E-ISSN 2963-9174

Indonesian Journal of Computer Science Research

Journal page: https://subset.id/index.php/IJCSR

Implementasi NFtables dan Fail2ban pada Linux Server untuk Pengembangan Adaptive Defense Firewall

Implementation of NFtables and Fail2ban on Linux Server for Adaptive Defense Firewall Development

Prasetyo Purnomo[1, a)*], Juarisman[2, b)], Irma Suwarning Dyastuti[3, c)] Eko Tri Anggono[4,d)]

[1,2,3)]Program Studi Informatika, Fakultas Ilmu Komputer [4)] Program Studi Manajemen Informatika, Fakultas Ilmu Komputer Universitas Amikom, Jl. Ringroad Utara, Sleman, Yogyakarta, Indonesia

[a)] *Corresponding author:* Erna Daniati. Email: *prasetyo.p@amikom.ac.id*
Received: 1 Agustus 2026; Accepted: 10 Agustus 2026

K E Y W O R D S

Kata kunci:
Adaptive defense
Fail2ban
Firewall
NFtables
Linux server

Keywords:
Adaptive defense
Fail2ban
Firewall
NFtables
Linux server

A B S T R A K

Abstrak. Manajemen keamanan server Linux saat ini menghadapi tantangan berupa kompleksitas konfigurasi baris perintah dan respons yang lambat terhadap serangan otomatis. Penelitian ini bertujuan untuk mengembangkan prototipe aplikasi web *Adaptive Defense Firewall* yang mengintegrasikan *firewalld, nftables,* dan *fail2ban* secara terpusat. Pentingnya penelitian ini terletak pada penyediaan manajemen visual yang memudahkan administrator dalam memitigasi intrusi jaringan secara *real-time*. Kebaruan yang ditawarkan adalah fitur *Adaptive Defense Levels* (profil *Safe, Moderate, Aggressive, Under Attack*) yang mampu menyesuaikan parameter mitigasi *fail2ban* dan pemblokiran tingkat kernel oleh *nftables* secara dinamis. Metodologi menggunakan pendekatan *prototipe* yang mencakup analisis kebutuhan, perancangan arsitektur, implementasi React.js dan Python, serta pengujian penetrasi. *Dataset* pengujian dihasilkan dari simulasi serangan terkendali menggunakan program Hydra untuk *brute-force* pada layanan ssh dan ftp, serta Nmap untuk *port scanning*. Hasil penelitian menunjukkan bahwa sistem efektif mendeteksi dan memblokir seluruh alamat IP penyerang. Waktu respons pemblokiran tercatat 1,030 detik untuk SSH, 0,301 detik untuk FTP, serta 0,492 detik untuk *port scanning* dengan rata-rata kenaikan resource CPU mencapai 2,8%–6,1% dan resource RAM mencapai 49,5%.

Abstract. *Linux server security management today faces the challenges of command-line configuration complexity and slow response to automated attacks. This research aims to develop a prototype of an Adaptive Defense Firewall web application that integrates firewalld, nftables, and fail2ban centrally. The importance of this research lies in the provision of visual management that makes it easier for administrators to mitigate network intrusions in real-time. The novelty offered is the Adaptive Defense Levels feature (Safe, Moderate, Aggressive, Under Attack profile), which is able to dynamically adjust fail2ban mitigation parameters and kernel-level blocking by nftables. The methodology uses a prototype approach that includes needs analysis, architectural design, React.js and Python implementation, and penetration testing. The test dataset was generated from a controlled attack simulation using the Hydra program for brute-force on SSH and ftp services, as well as Nmap for port scanning. The results of the study showed that the system effectively detected and blocked all the IP addresses of the attacker. The blocking response time was recorded at 1,030 seconds for SSH, 0.301 seconds for FTP, and 0.492 seconds for port*

scanning, with an average increase in CPU resources of 2.8%–6.1% and RAM resources of 49.5%.

1. Pendahuluan

Perkembangan infrastruktur Virtual Private Server (VPS) saat ini diiringi dengan peningkatan ancaman keamanan siber, khususnya serangan otomatis berbasis botnet seperti brute force dan port scanning [1]-[4]. Untuk memitigasi serangan ini, administrator umumnya mengandalkan pemfilteran paket melalui antarmuka baris perintah (CLI) yang memiliki kompleksitas konfigurasi tinggi dan rentan terhadap human error [5]. Sebagai alternatif, berbagai alat bantu berbasis Graphical User Interface (GUI) seperti Cockpit atau Webmin telah dikembangkan untuk menyederhanakan manajemen server. Namun, fungsionalitas manajemen firewall pada alat bantu eksisting tersebut masih sangat terbatas pada aturan allow/deny yang bersifat statis. Kondisi tersebut menunjukkan kebutuhan akan mekanisme respons yang dapat menyesuaikan parameter mitigasi berdasarkan bukti serangan pada log secara lebih cepat dan terukur [1], [6].

Untuk menutupi celah fungsionalitas tersebut, penelitian ini bertujuan untuk membangun sebuah aplikasi manajemen keamanan jaringan berbasis antarmuka web. Sistem ini dibuat untuk menggabungkan komponen keamanan Linux ke dalam satu ekosistem yang terpusat. Konsep ini menerapkan kemudahan manajemen zona yang ditawarkan oleh firewalld (sebagai lapis kebijakan) dengan kecepatan serta efisiensi pemfilteran paket tingkat dasar dari nftables (sebagai lapis eksekusi). Penggabungan dua teknologi ini bertujuan memberikan kontrol visual yang terpusat dan intuitif bagi administrator tanpa harus menurunkan performa sistem operasi.

Kebaruan utama dalam penelitian ini terletak pada integrasi sistem deteksi intrusi aktif yang dilengkapi dengan logika Adaptive Defense Levels. Berbeda dengan sistem firewall statis, fitur pertahanan adaptif ini mampu melakukan auto-scaling response dengan mengubah parameter keamanan, seperti batas maksimal percobaan (maxretry) dan durasi pemblokiran (bantime), dan secara otomatis dikategorikan ke dalam tiga tingkatan (Moderate, Aggressive, Under Attack) berdasarkan kondisi ancaman seketika [7]. Integrasi deteksi berbasis log, pemblokiran IP otomatis, dan visualisasi insiden terpusat diharapkan meningkatkan kecepatan respons administrator terhadap ancaman berulang [8], [9].

2. Tinjauan Pustaka

Pemanfaatan *Fail2Ban* sebagai *Intrusion Prevention System* (IPS) telah secara empiris terbukti tangguh dalam memitigasi serangan jaringan yang bersifat otomatis. Penelitian oleh Ridho, dkk. (2025) [10] menunjukkan bahwa parameter *maxretry* dan *bantime* pada f*ail2ban* secara signifikan mampu mendeteksi pola percobaan akses ilegal dan memblokir IP penyerang pada kasus serangan *brute force*. Pada layanan ssh, parameter *maxretry*, *findtime*, dan *bantime* memungkinkan sistem mengenali kegagalan autentikasi berulang lalu memblokir sumber serangan [11]. Pengujian keamanan ssh dengan Hydra juga memperlihatkan bahwa pembatasan percobaan autentikasi diperlukan untuk menurunkan peluang keberhasilan serangan *brute force* [12], [13], [14].

Pada lapisan pemfilteran paket, aturan firewall diperlukan untuk membatasi lalu lintas tidak sah serta mengurangi dampak serangan *DoS/DDoS* dan *port scanning* [15]–[17]. Nftables menyediakan kerangka pemfilteran paket Linux yang dapat digunakan untuk menerapkan aturan, *set*, dan aksi pemblokiran pada tingkat kernel. Efisiensi pada lapisan kernel ini merupakan faktor yang sangat krusial ketika server mengalami serangan yang berulang, sebagaimana penelitian yang dilakukan oleh Utomo, dkk. (2024) [11] dalam analisis implementasi keamanan jaringannya. Penelitian terdahulu juga menunjukkan bahwa analisis log dan karakteristik aliran jaringan dapat dipakai untuk membedakan perilaku autentikasi normal dan serangan *brute force*.

Walaupun penelitian terdahulu membuktikan efektivitas fail2ban, IDS/IPS, dan firewall dalam menangani serangan umum, sebagian besar menerapkan parameter mitigasi secara tetap atau hanya menguji satu komponen keamanan [18]. Perbedaan mendasar antara penelitian ini dengan studi-studi sebelumnya terletak pada penerapan arsitektur *iptables* dan *nftables* dalam bentuk aplikasi web serta integrasi *fail2ban* dengan fitur *Adaptive Defense Levels*. Pada penelitian terdahulu, parameter pemblokiran seperti *maxretry* dan *bantime* masih dikonfigurasi secara statis dan manual oleh administrator. Sebaliknya, penelitian ini merancang sistem pertahanan yang bersifat *auto-scaling*, di mana profil keamanan dapat disesuaikan (dari mode *Moderate* hingga *Under Attack*) untuk merespon intensitas serangan secara *real-time*. Selain itu, manajemen *firewalld, nftables* dan *fail2ban* yang disajikan melalui satu dashboard terpusat dan dilengkapi dengan data log berupa informasi detail serangan memberikan inovasi yang cukup lengkap sebagai salah satu alat pengambilan keputusan bagi seorang system administrator.

3. Metode Penelitian

Metode penelitian yang digunakan dalam pengembangan sistem ini mengadopsi model *Prototyping*. Model ini dipilih untuk memfasilitasi integrasi dan pengujian modul keamanan secara bertahap. Fokus utama pada metodologi ini adalah penerapan arsitektur Adaptive Defense Firewall yang dikendalikan melalui sebuah aplikasi antarmuka web secara terpusat.

3.1. Arsitektur Sistem

Sistem pada penelitian ini dirancang menggunakan arsitektur terpisah yang terdiri dari antarmuka pengguna, logika pemrograman antarmuka aplikasi (API), dan utilitas sistem operasi pada tingkat kernel. Pemisahan tugas ini secara spesifik diimplementasikan ke dalam tiga lapisan operasional utama yang saling terintegrasi untuk menjamin efisiensi mitigasi ancaman. Lapisan pertama adalah *policy layer (firewalld)*, Fungsinya untuk mengelola kebijakan jaringan secara statis dan persisten, serta menangani manajemen *zones*, pengaturan *services*, dan *port* untuk kebutuhan kontrol *firewall*. Lapisan kedua adalah *Execution Layer (nftables),* Fungsinya untuk bertindak sebagai mesin penyaring paket utama di tingkat kernel yang memantau status koneksi dan mengeksekusi aksi pemblokiran dengan performa tinggi. Sedangkan lapisan yang terakhir adalah *Intelligence Layer (fail2Ban),* fungsinya untuk memantau berkas log autentikasi seperti ssh dan ftp serta log aplikasi untuk mengenali pola kegagalan akses yang berulang, kemudian menjalankan aksi pemblokiran terhadap alamat IP penyerang

3.2. Mekanisme Kontrol Adaptive Defense Firewall

Mekanisme pertahanan pada sistem ini dibangun di atas konsep *Adaptive Defense*, yang memungkinkan untuk mengatur profil keamanan secara dinamis berdasarkan pola dan intensitas serangan. Mekanisme kontrol ini khusus untuk menangani serangan di level Layer 4 (Transport Level), sesuai dengan kategori OSI 7 layer. Ancaman seperti port scanning, ssh dan ftp bruteforce, serangan synflood, dan serangan DDoS tingkat jaringan dideteksi menggunakan kombinasi *nftables* dan *conntrack*. Pemilihan skenario tersebut didasarkan pada pola ancaman umum terhadap layanan jarak jauh dan jaringan yang telah diuji dalam penelitian sebelumnya [19], [20].

Tabel 1.
Parameter kontrol port scanning

Level Defense	Jail	maxretry	findtime	bantime	Filter	Sumber Log
Safe	safe	100	600 s	3600 s	scanners	app.log
Moderate	moderate	5	10 s	86400 s	scanners	app.log
Aggressive	aggressive	3	5 s	86400 s	scanners	app.log
Under Attack	under-attack	3	30 s	3600 s	scanners	app.log

Dari Tabel 1. di atas terlihat bahwa sistem ini memiliki 3 defense level, yaitu Moderate, Aggressive dan Under Attack serta 1 defense level default yaitu Safe, dengan masing-masing nilai parameter yang berbeda sesuai dengan tingkat keamanan yang diatur oleh system administrator

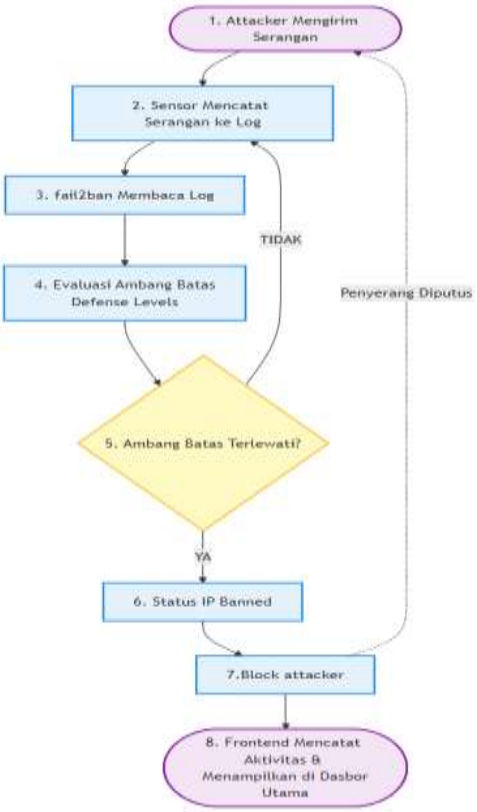
Tabel 2.
Parameter kontrol bruteforce

Service	Port	Jail	maxretry	findtime	bantime	Filter	Sumber Log
sshd	22	sshd	5	600 s	3600 s	sshd	journald
proftpd	21	proftpd	5	600 s	3600 s	proftpd	journald

Tabel 2. di atas terlihat bahwa aturan jail untuk ssh dan ftp memiliki nilai parameter yang sama.

3.3. Alur Kerja Penanganan Insiden

Guna memvalidasi efektivitas sistem deteksi dan respons, penelitian ini memetakan alur kerja penanganan insiden menggunakan pendekatan *Data Flow Diagram*. Alur pemrosesan data dimulai dari paket serangan masuk hingga alamat IP penyerang diblokir dan ditampilkan pada antarmuka web. Proses pertama adalah ketika sistem mendeteksi percobaan autentikasi layanan SSH yang gagal secara berulang akibat penggunaan kredensial yang salah oleh penyerang. Kegagalan autentikasi ini dicatat oleh *daemon* ssh ke dalam berkas log sistem operasi, seperti /var/log/auth.log atau /var/log/secure. Modul *fail2ban* kemudian membaca dan menyaring entri log tersebut untuk mendapatkan alamat IP penyerang. Mekanisme serupa juga diterapkan pada ancaman *port scanning*, di mana *nftables* mencatat paket SYN ke dalam berkas app.log, yang kemudian akan dibaca IP penyerangnya oleh modul fail2ban . Proses kedua adalah evaluasi ambang batas, yang berfungsi menghitung akumulasi kegagalan dari setiap alamat IP yang berhasil dikumpulkan. Jumlah ini dievaluasi oleh fungsi *threshold checker* yang terhubung langsung dengan aturan tingkat pertahanan *Adaptive Defense Firewall*. Apabila akumulasi kegagalan melampaui batas toleransi mode yang sedang aktif, sebagai contoh, 3 kali kegagalan pada mode *Under Attack* atau 5 kali pada mode *Moderat,* maka sistem akan secara otomatis memicu instruksi pemblokiran. Proses ketiga sekaligus terakhir adalah eksekusi pemblokiran alamat IP, yang dijalankan secara bersamaan oleh *nftables* dan *fail2ban*. Pada tahap ini, *nftables* mengeksekusi penolakan seluruh paket data dari IP yang dianggap berbahaya tersebut di tingkat jaringan, sementara *fail2ban* memasukkan alamat IP terkait ke dalam basis data agar riwayat pemblokiran dapat dicatat dan ditampilkan pada antarmuka dasbor pemantauan. Detail alur kerja penanganan insiden dapat dilihat pada Gambar 1 di bawah ini.



Gambar 1. Gambar Alur Kerja Penanganan Insiden

Gambar 1. di atas terlihat bahwa penanganan insiden serangan dimulai dari percobaan serangan oleh attacker berupa *brute force* dan *port scanning*. Sistem akan mencatat setiap serangan dan membandingkan dengan parameter yang sudah ditentukan. Jika melewati ambang batas, maka IP penyerang tersebut akan kena *banned* dan semua log akan dicatat oleh frontend untuk ditampilkan di dashboard utama

3.4. Alur Kerja Pencatatan Insiden dan Pemantauan Log

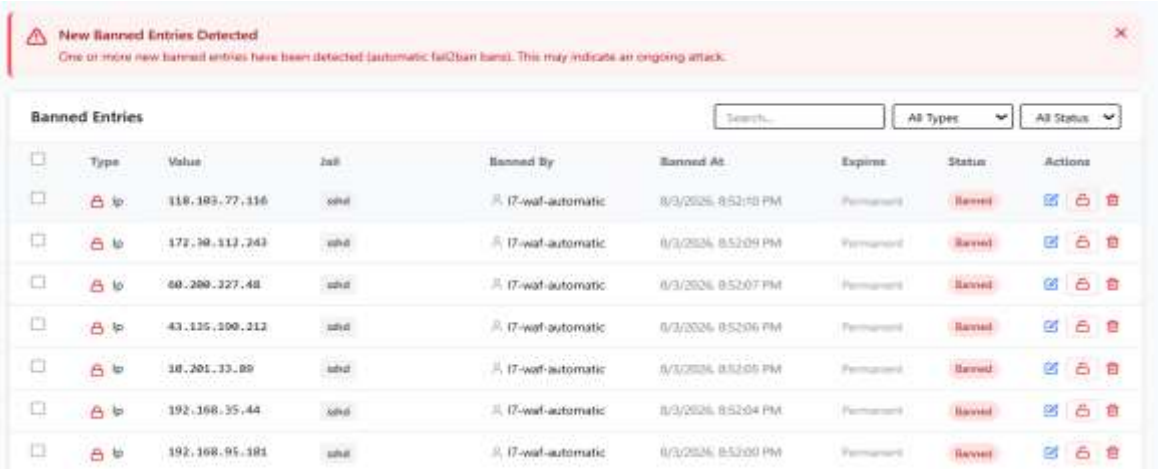
Alur kerja pencatatan insiden dirancang untuk memproses dan menyajikan isi dari berkas log secara terstruktur melalui beberapa tahapan utama. Tahapan pertama adalah pengumpulan berkas log, di mana sistem secara aktif menghimpun data dari berbagai sumber di dalam sistem operasi Linux. Sumber-sumber pencatatan tersebut meliputi journalctl (systemd) yang bertugas mencatat log metrik tingkat sistem dan aktivitas daemon, serta direktori /var/log/fail2ban/ dan /var/log/secure yang berisi aktivitas percobaan autentikasi. Selain itu, sistem juga mengumpulkan data dari /var/log/firewall-web/ untuk memantau log spesifik dari operasional backend aplikasi Adaptive Defense Firewall, serta log yang berhubungan dengan layanan spesifik, seperti /var/log/ftp, untuk mencatat aktivitas pada server FTP. Setelah tahap pengumpulan selesai, tahapan selanjutnya berfokus pada pemrosesan dan ekstraksi data. Seluruh data mentah yang dikumpulkan dari sumber-sumber tersebut kemudian diproses lebih lanjut oleh backend log manager. Dalam proses ini, sistem menggunakan teknik parsing berbasis regular expression untuk mengubah log mentah menjadi objek data yang terstruktur.

4. Hasil dan Pembahasan

Bab ini menjelaskan hasil implementasi sistem Adaptive Defense Firewall dan pengujian efektivitas mitigasinya terhadap insiden serangan siber berupa *brute force* dan port scanning.

4.1. Implementasi Web Antarmuka berbasis React.js dan Python

Sistem berhasil diimplementasikan dengan antarmuka berbasis React Js yang terhubung ke *backend* Python Flask. Penulis memilih React Js sebagai frontend karena mempunyai fitur yang sangat lengkap untuk membangun sebuah dashboard monitoring. Sedangkan pemilihan Python didasarkan pada kemampuan bahasa pemrograman ini untuk berkomunikasi dengan perintah-perintah program di level sistem operasi Linux melalui subproses.



Gambar 2. Gambar dashboard utama

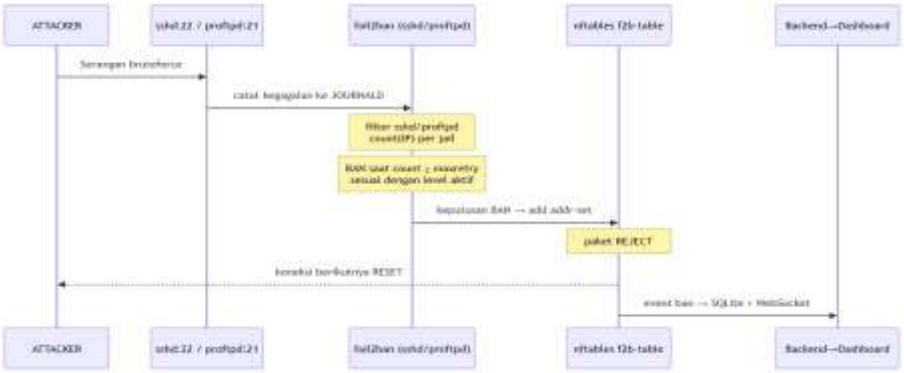
Gambar 2 di atas menampilkan dashboard utama web aplikasi Adaptive Defense Firewall. Dashboard ini digunakan untuk melakukan monitoring secara terpusat atas semua insiden keamanan yang terjadi. Web aplikasi ini juga dilengkapi dengan log yang menunjukkan aktivitas serangan yang terjadi, seperti tampak pada gambar di bawah ini.

Gambar 3. Halaman log

Gambar 3 menunjukkan dashboard utama pada web aplikasi *Adaptive Defense Firewall* yang digunakan untuk melakukan monitoring keamanan server secara terpusat. Dashboard menyajikan informasi mengenai insiden keamanan yang terdeteksi, jumlah penyerang unik, alamat IP yang sedang diblokir, serta negara asal serangan. Selain itu, dashboard menyediakan daftar *top attackers* berdasarkan jumlah aktivitas serangan yang teridentifikasi. Informasi tersebut memberikan gambaran kondisi keamanan server secara *real-time* dan membantu administrator dalam memantau serta merespons aktivitas yang berpotensi mengancam keamanan sistem.

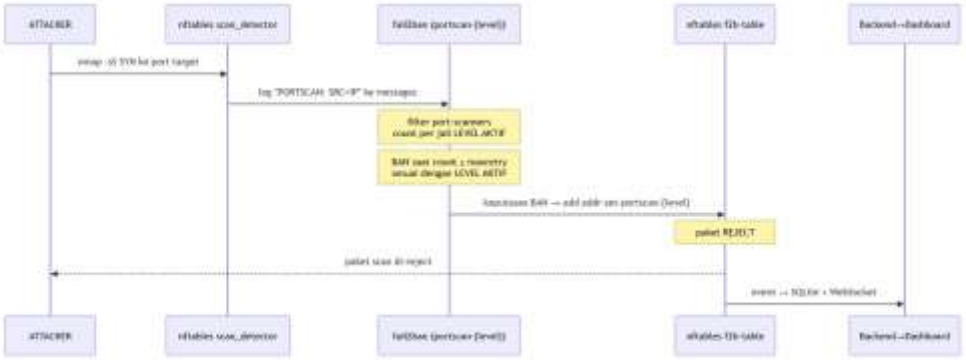
4.2. Pengujian Deteksi Serangan Berbasis Log dan Pemblokiran pada Lapisan Jaringan

Untuk memvalidasi keandalan mekanisme Adaptive Defense Firewall, pengujian dilakukan dengan mengeksekusi penetrasi *brute force* menggunakan alat otomatis *Hydra* dan Nmap dari alamat IP penyerang 192.168.55.19. Pengujian dilakukan dengan mensimulasikan serangan *brute force pada dua layanan krusial, yaitu Secure Shell (SSH) dan File Transfer Protocol (FTP), untuk mengukur mekanisme defense layer 4* yang sudah dikonfigurasi oleh sistem. Sistem diuji secara empiris menggunakan simulasi serangan yang terkontrol. Bruteforce tool seperti *Hydra* digunakan untuk mensimulasikan serangan *bruteforce* pada layanan SSH, sementara *Masscan* dan *nmap* digunakan untuk mensimulasikan pemindaian port. Pengujian difokuskan pada dua metrik utama, yaitu waktu yang dibutuhkan sistem sejak serangan dimulai hingga alamat IP terblokir di level kernel dan stabilitas penggunaan sumber daya server saat memproses volume log yang tinggi secara *real time*.



Gambar 4. Alur serangan bruteforce

Gambar 4 di atas menunjukkan bahwa jika terjadi serangan jenis brute force ke service ssh dan ftp, akan ditangani oleh fail2ban. Sementara nftables digunakan untuk memblokir IP penyerang dengan cara menolak paketnya. Begitu juga dengan jenis serangan port *scanning*. Jika ada indikasi dari tools Nmap atau Masscan, akan ditangani oleh fail2ban dan segera diproses oleh nftables untuk memblokir IP penyerang dengan cara menolak paketnya. Semua proses itu digambarkan secara jelas pada Gambar 5 di bawah ini.



Gambar 5. Alur serangan *port scanning*

Gambar 5 menunjukkan alur proses kerja Adaptive Defense Firewall dalam mendeteksi dan merespons aktivitas pemindaian port pada server Linux. Proses diawali ketika *attacker* melakukan pemindaian port menggunakan Nmap, kemudian aktivitas tersebut dideteksi oleh komponen nftables_scan_detector dan dicatat sebagai pesan log *PORTSCAN*. Log selanjutnya diproses oleh Fail2ban melalui mekanisme *filter port scan* untuk menghitung jumlah kejadian berdasarkan alamat IP sumber. Apabila jumlah kejadian memenuhi batas (*threshold*) yang telah ditentukan, Fail2ban melakukan tindakan *ban* dengan menambahkan alamat IP penyerang ke dalam nftables f2b-table. Selanjutnya, paket dari alamat IP tersebut ditolak (*REJECT*), sehingga aktivitas serangan tidak dapat dilanjutkan. Informasi mengenai kejadian pemblokiran kemudian dikirim ke Backend/Dashboard melalui SQLite dan WebSocket untuk ditampilkan pada dashboard monitoring.

4.2.1. Skenario 1: Serangan brute force pada layanan SSH

Serangan pertama ditargetkan pada layanan ssh (port 22). Gambar di bawah ini menunjukkan tabel serangan ssh dari IP penyerang 192.168.55.19 ke target serangan IP 192.168.55.191. Hasil pengujian dirangkum pada gambar di bawah ini

Skenario 1 — Serangan Brute-Force SSH

IP Penyerang	192.168.55.19
IP Target	192.168.55.191
Layanan / Port	SSH (Port 22)
Alat serangan	Hydra (-t 4, wordlist 40)
Ambang ban (maxretry)	5 kegagalan
Waktu deteksi (Found)	11:09:11.569
Waktu blokir (Ban)	11:09:12.599
Latensi deteksi → ban	1.030 detik
Eksekusi blokir	nftables (f2b-chain, priority -1) — reject
Durasi blokir (bantime)	1 jam
CPU saat serangan	avg 3.87% / peak 35.68%
RAM saat serangan	avg 48.6% / peak 49.48%

Gambar 6. Serangan bruteforce ssh

Berdasarkan hasil pengujian sesuai dengan gambar di atas, penyerang tercatat melakukan lima kali percobaan autentikasi yang gagal secara berulang, di mana akumulasi tersebut telah memenuhi ambang batas maxretry yang ditetapkan sistem, yakni sebanyak 5 kali. Menanggapi ancaman tersebut, modul fail2ban pada aturan jail sshd berhasil mendeteksi pola kegagalan autentikasi dan memicu pemblokiran. Waktu deteksi tersebut tercatat pada 11:09:11.569, sedangkan eksekusi pemblokiran dilakukan pada 11:09:12.599, sehingga sistem mencatatkan selisih waktu respons 1,030 detik. Pada tahap eksekusi blokir, sistem memasukkan alamat IP 192.168.55.19 ke dalam tabel nftables untuk menolak (reject) seluruh paket dari IP tersebut dengan durasi satu jam. Setelah pemblokiran aktif, seluruh upaya koneksi ssh berikutnya dari alamat IP penyerang tersebut langsung ditolak pada tingkat kernel.

4.2.2. Skenario 2: Serangan Brute-Force pada Layanan FTP

Serangan kedua ditargetkan pada layanan ftp (port 21). Gambar di bawah ini menunjukkan tabel serangan ftp dari IP penyerang 192.168.55.19 ke target serangan IP 192.168.55.191. Hasil pengujian dirangkum pada gambar di bawah ini.

Skenario 2 — Serangan Brute-Force FTP

IP Penyerang	192.168.55.19
IP Target	192.168.55.191
Layanan / Port	FTP (Port 21)
Alat serangan	Hydra (-t 4, wordlist 40)
Ambang ban (maxretry)	5 kegagalan
Waktu deteksi (Found)	11:28:19.186
Waktu blokir (Ban)	11:28:19.487
Latensi deteksi → ban	0.301 detik
Eksekusi blokir	nftables (f2b-chain, priority -1) — reject
Durasi blokir (bantime)	1 jam
CPU saat serangan	avg 3.27% / peak 28.5%
RAM saat serangan	avg 48.13% / peak 49.03%

Gambar 7. Serangan bruteforce ftp

Berdasarkan hasil pengujian sesuai dengan gambar di atas, penyerang tercatat melakukan lima kali percobaan autentikasi yang gagal secara berulang, di mana akumulasi tersebut telah memenuhi ambang batas maxretry yang ditetapkan sistem, yakni sebanyak 5 kali. Menanggapi ancaman tersebut, modul fail2ban pada aturan jail ftpd berhasil mendeteksi pola kegagalan autentikasi dan memicu pemblokiran. Waktu deteksi tersebut tercatat pada 11:28:19.487, sedangkan eksekusi pemblokiran dilakukan pada 11:28:19.487, sehingga sistem mencatatkan selisih waktu respons 0,301 detik. Pada tahap eksekusi blokir, sistem memasukkan alamat IP 192.168.55.19 ke dalam tabel nftables untuk menolak (reject) seluruh paket dari IP tersebut dengan durasi satu jam. Setelah pemblokiran aktif, seluruh upaya koneksi ftp berikutnya dari alamat IP penyerang tersebut langsung ditolak pada tingkat kernel.

4.2.3. Skenario 3: Serangan Port Scanning menggunakan Nmap

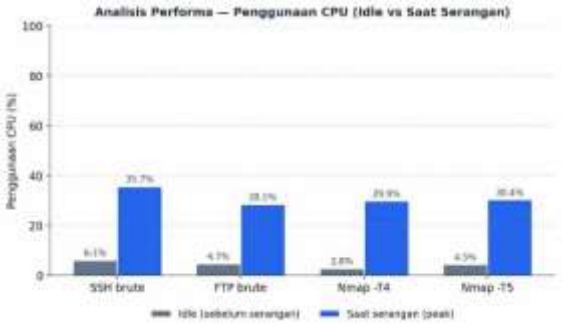
Serangan ketiga ini menguji ketahanan sistem terhadap upaya *network reconnaissance* dan *port scanning*. Gambar di bawah ini menunjukkan tabel serangan *port scanning* dari IP penyerang 192.168.55.19 ke target serangan IP 192.168.55.191

Gambar 8. Serangan port scanning

Berdasarkan hasil pengujian sesuai dengan gambar di atas, penyerang tercatat melakukan lima kali percobaan *portscanning* yang dilakukan secara berulang, di mana akumulasi tersebut telah memenuhi ambang batas *maxretry* yang ditetapkan sistem, yakni sebanyak 5 kali. Menanggapi ancaman tersebut, modul *fail2ban* pada aturan *jail portscan* berhasil mendeteksi pola *portscanning* dan memicu pemblokiran. Waktu deteksi tersebut tercatat pada 10:36:14.206, sedangkan eksekusi pemblokiran dilakukan pada 10:36:149.698, sehingga sistem mencatatkan selisih waktu respons 0.492 detik. Pada tahap eksekusi blokir, sistem memasukkan alamat IP 192.168.55.19 ke dalam tabel *nftables* untuk menolak (*reject*) seluruh paket dari IP tersebut dengan durasi satu jam. Setelah pemblokiran aktif, seluruh upaya *portscanning* berikutnya dari alamat IP penyerang tersebut langsung ditolak pada tingkat kernel

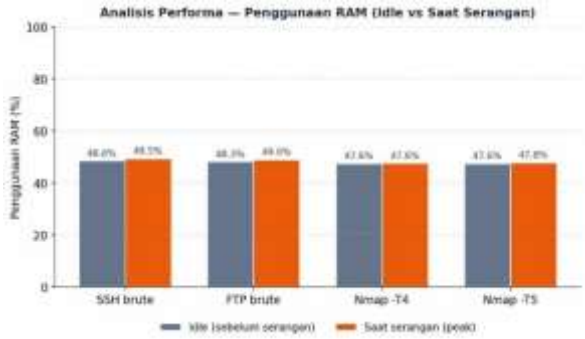
4.3. Análisis Performa Sistem

Keunggulan utama aplikasi Adaptive Defense Firewall terbukti dari performa sistem dalam menangani dua jenis serangan yang berbeda, yaitu brute force pada ssh dan ftp serta serangan port scanning. Gambar di bawah ini menunjukkan penggunaan resource sistem pada kondisi sebelum serangan dan saat serangan terjadi.



Gambar 9. Resource CPU

Gambar 9 di atas memperlihatkan perbandingan penggunaan CPU server dalam dua kondisi utama, yaitu saat sebelum serangan dan saat serangan berlangsung untuk ketiga skenario pengujian yang dilakukan. Pada kondisi normal atau *idle*, persentase CPU yang digunakan rata-rata hanya berkisar antara 2,8% hingga 6,1%.. Namun, ketika terjadi serangan, CPU mengalami lonjakan beban sesaat hingga menyentuh angka 28,5% hingga 35,7%. Lonjakan daya komputasi ini adalah akibat tingginya intensitas pemrosesan paket data dan pencatatan log yang dilakukan secara bersamaan oleh *daemon* layanan seperti *sshd* dan *proftpd*, modul *fail2ban*, serta *nftables.* Sedangkan penggunaa RAM terlihat pada gambar di bawah ini.



Gambar 10. Resource RAM

Gambar 10 di atas memperlihatkan perbandingan penggunaan RAM server dalam dua kondisi utama, yaitu saat sebelum serangan dan saat serangan berlangsung untuk ketiga skenario pengujian yang dilakukan. Pada kondisi normal atau *idle*, persentase RAM yang digunakan rata-rata hanya berkisar antara 47,6% hingga 48,6%. Namun, ketika terjadi serangan, RAM mengalami lonjakan beban sesaat hingga menyentuh angka paling tinggi 49,5%. Hal ini mengindikasikan bahwa seluruh tahapan pemrosesan deteksi serangan dan eksekusi pemblokiran yang dikoordinasikan oleh *fail2ban* beserta *nftables* membebani kapasitas RAM server. Dengan demikian, berdasarkan hasil pengujian kedua sumber daya tersebut, sistem *Adaptive Defense Firewall* terbukti ringan dan mampu memberikan respons pertahanan secara *real-time* tanpa mengorbankan stabilitas kinerja server.

5. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian, penelitian ini menyimpulkan bahwa pengembangan aplikasi *Adaptive Defensive Firewall* berbasis web berhasil mengatasi kendala konfigurasi berbasis perintah (*Command Line Interface*). Aplikasi ini menyediakan antarmuka visual yang intuitif untuk mengelola keamanan jaringan secara terpusat, dilengkapi dengan fitur pemantauan insiden *real-time* berbasis *websocket*, visualisasi serangan, serta pencatatan log secara otomatis. Fungsionalitas terpusat ini terbukti secara signifikan mampu mengurangi risiko kesalahan identifikasi ancaman oleh administrator sistem saat melakukan tugas pengawasan dan pengambilan keputusan operasional. Kesimpulan selanjutnya menyoroti keberhasilan integrasi antara perangkat *fail2ban* dan *nftables* pada *layer network* yang menghasilkan arsitektur pertahanan yang kuat dan efisien. Kolaborasi kedua komponen ini pada berbagai tingkatan memungkinkan sistem untuk memberikan respons mitigasi secara otomatis, di mana *fail2ban* bertugas mendeteksi anomali akses secara instan dan *nftables* mengeksekusi pemblokiran ancaman dengan toleransi waktu penanganan yang sangat singkat. Proses penolakan paket yang dilakukan secara langsung pada tingkat kernel ini juga berdampak sangat positif terhadap efisiensi sumber daya server, karena terbukti hanya membutuhkan konsumsi sumber daya sistem yang sangat kecil. Meskipun sistem telah beroperasi secara optimal pada tingkat jaringan, masih terdapat beberapa kekurangan yang direkomendasikan untuk disempurnakan pada penelitian selanjutnya. Penelitian mendatang disarankan untuk memperluas cakupan deteksi ancaman hingga ke *layer application* guna menangkal serangan seperti *SQL Injection* dan *Cross-Site Scripting*, yang dapat dicapai melalui integrasi *Web Application Firewall* (WAF) seperti ModSecurity dengan *fail2ban*. Selain itu, arsitektur *backend* juga perlu dikembangkan agar lebih kompatibel dengan berbagai distribusi Linux selain CentOS, serta disarankan untuk dikemas ke dalam bentuk instalasi berbasis Docker guna menjamin interoperabilitas dan mempermudah implementasi oleh lebih banyak administrator sistem.

Daftar Pustaka

[1] M. Nadeem, A. Arshad, S. Riaz, S. S. Band, and A. Mosavi, "Intercept the cloud network from brute force and DDoS attacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, pp. 152300–152309, 2021, doi: 10.1109/ACCESS.2021.3126535.

[2] J. Park, J. Kim, B. B. Gupta, and N. Park, "Network log-based SSH brute-force attack detection model," *Computers, Materials & Continua*, vol. 68, no. 1, pp. 887–901, 2021, doi: 10.32604/cmc.2021.015172.

[3] P. Khandait, N. Tiwari, and N. Hubballi, "Who is trying to compromise your SSH server? An analysis of authentication logs and detection of bruteforce attacks," in *Adjunct Proceedings of the 2021 International Conference on Distributed Computing and Networking*, 2021, pp. 127–132, doi: 10.1145/3427477.3429772.

[4] M. K. Putro, "Analisis komparatif kinerja komputasi brute force pada arsitektur GPU NVIDIA Turing, Ampere, dan Ada Lovelace," The Indonesian Journal of Computer Science Research, vol. 5, no. 1, pp. 79–85, Jan. 2026, doi: 10.59095/ijcsr.v5i1.253.

[5] T. O. Agboola, J. Adegede, and J. G. Jacob, "Balancing usability and security in secure system design: A comprehensive study on principles, implementation, and impact on usability," *International Journal of Computing Sciences Research*, vol. 8, pp. 2995–3009, 2024, doi: 10.25147/ijcsr.2017.001.1.199.

[6] F. Fachri, "Optimasi keamanan web server terhadap serangan brute-force menggunakan penetration testing," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 1, pp. 51–58, 2023, doi: 10.25126/jtiik.20231015872.

[7] A. D. Batistuta, A. H. Hendrawan, and Ritzkal, "Analisis keamanan jaringan server terhadap serangan dictionary menggunakan tools Fail2Ban dengan notifikasi Telegram," *INFOTECH Journal*, vol. 10, no. 1, pp. 64–73, 2024, doi: 10.31949/infotech.v10i1.8730.

[8] Farhannullah and M. Hardjianto, "Sistem monitoring serangan SSH dengan metode intrusion prevention system (IPS) Fail2ban menggunakan Python pada sistem operasi Linux," *Jurnal TICOM: Technology of Information and Communication*, vol. 11, no. 1, pp. 33–38, 2022, doi: 10.70309/ticom.v11i1.68.

[9] F. A. Ruambo, E. E. Masanga, B. Lufyagila, *et al.*, "Brute-force attack mitigation on remote access services via software-defined perimeter," *Scientific Reports*, vol. 15, Art. no. 18599, 2025, doi: 10.1038/s41598-025-01080-5.

[10] M. Ridho, A. Hafizh, I. Dani, and T. Ariyadi, "Peningkatan keamanan SSH server berbasis Linux melalui implementasi Fail2Ban dan uji serangan brute force," *Jurnal Penelitian Multidisiplin Bangsa*, vol. 1, no. 12, pp. 2206–2214, 2025

[11] B. R. Utomo, N. H. Jati, A. K. Jati, I. A. Saputro, and M. H. Purwidiantoro, "Analisis implementasi keamanan jaringan dengan Fail2ban terhadap serangan bruteforce," *Prosiding Seminar Nasional Amikom Surakarta*, vol. 2, pp. 1211–1223, 2024

[12] D. R. Az Zahra, F. P. Ilham, H. N. Ramdhani, and A. Setiawan, "Penerapan dan pengujian keamanan SSH pada server Linux menggunakan Hydra," *Journal of Internet and Software Engineering*, vol. 1, no. 3, 2024, doi: 10.47134/pjise.v1i3.2627.

[13] Y. Mulyanto, H. Herfandi, and R. C. Kirana, "Analisis keamanan wireless local area network terhadap serangan brute force dengan metode penetration testing," *Jurnal Informatika Teknologi dan Sains*, vol. 4, no. 1, pp. 26–35, 2022, doi: 10.51401/jinteks.v4i1.1528.

[14] J. Hancock, T. M. Khoshgoftaar, and J. L. Leevy, "Detecting SSH and FTP brute force attacks in big data," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 760–765, doi: 10.1109/ICMLA52953.2021.00126.

[15] I. Rahmadaniar, D. A. A. Tondang, B. S. Fernando, and A. Setiawan, "Implementasi firewall menggunakan iptables untuk melindungi server dari serangan DDoS," *Journal of Internet and Software Engineering*, vol. 1, no. 3, 2024, doi: 10.47134/pjise.v1i3.2564.

[16] N. Nuroji, "Penerapan intrusion detection and prevention system (IDPS) pada jaringan komputer sebagai pencegahan serangan port-scanning," *Journal of Data Science and Information Systems*, vol. 1, no. 2, pp. 41–49, 2023, doi: 10.58602/dimis.v1i2.35.

[17] M. R. H. Tambunan and S. N. Neyman, "Implementasi firewall pada Linux untuk pencegahan dari serangan DoS," *Journal of Technology and System Information*, vol. 1, no. 4, 2024, doi: 10.47134/jtsi.v1i4.2648.

[18] O. E. Taylor and P. S. Ezekiel, "Firewall defense and response policy towards resisting attacks on network logs," *International Journal of Computing Sciences Research*, vol. 8, pp. 2886–2904, 2024

[19] S. K. Wanjau, G. M. Wambugu, and G. N. Kamau, "SSH-brute force attack detection model based on deep learning," *International Journal of Computer Applications Technology and Research*, vol. 10, no. 1, pp. 42–50, 2021, doi: 10.7753/IJCATR1001.1008.

[20] N. Tiwari and N. Hubballi, "Secure socket shell bruteforce attack detection with Petri net modeling," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 697–710, 2023, doi: 10.1109/TNSM.2022.3212591.