meneruskan log Suricata ke Wazuh menggunakan protokol TCP. Penggunaan protokol TCP dipilih karena protokol UDP menyebabkan log terpotong atau tidak terkirim secara utuh. Protokol UDP tidak memiliki mekanisme verifikasi pengiriman paket, selain itu pengiriman melalui UDP memiliki batas ukuran *byte* yang dapat dikirim.



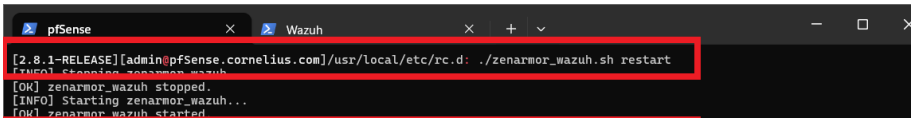
Gambar 4. 33 Log Suricata Berhasil Diterima Wazuh

Zenarmor fitur native untuk melakukan pengiriman log langsung menuju platform SIEM. Namun fitur tersebut hanya tersedia pada versi berlisensi, sedangkan versi gratis hanya menyediakan opsi untuk menyimpan log pada database SQLite lokal. Untuk mengatasi keterbatasan tersebut, dibuat script Python yang bertugas mengambil log dari database SQLite Zenarmor.



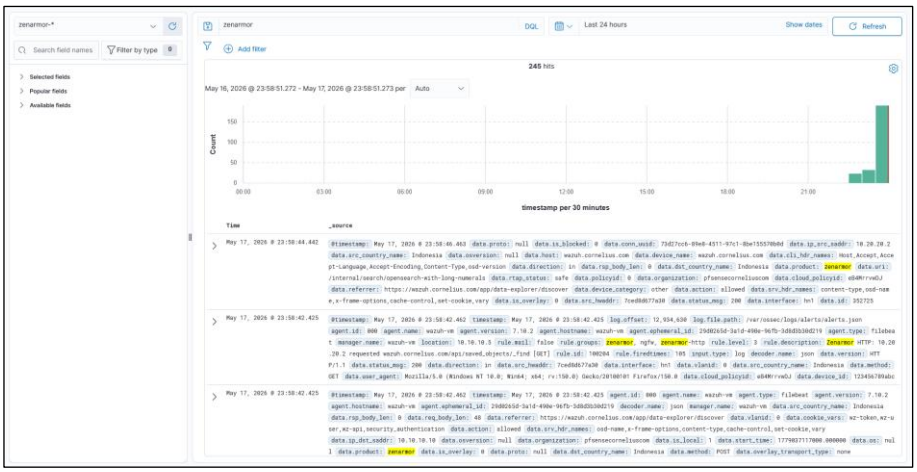
Gambar 4. 34 Script Pengambilan Log Zenarmor

Script tersebut secara berkala membaca database SQLite yang digunakan oleh Zenarmor untuk menyimpan log trafik. Data yang diperoleh kemudian diproses dan diformat ulang agar sesuai dengan format log yang dapat dikenali oleh Wazuh. Pendekatan ini memungkinkan informasi keamanan dari Zenarmor tetap dapat dikirim dan dianalisis melalui platform SIEM tanpa memerlukan lisensi tambahan.



Gambar 4. 35 Script Mulai Otomatis Zenarmor

Agar proses tersebut berjalan secara otomatis, dibuat bash script pada direktori `/usr/local/etc/rc.d` di pfSense untuk menjalankan script Python setiap sistem pfSense melakukan booting. Melalui script tersebut, proses pengambilan dan pengiriman log Zenarmor dapat berjalan secara otomatis tanpa perlu dijalankan secara manual oleh administrator.

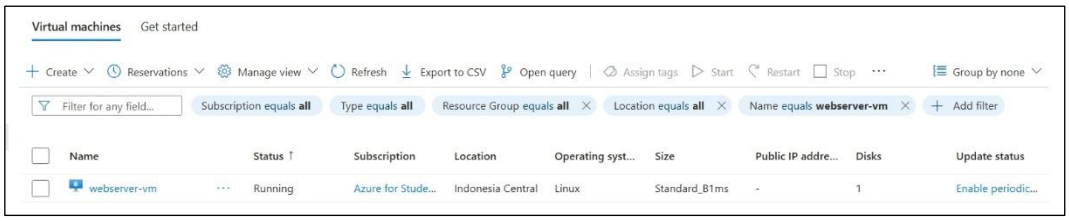


Gambar 4. 36 Log Zenarmor pada Wazuh SIEM

Gambar 4.36 menunjukkan bahwa log Zenarmor berhasil diterima dan ditampilkan pada Wazuh. Dengan terintegrasinya pfSense, Suricata, dan Zenarmor ke dalam Wazuh, administrator dapat melakukan monitoring keamanan secara terpusat serta memperoleh visibilitas yang lebih baik terhadap aktivitas jaringan dan insiden keamanan yang terjadi pada sistem

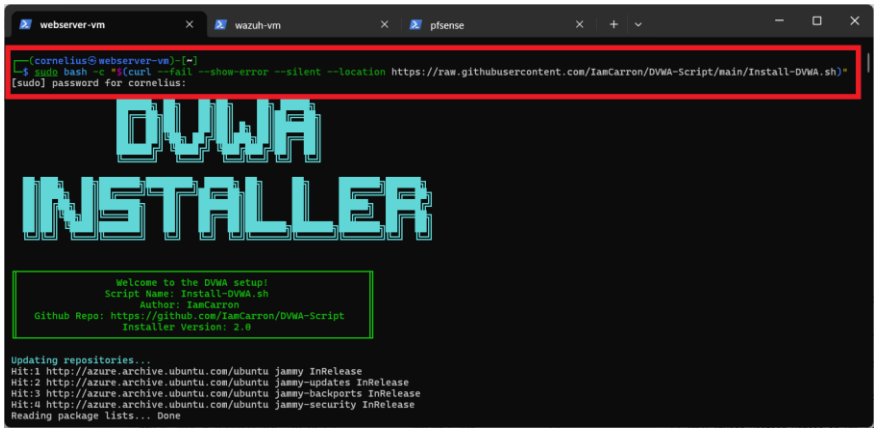
4.3.7 Instalasi Web Server

Web Server diimplementasikan sebagai target utama pada skenario pengujian keamanan yang dilakukan dalam penelitian ini. Server menggunakan sistem operasi Ubuntu Server dan menjalankan aplikasi *Damn Vulnerable Web Application* (DVWA) sebagai target simulasi serangan. DVWA dipilih karena menyediakan berbagai kerentanan yang umum ditemukan pada aplikasi web sehingga dapat digunakan untuk menguji kemampuan deteksi dan pencegahan yang dimiliki oleh NGFW.



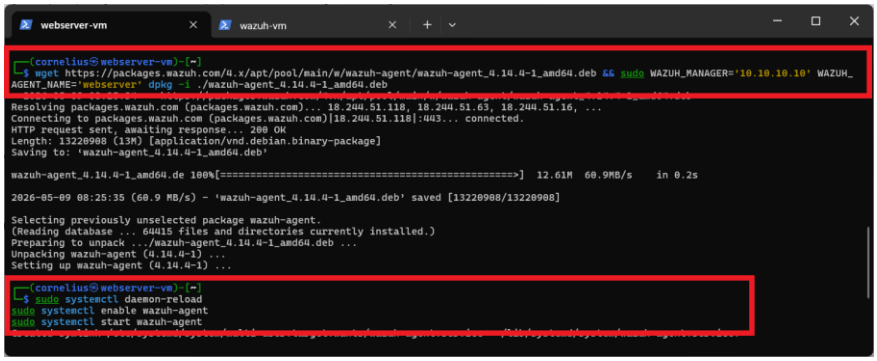
Gambar 4. 37 Deployment Virtual Machine Web Server pada Azure

Gambar 4.37 menunjukkan virtual machine yang digunakan sebagai Web Server pada lingkungan Microsoft Azure. Server dikonfigurasi dengan spesifikasi 1 vCPU dan 2 GB RAM yang dinilai cukup untuk menjalankan layanan web dan mendukung kebutuhan pengujian pada penelitian ini. Setelah Virtual Machine selesai dikonfigurasi, dilakukan instalasi DVWA beserta dependensi yang dibutuhkan. Instalasi dilakukan secara otomatis menggunakan skrip untuk mempermudah proses deployment dan memastikan konfigurasi yang digunakan tetap konsisten.



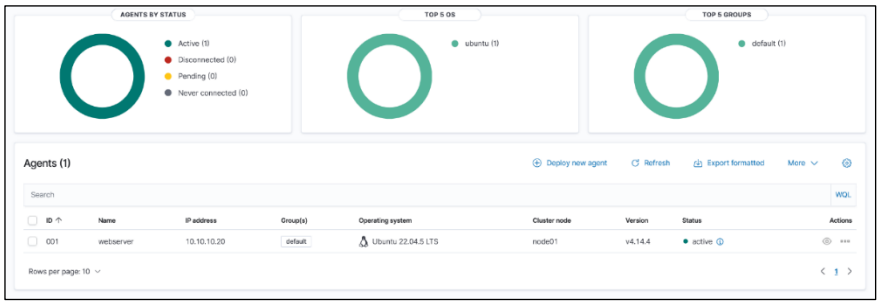
Gambar 4. 38 Instalasi DVWA

Selain aplikasi web target, pada server ini juga dipasang Wazuh Agent yang bertindak sebagai Host-based Intrusion Detection System (HIDS). Pemasangan agen ini dilakukan untuk memantau aktivitas pengguna, serta mendeteksi request ataupun aktivitas yang terindikasi *malicious*.



Gambar 4. 39 Instalasi Wazuh Agent

Apabila tahap instalasi berhasil, maka server yang telah terpasang wazuh agent akan tampil sebagai agent yang aktif. Gambar 4.40 menunjukkan bahwa Wazuh Agent berhasil terhubung dengan Wazuh Server dan terdeteksi sebagai agent aktif pada dashboard.



Gambar 4. 40 Wazuh Agent Terdeteksi pada Dashboard SIEM

4.4 Pengujian

Pengujian dilakukan untuk mengevaluasi kemampuan sistem dalam mendeteksi serangan dan merespon serangan yang terjadi. Selain itu, pengujian juga dilakukan untuk mengetahui pengaruh implementasi sistem keamanan terhadap performa jaringan dan kestabilan layanan. Ditentukan parameter-parameter keberhasilan untuk mengetahui apakah fungsional dan performa sistem sudah sesuai dengan yang diharapkan

4.4.1 Deskripsi Pengujian

Pengujian sistem dilakukan untuk mengevaluasi fungsi keamanan, monitoring, dan performa sistem yang telah dibangun. Pengujian dilakukan pada seluruh komponen dan dibagi menjadi beberapa jenis yaitu pengujian fungsionalitas sistem, pengujian keamanan sistem, pengujian *Quality of Service* (QoS), serta pengujian monitoring serangan harian. Setiap pengujian dilakukan dengan parameter dan skenario yang berbeda untuk mengetahui kemampuan sistem dalam mendeteksi, memblokir, dan memonitor aktivitas keamanan jaringan.

4.4.1.1 Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan untuk memastikan setiap komponen berjalan, dikonfigurasi dengan baik, dan terintegrasi sesuai dengan arsitektur yang dibuat. Pengujian dilakukan dengan memastikan komponen firewall pfSense, Suricata, Zenarmor, proses *forwarding* log, serta SIEM dapat berjalan sesuai fungsi masing-masing. Sistem dianggap berhasil apabila sebagian besar komponen berfungsi sesuai dengan parameter yang ada.

4.4.1.2 Pengujian Ancaman Eksternal

Pengujian ancaman eksternal dilakukan untuk memastikan bahwa sistem dapat mengetahui, menerima, dan merespon terhadap ancaman yang dari luar berupa aktivitas *web scanning* dan *exploitation*. Sistem dianggap berhasil apabila mampu mendeteksi ancaman, menghasilkan *alert*, serta melakukan blocking terhadap ancaman yang ada. Pengujian dilakukan menggunakan bantuan tools *penetration testing* yaitu nmap, ffuz, sqlmap, dan hping3 untuk menguji 4 jenis serangan yaitu port scanning, directory scanning, SQL injection, dan Denial of Service (DoS).

4.4.1.3 Pengujian Ancaman Internal

Pengujian ancaman internal dilakukan untuk mengetahui kemampuan sistem dalam mendeteksi dan melakukan filtering terhadap trafik yang berasal dari internal, baik berupa akses terhadap aplikasi dan situs web yang tidak sesuai dengan kebijakan internal, maupun akses menuju domain atau URL yang bersifat malicious. Pengujian terhadap akses aplikasi dan situs web bertujuan untuk mengukur efektivitas fitur *Application Control* dan *Web Control* dalam membatasi kategori aplikasi maupun kategori situs web yang telah ditentukan pada kebijakan keamanan yang diterapkan. Sementara itu, pengujian terhadap domain dan URL malicious bertujuan untuk mensimulasikan skenario ketika perangkat pada jaringan internal terinfeksi dan mencoba mengunduh *malware* maupun backdoor, guna mengetahui kemampuan fitur *Threat protection* dalam mendeteksi aktivitas mencurigakan tersebut serta memutus koneksi secara paksa.

Daftar domain dan URL malicious yang digunakan pada pengujian ini diperoleh dari URLhaus, yaitu platform berbagi Indicators of Compromise (IoC) yang menyediakan informasi mengenai domain dan URL yang diketahui digunakan untuk distribusi *malware* maupun aktivitas berbahaya lainnya. Adapun daftar kategori aplikasi dan situs web yang diuji mengacu pada kategori yang telah dibatasi melalui kebijakan *Application Control* dan *Web Control* pada Zenarmor

4.4.1.4 Pengujian Akurasi Deteksi

Pengujian akurasi deteksi dilakukan untuk mengukur serangan yang berhasil dikenali dan dideteksi oleh firewall. Pengujian ini bertujuan untuk mengetahui tingkat akurasi firwall dalam mendeteksi aktivitas serangan yang dikirim menuju Web Server serta mengidentifikasi kemungkinan terjadinya *false positive* dan *false negative* selama proses deteksi.

Pengujian dilakukan dengan mengirimkan sejumlah *request malicious* dan *request benign* menuju Web Server. *Request malicious* terdiri dari *payload* dan pola akses yang umum digunakan dalam serangan aplikasi web, seperti SQL *Injection*, *Cross Site Scripting* (XSS), *Local File Inclusion* (LFI), dan *Directory Traversal*. Sementara itu, *request benign* terdiri dari *endpoint* yang umum diakses oleh pengguna normal saat menggunakan aplikasi web. Seluruh request yang dikirim

akan diamati melalui *alert* yang dihasilkan Suricata, log trafik Zenamor, log yang diterima oleh Wazuh Agent HIDS pada Server. Hasil pengujian kemudian diklasifikasikan ke dalam kategori *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

4.4.1.5 Pengujian Serangan Harian

Pengujian serangan harian dilakukan untuk mengamati aktivitas serangan yang diterima server dari internet publik. Pengujian ini bertujuan untuk memperoleh gambaran mengenai ancaman nyata yang diterima oleh server serta mengevaluasi kemampuan sistem keamanan dalam mendeteksi dan merespons aktivitas yang terindikasi sebagai serangan.

Pengujian dianggap berhasil apabila sistem mampu mengumpulkan dan menampilkan informasi serangan yang terjadi selama periode pengamatan, serta memberikan visibilitas yang memadai terhadap aktivitas yang berpotensi membahayakan server..

4.4.1.6 Pengujian Quality of Service (QoS)

Pengujian *Quality of Service* (QoS) dilakukan untuk mengetahui pengaruh implementasi sistem keamanan terhadap performa jaringan. Parameter yang diuji meliputi *delay*, *jitter*, *throughput*, dan packet loss sebelum dan sesudah fitur keamanan diaktifkan. Pengujian dianggap berhasil apabila sistem keamanan tetap dapat berjalan tanpa menyebabkan penurunan performa jaringan yang signifikan.

4.4.2 Prosedur Pengujian

4.4.2.1 Prosedur Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan mencoba dan memastikan setiap fungsi dari komponen dengan menggunakan *black box testing*. Tabel 4.5 menunjukkan detail pengujian fungsionalitas sistem.

Tabel 4. 5 Skenario Pengujian Fungsionalitas

ID	Skenario Pengujian	Tujuan Pengujian
PF-01	Pengujian koneksi VPN	Memastikan client dapat terhubung ke jaringan internal melalui VPN

PF-02	Pengujian akses dashboard Wazuh dengan VPN aktif	Memastikan dashboard Wazuh bekerja dan dapat diakses melalui VPN
PF-03	Pengujian akses dashboard pfSense dengan VPN aktif	Memastikan dashboard pfSense berkerja dan dapat diakses melalui VPN
PF-04	Pengujian konektivitas internet server internal	Memastikan server internal dapat mengakses internet
PF-05	Pengujian akses Web service melalui internet	Memastikan Web Server dapat diakses dari internet melalui NAT dan *routing* firewall
PF-06	Pengujian pemblokiran serangan oleh Suricata	Memastikan Suricata mendeteksi merespon terhadap serangan yang masuk
PF-07	Pengujian forwarding log Suricata	Memastikan log/*alert* Suricata dikirim ke Wazuh SIEM
PF-08	Pengujian melakukan request menuju domain *malicious*	Memastikan Zenarmor mendeteksi dan merespon merespon aktivitas mencurigakan
PF-09	Pengujian forwarding log Zenarmor	Memastikan log Zenarmor dikirim ke Wazuh SIEM dengan bantuan script

4.4.2.2 Prosedur Pengujian Ancaman Eksternal

Pengujian ancaman eksternal dilakukan untuk mengetahui sistem dapat mendeteksi dan merespon terhadap serangan yang berasal dari jaringan ekternal ataupun internet. Pengujian dilakukan dengan mengirimkan request ataupun parameter *malicious* menuju layanan web server DVWA yang dapat diakses melalui internet. Seluruh serangan dilakukan menggunakan host external yaitu virtual machine yang berada pada Azure dan mengirimkan serangan dari internet untuk mensimulasikan serangan dari internet publik.

Seluruh pengujian yang dilakukan memiliki 2 kondisi yaitu ketika layanan firewall dan sistem keamanan aktif dan ketika layanan nonaktif. Selain itu, beberapa pengujian dilakukan dengan tingkat intensitas berbeda seperti peningkatan jumlah *request*, *concurrency*, ataupun rate pengiriman paket. Hal ini bertujuan untuk mengetahui efektivitas sistem keamanan dalam menghadapi variasi intensitas serangan.

Pengujian *port scanning* dilakukan untuk mengidentifikasi port terbuka dan layanan yang berjalan pada server target. Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam mendeteksi aktivitas *reconnaissance* yang umum dilakukan penyerang sebelum melakukan eksploitasi lebih lanjut.

Tools yang digunakan pada pengujian ini adalah Nmap dengan mode *scanning* TCP SYN *Scan* dan *service enumeration*. Tabel 4.6 menunjukkan skenario dan *command* yang digunakan pada pengujian port scanning.

Tabel 4. 6 Skenario dan Command Pengujian Port Scanning

ID	Skenario	Tools	Command
EX-01	TCP SYN Scan dengan firewall nonaktif	nmap	`nmap -sS -Pn -T3 --max-retries 0 --min-rate 200 --top-ports 1000 <target>`
EX-02	TCP SYN Scan dan service enumeration dengan firewall nonaktif	nmap	`nmap -sS -Pn -T5 --max-retries 0 -sV -p- <target>`
EX-03	TCP SYN Scan dengan firewall aktif	nmap	`nmap -sS -Pn -T3 --max-retries 0 --min-rate 200 --top-ports 1000 <target>`
EX-04	TCP SYN Scan dan service enumeration dengan firewall aktif	nmap	`nmap -sS -Pn -T5 --max-retries 0 -sV -p- <target>`

Pengujian SQL *Injection* dilakukan untuk mengetahui kemampuan sistem keamanan dalam mendeteksi dan memblokir *payload* SQL Injection yang dikirim menuju aplikasi DVWA. Pengujian dilakukan menggunakan tools sqlmap untuk melakukan enumerasi *database* serta mengirimkan serangan SQL *Injection* secara otomatis.

Tabel 4. 7 Skenario dan Command Pengujian SQL Injection

ID	Skenario	Tools	Command
EX-05	SQL Injection dengan firewall nonaktif	sqlmap	`sqlmap -u <target> --cookie=<cookie-target> security=low" --batch --dbs --random-agent --flush-session --timeout=5 --retries=2`

EX-06	SQL Injection intensitas tinggi dengan firewall nonaktif	sqlmap	`sqlmap -u <target> --cookie=<cookie-target> --batch --level=5 --risk=3 --dbs --dump --random-agent --flush-session --timeout=5 --retries=2`
EX-07	SQL Injection dengan firewall aktif	sqlmap	`sqlmap -u <target> --cookie=<cookie-target> security=low" --batch --dbs --random-agent --flush-session --timeout=5 --retries=2`
EX-08	SQL Injection intensitas tinggi dengan firewall aktif	sqlmap	`sqlmap -u <target> --cookie=<cookie-target> --batch --level=5 --risk=3 --dbs --dump --random-agent --flush-session --timeout=5 --retries=2`

Pengujian *directory scanning* dilakukan untuk mengetahui kemampuan sistem keamanan dalam mendeteksi aktivitas enumerasi direktori dan file sensitif yang dapat diakses secara publik pada web server. Pengujian dilakukan menggunakan *tool* FFUF dengan kombinasi *wordlist* yang berasal dari berbagai repositori *security* testing. Penggunaan beberapa wordlist bertujuan untuk meningkatkan variasi enumerasi direktori.

Tabel 4. 8 Skenario dan Command Pengujian Directory Scanning

ID	Skenario	Tools	Command
EX-09	Directory scanning dengan firewall tidak aktif	FFUF	`ffuf -u <target> -w wordlists.txt -ac -c -noninteractive -t 50 -timeout 1 -rate 50`
EX-10	Directory scanning intensitas tinggi dengan firewall tidak aktif	FFUF	`ffuf -u <target> -w wordlists.txt -ac -c -noninteractive -t 100 -timeout 1 -rate 50`
EX-11	Directory scanning dengan firewall aktif	FFUF	`ffuf -u <target> -w wordlists.txt -ac -c -noninteractive -t 50 -timeout 1 -rate 50`

EX-12	Directory scanning intensitas tinggi dengan firewall aktif	FFUF	`ffuf -u <target> -w wordlists.txt -ac -c -noninteractive -t 100 -timeout 1 -rate 50`

Pengujian *Denial of Service* (DoS) dilakukan untuk mengetahui kemampuan sistem keamanan dalam mendeteksi dan memitigasi serangan *flooding* terhadap server target. Pengujian dilakukan dengan mengirimkan trafik dalam jumlah besar menggunakan paket TCP SYN secara terus menerus menuju server target. Tool yang digunakan pada pengujian ini adalah hping3. Selain melakukan pengamatan terhadap *alert* keamanan yang dihasilkan, pengujian ini juga mengukur ketersediaan layanan server ketika serangan berlangsung untuk mengetahui apakah layanan masih dapat diakses oleh pengguna normal. Selain itu dilakukan pengukuran terhadap penggunaan *resource* server target berupa penggunaan CPU dan memori RAM. Tabel 4.9 menunjukkan skenario dan perintah yang digunakan dalam pengujian DoS.

Tabel 4. 9 Skenario dan Command Pengujian DoS

ID	Skenario	Tools	Command
EX-13	SYN Flood dengan firewall nonaktif	hping3	`hping3 -S -p 80 --flood <target>`
EX-14	SYN Flood dengan firewall aktif	hping3	`hping3 -S -p 80 --flood <target>`

4.4.2.3 Prosedur Pengujian Ancaman Internal

Pengujian ancaman internal dilakukan dengan menjalankan script Python untuk melakukan request secara otomatis terhadap daftar URL dan domain yang telah disiapkan, yaitu kategori aplikasi/situs web yang dibatasi maupun domain/URL malicious dari URLhaus. Pengujian dilakukan dalam dua skenario, yaitu kondisi keamanan firewall nonaktif dan aktif, guna menguji fitur *Web Control*, *Application Control*, serta Threat Protection.

Pada skenario firewall nonaktif, Zenarmor dan Suricata dinonaktifkan terlebih dahulu untuk mensimulasikan kondisi trafik jaringan sebelum penerapan firewall,

sehingga dapat diketahui apakah seluruh request ke daftar URL/domain berhasil diakses. Selanjutnya, pada skenario firewall aktif, Zenarmor dan Suricata diaktifkan kembali untuk menguji apakah firewall dapat memblokir request yang tidak sesuai dengan kebijakan keamanan yang telah diterapkan. Hasil dari kedua skenario tersebut kemudian dibandingkan untuk mengetahui efektivitas fitur *Application Control*, *Web Control*, dan *Threat protection* dalam mendeteksi dan memblokir trafik yang melanggar kebijakan. Tabel 4.10 menunjukkan skenario pengujian ancaman internal yang dilakukan pada penelitian ini.

Tabel 4. 10 Skenario Pengujian Ancaman Internal

ID	Skenario	Keterangan
IN-01	Akses 20 website yang tidak sesuai dengan kebijakan saat keamanan firewall nonaktif	Melakukan akses ke website yang termasuk dalam kategori yang dibatasi oleh kebijakan ketika firewall dalam kondisi nonaktif.
IN-02	Akses 20 website yang tidak sesuai dengan kebijakan saat keamanan firewall aktif	Melakukan akses ke website yang termasuk dalam kategori yang dibatasi oleh kebijakan ketika firewall dalam kondisi aktif, untuk menguji efektivitas fitur *Application Control* dan *Web Control*.
IN-03	Akses 14.xxx domain atau URL malicious saat keamanan firewall nonaktif	Melakukan akses ke domain atau URL yang telah teridentifikasi sebagai malicious ketika firewall dalam kondisi nonaktif.
IN-04	Akses 14.xxx domain atau URL malicious saat keamanan firewall aktif	Melakukan akses ke domain atau URL yang telah teridentifikasi sebagai malicious ketika firewall dalam kondisi aktif, untuk menguji kemampuan fitur *Threat protection* dalam mendeteksi dan memblokir akses ke sumber yang mengandung ancaman keamanan, seperti *malware* maupun phishing.

4.4.2.4 Prosedur Pengujian Akurasi Deteksi

Pengujian akurasi deteksi dilakukan untuk mengevaluasi kemampuan NGFW dalam membedakan antara trafik yang bersifat *malicious* dan trafik yang bersifat

benign. Pengujian ini bertujuan untuk mengukur efektivitas mekanisme deteksi yang diterapkan oleh Suricata dalam mengidentifikasi aktivitas yang berpotensi membahayakan serta meminimalkan kesalahan deteksi terhadap trafik normal.

Pengujian dilakukan dengan mengirimkan sejumlah request HTTP menuju aplikasi DVWA menggunakan tool FFUF. Total pengujian terdiri dari 3.000 request yang terbagi menjadi dua kategori. Sebanyak 2.000 request menggunakan *payload* yang mewakili berbagai teknik serangan aplikasi web seperti SQL Injection, Cross-Site Scripting (XSS), Local File Inclusion (LFI), dan Directory Traversal. *Payload* tersebut digunakan sebagai representasi trafik *malicious* yang diharapkan dapat dideteksi oleh sistem keamanan. Sementara itu, 1.000 request lainnya menggunakan *payload benign* yang merepresentasikan aktivitas normal pengguna ketika mengakses layanan web. Tabel 4.11 merangkum pengujian yang dilakukan.

Tabel 4. 11 Skenario Pengujian Akurasi Deteksi

ID	Skenario	Tools	Jumlah
AD-01	Pengiriman request menggunakan *payload* serangan aplikasi web	ffuf	2.000
AD-02	Pengiriman request menggunakan *payload benign*	ffuf	1.000

4.4.2.5 Prosedur Pengujian Serangan Harian

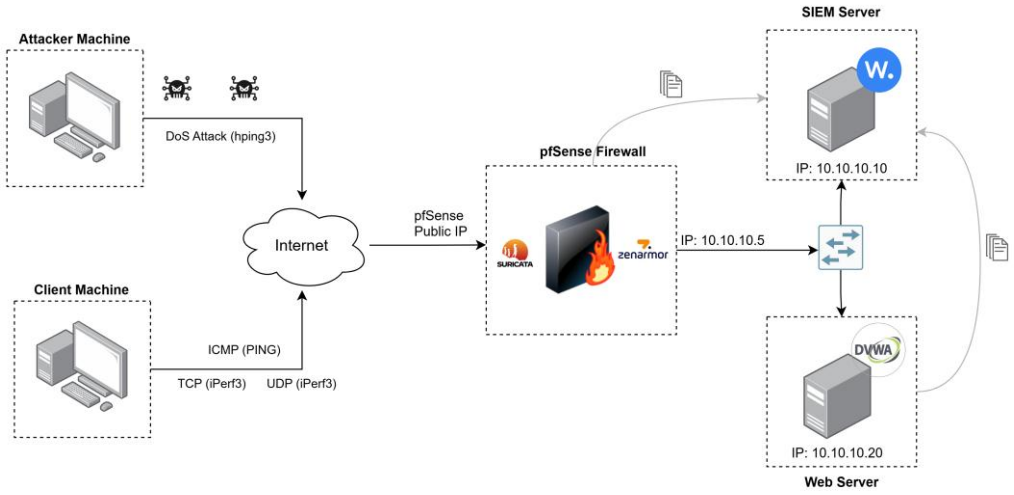
Pengujian serangan harian dilakukan dengan menempatkan Web Server pada jaringan publik sehingga dapat diakses secara langsung melalui internet. Selama periode pengamatan, seluruh aktivitas jaringan yang diterima oleh server dicatat dan dianalisis menggunakan Suricata, Zenarmor, dan Wazuh SIEM.

Data yang dikumpulkan meliputi jumlah *alert* yang diterima. Seluruh data hasil pengamatan kemudian direkap dan dikelompokkan berdasarkan kategori serangan untuk mengetahui pola ancaman yang diterima oleh Web Server selama berada pada jaringan publik.

4.4.2.6 Prosedur Pengujian Quality of Service (QoS)

Pengujian Quality of Service (QoS) dilakukan untuk mengetahui pengaruh implementasi sistem keamanan terhadap performa jaringan. Pengujian dilakukan

dengan membandingkan kualitas jaringan ketika layanan keamanan dinonaktifkan, ketika layanan keamanan seperti Suricata dan Zenarmor diaktifkan, ketika firewall menerima serangan yaitu berupa DoS, dan keadaan firewall pasca serangan. Parameter yang diuji meliputi *delay*, *jitter*, *throughput*, dan *packet loss*.



Gambar 4. 41 Topologi Pengujian QoS

Pengukuran dilakukan menggunakan satu virtual machine tambahan yang ditempatkan pada lingkungan Microsoft Azure yaitu Client Machine. Machine tersebut berperan sebagai pengguna yang sah dan melakukan pengujian menuju Web Server melalui firewall. Penempatan Client Machine pada lingkungan cloud bertujuan untuk meminimalkan pengaruh faktor eksternal seperti keterbatasan bandwidth ataupun kualitas jaringan internet lokal.

Pada skenario Firewall Mengalami Serangan, trafik pengujian QoS dikirimkan oleh Client Machine bersamaan dengan trafik serangan DoS yang dikirimkan dari Attacker Machine. Skenario tersebut mensimulasikan situasi ketika pengguna yang sah tetap mengakses layanan pada saat sistem sedang menerima serangan.

Pengukuran pengukuran *delay* dilakukan dengan mengirimkan paket ICMP dari Client Machine menuju Web Server dan menghitung nilai rata-rata *Round Trip Time* (RTT) yang diperoleh. Nilai RTT kemudian dibagi 2 untuk mendapatkan perkiraan *One Way Delay* (OWD). Pada pengukuran *throughput* digunakan iperf3 dengan protokol TCP, sedangkan pada pengukuran *jitter* dan packetloss digunakan protokol UDP. Pengujian dilakukan sebanyak 3 kali pada masing-masing kondisi untuk

memastikan hasil yang diperoleh konsisten dan tidak dipengaruhi oleh kondisi sesaat pada jaringan maupun sistem.

4.4.3 Data Hasil Pengujian

Data hasil pengujian menunjukkan data-data hasil dari setiap pengujian yang telah dilakukan.

4.4.3.1 Hasil Pengujian Fungsionalitas

a. PF-01: Koneksi VPN

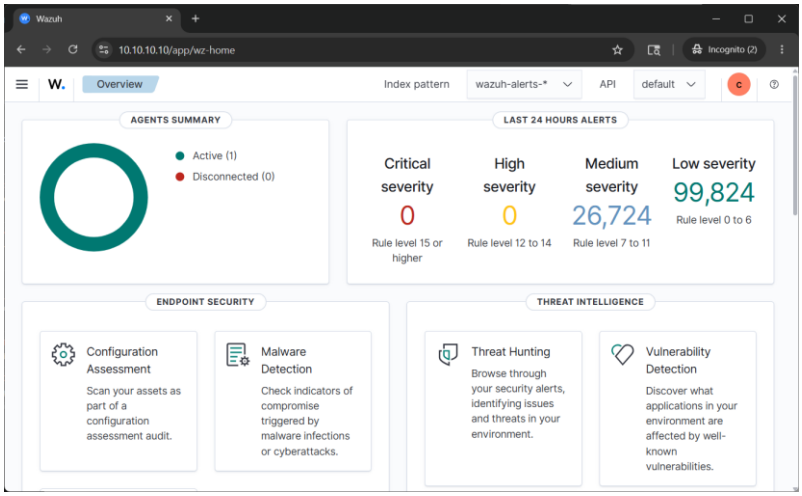
Pengujian dilakukan dengan melakukan koneksi ke layanan OpenVPN yang telah dikonfigurasi pada pfSense. Koneksi VPN diperlukan oleh administrator untuk mengakses service internal. Berdasarkan hasil pengujian, klient berhasil terhubung ke jaringan VPN dan memperoleh alamat IP virtual dari server OpenVPN.

Gambar 4. 42 Klien Terhubung ke OpenVPN

b. PF-02: Akses Dashboard Wazuh

Pengujian dilakukan dengan mengakses dashboard Wazuh melalui koneksi VPN yang telah aktif. Percobaan dilakukan untuk memverifikasi sistem

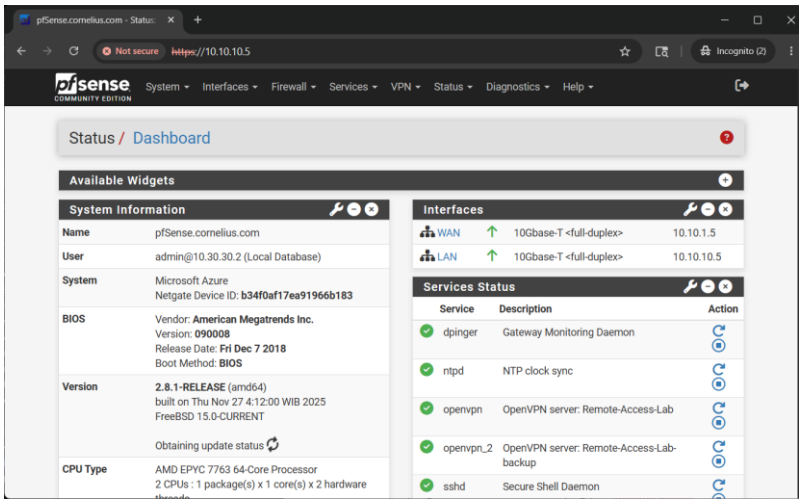
SIEM bekerja dan administrator dapat mengakses dashboard tersebut melalui koneksi VPN. Hasil pengujian menunjukkan bahwa dashboard Wazuh dapat diakses melalui jaringan VPN.



Gambar 4. 43 Dashboard Wazuh Berhasil Diakses Melalui VPN

c. PF-03: Akses Dashboard pfSense

Pada pengujian ini dilakukan akses ke dashboard administrasi pfSense melalui koneksi VPN. Berdasarkan hasil pengujian, dashboard pfSense dapat diakses melalui alamat IP internal firewall (10.10.10.5) ketika klien terhubung ke VPN.

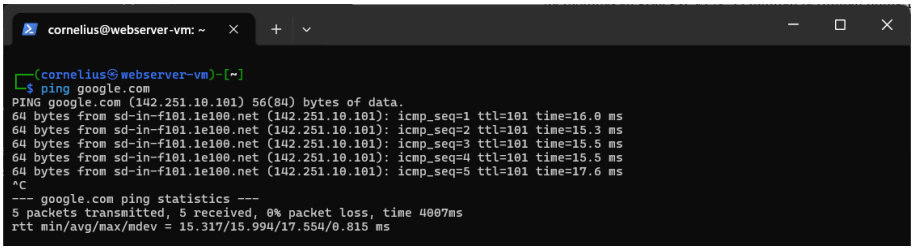


Gambar 4. 44 Dashboard pfSense Berhasil Diakses Melalui VPN

d. PF-04: Konektivitas Internet Server Internal

Pengujian dilakukan dengan mengirimkan paket ICMP dari Web Server menuju domain eksternal untuk memverifikasi konfigurasi *routing* dan

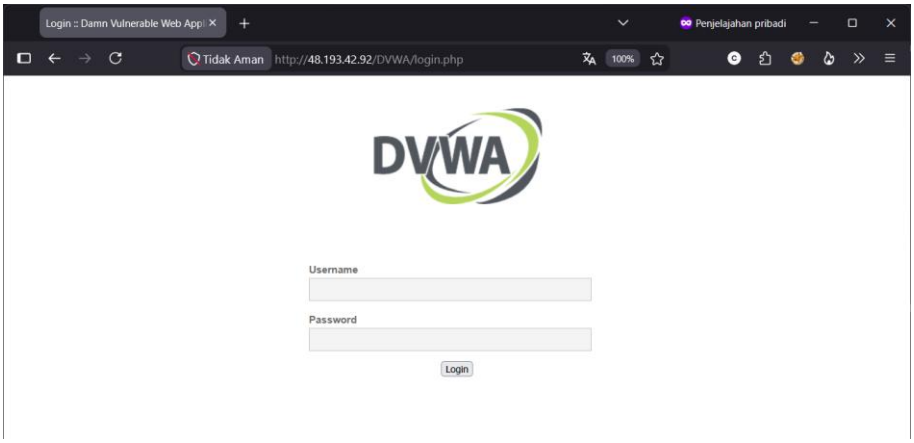
gateway pada pfSense. Hasil pengujian menunjukkan bahwa server internal berhasil terhubung ke internet dan mampu melakukan komunikasi jaringan internal dengan normal.



Gambar 4. 45 Web Server Terhubung dengan Internet

e. PF-05: Akses Web Service

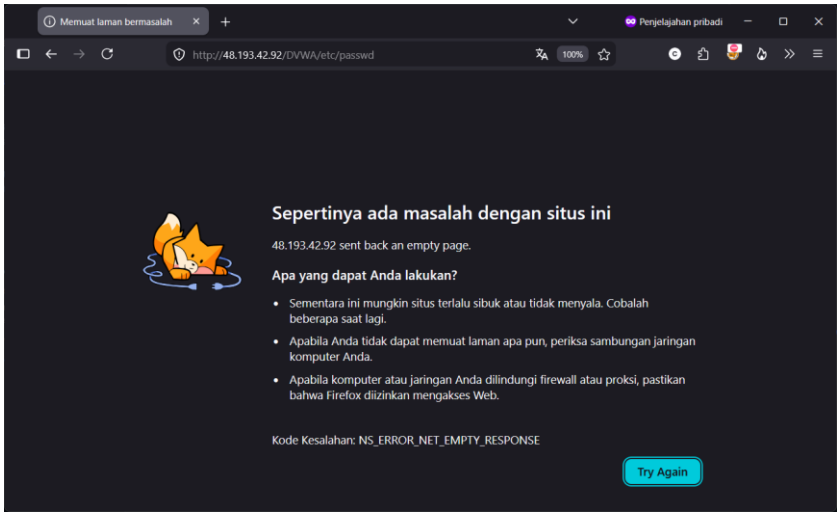
Pengujian dilakukan dengan mengakses aplikasi DVWA melalui alamat IP publik yang terhubung ke pfSense. Hasil pengujian menunjukkan bahwa request dari internet dapat diteruskan menuju Web Server sehingga layanan web dapat diakses dengan normal.



Gambar 4. 46 DVWA dapat Diakses melalui Internet

f. PF-06: Pemblokiran Serangan oleh Suricata IPS

Pengujian dilakukan dengan mengirimkan request yang mengandung *payload malicious* menuju ke Web Server. Pengujian bertujuan untuk memastikan fitur *Intrusion prevention system* (IPS) pada Suricata dapat memblokir ancaman secara otomatis. Hasil pengujian menunjukkan bahwa Suricata mendeteksi aktivitas tersebut dan melakukan pemblokiran sesuai dengan *rule* yang aktif.



Gambar 4. 47 Request Serangan Diblokir oleh Suricata

g. PF-07: Forwarding Log Suricata

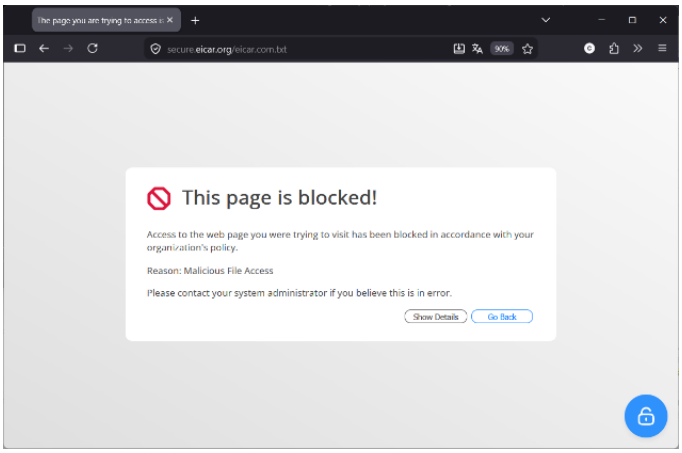
Pengujian dilakukan dengan menghasilkan *alert* yang terdeteksi oleh Suricata, kemudian memverifikasi apakah log berhasil dikirim dan ditampilkan pada dashboard Wazuh. Pengujian bertujuan untuk memastikan integrasi antara sistem Suricata dengan SIEM berjalan dengan baik. Hasil pengujian menunjukkan bahwa *alert* yang dihasilkan Suricata berhasil ditampilkan pada dashboard Wazuh.

Gambar 4. 48 *Alert* Suricata Berhasil Diterima oleh Wazuh

h. PF-08: Pemblokiran Akses ke Domain Malicious

Pada pengujian ini dilakukan percobaan akses menuju domain berbahaya atau berkaitan dengan *malware*. Pengujian bertujuan untuk memastikan fitur Zenarmor dapat mengenali akses menuju domain *malicious* dan melakukan pemblokiran. Hasil pengujian menunjukkan bahwa akses ke domain tersebut

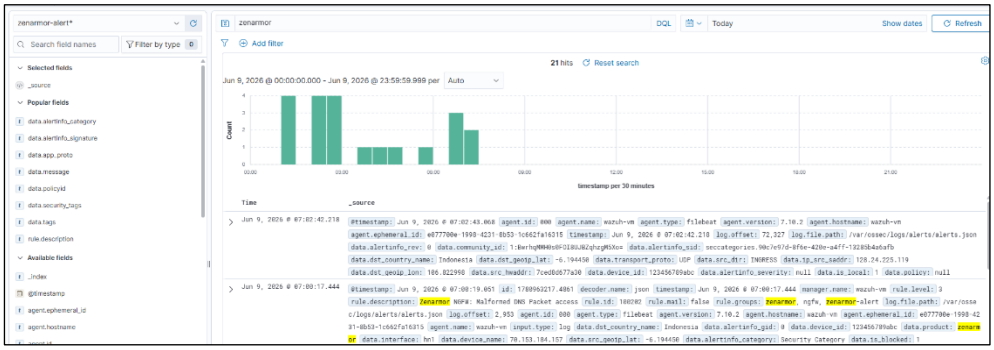
diblokir dan pengguna diarahkan ke halaman blokir yang disediakan oleh Zenarmor.



Gambar 4. 49 Akses ke Domain Malicious Diblokir oleh Zenarmor

i. PF-09: forwarding log Zenarmor

Pengujian dilakukan dengan menghasilkan *alert* yang terdeteksi oleh Zenarmor, kemudian memverifikasi apakah log berhasil dikirim dan ditampilkan pada dashboard Wazuh. Pengujian bertujuan untuk memastikan integrasi antara Zenarmor dengan SIEM berjalan dengan baik. Berdasarkan hasil pengujian, *alert* yang dihasilkan Zenarmor berhasil diterima dan ditampilkan pada dashboard Wazuh.



Gambar 4. 50 *Alert* Zenarmor Berhasil Diterima oleh Wazuh

Hasil keseluruhan pengujian fungsionalitas dirangkum pada Tabel 4.12.

Tabel 4. 12 Hasil Pengujian Fungsionalitas

ID	Skenario Pengujian	Hasil	Status
PF-01	Pengujian koneksi VPN	Client berhasil terhubung VPN	Berhasil

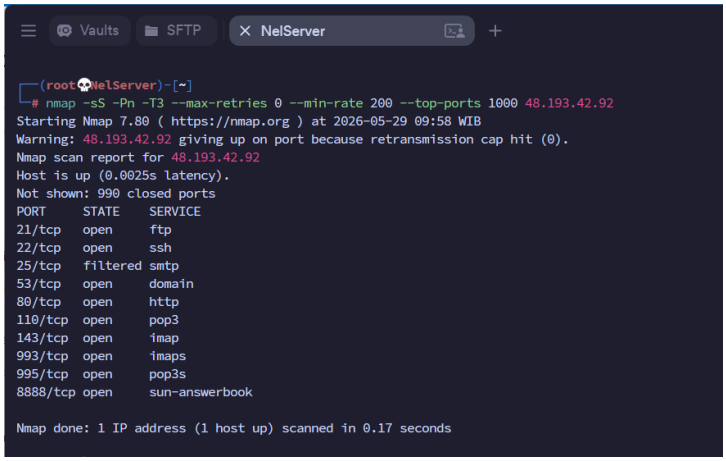
PF-02	Pengujian akses dashboard Wazuh dengan VPN aktif	Wazuh dashboard (10.10.10.10) berhasil diakses melalui VPN	Berhasil
PF-03	Pengujian akses dashboard pfSense dengan VPN aktif	pFsense dashboard (10.10.10.5) berhasil diakses melalui VPN	Berhasil
PF-04	Pengujian konektivitas internet server internal	Web server berhasil melakukan ping menuju google	Berhasil
PF-05	Pengujian akses Web service melalui internet	Web service DVWA dapat diakses melalui internet	Berhasil
PF-06	Pengujian pemblokiran serangan oleh Suricata	Suricata berhasil memblokir request malicious menuju server	Berhasil
PF-07	Pengujian forwarding log Suricata	Log Suricata berhasil dikirim menuju SIEM Wazuh	Berhasil
PF-08	Pengujian melakukan request menuju domain *malicious*	Website tidak dapat diakses oleh client dan sistem menampilkan halaman blokir dari Zenarmor.	Berhasil
PF-09	Pengujian forwarding log Zenarmor	*Alert* dari Zenarmor berhasil diterima pada dashboard Wazuh.	Berhasil

4.4.3.2 Hasil Pengujian Ancaman Eksternal

Berikut adalah data yang didapat dari hasil ancaman eksternal

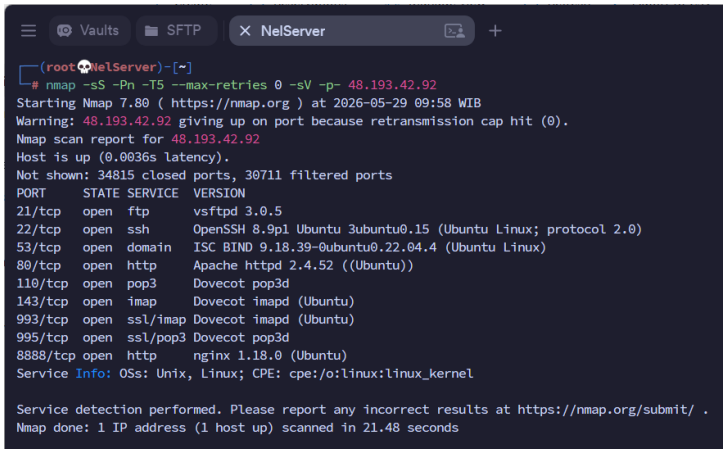
a. Pengujian Port Scanning

Pengujian port scanning dilakukan menggunakan Nmap pada skenario EX-01 sampai EX-04. Pengujian dilakukan pada dua kondisi, yaitu ketika fitur keamanan firewall dinonaktifkan dan ketika Suricata IPS diaktifkan. Tujuan pengujian ini adalah untuk mengetahui pengaruh implementasi NGFW terhadap kemampuan penyerang dalam mengidentifikasi layanan yang berjalan pada server target. Pada pengujian EX-01 dilakukan TCP SYN Scan terhadap 1.000 port yang umum digunakan oleh layanan jaringan. Hasil pengujian ditunjukkan pada Gambar 4.51.



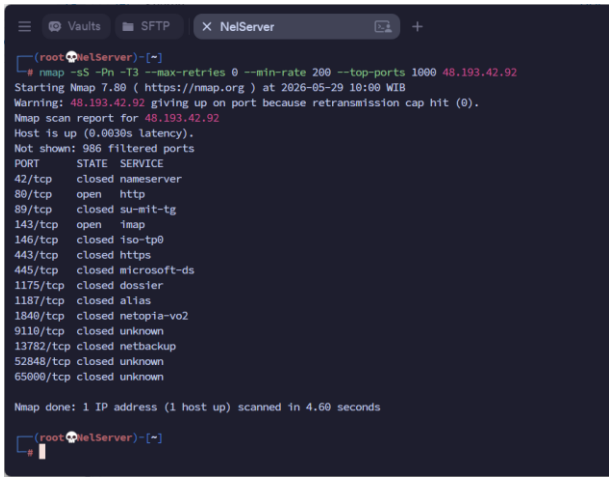
Gambar 4. 51 Hasil Pengujian EX-01 TCP SYN Scan

Selanjutnya pada pengujian EX-02 dilakukan TCP SYN Scan yang disertai enumerasi layanan untuk memperoleh informasi tambahan mengenai service yang berjalan pada server target. Hasil pengujian ditunjukkan pada Gambar 4.52.



Gambar 4. 52 Hasil Pengujian EX-02 TCP SYN Scan dan Service Scan

Ketika fitur keamanan Suricata IPS diaktifkan, hasil yang berbeda diperoleh pada pengujian EX-03. Berdasarkan hasil scanning yang ditunjukkan pada Gambar 4.53, jumlah port yang berhasil diidentifikasi mengalami penurunan dibandingkan kondisi sebelumnya.



Gambar 4. 53 Hasil Pengujian EX-03 TCP SYN Scan

Hasil serupa diperoleh pada pengujian EX-04. Meskipun Nmap masih dapat menemukan beberapa port yang merespons, jumlah layanan yang berhasil diidentifikasi lebih sedikit dibandingkan kondisi ketika fitur keamanan dinonaktifkan.

Gambar 4. 54 Hasil Pengujian EX-04 TCP SYN Scan dan Service Scan

Berdasarkan hasil pengujian, pada kondisi firewall nonaktif yaitu EX-01 dan EX-02, nmap dapat melihat port yang terbuka serta versi layanan yang berjalan pada server. Ketika keamanan firewall aktif berupa Suricata, pada pengujian EX-04 dan EX-03 jumlah port yang berhasil diidentifikasi mengalami penurunan menjadi dua hingga tiga port. Ringkasan hasil pengujian ditunjukkan pada Tabel 4.13.

Tabel 4. 13 Hasil Pengujian NMAP

Skenario	Port Terbuka	Detail Port
TCP SYN Scan (Keamanan Firewall Nonaktif)	9	FTP (21) SSH (22) DNS (53)

		HTTP (80) POP3 (110) IMAP (143) IMAPS (993) POP3S (995) HTTP Alternate (8888)
TCP SYN Scan dan Enumerasi Layanan (Keamanan Firewall Nonaktif)	9	FTP (21) SSH (22) DNS (53) HTTP (80) POP3 (110) IMAP (143) IMAPS (993) POP3S (995) HTTP Alternate (8888)
TCP SYN Scan (Keamanan Firewall aktif)	2	HTTP (80) IMAP (143)
TCP SYN Scan dan Enumerasi Layanan (Keamanan Firewall aktif)	3	DNS (53) POP3 (995) HTTP (8080)

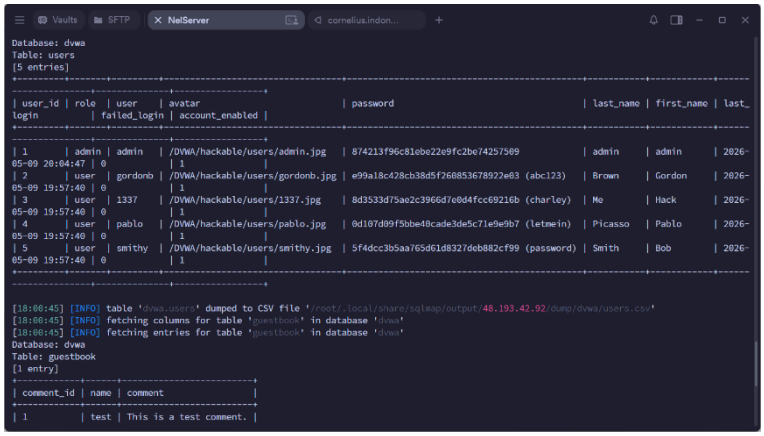
b. SQL Injection

Pengujian SQL *Injection* dilakukan menggunakan SQLmap dengan target aplikasi DVWA yang berjalan pada Web Server. Pengujian dilakukan untuk mengetahui kemampuan sistem dalam mendeteksi *payload* SQL *Injection* yang dikirim menuju aplikasi web.

Pada kondisi ketika keamanan firewall dinonaktifkan, *payload* SQL *Injection* yang dikirim menggunakan SQLmap berhasil mencapai Web Server. Aktivitas tersebut tercatat oleh Wazuh Agent HIDS yang terpasang pada server dan menghasilkan yang berkaitan dengan adanya eksploitasi.

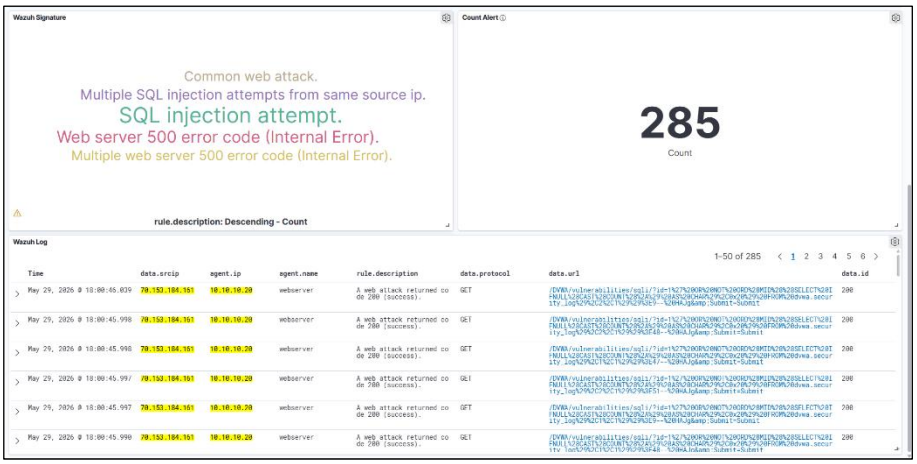
Gambar 4. 55 Pengujian EX-05

Pada pengujian EX-05, SQLmap berhasil memperoleh informasi mengenai struktur database yang digunakan oleh aplikasi. Hal ini menunjukkan bahwa *payload* yang dikirim berhasil diproses oleh Web Server.



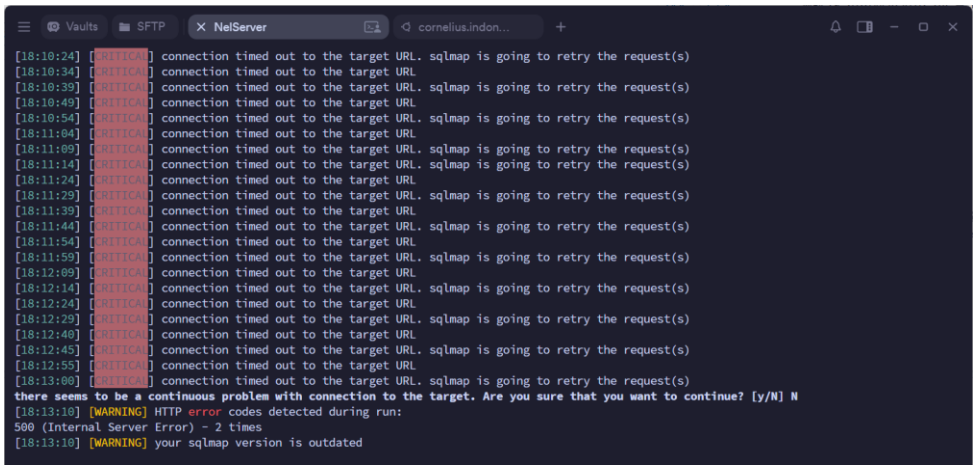
Gambar 4. 56 Pengujian EX-06

Pada pengujian EX-06, SQLmap berhasil mendapatkan informasi *user* yang tersimpan pada *database*. Gambar 4.57 menunjukkan, aktivitas SQL *Injection* yang berhasil mencapai Web Server menghasilkan *alert* pada Wazuh Agent HIDS. *Alert* tersebut menunjukkan bahwa *request* telah diproses oleh aplikasi dan tercatat pada sisi host.



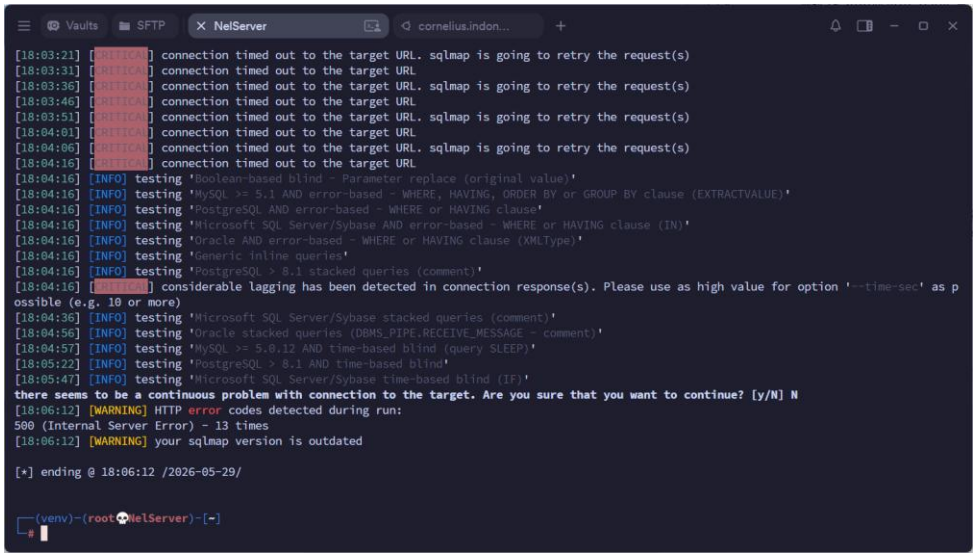
Gambar 4. 57 *Alert* Wazuh HIDS Pada Pengujian EX-06

Pengujian selanjutnya dilakukan dengan mengaktifkan Suricata IPS pada firewall untuk mengetahui kemampuan sistem dalam mencegah serangan SQL *Injection*.



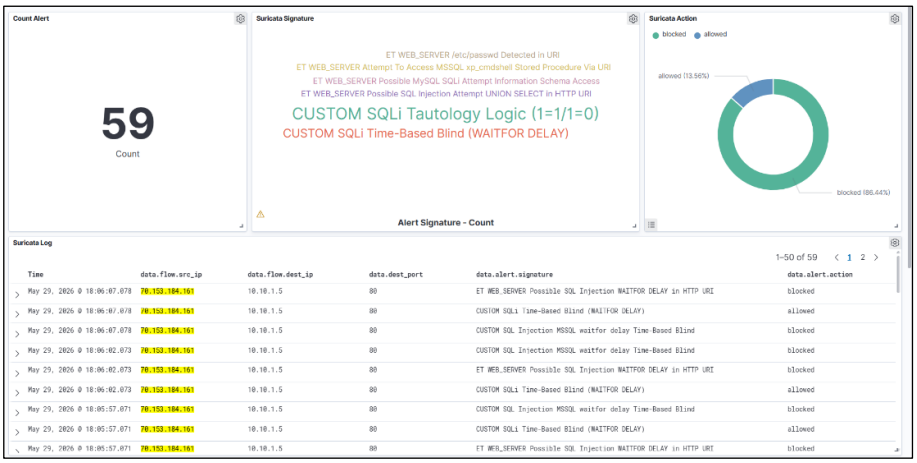
Gambar 4. 58 Pengujian EX-07

Pada pengujian EX-07 dan EX-08, sqlmap tidak berhasil menampilkan informasi struktur database maupun data pengguna dari Web Server. Selain itu, tidak ditemukan request SQL Injection yang tercatat pada log Wazuh Agent di Web Server sehingga dapat dikatakan tidak terdapat request SQL Injection yang lolos dan terdeteksi oleh HIDS.



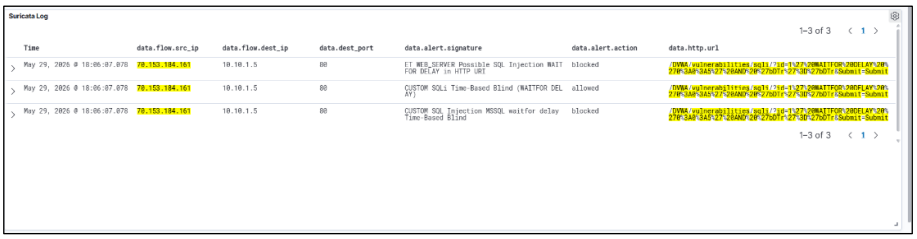
Gambar 4. 59 Pengujian EX-08

Pada pengujian EX-07, terdapat *alert* Suricata yang tidak mendapatkan *action* "*Blocked*" oleh Suricata. Namun Wazuh Agent HIDS mencatat ada 0 request *malicious* yang terdeteksi pada Web Server.



Gambar 4. 60 *Alert* Suricata pada Wazuh SIEM

Berdasarkan Gambar 4.61, request yang sama dapat memicu lebih dari satu *rule* Suricata dengan aksi yang berbeda. Namun demikian, hasil pengujian menunjukkan bahwa tidak terdapat request SQL Injection yang tercatat pada Web Server ketika Suricata IPS diaktifkan. Hal ini mengindikasikan bahwa mekanisme pencegahan yang diterapkan oleh Suricata berhasil menghentikan eksploitasi sebelum request mencapai aplikasi target.



Gambar 4. 61 *Alert* Suricata dari Request yang Sama

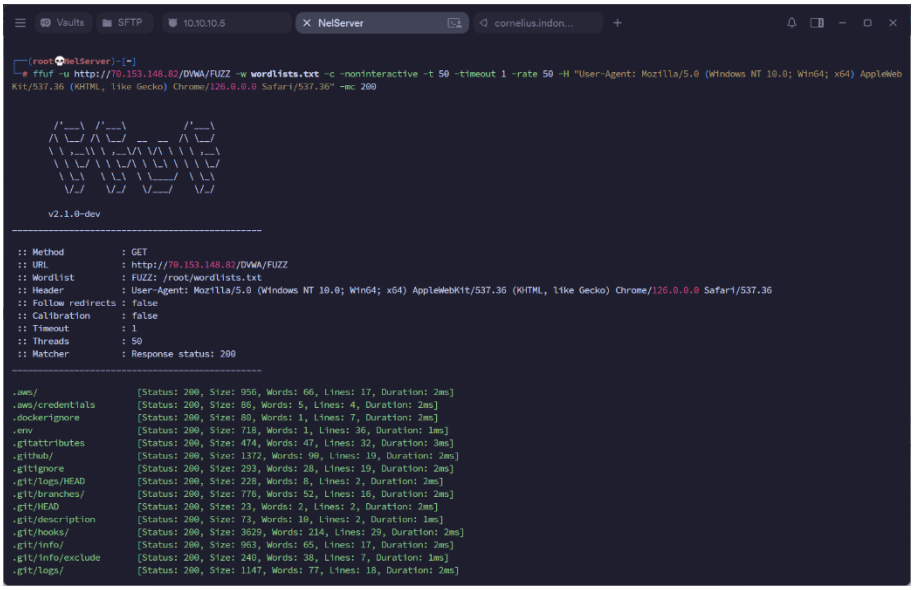
Ringkasan hasil pengujian SQL Injection ditunjukkan pada Tabel 4.14.

Tabel 4. 14 Hasil Pengujian SQL Injection

ID	*Alert* Suricata	*Alert* Wazuh	Hasil
EX-05	0	79	Terdapat 79 request yang diterima web server dan tercatat HIDS wazuh agent
EX-06	0	285	Terdapat 285 request yang diterima Web Server dan tercatat oleh Wazuh Agent.
EX-07	59	0	Tidak terdapat request yang diterima Web Server. Suricata menghasilkan 59 *alert*, terdiri dari 8 *alert* dengan *action alert* dan 51 *alert* dengan *action* drop.
EX-08	43	0	Tidak terdapat request yang diterima Web Server. Suricata menghasilkan 43 *alert* selama pengujian berlangsung.

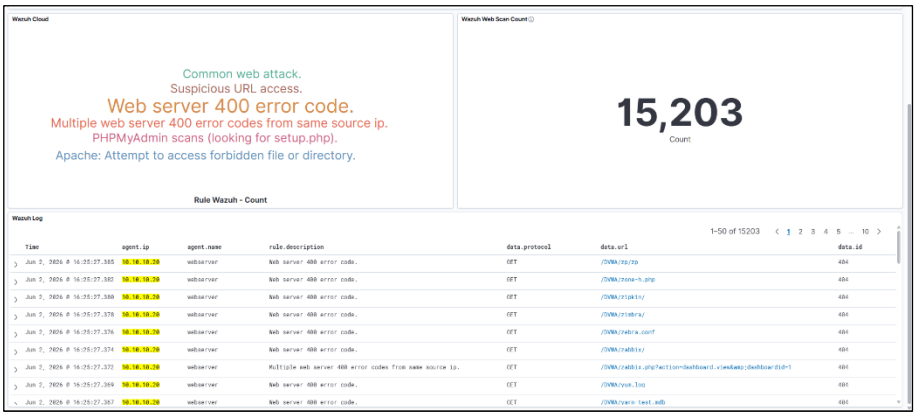
c. Pengujian Direktori Scanning

Pengujian directory scanning dilakukan menggunakan tools FFUF dengan target web service DVWA yang berjalan pada Web Server. Pada kondisi firewall nonaktif, proses directory scanning berhasil dijalankan menggunakan wordlist yang telah disiapkan. Seluruh request yang dikirim oleh FFUF diteruskan menuju Web Server dan diproses oleh web service. Hasil pengujian menunjukkan bahwa FFUF berhasil menemukan direktori dan file yang dapat diakses oleh publik melalui web server. Selain itu, aktivitas request yang dihasilkan selama proses scanning juga tercatat pada Wazuh Agent yang terpasang pada Web Server.



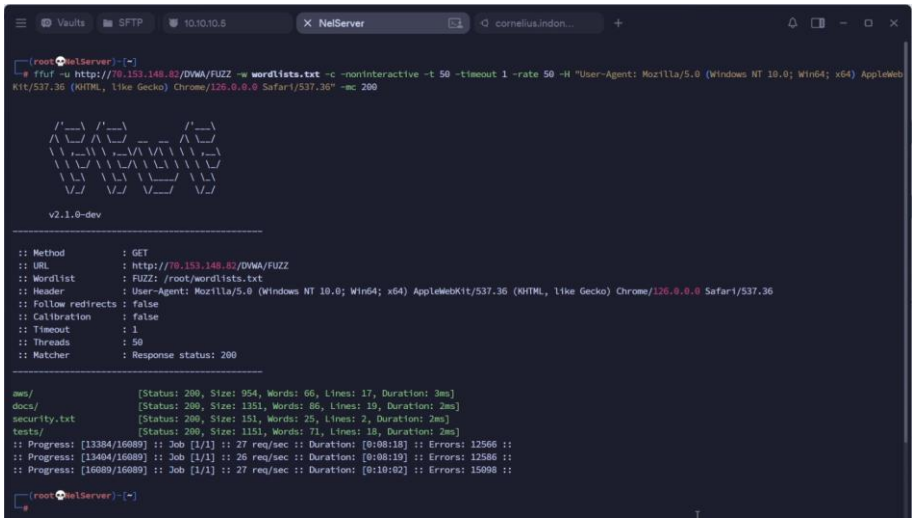
Gambar 4. 62 Pengujian EX-09

Aktivitas *request* direktori scanning tersebut diteruskan menuju web server dan aktivitasnya tercatat oleh Wazuh HIDS



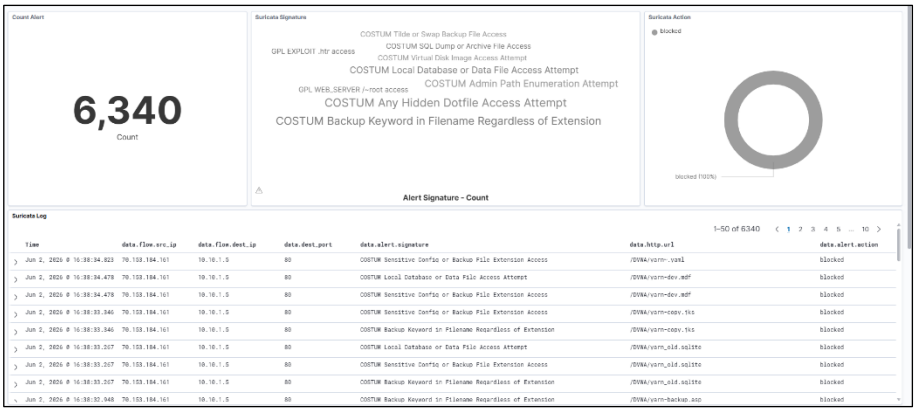
Gambar 4. 63 *Alert* Wazuh HIDS Pada Pengujian EX-09

Pengujian berikutnya dilakukan dengan mengaktifkan Suricata IPS pada firewall. Pada pengujian EX-11, FFUF masih dapat memperoleh beberapa hasil *scanning* berupa direktori dan file yang dapat diakses oleh aplikasi web.



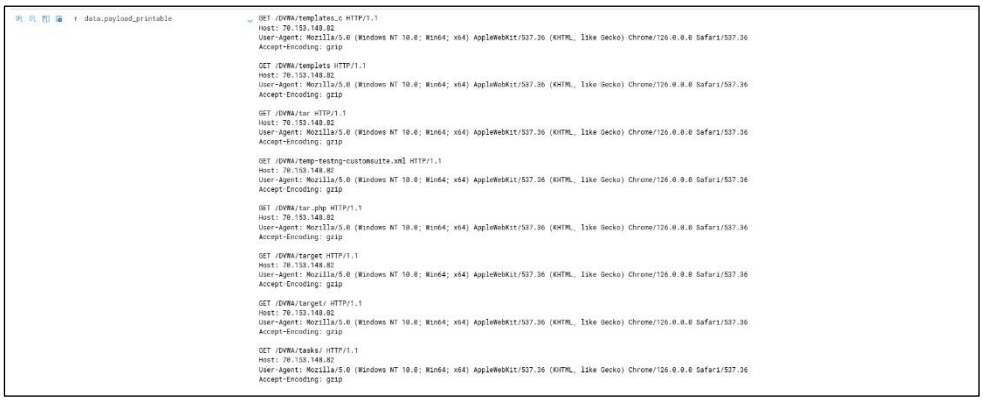
Gambar 4. 64 Pengujian EX-11

Berdasarkan hasil pengujian, ditemukan empat direktori atau file yang berhasil ditemukan selama proses *scanning* berlangsung. Selain itu, Suricata menghasilkan sejumlah *alert* keamanan yang berkaitan dengan aktivitas *scanning* aplikasi web.



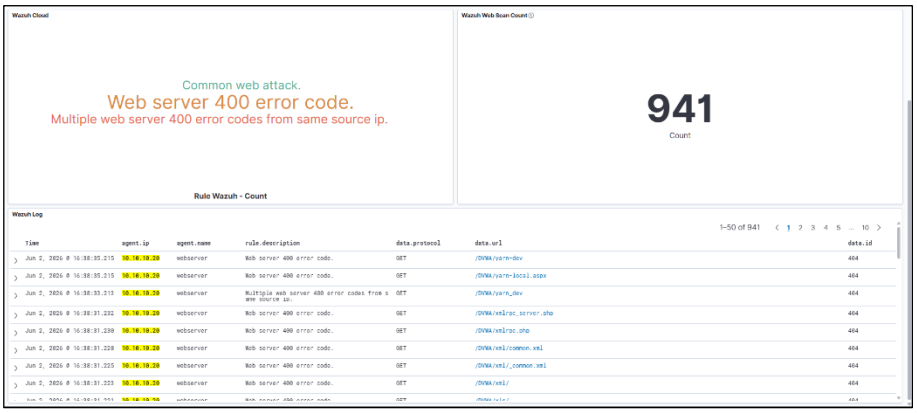
Gambar 4. 65 *Alert* Suricata pada Pengujian EX-11

Perlu diketahui bahwa meskipun Suricata berhasil mendeteksi dan memblokir seluruh request, jumlah *alert* yang dibuat Suricata tidak akan sama dengan jumlah request yang dilakukan. Hal ini karena *rule* Suricata memiliki mekanisme threshold untuk mengelompokan request serupa kedalam 1 *alert*.



Gambar 4. 66 Trafik dalam 1 *Alert* Suricata

Wazuh Agent HIDS yang terpasang pada Web Server juga menemukan adanya aktivitas mencurigakan. Dari pengujian EX-11 terdapat 941 *alert* yang dibuat Wazuh HIDS untuk request yang masih lolos dari Suricata dan dianggap mencurigakan.



Gambar 4. 67 *Alert* Wazuh HIDS pada Pengujian EX-11

Tabel 4.15 menunjukkan hasil dari pengujian directory *scanning* yang dilakukan.

Tabel 4. 15 Hasil Pengujian Directory *Scanning*

ID	*Alert* Suricata	*Alert* Wazuh	Hasil
EX-09	0	15,203	Ditemukan 50 direktori atau file dari hasil *scanning* menggunakan FFUF. Wazuh menghasilkan 15.203 *alert* dan Suricata tidak menghasilkan *alert*.
EX-10	0	15,203	Ditemukan 50 direktori atau file dari hasil *scanning* menggunakan FFUF. Wazuh

			menghasilkan 15.203 *alert* dan Suricata tidak menghasilkan *alert*.
EX-11	6,340	941	Ditemukan 4 direktori atau file dari hasil *scanning* menggunakan FFUF. Suricata menghasilkan 6.340 *alert* dan Wazuh menghasilkan 941 *alert*.
EX-12	531	130	Tidak ditemukan direktori maupun file dari hasil *scanning* menggunakan FFUF. Suricata menghasilkan 531 *alert* dan Wazuh menghasilkan 130 *alert*.

d. Denial of Service (DoS)

Pengujian Denial of Service (DoS) dilakukan menggunakan metode SYN Flood untuk memperoleh data penggunaan sumber daya Web Server selama menerima serangan. Selain itu, pengujian ini juga dilakukan untuk memperoleh data perbandingan penggunaan CPU dan RAM pada kondisi firewall nonaktif dan aktif.

Pada tahap pertama, pengujian dilakukan dengan menonaktifkan sistem keamanan firewall. Sebelum serangan dijalankan, dilakukan pengambilan data penggunaan CPU dan RAM pada kondisi normal sebagai data pembanding. Selanjutnya dilakukan serangan SYN Flood dan penggunaan sumber daya server dicatat selama proses pengujian berlangsung. Data hasil pengukuran ditunjukkan pada Tabel 4.16.

Tabel 4. 16 Pengukuran Resource Web Server pada EX-13

Time	Keadaan Normal			SYN Flood		
	CPU%	RAM	RAM%	CPU%	RAM	RAM%
1	3,9%	643	33,8%	5,8%	744	39,1%
2	4,7%	643	33,8%	6,7%	799	42,0%
3	4,8%	643	33,8%	3,0%	803	42,2%
4	3,8%	643	33,8%	5,8%	845	44,5%
5	5,7%	643	33,8%	4,9%	859	45,2%
6	3,8%	643	33,8%	5,8%	931	49,0%
7	4,7%	644	33,9%	5,8%	968	50,9%
8	3,8%	644	33,9%	3,9%	989	52,0%
9	4,7%	644	33,9%	5,8%	1068	56,2%
10	4,8%	644	33,9%	4,9%	1108	58,3%

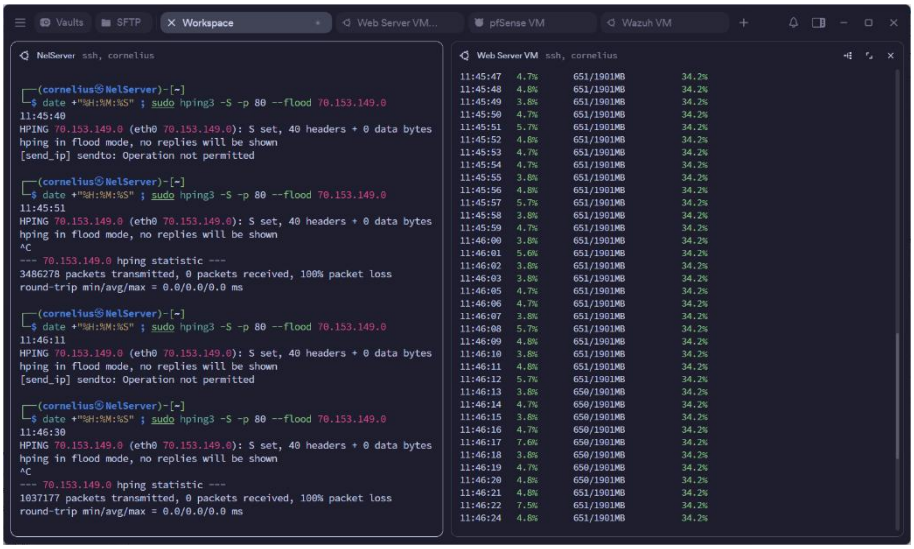
11	4,7%	644	33,9%	4,9%	1117	58,8%
12	4,8%	644	33,9%	3,9%	1131	59,5%
13	3,8%	644	33,9%	4,8%	1149	60,4%
14	4,7%	644	33,9%	4,7%	1173	61,7%
15	4,8%	643	33,8%	3,8%	1210	63,7%
16	4,7%	644	33,9%	7,5%	1270	66,8%
17	3,8%	643	33,8%	3,9%	1325	69,7%
18	4,7%	643	33,8%	4,8%	1357	71,4%
19	4,8%	643	33,8%	6,7%	1251	65,8%
20	4,8%	643	33,8%	4,8%	1258	66,2%
21	4,7%	643	33,8%	3,9%	1255	66,0%
22	3,8%	643	33,8%	4,8%	1255	66,0%
23	4,8%	643	33,8%	5,7%	1274	67,0%
24	5,6%	643	33,8%	4,8%	1255	66,0%
25	3,8%	643	33,8%	2,9%	1255	66,0%
26	4,7%	644	33,9%	5,8%	1306	68,7%
27	3,8%	644	33,9%	4,8%	1326	69,8%
28	3,8%	644	33,9%	3,9%	1302	68,5%
29	5,7%	644	33,9%	4,8%	1320	69,4%
30	4,7%	644	33,9%	4,8%	1317	69,3%
31	4,8%	644	33,9%	4,8%	1312	69,0%
32	3,8%	644	33,9%	4,8%	1336	70,3%
33	4,7%	644	33,9%	3,8%	1259	66,2%
34	5,7%	644	33,9%	4,8%	1254	66,0%
35	3,8%	644	33,9%	4,8%	1254	66,0%
36	4,7%	644	33,9%	4,8%	1260	66,3%
37	9,4%	644	33,9%	3,8%	1266	66,6%
38	3,8%	644	33,9%	5,7%	1293	68,0%
39	3,8%	644	33,9%	4,8%	1300	68,4%
40	4,7%	644	33,9%	4,8%	1290	67,9%
41	3,8%	644	33,9%	4,8%	1319	69,4%
42	4,8%	644	33,9%	4,8%	1347	70,9%
43	5,7%	644	33,9%	4,8%	1321	69,5%
44	3,8%	644	33,9%	4,7%	1331	70,0%
45	3,8%	644	33,9%	6,8%	1385	72,9%
46	3,8%	644	33,9%	3,8%	1275	67,1%
47	5,7%	644	33,9%	6,5%	1313	69,1%
48	3,8%	643	33,8%	2,9%	1277	67,2%
49	4,7%	643	33,8%	5,7%	1278	67,2%
50	4,7%	643	33,8%	4,9%	1266	66,6%
51	3,8%	643	33,8%	4,8%	1344	70,7%
52	4,7%	643	33,8%	4,8%	1216	64,0%
53	3,8%	643	33,8%	3,9%	1255	66,0%
54	3,8%	643	33,8%	5,7%	1258	66,2%
55	4,7%	643	33,8%	3,8%	1255	66,0%
56	4,7%	642	33,8%	4,8%	1279	67,3%
57	3,8%	642	33,8%	4,7%	1295	68,1%
58	4,7%	642	33,8%	5,7%	1195	62,9%
59	5,9%	644	33,9%	4,8%	1135	59,7%
60	4,8%	644	33,9%	4,8%	1092	57,4%

Berdasarkan data pengujian yang diperoleh, penggunaan CPU dan RAM Web Server mengalami perubahan selama serangan SYN Flood berlangsung. Untuk mempermudah pengamatan, nilai minimum, maksimum, dan rata-rata dari hasil pengukuran dirangkum pada Tabel 4.17.

Tabel 4. 17 Rangkuman Hasil Pengukuran Pengujian EX-13

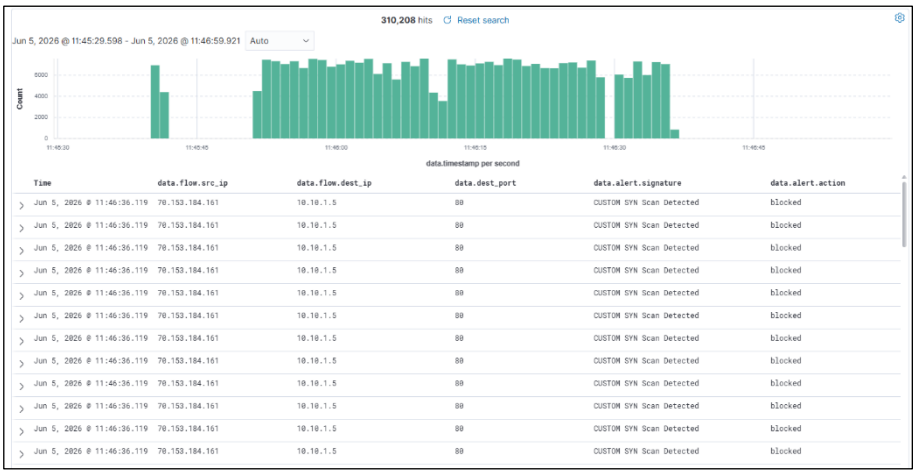
	Keadaan Normal			SYN Flood		
	CPU%	**RAM**	**RAM%**	**CPU%**	**RAM**	**RAM%**
MIN	3,8	642	33,8%	2,9%	744	39,1%
MAX	9,4	644	33,9%	7,5%	1385	72,9%
AVG	4,57	643,50	33,86%	4,88%	1208,00	63,55%

Pengujian berikutnya dilakukan dengan mengaktifkan Suricata IPS pada firewall. Pengambilan data dilakukan menggunakan metode yang sama, yaitu dengan membandingkan kondisi normal dan kondisi saat menerima serangan SYN Flood.



Gambar 4. 68 Pengujian EX-14

Dari pengujian yang dilakukan, pada SIEM Wazuh terlihat bahwa sistem berhasil mendeteksi adanya DoS SYN Scan dan melakukan block pada trafik tersebut.



Gambar 4. 69 *Alert* Suricata pada Wazuh pada Pengujian EX-14

Keseluruhan data hasil pengukuran ditunjukkan pada Tabel 4.18.

Tabel 4. 18 Pengukuran Resource Web Server pada EX-14

ini	Keadaan Normal			SYN Flood		
	CPU%	RAM	RAM%	CPU%	RAM	RAM%
1	4,7%	660	34,7%	4,8%	660	34,2%
2	3,8%	660	34,7%	4,7%	660	34,2%
3	4,8%	660	34,7%	3,8%	660	34,2%
4	4,7%	660	34,7%	5,7%	660	34,2%
5	4,8%	660	34,7%	4,7%	660	34,2%
6	3,8%	660	34,7%	3,8%	660	34,2%
7	4,7%	660	34,7%	4,8%	660	34,2%
8	5,7%	660	34,7%	4,7%	660	34,2%
9	4,8%	660	34,7%	4,8%	660	34,2%
10	4,7%	660	34,7%	3,8%	660	34,2%
11	4,7%	660	34,7%	4,7%	660	34,2%
12	3,8%	660	34,7%	5,7%	660	34,2%
13	4,8%	660	34,7%	4,8%	660	34,2%
14	5,7%	660	34,7%	4,7%	660	34,2%
15	3,8%	653	34,4%	4,7%	653	34,2%
16	4,7%	653	34,4%	3,8%	653	34,2%
17	3,8%	653	34,4%	4,8%	653	34,2%
18	5,6%	653	34,4%	5,7%	653	34,2%
19	3,8%	653	34,4%	3,8%	653	34,2%
20	3,8%	650	34,2%	4,7%	650	34,2%
21	4,7%	650	34,2%	3,8%	650	34,2%
22	4,7%	651	34,2%	5,6%	651	34,2%
23	3,8%	651	34,2%	3,8%	651	34,2%
24	5,7%	651	34,2%	3,8%	651	34,2%
25	4,8%	651	34,2%	4,7%	651	34,2%
26	3,8%	651	34,2%	4,7%	651	34,2%
27	4,8%	651	34,2%	3,8%	651	34,2%
28	5,7%	651	34,2%	5,7%	651	34,2%
29	3,8%	651	34,2%	4,8%	651	34,2%

30	4,7%	651	34,2%	3,8%	651	34,2%
31	3,8%	651	34,2%	4,8%	651	34,2%
32	4,7%	651	34,2%	5,7%	651	34,2%
33	7,6%	651	34,2%	3,8%	651	34,2%
34	3,8%	651	34,2%	4,7%	651	34,2%
35	4,7%	651	34,2%	3,8%	651	34,2%
36	4,8%	651	34,2%	4,7%	651	34,2%
37	4,8%	651	34,2%	7,6%	651	34,2%
38	7,5%	651	34,2%	3,8%	651	34,2%
39	4,8%	651	34,2%	4,7%	651	34,2%
40	3,8%	651	34,2%	4,8%	651	34,2%
41	4,7%	651	34,2%	4,8%	651	34,2%
42	4,8%	650	34,2%	7,5%	650	34,2%
43	4,8%	650	34,2%	4,8%	650	34,2%
44	4,7%	650	34,2%	3,8%	650	34,2%
45	4,8%	650	34,2%	4,7%	650	34,2%
46	4,7%	650	34,2%	4,8%	650	34,2%
47	3,8%	651	34,2%	4,8%	651	34,2%
48	5,7%	651	34,2%	4,7%	651	34,2%
49	4,8%	651	34,2%	4,8%	651	34,2%
50	3,8%	651	34,2%	4,7%	651	34,2%
51	5,7%	651	34,2%	3,8%	651	34,2%
52	5,7%	651	34,2%	5,7%	651	34,3%
53	3,8%	651	34,2%	4,8%	651	34,3%
54	4,8%	651	34,2%	3,8%	651	34,3%
55	4,7%	651	34,2%	5,7%	651	34,2%
56	4,8%	651	34,2%	5,7%	651	34,2%
57	6,5%	651	34,2%	3,8%	651	34,7%
58	3,8%	651	34,2%	4,8%	651	34,7%
59	4,7%	651	34,2%	4,7%	651	34,7%
60	3,8%	651	34,2%	4,8%	651	34,7%

Selain dilakukan monitoring terhadap penggunaan sumber daya server, selama pengujian berlangsung juga dilakukan pengamatan terhadap *alert* keamanan yang dihasilkan oleh sistem. *Alert* yang dihasilkan Suricata diteruskan menuju Wazuh SIEM dan ditampilkan pada dashboard monitoring.

Tabel 4. 19 Rangkuman Rata-Rata Pengujian EX-14

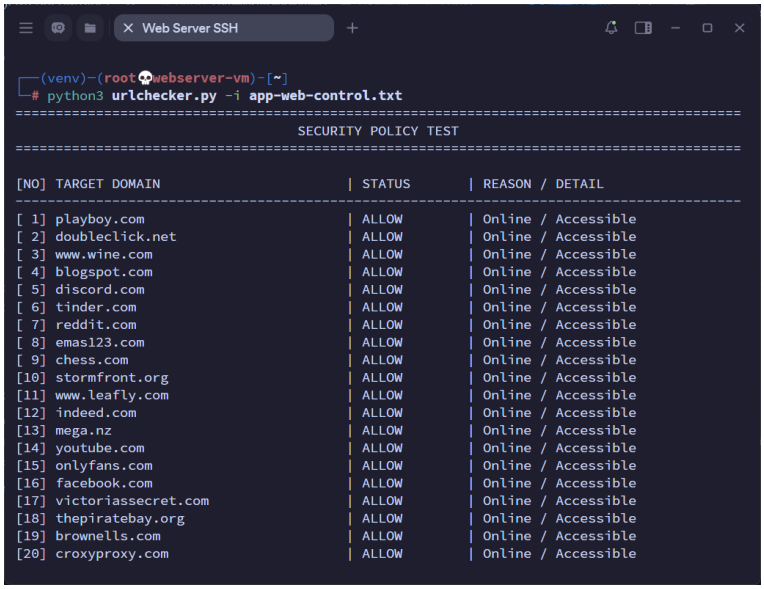
	Keadaan Normal			**SYN Flood**		
	CPU%	**RAM**	**RAM%**	**CPU%**	**RAM**	**RAM%**
MIN	3,8%	650	34,2%	3,8%	650	34,2%
MAX	7,6%	660	34,7%	7,6%	660	34,7%
AVG	4,7%	653,17	34,3%	4,72%	651,52	34,2%

4.4.3.3 Hasil Pengujian Ancaman Internal

Berikut adalah data yang didapat dari hasil ancaman internal

a. Pengujian IN-01

Pengujian IN-01 dilakukan dengan menjalankan script untuk mengakses 20 website ketika keamanan firewall nonaktif.



Gambar 4. 70 Hasil Pengujian IN-01

Hasil dari pengujian IN-01 akses ke website dirangkum pada Tabel 4.20.

Tabel 4. 20 Rangkuman Hasil Pengukuran Pengujian IN-01

No.	Website	Status
1	playboy.com	Dapat Diakses
2	doubleclick.net	Dapat Diakses
3	wine.com	Dapat Diakses
4	blogspot.com	Dapat Diakses
5	discord.com	Dapat Diakses
6	tinder.com	Dapat Diakses
7	reddit.com	Dapat Diakses
8	emas123.com	Dapat Diakses
9	steam.com	Dapat Diakses
10	stormfront.org	Dapat Diakses
11	leafly.com	Dapat Diakses

12	indeed.com	Dapat Diakses
13	mega.nz	Dapat Diakses
14	youtube.com	Dapat Diakses
15	onlyfans.com	Dapat Diakses
16	facebook.com	Dapat Diakses
17	victoriassecret.com	Dapat Diakses
18	thepiratebay.org	Dapat Diakses
19	brownells.com	Dapat Diakses
20	croxyproxy.com	Dapat Diakses

b. Pengujian IN-02

Pengujian IN-02 dilakukan dengan menjalankan script untuk mengakses 20 website ketika keamanan firewall aktif yaitu ketika Suricata dan Zenarmor diaktifkan.

Gambar 4. 71 Hasil Pengujian IN-02

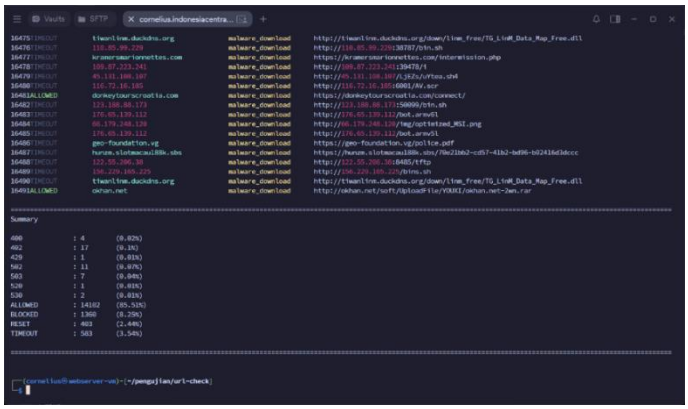
Seluruh situs yang dicoba diakses menghasilkan status Block/Reset, yaitu akses ke situs tersebut menghasilkan halaman block dari Zenarmor ataupun reset ketika Zenarmor memutus koneksi secara diam-diam. Hasil dari pengujian ini dirangkum pada Tabel 4.21.

Tabel 4. 21 Rangkuman Hasil Pengukuran Pengujian IN-02

No.	Website	Status
1	playboy.com	Blocked
2	doubleclick.net	Reset
3	wine.com	Blocked
4	blogspot.com	Blocked
5	discord.com	Blocked
6	tinder.com	Blocked
7	reddit.com	Blocked
8	emas123.com	Blocked
9	steam.com	Blocked
10	stormfront.org	Blocked
11	leafly.com	Blocked
12	indeed.com	Blocked
13	mega.nz	Blocked
14	youtube.com	Blocked
15	onlyfans.com	Blocked
16	facebook.com	Blocked
17	victoriassecret.com	Blocked
18	thepiratebay.org	Blocked
19	brownells.com	Blocked
20	croxyproxy.com	Blocked

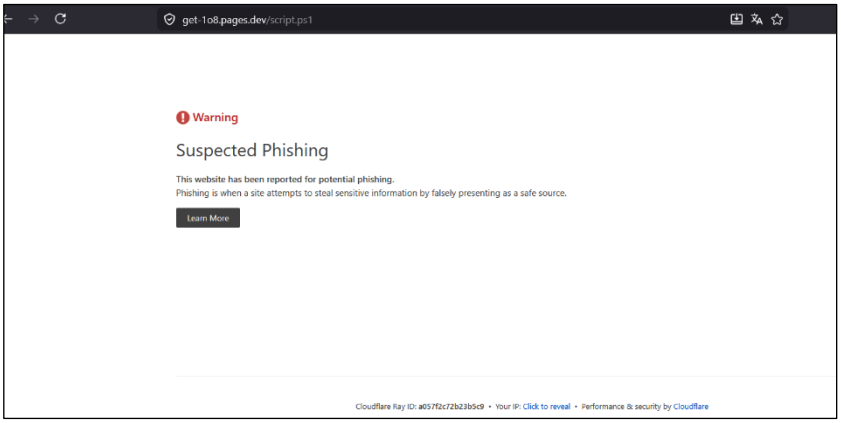
c. Pengujian IN-03

Pengujian IN-03 dilakukan dengan mengirimkan request menuju 16.491 URL ataupun domain malicious menggunakan script Python. Pengujian ini bertujuan untuk mengevaluasi kemampuan Zenarmor dalam memfilter akses terhadap URL dan domain dalam jumlah besar.



Gambar 4. 72 Hasil Pengujian IN-03

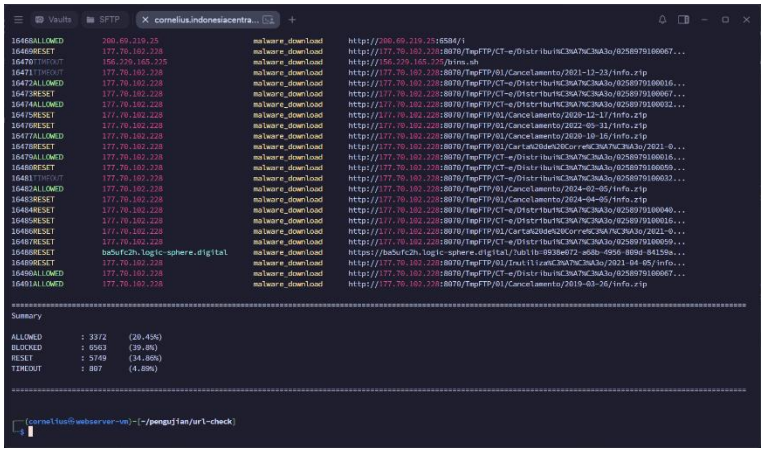
Pada kondisi ketika policy keamanan belum diaktifkan, terdapat 14.145 *request* yang berhasil diteruskan menuju domain tujuan dan memperoleh respons dari server tujuan. Meskipun dalam keadaan keamanan firewall tidak aktif, beberapa domain menampilkan halaman blocked ataupun connection reset yang dapat disebabkan koneksi tidak diteruskan oleh penyedia layanan ataupun ISP untuk alasan keamanan.



Gambar 4. 73 Halaman Blokir oleh *Cloud*flare

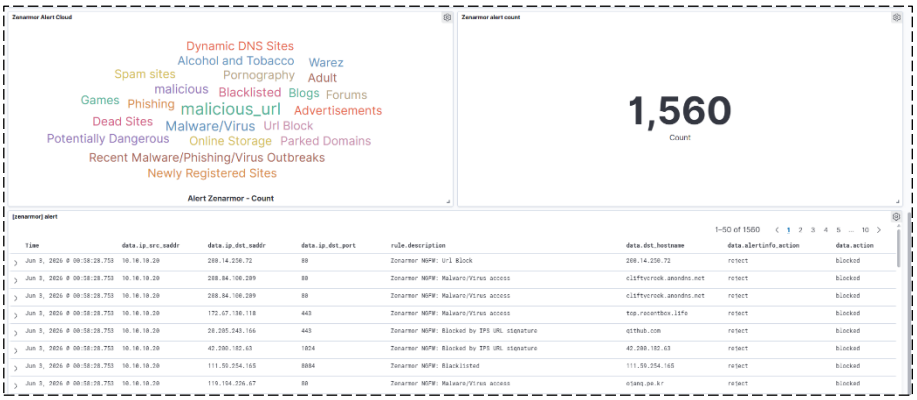
d. Akses Domain dan URL Malicious

Pengujian IN-04 dilakukan dengan mengirimkan request menuju sejumlah domain dan URL *malicious* menggunakan script Python yang telah disiapkan sebelumnya.



Gambar 4. 74 Hasil Pengujian IN-04

Setelah Zenarmor dan Suricata diaktifkan, jumlah request yang berhasil diblokir meningkat secara signifikan sehingga koneksi menuju sebagian besar domain *malicious* tidak dapat dilanjutkan.



Gambar 4. 75 *Alert* Zenarmor dari Pengujian IN-04

Jumlah request yang berhasil diblokir meningkat dari 1.763 menjadi 12.312 request setelah keamanan firewall diaktifkan. Sebaliknya, jumlah request yang berhasil diteruskan menuju domain tujuan menurun dari 14.145 menjadi 3.372.

Hasil pengujian ancaman internal menunjukkan bahwa fitur App Control, *Web Control* dan *Threat protection* yang diterapkan pada Zenarmor membatasi akses menuju domain berbahaya maupun kategori website yang dibatasi melalui policy keamanan. Ringkasan hasil pengujian ditunjukkan pada Tabel 4.22.

Tabel 4. 22 Rangkuman Hasil Pengujian Ancaman Internal

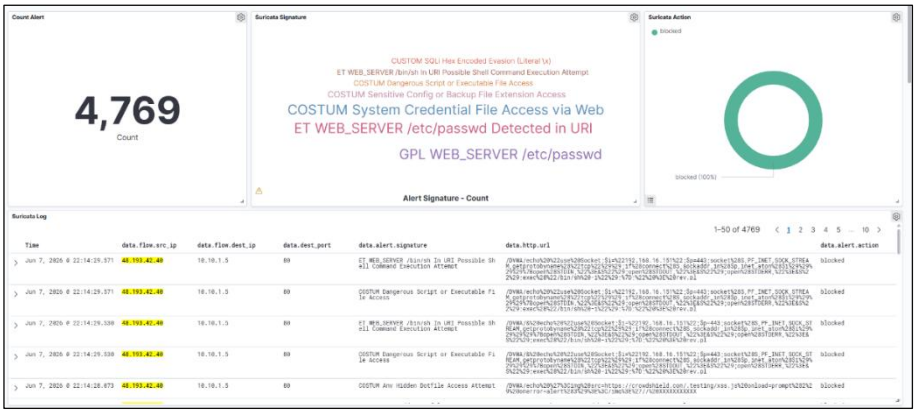
ID	Skenario	Allow	Block	Hasil
IN-01	Akses 20 website yang tidak sesuai dengan kebijakan saat keamanan firewall nonaktif	20	0	Domain berhasil diakses oleh client.
IN-02	Akses 20 website yang tidak sesuai dengan kebijakan saat keamanan firewall aktif	0	20	Domain tidak dapat diakses dan ditampilkan halaman blokir Zenarmor.
IN-03	Akses 16.491 domain atau URL malicious saat	14,145	1,763	Website kategori Proxy tidak dapat diakses sesuai policy yang diterapkan.

	keamanan firewall nonaktif			
IN-04	Akses 16.491 domain atau URL malicious saat keamanan firewall aktif	3,372	12,312	Sebagian domain berhasil diblokir dan menghasilkan *alert* pada sistem monitoring.

4.4.3.4 Hasil Pengujian Akurasi Deteksi

Pengujian akurasi deteksi dilakukan untuk mengukur kemampuan NGFW dalam membedakan aktivitas serangan dan aktivitas legitimate. Pengujian dilakukan dengan mengirimkan 2.000 request *malicious* dan 1.000 request benign menuju Web Server menggunakan tool FFUF.

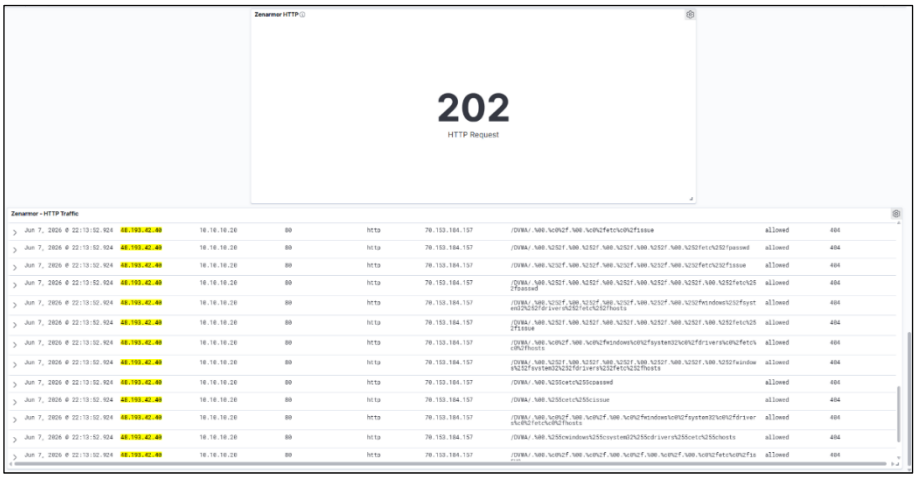
Pengiriman request *malicious* dilakukan menggunakan wordlist yang berisi *payload* serangan aplikasi web dan akses menuju direktori sensitif. Selama proses pengujian, Suricata melakukan inspeksi terhadap seluruh request yang diterima. Request yang berhasil dikenali sebagai serangan menghasilkan *alert* pada Suricata dan diteruskan ke Wazuh SIEM. Gambar 4.76 menunjukkan *alert* yang dihasilkan Suricata terhadap request *malicious* yang dikirimkan.



Gambar 4. 76 *Alert* Suricata pada Pengujian AD-01

Berdasarkan hasil pengujian, seluruh request yang menghasilkan *alert* pada Suricata memperoleh *action* "*blocked*". Namun untuk mengetahui request *malicious* yang tidak terdeteksi, diperlukan pengamatan terhadap trafik HTTP yang berhasil diteruskan menuju Web Server.

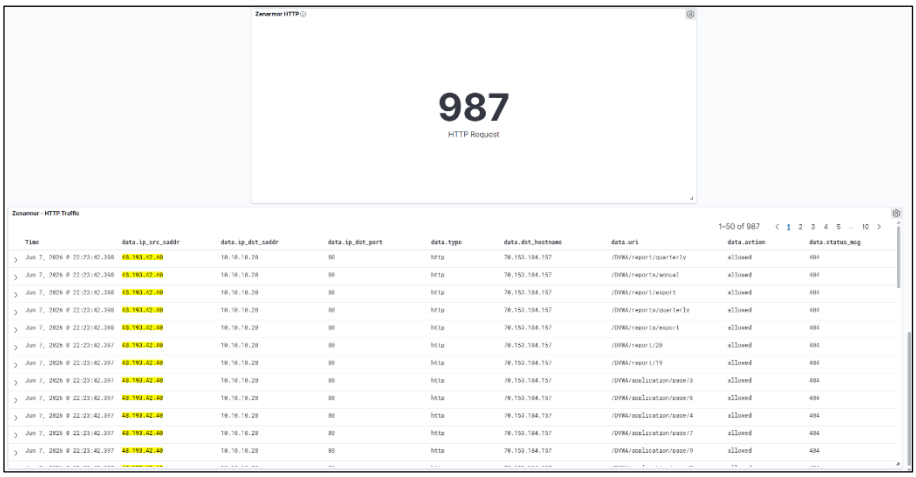
Pada sistem yang dibangun, seluruh trafik HTTP yang berhasil melewati Suricata akan dicatat oleh Zenarmor. Oleh karena itu, request *malicious* yang masih muncul pada log HTTP Zenarmor dapat dikategorikan sebagai request yang tidak terdeteksi atau tidak diblokir oleh *rule* Suricata. Gambar 4.77 menunjukkan contoh request *malicious* yang berhasil diteruskan menuju Web Server dan tercatat pada log Zenarmor.



Gambar 4. 77 Trafik HTTP yang Lolos pada Pengujian AD-01

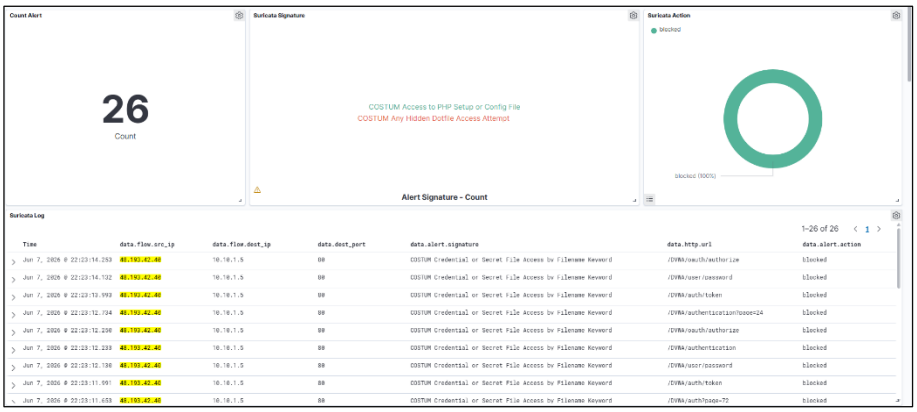
Selain request *malicious*, request benign dikirim untuk mengukur false positive. Request benign terdiri dari URL dan resource yang umum diakses oleh pengguna normal saat menggunakan aplikasi web.

Hasil pengujian lainnya yaiut dengan mengirimkan *request benign*, sebagian besar trafik berhasil diteruskan menuju Web Server dan tercatat sebagai trafik HTTP normal pada Zenarmor. Hal ini menunjukkan bahwa *request* tersebut tidak dianggap sebagai aktivitas serangan oleh sistem keamanan.



Gambar 4. 78 Trafik HTTP *Request Benign* pada Pengujian AD-02

Terdapat *request benign* yang menghasilkan *alert* pada Suricata. *Alert* tersebut dikategorikan sebagai *false positive* karena request yang dikirim tidak mengandung *payload* berbahaya namun terdeteksi sebagai aktivitas serangan.



Gambar 4. 79 *Alert* Suricata pada Pengujian AD-02

Hasil pengujian akurasi deteksi dirangkum pada Tabel 4.23.

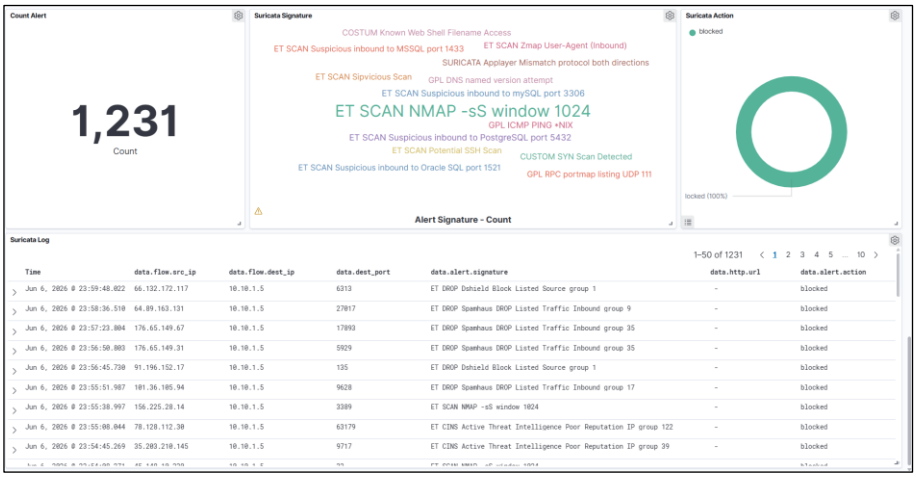
Tabel 4. 23 Hasil Pengujian Akurasi Deteksi

ID	Skenario	Request Terblokir	Request Diteruskan
AD-01	Mengirimkan 2.000 request *malicious*	1798	202
AD-02	Mengirimkan 1.000 request benign	13	987

4.4.3.5 Hasil Pengujian Serangan Harian

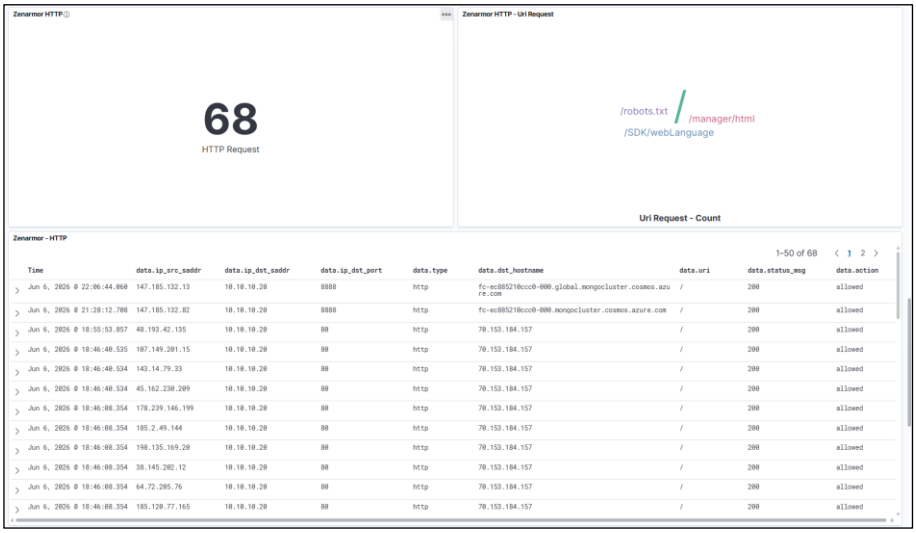
Pengujian serangan harian dilakukan dengan menempatkan Web Server pada jaringan publik selama 7 hari. Selama periode tersebut, seluruh aktivitas serangan yang berasal dari internet dipantau menggunakan Suricata IPS dan Zenarmor. Tujuan pengujian ini adalah untuk mengetahui ancaman yang diterima oleh Web Server pada lingkungan nyata serta mengevaluasi kemampuan sistem dalam melakukan monitoring dan pencatatan aktivitas keamanan.

Pada hari ke-1, Suricata mencatat terdapat 1,231 *alert*, dimana sekitar 39 *alert* diantaranya merupakan request yang berisi *payload malicious*



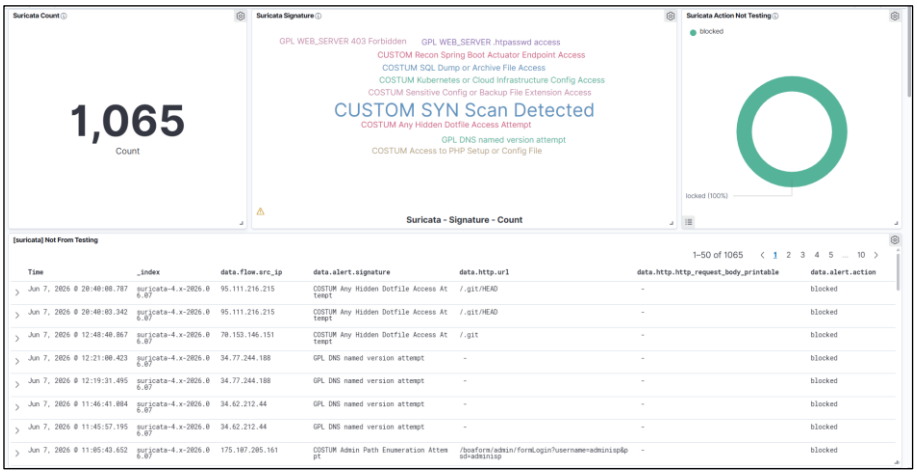
Gambar 4. 80 *Alert* Suricata pada Pengujian Serangan Harian

Zenarmor mencatat terdapat 68 request yang diteruskan menuju Web Server seperti yang ditunjukkan pada Gambar 4.81.



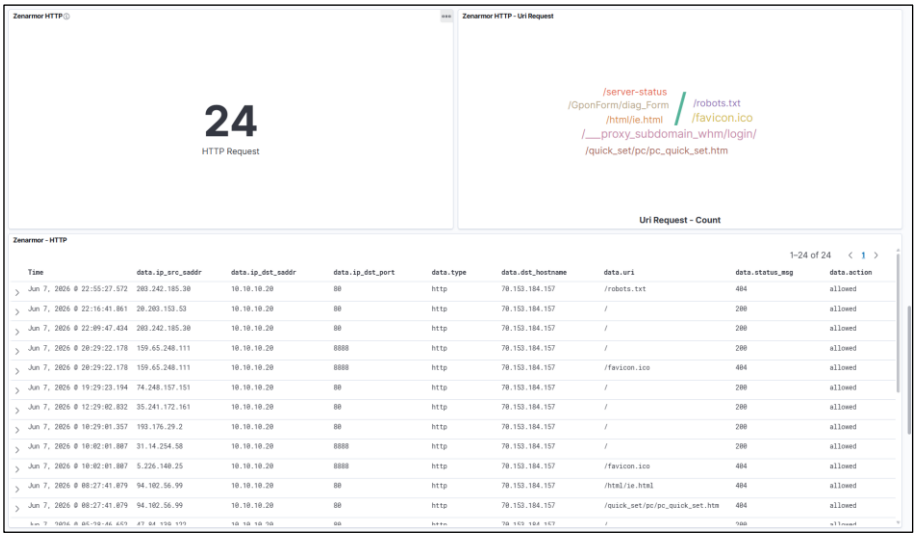
Gambar 4. 81 Aktivitas yang Tercatat oleh Zenarmor pada Hari Pertama

Pada hari ke 2, terdapat 1,065 *alert* pada Suricata yang mana sebanyak 35 request mengandung *payload malicious*.



Gambar 4. 82 *Alert* Suricata Pengujian Serangan Harian ke-2

Sementara itu, Zenarmor mencatat sebanyak 24 request yang diteruskan menuju Web Server selama periode pengamatan hari kedua.



Gambar 4. 83 Aktivitas yang Tercatat oleh Zenarmor pada Hari Kedua

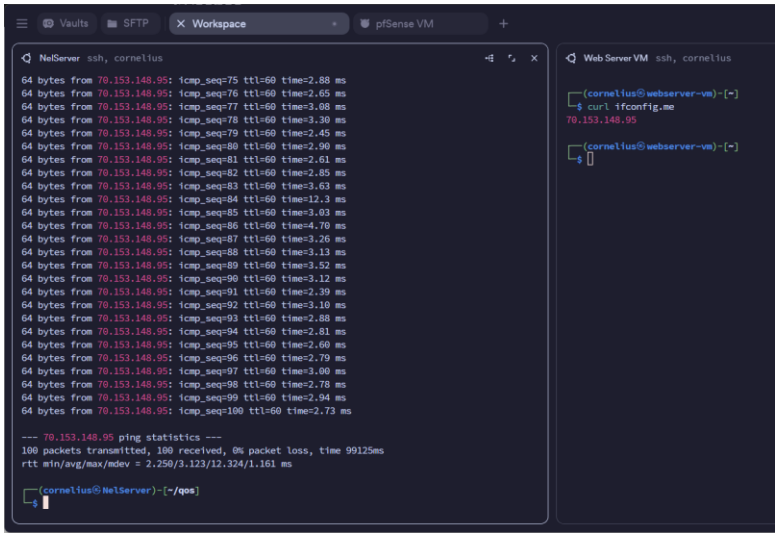
4.4.3.6 Hasil Pengujian Quality of Service (QoS)

Pengujian Quality of Service (QoS) dilakukan untuk memperoleh data kualitas jaringan pada tiga kondisi pengujian, yaitu sebelum sistem keamanan diaktifkan, setelah sistem keamanan diaktifkan, dan ketika sistem keamanan menerima serangan. Parameter yang diamati meliputi *delay*, *packet loss*, *throughput*.

Pengujian dilakukan pada empat kondisi, yaitu sebelum implementasi sistem keamanan, setelah sistem keamanan diaktifkan, ketika firewall menerima serangan, dan setelah serangan berhasil ditangani

a. Pengujian *Delay*

Pengujian *delay* dilakukan dengan mengambil nilai rata rata dari pengiriman paket ICMP.



Gambar 4. 84 Hasil Pengujian *Delay* Menggunakan ICMP

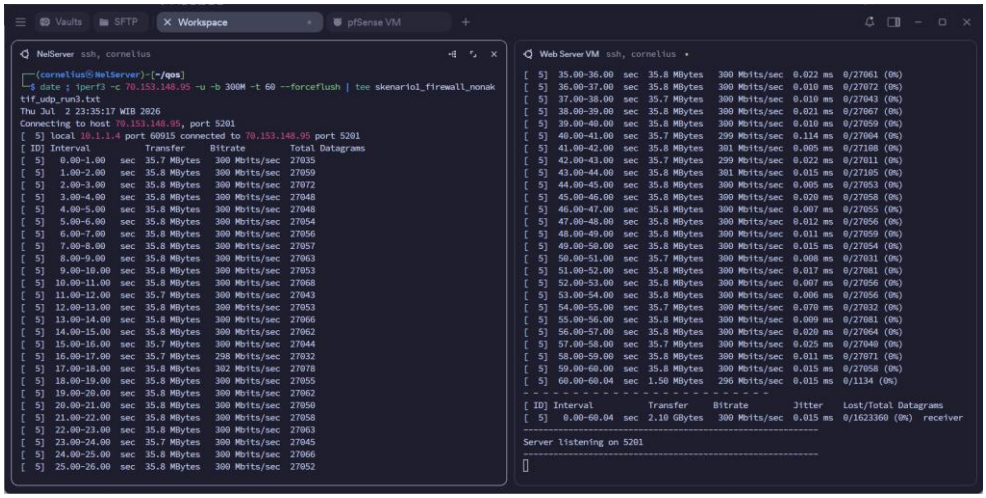
Hasil pengukuran *delay* ditunjukkan pada Tabel 4.24.

Tabel 4. 24 Hasil Pengujian *Delay* dalam *Delay* One Way *Delay* (OWD).

Pengujian	**Keamanan Firewall Nonaktif (ms)**	**Keamanan Firewall Aktif (ms)**	**Firewall Mengalami Serangan (ms)**	**Firewall Setelah Serangan (ms)**
1	1.562	1.559	1.559	1.581
2	1.538	1.590	1.787	1.557
3	1.500	1.690	1.555	2.936

b. Pengujian Jitter dan Packet Loss

Pengujian *jitter* dan *packet loss* dilakukan menggunakan aplikasi iPerf3 dengan protokol UDP. Nilai *jitter* diperoleh dari variasi waktu kedatangan paket (packet *delay* variation) yang dihitung secara otomatis oleh iPerf3 selama proses transmisi data berlangsung.



Gambar 4. 85 Pengujian Jitter dan Packet Loss

Hasil pengukuran *jitter* ditunjukkan pada Tabel 4.25.

Tabel 4. 25 Hasil Pengujian Jitter

Pengujian	Keamanan Firewall Nonaktif (ms)	Keamanan Firewall Aktif (ms)	Firewall Mengalami Serangan (ms)	Firewall Setelah Serangan (ms)
1	0.018	0.032	0.046	0.026
2	0.017	0.024	0.021	0.023
3	0.015	0.007	0.015	0.046

Persentase *packet loss* diperoleh dari perbandingan jumlah paket UDP yang hilang dari total paket UDP yang dikirim yang diambil aari hasil pengujian iPerf3. Hasil pengukuran *packet loss* ditunjukkan pada Tabel 4.26.

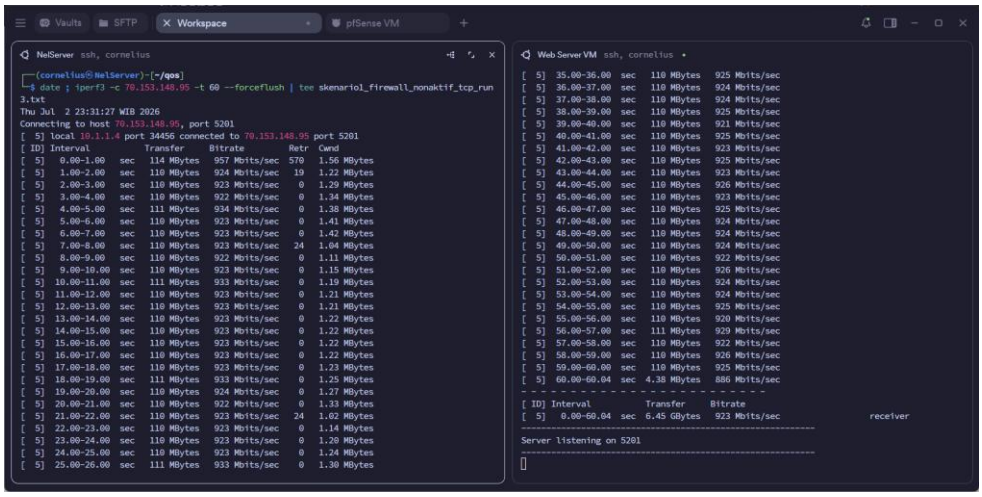
Tabel 4. 26 Hasil Pengujian Packet Loss

Pengujian	Keamanan Firewall Nonaktif (%)	Keamanan Firewall Aktif (%)	Firewall Mengalami Serangan (%)	Firewall Setelah Serangan (%)
1	0	0	0	2.6%
2	0	0	0	0.2%
3	0	0	0	0.66%

c. Pengujian *Throughput*

Pengujian *throughput* dilakukan menggunakan aplikasi iPerf3 dengan protokol TCP untuk mengukur kecepatan transfer data antara Client dan Web

Server yang melewati perangkat firewall. Nilai *throughput* diperoleh dari rata-rata bandwidth yang berhasil ditransmisikan selama durasi pengujian.



Gambar 4. 86 Pengujian Throughput

Hasil pengukuran *throughput* ditunjukkan pada Tabel 4.27.

Tabel 4. 27 Hasil Pengujian Throughput

Pengujian	Keamanan Firewall Nonaktif (Mbps)	Keamanan Firewall Aktif (Mbps)	Firewall Mengalami Serangan (Mbps)	Firewall Setelah Serangan (Mbps)
1	912	885	799	715
2	924	847	828	801
3	923	906	820	659

4.4.4 Analisis Data

4.4.4.1 Analisis Pengujian Fungsionalitas

Berdasarkan hasil pengujian fungsionalitas pada PF-01 hingga PF-09, seluruh komponen sistem berhasil beroperasi sesuai dengan perancangan yang telah dibuat. Klien dapat terhubung dengan VPN dan berhasil untuk mengakses layanan internal, seperti dashboard administrasi pfSense maupun Wazuh. Selain itu pengujian konektivitas internet pada jaringan internal berjalan dengan baik dimana layanan Web Server berhasil diakses melalui internet melalui mekanisme *routing* dan port forwarding yang dikonfigurasi pada pfSense dan Server internal berhasil mengakses internet melalui *gateway* pfSense firewall.

Pengujian juga menunjukkan bahwa fungsi Suricata IPS mampu melakukan deteksi dan pemblokiran terhadap trafik *malicious* yang dikirim menuju Web Server. *Alert* yang dihasilkan Suricata berhasil diteruskan menuju Wazuh SIEM melalui mekanisme log forwarding yang telah dikonfigurasi. Pada Zenarmor, fitur *Web Control* berhasil memblokir akses menuju domain *malicious* dan log yang dihasilkan juga berhasil diteruskan menuju Wazuh SIEM. Hasil tersebut menunjukkan bahwa integrasi antara pfSense, Suricata, Zenarmor, dan Wazuh telah berjalan sesuai dengan rancangan sistem yang dibuat.

Tabel 4. 28 Hasil Pengujian Fungsionalitas

ID	Skenario Pengujian	Status
PF-01	Pengujian koneksi VPN	Berhasil
PF-02	Pengujian akses dashboard Wazuh dengan VPN aktif	Berhasil
PF-03	Pengujian akses dashboard pfSense dengan VPN aktif	Berhasil
PF-04	Pengujian konektivitas internet server internal	Berhasil
PF-05	Pengujian akses Web service melalui internet	Berhasil
PF-06	Pengujian pemblokiran serangan oleh Suricata	Berhasil
PF-07	Pengujian forwarding log Suricata	Berhasil
PF-08	Pengujian melakukan request menuju domain *malicious*	Berhasil
PF-09	Pengujian forwarding log Zenarmor	Berhasil

4.4.4.2 Analisis Efektivitas Pemblokiran

Efektivitas pemblokiran dianalisis berdasarkan kemampuan sistem dalam menghentikan aktivitas serangan sebelum mencapai target server. Analisis dilakukan dengan membandingkan hasil pengujian pada kondisi firewall nonaktif dan setelah fitur keamanan NGFW diaktifkan. Parameter yang diamati meliputi keberhasilan eksploitasi serangan, jumlah resource yang berhasil ditemukan selama proses *reconnaissance*, serta kemampuan sistem dalam memblokir akses menuju domain berbahaya.

a. Port *Scanning*

   Pengujian port *scanning* menunjukkan bahwa NGFW mampu mengurangi informasi yang berhasil diperoleh selama proses *reconnaissance*. Pada

kondisi keamanan firewall nonaktif, Nmap berhasil menemukan seluruh port dan layanan yang terbuka pada Web Server.

Setelah sistem keamanan firewall diaktifkan yaitu Suricata IPS, jumlah port yang berhasil ditemukan terbuka mengalami penurunan. Selain menghasilkan *alert* keamanan, sebagian paket *scanning* juga berhasil difilter oleh sistem sehingga proses identifikasi layanan menjadi lebih terbatas. Hasil ini menunjukkan bahwa implementasi NGFW mampu mengurangi efektivitas *reconnaissance* yang dilakukan attacker. Efektivitas pemblokiran port *scanning* dapat dihitung menggunakan persamaan:

$$Blocking\ Rate = \frac{Port_{\text{before}} - Port_{\text{after}}}{Port_{before}} \times 100\%$$

$$Blocking\ Rate = \frac{9\ -2}{9} \times 100\% = 77{,}78\%$$

Nilai tersebut menunjukkan bahwa implementasi NGFW berhasil mengurangi informasi port yang dapat diperoleh attacker sebesar 77,78%.

b. SQL Injection

Hasil pengujian SQL Injection menunjukkan perbedaan antara kondisi sebelum dan sesudah sistem keamanan diaktifkan. Pada kondisi keamanan firewall nonaktif, request SQL Injection berhasil mencapai aplikasi Web Server sehingga sqlmap dapat memperoleh informasi struktur database dan data pengguna. Aktivitas tersebut terdeteksi pada log Web Server dan menghasilkan *alert* pada Wazuh Agent.

Ketika keamanan firewall diaktifkan, request SQL Injection berhasil dideteksi dan diblokir. Hal ini terlihat pada output sqlmap yang tidak mendapatkan struktur database maupun data pengguna. Selain itu tidak ditemukannya request SQL Injection pada log Web Server. Meskipun ditemukan beberapa *alert* Suricata dengan *action alert*, ditemukan bahwa request yang sama juga diproses oleh *rule* lain dengan *action drop*. Namun Wazuh HIDS tidak mendeteksi adanya *SQL Injection* yang diterima server sehingga dapat disimpulkan bahwa request yang mendapatkan *action alert* berhasil diblokir melalui *rule* yang memiliki *action drop*. Berdasarkan hasil tersebut,

efektivitas pemblokiran SQL Injection menggunakan sqlmap dapat dinyatakan mencapai 100%.

c. Directory *Scanning*

Pengujian directory *scanning* dilakukan menggunakan FFUF dengan wordlist berisi 16,089 request. Pada kondisi firewall nonaktif, FFUF berhasil menemukan 50 direktori ataupun file yang dapat diakses pada Web Server. Meskipun tidak semua file ataupun direktori bersifat *malicious*, informasi dapat digunakan attacker untuk enumerasi dan serangan lebih lanjut.

Setelah keamanan firewall diaktifkan, jumlah file dan direktori yang ditemukan mengalami penurunan, yaitu menjadi 4. Efektivitas pemblokiran directory *scanning* yang terjadi yaitu

$$Blocking\ Rate = \frac{Directory_{\text{before}} - Directory_{\text{after}}}{Directory_{before}} \times 100\%$$

$$Blocking\ Rate = \frac{50\ -4}{50} \times 100\% = 92\%$$

Nilai tersebut menunjukkan bahwa implementasi NGFW mampu mengurangi keberhasilan proses directory enumeration sebesar 92% selama pengujian berlangsung.

d. Akses Domain dan URL Malicious

Pengujian dilakukan dengan menggunakan script python untuk melakukan akses dan koneksi ke URL dan domain yang dikategorikan *malicious*. Ketika keamanan firewall tidak aktif, sebagian besar domain berhasil diakses. Namun ketika keamanan firewall diaktifkan, jumlah koneksi yang berhasil di blokir meningkat. Efektivitas pemblokiran dihitung dengan:

$$Blocking\ Rate = \frac{Allow_{\text{before}} - Allow_{\text{after}}}{Allow_{before}} \times 100\%$$

$$Blocking\ Rate = \frac{14.145\ -3.372}{14.145} \times 100\% = 76{,}16\%$$

Nilai tersebut menunjukkan bahwa Zenarmor berhasil memblokir sekitar 76,16% koneksi menuju domain ataupun url malicious dibandingkan keadaan awal.

e. Analisis Pemblokiran Serangan DoS

Berdasarkan pengujian DoS yang dilakukan, terlihat perbedaan bebas Web Server pada kondisi sebelum menggunakan NGFW dan setelah menggunakan NGFW.

Tabel 4. 29 Hasil Pengujian DoS ketika Firewall Nonaktif

	Keadaan Normal			**SYN Flood**		
	CPU%	**RAM**	**RAM%**	**CPU%**	**RAM**	**RAM%**
MIN	3,8%	642	33,8%	2,9%	744	39,1%
MAX	9,4%	644	33,9%	7,5%	1385	72,9%
AVG	4,57%	643,50	33,9%	4,88%	1208,00	63,55%

Berdasarkan data pengujian yang terdapat pada Tabel 4.29, dalam keadaan normal penggunaan RAM rata-rata server sebanyak 33.9% dan CPU sebanyak 4.57%. Kemudian ketika mendapatkan serangan DoS, rata-rata penggunaan CPU sebesar 4.88% dan RAM 63.55% yang menunjukkan adanya peningkatan.

Tabel 4. 30 Hasil Pengujian DoS ketika Firewall Aktif

	Keadaan Normal			**SYN Flood**		
	CPU%	**RAM**	**RAM%**	**CPU%**	**RAM**	**RAM%**
MIN	3,8%	650	34,2%	3,8%	650	34,2%
MAX	7,6%	660	34,7%	7,6%	660	34,7%
AVG	4,7%	653,17	34,3%	4,72%	651,52	34,2%

Kemudian setelah firewall NGFW aktif, berdasarkan data Tabel 4.30, adanya serangan DoS tidak memberikan dampak signifikan terhadap penggunaan resource Web Server. Saat terkena SYN Flood, rata-rata penggunaan CPU sebesar 4,72% yang tidak jauh berbeda dengan rata-rata CPU dalam kondisi normal, sedangkan penggunaan RAM tetap di sekitar 34%. Hal ini menunjukkan penggunaan NGFW tidak hanya membantu mendeteksi aktivitas serangan tetapi juga mengurangi beban yang diterima oleh Web Server.

Berdasarkan pengujian yang telah dilakukan, implementasi NGFW berhasil meningkatkan keamanan jaringan melalui kemampuan deteksi dan pencegahan terhadap berbagai jenis ancaman. Sistem mampu mengurangi efektivitas *reconnaissance* melalui port *scanning* dan *directory scanning*, mencegah

eksploitasi SQL Injection, membatasi akses menuju domain *malicious*, serta mengurangi dampak serangan DoS terhadap Web Server.

4.4.4.3 Analisis Akurasi Deteksi

Analisis akurasi deteksi dilakukan untuk mengevaluasi kemampuan sistem keamanan dalam membedakan aktivitas *malicious* dan aktivitas benign. Pengukuran dilakukan berdasarkan hasil pengujian yang telah disajikan pada Tabel 4.31 dengan mengklasifikasikan hasil deteksi ke dalam kategori *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Tabel 4. 31 Confusion Matrix Pengujian Akurasi Deteksi

		Predicted Payload	
		Malicious	*Benign*
Actual Payload	*Malicious*	1798 (TP)	202 (FN)
	Benign	13 (FP)	987 (TN)

Berdasarkan hasil pengujian, dari 2,000 request *malicious* yang dikirimkan, sebanyak 1,798 berhasil dilakukan blokir terhadap paket trafik. Nilai tersebut dikategorikan sebagai *True Positive* karena sistem berhasil mengenali trafik berbahaya. Terdapat 202 *request* yang tidak berhasil terdeteksi dan diteruskan menuju Web Server sehingga dikategorikan sebagai *False Negative* (FN).

Pada pengujian *request* benign, sebanyak 987 *request* diteruskan menuju Web Server dan tidak terdapat terdeteksi sebagai ancaman sehingga dikategorikan sebagai *True Negative* (TN), sedangkan 13 *request* menghasilkan *alert* dan diblokir oleh sistem keamanan sehingga dikategorikan sebagai *False Positive* (FP).

Akurasi:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Accuracy = \frac{1798 + 987}{1798 + 987 + 13 + 202}$$

$$Accuracy = \frac{2785}{3000}$$

$$Accuracy = 92.83\%$$

Detection Rate (Recall):

$$Recall = \frac{TP}{TP + FN}$$
$$Recall = \frac{1798}{1798 + 202}$$
$$Recall = 89.90\%$$

False Positive Rate:

$$FPR = \frac{FP}{FP + TN}$$
$$FPR = \frac{13}{13 + 987}$$
$$FPR = 1.30\%$$

Didapatkan nilai akurasi sebesar 92,83% yang berarti sistem keamanan mampu melakukan klasifikasi terhadap trafik jaringan *benign* dan *malicious* dengan baik. Nilai precision sebesar 99,28% menunjukkan bahwa sebagian besar *alert* yang dihasilkan sistem merupakan deteksi yang valid. Nilai recall sebesar 89,90% menunjukkan bahwa sebagian besar aktivitas serangan berhasil dideteksi oleh sistem keamanan meskipun masih terdapat sejumlah *request malicious* yang berhasil lewat dan diteruskan menuju Web Server.

Berdasarkan hasil pengujian didapatkan nilai accuracy sebesar 92,83%. Selain itu, sistem memiliki detection rate sebesar 89,90%, yang menunjukkan bahwa sebagian besar serangan berhasil dideteksi dan diblokir. Pada trafik benign diperoleh false positive rate sebesar 1,30%, yang menunjukkan bahwa hanya sebagian kecil trafik normal yang salah dikategorikan sebagai ancaman.

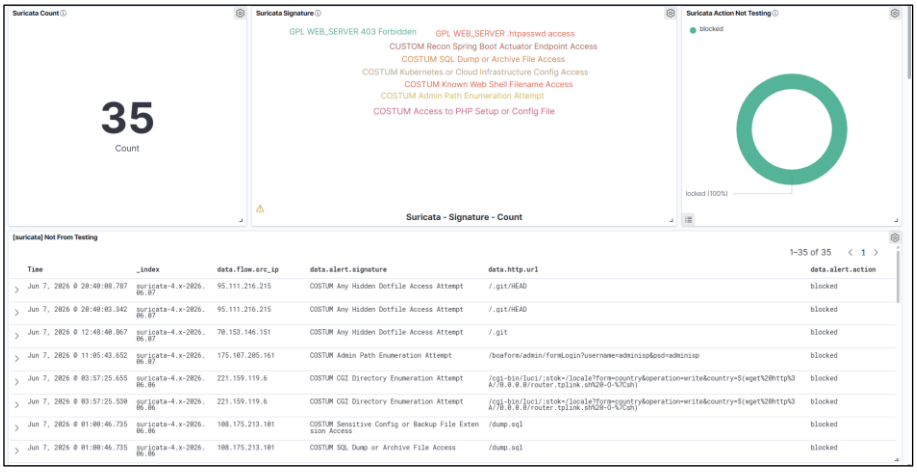
4.4.4.4 Analisis Serangan Harian

Melalui pengujian serangan harian selama 7 hari yaitu mengamati akses trafik yang mengakses server dari internet, server banyak menerima banyak aktivitas yang dapat dikategorikan sebagai *suspicious* dan *malicious*. Data pengujian serangan harian ditampilkan pada Tabel 4.32.

Tabel 4. 32 Hasil Pengujian Serangan Harian

Hari	*Alert* Suricata		Trafik Zenarmor		Total
	Suspicious	*Malicious*	*Suspicious*	*Malicious*	
1	1.192	39	68	0	1.299
2	1.030	35	24	0	1.089
3	2.202	8	11	0	2.221
4	3.130	18	160	0	3.308
5	5.324	10	110	0	5.444
6	3.175	10	109	0	3.294
7	1.240	9	16	0	1.265

Berdasarkan hasil pengamatan selama tujuh hari, Suricata menghasilkan sejumlah alert yang dikategorikan sebagai suspicious dan malicious. Aktivitas suspicious menunjukkan adanya trafik yang memiliki karakteristik tidak normal berdasarkan pola komunikasi misalnya aktivitas scanning ataupun permintaan terhadap file, direktori, atau resource yang umum digunakan untuk mengidentifikasi kerentanan pada server. Sementara itu, aktivitas yang dikategorikan sebagai malicious menunjukkan adanya indikasi trafik yang memiliki karakteristik serangan dan mengandung pola eksploitasi terhadap layanan yang berpotensi membahayakan sistem.



Gambar 4. 87 *Request* yang Terdeteksi pada Pengujian Akurasi

Zenarmor tidak mencatat adanya *request* yang dikategorikan sebagai *malicious* selama periode pengujian. Hal ini menunjukkan bahwa sebagian besar aktivitas

yang diteruskan menuju Web Server hanya berupa *request* suspicious yang tidak mengandung *payload* berbahaya. Sementara itu, *request* yang teridentifikasi sebagai *malicious* telah terlebih dahulu dideteksi dan diblokir oleh Suricata.

Melalui pengujian ini, dapat menunjukkan bahwa meskipun tidak dilakukan pengujian serangan secara langsung, Web Server yang terhubung ke internet tetap menerima berbagai aktivitas mencurigakan setiap hari.

4.4.4.5 Analisis *Quality of Service*

Berdasarkan pengujian QoS yang dilakukan, didapatkan data yang menunjukkan adanya perbedaan akibat implementasi sistem NGFW. Data rata-rata parameter ditunjukkan pada Tabel 4.33.

Tabel 4. 33 Hasil Rata-Rata Pengujian *Quality of Service* (QoS)

Parameter	**Keamanan Firewall Nonaktif**	**Keamanan Firewall Aktif**	**Firewall Mengalami Serangan**	**Firewall Setelah Serangan (ms)**
Delay	1.533 ms	1.648 ms	1.634 ms	2.025 ms
Jitter	0.017 ms	0.021 ms	0.027 ms	0.032 ms
Packet Loss	0	0	0	1,15 %
Throughput	919.7 Mbps	879.3 Mbps	815.7 Mbps	725.0 Mbps

Berdasarkan pengujian QoS yang dilakukan, terjadi penurunan pada *throughput* dibandingkan ketika keamanan nonaktif yaitu ketika Suricata dan Zenarmor tidak aktif. Hal ini dapat disebabkan karena seluruh trafik perlu diinspeksi terlebih dahulu oleh firewall. Setelah firewall diaktifkan, *throughput* turun sebesar 4,4%, dan penurunan semakin besar ketika sistem menerima serangan DoS yaitu sebesar 11,3%. Penurunan *throughput* pada kondisi setelah serangan sebesar 21,2%, yang menunjukkan kondisi firewall yang belum kembali normal sepenuhnya setelah serangan berhenti. Hal serupa juga terjadi pada parameter lain seperti *delay*, *jitter*, dan *packet loss* yang mengalami kenaikan di setiap skenario.

Tabel 4. 34 Standar Parameter QoS TIPHON

Index	**Kategori**	***Delay***	**Jitter**	**Packet Loss**	**Throughput**
4	Sangat Bagus	<150 ms	0 ms	0 – 2 %	>2,1 Mbps

3	Bagus	150 - 300 ms	1 - 75 ms	3 – 14 %	1,2 – 2,1 Mbps
2	Sedang	300 - 450 ms	76 - 125 ms	12 – 24 %	338 – 1.200 Kbps
1	Jelek	> 450 ms	>225 ms	>25%	0–338 Kbps

Berdasarkan perbandingan dengan standar TIPHON pada Tabel 4.34, seluruh parameter QoS pada keempat skenario pengujian berada dalam kategori Sangat Bagus dengan indeks 4. Nilai *delay* tertinggi sebesar 2,025 ms yang berada di bawah batas 150 ms, *jitter* tertinggi sebesar 0,032 ms berada di bawah batas 75 ms, *packet loss* tertinggi sebesar 1,15% masih dalam rentang 0–2%, dan *throughput* terendah sebesar 725,0 Mbps berada di atas batas minimal 2,1 Mbps.

Secara keseluruhan, hasil pengujian menunjukkan bahwa implementasi NGFW memberikan dampak terhadap performa jaringan, terutama pada *throughput*. Namun demikian, seluruh parameter tetap berada pada kategori sangat bagus berdasarkan standar TIPHON sehingga layanan jaringan masih dapat beroperasi dengan baik meskipun seluruh mekanisme keamanan diaktifkan.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Sistem keamanan berbasis *Next-Generation Firewall* (NGFW) berhasil diimplementasikan menggunakan pfSense yang diintegrasikan dengan Suricata IPS dan Zenarmor pada lingkungan *Cloud* Azure. Sistem yang dibangun menempatkan NGFW sebagai *gateway* utama sehingga seluruh trafik baik masuk ataupun keluar akan melewati proses inspeksi dan filtering sebelum diteruskan menuju tujuan.
2. Integrasi antara pfSense, Suricata, Zenarmor, dan Wazuh berhasil dilakukan. Log keamanan yang dihasilkan oleh Suricata dapat dikirimkan menggunakan Syslog-ng, sedangkan log Zenarmor berhasil diintegrasikan menggunakan *script* python yang membaca data dari database internal Zenarmor. Seluruh log dikumpulkan pada Wazuh SIEM sehingga *alert* dapat dipantau secara terpusat.
3. Berdasarkan pengujian ancaman eksternal, sistem mampu mendeteksi dan memblokir berbagai aktivitas serangan seperti port *scanning* dan SQL Injection sebelum mencapai Web Server. Pada pengujian SQL Injection, seluruh *request malicious* berhasil diblokir dan tidak ditemukan aktivitas serangan yang tercatat pada Web Server melalui Wazuh Agent HIDS.
4. Hasil pengujian serangan harian menunjukkan bahwa Web Server yang terhubung langsung ke internet menerima berbagai aktivitas mencurigakan setiap hari, mulai dari koneksi yang berasal dari alamat IP dengan reputasi buruk hingga *request* yang mengandung *payload* yang terindikasi sebagai upaya eksploitasi. Seluruh aktivitas tersebut berhasil dimonitor dan *request* berbahaya berhasil diblokir oleh sistem keamanan.

5.2 Saran

Adapun beberapa saran yang dapat dilakukan pada penelitian selanjutnya, antara lain:

1. Penelitian selanjutnya dapat menggunakan SIEM lain baik berbasis komersial ataupun *open source* untuk dapat dibandingkan kemampuan serta performa sistem.
2. Integrasi sistem keamanan dapat diperluas dengan menambahkan sumber log lain seperti WAF maupun NDR.
3. Implementasi dapat dilakukan pada lingkungan jaringan yang berbeda misalnya pada lingkungan on-premises, untuk mengetahui kemampuan sistem dalam menangani trafik yang lebih besar.
4. Penelitian selanjutnya dapat melakukan evaluasi terhadap performa dan efektivitas implementasi NGFW berbasis *open source* serta membandingkannya dengan solusi NGFW komersial.

DAFTAR PUSTAKA

Ang, S., Huy, S. & Janarthanan, M., 2025, 'Utilizing IDS and IPS to Improve Cybersecurity Monitoring Process', *Journal of Cyber Security and Risk Auditing*, 2025(3), 77.

Badan Pusat Statistik, 2024, *Statistik Telekomunikasi Indonesia 2024*, BPS, Badan Pusat Statistik.

Bezas, K. & Filippidou, F., 2023, 'Comparative Analysis of Open Source Security Information & Event Management Systems (SIEMs)', *Comparative Analysis of Open Source Security Information & Event ManagementSystems (SIEMs)*, 12, 444–468.

Gupta, P., 2024, *Review of Next-Generation Firewalls*.

Liang, J. & Kim, Y., 2022, *Evolution of Firewalls: Toward Securer Network Using Next Generation Firewall*, *2022 IEEE 12th Annual Computing and Communication Workshop and Conference, CCWC 2022*, 752–759, Institute of Electrical and Electronics Engineers Inc.

Manzoor, J., Waleed, A., Jamali, A.F. & Masood, A., 2024, 'Cybersecurity on a budget: Evaluating security and performance of *open-source* SIEM solutions for SMEs', *PLOS ONE*, 19(3 March).

Muhamad Malik Matin, I., Kurniawan, A., Rosyida Zain, A., Agustin, M., Informatika dan Komputer, T. & Negeri Jakarta, P., 2023, 'Sistem Monitoring Keamanan Pada Data Center Berbasis Security Information And Event Management (Siem) Dengan Wazuh dan IDS', *Journal of Innovative and Creativity*, 5(1), 2025.

Open Information Security Foundation (OISF), 2026, *Suricata User Guide*.

Patil, M. & Mohurle, S., 2017, 'The Empirical Study of the Evolution of the Next Generation Firewalls', *International Journal of Trend in Scientific Research and Development*, 1(5), 193–196.

Satilmis, H., Akleylek, S. & Yuce Tok, Z., 2025, 'Development of Various Stacking Ensemble-Based HIDS Using ADFA Datasets', *IEEE Open Journal of the Communications Society*, 6, 1170–1189.

Sulaiman, F.S., Seta, H.B. & Falih, N., 2021, *Exploitation Prevention on Network Printer with Signature-Based Suricata on PfSense*, *Proceedings - 3rd International Conference on Informatics, Multimedia, Cyber, and Information System, ICIMCIS 2021*, 35–39, Institute of Electrical and Electronics Engineers Inc.

Wazuh, 2024, *Wazuh Documentation*.

Wiguna, R., 2024, *Analisis Implementasi dan Integrasi Suricata dengan Scirius dan ntopng untuk Pendeteksian dan Pemblokiran Serangan pada Server dengan Notifikasi WhatsApp*.

Wiradyaksa, I.D., Putri, D.H., Iqbal, R.M., Astari, N.H., Karna, N. & Dewanta, F., 2024, *Design and Implementation of Automated Web Application Firewall, Rate Limiting, and Intrusion Detection System for Cyber Defense*, *2024 8th International Conference on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2024*, 256–261, Institute of Electrical and Electronics Engineers Inc.

Zenarmor, 2024, *What is Web Filtering? Types, Benefits, Challenges and Best Practices*.

Zenarmor, 2026, *Welcome to the Zenarmor User Guide*.

DAFTAR RIWAYAT HIDUP

Cornelius Yuli Rosdianto, lahir pada tahun 2004, merupakan anak kedua dari tiga bersaudara. Penulis menyelesaikan pendidikan menengah pertama di SMPN 1 Cikarang Barat dan pendidikan menengah atas di SMAN 4 Tambun Selatan. Pada tahun 2022, penulis melanjutkan pendidikan di Program Studi Teknologi Multimedia dan Jaringan, Jurusan Teknik Informatika dan Komputer, Politeknik Negeri Jakarta. Selama menempuh pendidikan, penulis memiliki minat pada bidang jaringan komputer dan keamanan siber.