final-journal-saputra.docx

by Tanner Gaskins

Submission date: 04-Jul-2025 09:36AM (UTC+0200)
Submission ID: 2710036171
File name: final-journal-saputra.docx (4.52M)
Word count: 4171
Character count: 26770

Implementasi Deteksi Serangan DoS Real-Time dan Mitigasi Otomatis pada Server VPS Berbasis C

Saputra Budianto [1*] , Hamzah Setiawan [2] , M.Alfan Rosyid [3]
[123] Universitas Muhammadiyah Sidoarjo, Jurusan Informatika, Sidoarjo, Jawa Timur, Indonesia
saputrabudianto23@gmail.com[1], hamzahsetiawan@umsida.ac.id[2],malfanrosyid@umsida.ac.id[3]

Abstrak

Di era digital, keamanan menjadi aspek penting dalam menjaga sistem IT, terutama pada Virtual Private Server (VPS) yang paling sering terpapar ancaman siber. Denial-of-Service (DoS) merupakan risiko yang dapat dikurangi dengan mengganti server dengan protokol yang lebih canggih seperti TCP, UDP, dan ICMP, serta menerapkan sistem yang mendeteksi dan mengurangi DoS secara real time dan otomatis menggunakan bahasa pemrograman C. Proses pengembangan sistem dengan menggunakan metodologi Agile Scrum memungkinkan proses yang iteratif, fleksibel, dan fleksibel. Sprint meliputi analisis antrean server dengan libpcap, manajemen log dengan SQLite, pemblokiran IP otomatis dengan iptables, dan pembaruan log melalui log sistem. Studi ini menunjukkan bahwa sistem dapat mendeteksi dan mengurangi HTTP Flood, ICMP Flood, dan Slowloris dalam waktu 0,5 detik dengan CPU dan memori yang rendah. Meskipun tidak ada integrasi visual real-time atau notifikasi real-time untuk manajemen, sistem ini efisien dan efektif dalam memproses data dengan cepat. Studi ini menyimpulkan bahwa penggunaan bahasa pemrograman C dalam pengembangan keamanan VPS sangat penting untuk mitigasi dan pemulihan yang cepat. Fase pengembangan meliputi deteksi tingkat aplikasi (lapisan 7), visualisasi dasbor, dan pembelajaran mesin untuk mengidentifikasi kerentanan dan kompromi dengan cepat.

Kata kunci— Serangan Siber, DoS, Agile Scrum, Linux, Server

Abstract

In the digital era, security is an important aspect in maintaining IT systems, especially on Virtual Private Servers (VPS) which are most often exposed to cyber threats. Denial-of-Service (DoS) is a risk that can be reduced by replacing servers with more sophisticated protocols such as TCP, UDP, and ICMP, and implementing a system that detects and reduces DoS in real time and automatically using the C programming language. The system development process using the Agile Scrum methodology allows for an iterative, flexible, and flexible process. Sprints include server queue analysis with libpcap, log management with SQLite, automatic IP blocking with iptables, and log updates via system logs. This study shows that the system can detect and reduce HTTP Flood, ICMP Flood, and Slowloris in 0.5 seconds with low CPU and memory. Although there is no real-time visual integration or real-time notification for management, the system is efficient and effective in processing data quickly. This study concludes that the use of the C programming language in VPS security development is essential for rapid mitigation and recovery. The development phase includes application-level (layer 7) detection, dashboard visualization, and machine learning to quickly identify vulnerabilities and compromises

.

Keywords— Cyberattack, DoS, Agile Scrum, Linux, Server

1. PENDAHULUAN

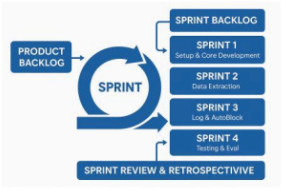
Di era dunia serba dengan kemudahan untuk berkomunikasi satu sama lain, menciptakan peluang bagi para pelaku kejahatan untuk melakukan tindakan ilegal. Berbagai port terbuka pada setiap perangkat yang terhubung dengan jaringan internet terekspos sehingga para pelaku kejahatan siber dapat melakukan kejahatan melalui celah-celah pada port yang terbuka tersebut. Salah satunya adalah teknik serangan DoS (*Denial of Service)*[1]. Pada dasarnya, serangan DoS dilakukan oleh penyerang untuk melumpuhkan sistem target dengan menghabiskan sumber daya yang dimiliki oleh target [2].

Serangan DoS merupakan serangan yang fatal dan sering digunakan oleh para pelaku kejahatan dunia maya, dimana serangan tersebut cukup efektif untuk mengganggu lalu lintas jaringan pada komputer server yang menjadi target, sehingga pengguna tidak dapat mengakses komputer server tersebut. Serangan DoS yang menargetkan DNS (*Domain Name Server*) yang merupakan peran krusial, karena merupakan infrastruktur pendukung berbagai aplikasi, platform pendistribusian konten, dan layanan keamanan sistem. Serangan ini bekerja dengan cara di mana penyerang atau penyerang melakukan permintaan terus [3]-menerus ke server atau disebut juga serangan banjir. Serangan ini dilakukan melalui protokol TCP (*Transmission Control Protocol*), UDP (*User Datagram Protocol*), dan ICMP (*Internet Control Message Protocol*) [4].

Serangan DoS dapat diklasifikasikan berdasarkan lapisan-lapisan dalam model OSI (*Open System Interconnection*). Model OSI merupakan suatu kerangka konseptual yang membagi fungsi komunikasi jaringan ke dalam tujuh lapisan, meliputi Lapisan 1 *(Physical Layer*), seperti kabel dan perangkat keras lainnya. Lapisan 2 (*Data Link*), lapisan yang menangani pengiriman data antar perangkat yang berdekatan. Lapisan 3 (*Network*), lapisan yang menangani pengalamatan dan perutean alamat IP. Lapisan 4 (*Transport*), lapisan yang menangani pengiriman data antar aplikasi. Lapisan 5 (*Session*), lapisan yang mengatur sesi-sesi komunikasi. Lapisan 6 (*Presentation*), lapisan yang menangani pemformatan dan enkripsi data. Lapisan 7 (*Application*), lapisan yang menyediakan jaringan bagi aplikasi pengguna. Misalnya, serangan pada lapisan Application akan lebih sulit dideteksi, serangan pada lapisan Application bekerja dengan cara membanjiri banyak permintaan, seolah-olah banyak pengguna yang mengakses, sehingga sulit dihentikan. Contoh kasus seperti serangan yang menggunakan teknik *HTTP Flood seperti DoS Attack Tool Slowloris* dan *Slow HTTP Test* [5].

Mitigasi atau penanggulangan yang dapat dilakukan untuk mencegah serangan DoS salah satunya dengan melakukan pemantauan lalu lintas jaringan pada web server yang tujuan utamanya adalah untuk mendeteksi adanya aktivitas yang mencurigakan pada Web Server. Untuk pengembangan sistem pertahanan server dari serangan DoS lebih cocok menggunakan bahasa pemrograman C yang mempertimbangkan faktor efektivitas dan kecepatan. Dalam hal ini bahasa pemrograman C memiliki kelebihan dalam hal pengendalian perangkat keras komputer yang sangat dibutuhkan untuk proses pengembangan alat pertahanan dari serangan DoS [6].

2. METODE PENELITIAN



Gambar 1 Metode Agile Scrum

Scrum merupakan metodologi yang digunakan dalam pengembangan sistem, yang diadaptasi dari metode agile, yang digunakan untuk mengelola dan mengendalikan proses perubahan sistem. Dalam penelitian ini, Scrum digunakan untuk memitigasi dan mendeteksi serangan DoS pada server VPS secara real time dan otomatis menggunakan bahasa C[7]. Metodologi Scrum membantu mengembangkan persyaratan sistem secara kompleks, agile, dan tepat waktu [8].

Hasil penelitian ini dikelola melalui beberapa sprint dengan eksekusi berurutan dan berkelanjutan pada deteksi data abnormal, pelacakan IP otomatis, manajemen log, dan notifikasi real-time [9]. Prioritas fitur ditetapkan dan dieksekusi, dan evaluasi dilakukan melalui Sprint Review dan Retrospective untuk memastikan sistem dapat beradaptasi dengan risiko baru atau kebutuhan mitigasi baru [10].

Pendekatan Scrum dalam proyek ini meliputi *User Story*, *Product Backlog*, *Sprint*, *Sprint Backlog*, dan *Sprint Review & Retrospective* [11]. *User Story* menjelaskan persyaratan sistem pengguna, *Product Backlog* menjelaskan fitur yang dibutuhkan, *Sprint* menjelaskan timeline untuk setiap fitur, *Sprint Backlog* menjelaskan *Product Backlog*, dan Sprint *Review & Retrospective* mengalir setiap *Sprint* untuk mengidentifikasi area yang membutuhkan [12]. Dengan Scrum sebagai kerangka kerja, peneliti dapat mengaktifkan, secara fleksibel, dan mengarahkan proses pengembangan sistem tanpa batasan waktu, sambil memastikan bahwa sistem memenuhi tingkat risiko DoS saat ini [13].

3. PEMBAHASAN

Dalam mengembangkan penelitian ini metodologi Agile Scrum diadopsi karena ketersediaannya di lapangan. Pendekatan ini harus dipilih untuk memastikan fleksibilitas anggota sambil mempertahankan pendekatan iteratif dan inkremental. Pemrosesan sistem diperlukan untuk mendukung proses rantai pasokan secara real-time dan efisien.

3.1 Struktur Tim untuk Peran Scrum

Gambar 3. Diagram Proses

Implementasi Scrum melibatkan tiga peran utama: Product Owner, Scrum Master, dan Development. Pemahaman dari product owner adalah tindakan yang jelas, eksekusi working memory, dan perbaikan proses. Scrum Master bertanggung jawab untuk memastikan bahwa semua hal dalam Scrum dieksekusi sesuai dengan prinsip-prinsip sehingga kita bekerja menuju tujuan kita. Peneliti pengembangan memberikan jawaban tentang sistem logika, pemrograman C, dan pengujian sistem.

3.2 Backlog Produk dan Sprint yang Direncanakan

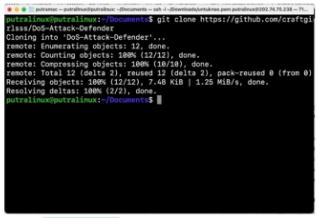
Gambar 4. Diagram Alir Program

Peneliti memiliki produk seperti membuat paket data menggunakan libpcap, membuat paket data untuk IP, port, dan protokol, mengembangkan sistem validasi berbasis batas paket, mengakses SQLite, mengimplementasikan firewall dengan iptables, dan meningkatkan log tingkat

sistem. Penundaan ini menyebabkan beberapa sprint dengan banyak tantangan.

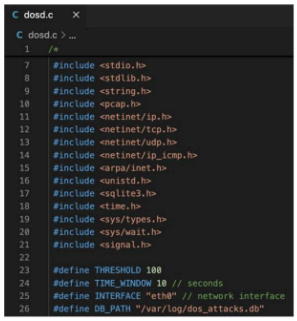
3.3 Sprint 1 – Pengaturan Awal dan Pengembangan Fitur Inti

Sebelum membahas fungsi dan kegunaan dari kode yang telah dibuat oleh penulis, Langkah awal clone terlebih dahulu repository penulis yang telah dipublikasikan ke platform Github. dengan cara membuka terminal. Berikut merupakan url github Tool DoS Attack Defender yang telah dikembangkan penulis. Ikuti perintah berikut untuk dapat mengkloning sebuah repository github, pastikan sudah install github terlebih dahulu di komputer masing masing. Perintahnya sebagai berikut git clone https://github.com/craftgirlsss/DoS-Attack-Defender. Jika berhasil maka akan muncul gambar sebagai berikut.



Gambar 5. Cloning Repository

Pada pengembangan tool DoS Defender, pada file dosd.c, penulis menggunakan Bahasa pemorgraman C, yang mana memiliki akses khusus pada suatu kernel OS, yang mana lebih efektif untuk pengembangan tool tersebut.



Gambar 6. Setup Package C

Pada gambar 5 terlihat bahwasanya pada kode tool tersebut membutuhkan sebuah library PCAP, SQLite3, dan lain lain yang dibutuhkan untuk mentracking request yang masuk. Yang kedua penulis mendefinisikan variable Threshold sebanyak 100 yang dimana untuk membatasi request per ms yang ditargetkan kepada sebuah IP Address yang mengakses server pada port 80. Lalu terdapat sebuah definisi eth0 yang mana merujuk pada sebuah network di kernel OS, terutama kernel Linux dan macOS.

Gambar 7. Struktur Data untuk Pelacakan IP Address

Pada gambar 6, terlihat sebuah struct baru bernama IPLog. Struktur ini digunakan untuk menyimpan informasi terkait setiap alamat IP yang sedang dilacak oleh system. Lalu terdapat sebuah deklarasi variabel-variabel global yang akan digunakan di seluruh program berupa #define MAX_IPS 1024 yang difungsikan sebagai jumlah maksimum IP yang bisa dilacak secara bersamaan. Program ini hanya bisa melacak hingga

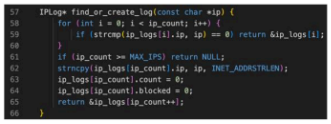
1024 alamat IP yang berbeda. Selanjutnya terdapat IPLog ip_logs[MAX_IPS], yaitu Sebuah array global bertipe IPLog yang dapat menampung sebuah data hingga MAX_IPS (1024) entri IPLog. Ini adalah tempat di mana semua data pelacakan IP disimpan. Selanjutnya juga terdapat sqlite3 *db, yang berupa pointer global bertipe sqlite3 *. Pointer ini akan menyimpan "handle" ke database SQLite setelah dibuka oleh sqlite3_open di fungsi main. Karena ini global, semua fungsi lain (seperti log_to_sqlite) dapat mengakses database ini.

Gambar 8. Fungsi Log

Pada gambar 7 merupakan fungsi void log_to_sqlite(const char *ip, const char *protocol) adalah fungsi yang secara spesifik bertugas untuk mencatat atau merekam informasi mengenai aktivitas mencurigakan atau serangan yang terdeteksi ke dalam sebuah database SQLite. Tujuan Utama yaitu mencatat detail sebuah serangan (atau kejadian yang dianggap perlu dicatat) ke dalam database.

Gambar 9. Fungsi Blokir IP Address

Ini adalah fungsi yang sangat penting dalam konteks keamanan jaringan, terutama jika program ini bertujuan sebagai sistem pencegahan intrusi (Intrusion Prevention System / IPS) sederhana. Fungsi void block_ip(const char *ip) memiliki tujuan tunggal yaitu untuk memblokir alamat IP tertentu pada system.



Gambar 10. Fungsi Blokir IP Address

Pada gambar 9 terlihat sebuah Fungsi IPLog* find_or_create_log(const char *ip). Fungsi ini merupakan sebuah fungsi yang bertugas untuk mencari atau membuat entri log baru untuk sebuah alamat IP tertentu.

Gambar 11. Fungsi Analisis Paket

Secara sederhana, analyze_packet() ini merupakan "otak" dari sistem IDS/IPS sederhana ini. Ia bertanggung jawab untuk memeriksa setiap paket jaringan, melacak aktivitas setiap IP, mendeteksi pola serangan frekuensi tinggi, dan kemudian memblokir IP yang menjadi sumber serangan tersebut.

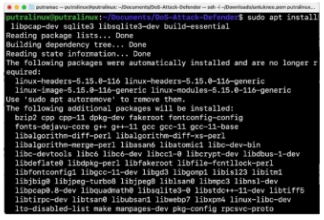
Gambar 12. Fungsi Paket Handler

Ini adalah bagian standart dari cara kerja program packet sniffer atau network analyzer yang menggunakan package libpcap. Secara singkat, packet_handler adalah gerbang masuk untuk setiap paket jaringan yang ditangkap. Tugas utamanya adalah menerima data mentah paket, melompati header layer-2 (Ethernet)

untuk menemukan awal header IP Address, dan meneruskan header IP tersebut ke fungsi analyze_packet untuk diproses lebih lanjut dan menentukan apakah ada ancaman yang perlu ditindak lanjuti.

Gambar 13. Fungsi Main

Yang terakhir adalah Fungsi int main() adalah titik awal eksekusi program. Ini adalah tempat di mana semua inisialisasi dilakukan dan loop utama program dimulai. Fungsi main ini adalah fungsi orkestrator yang dimana menginisialisasi semua komponen yang diperlukan (penangkap paket dan database), memulai proses penangkapan dan analisis paket secara berkelanjutan, dan kemudian melakukan pembersihan sumber daya saat program dihentikan. Ini adalah kerangka dasar untuk aplikasi pemantau jaringan dan sistem IPS sederhana ini.



Gambar 14. Setup Instalasi Package Linux

Fase ini mencakup menginstal dan mengonfigurasi lingkungan server Linux untuk pengumpulan package yang dibutuhkan oleh tool, untuk dapat menjalankan perintah libpcap untuk pengumpulan dan migrasi data, dan menjalankan Sprint Results.

Gambar 15. Kompilasi Kode C

Langkah ini merupakah Langkah umum untuk kompilasi suatu program C untuk dijadikan sebuah executable program untuk dapat dijalankan. Proses kompilasi menggunakan compiler Bernama GCC (GNU Compiler Collection) yaitu compiler yang sangat popular dan banyak digunakan untuk Bahasa pemorgraman C, C++, Objective-C, Fortran, Ada, Go dan D. Berikut perintahnya, gcc -o dosd dosd.c -lpcap -lsqlite3. Pada perintah tersebut terdapat sebuah tambahan perintah yang mana berupa -lpcap, dan -lsqlite3 yang mana merupakan perintah untuk opsi untuk menghubungkan program dengan library libpcap dan menghubungkan program dengan library SQLite3.

Gambar 16. Pemindahan File Compiled

Pada gambar tersebut terdapat perintah sudo mv dosd /usr/local/bin/, yaitu perintah untuk memindahkan file hasil kompilasi proses sebelumnya ke dalam folder /usr/local/bin/ pada kernel Linux.

Gambar 17. Penulisan File dosd.service

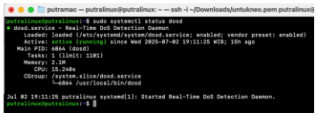
Pada gambar 17, terlihat sebuah aplikasi nano yang dimana fungsi aplikasi tersebut merupakan sebuah text editor bawaan UNIX Kernel. Penulis membuat sebuah file dengan nama dosd.service yang dibuat pada folder /etc/systemd/system/ dengan perintah sebagai berikut. cd /etc/systemd/system/ && sudo nano dosd.service.

Gambar 18. Penulisan File dosd.service

Perintah sudo systemctl daemon-reexec adalah perintah yang digunakan dalam system Linux yang menggunakan system sebagai init systemd. Perintah ini memerintahkan systemd yang berjalan sebagai daemon di latar belakang untuk membaca ulang sebuah unit file yang mungkin telah dimodifikasi, dan melakukan re-exec dirinya sendiri yang artinya proses sytemd akan memuat ulang kodenya sendiri dan memulai ulang tanpa me-reboot seluruh system.

Gambar 19. Perintah Running Daemon

Perintah ini juga terkait dengan systemd dan digunakan untuk mengelola layanan pada system Linux. Perintah ini akan melakukan dua hal utama, yaitu mengatur agar layanan dosd dimulai secara otomatis setiap kali system boot. Ini memastikan bahwa program detector DoS akan selalu berjalan seletah system dihidupkan ulang tanpa perintah intervansi manual. Yang kedua segera memulai layanan dosd pada saat itu juga, berarti tanpa perlu menjalankan sudo systemctl start dosd secara terpisah setelah mengaktifkannya. Untuk memastikan bahwasanya tool tersebut sudah berjalan pada daemon server, kita bisa melihat proses tersebut dengan melakukan perntah sudo systemctl status dosd, jika berhasil berjalan, maka akan tampil seperti gambar berikut.



Gambar 20. Cek Status dosd

3.4 Sprint 2 – Logika ekstraksi dan pengambilan informasi

Pada fase ini, sistem mengekstrak informasi penting dari paket, seperti IP sumber, protokol, dan port tujuan, melacak frekuensi paket menggunakan struktur data HashMap, menetapkan ambang batas (100 paket per 2 detik) sebagai parameter keamanan, dan mengirimkan hash.

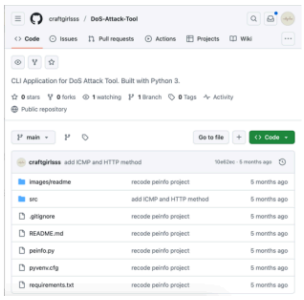
3.5 Sprint 3 – Penyimpanan Log untuk Pemblokir Otomatis

Pada fase ini, integrasi iptables dapat secara otomatis memblokir alamat IP, menghubungkan ke SQLite, menambahkan fitur pencatatan ke sistem, dan Sprint hail: Secara otomatis memblokir dan mencatat alamat IP yang diblokir oleh sistem.

3.6 Sprint 4 – Pengujian dan Evaluasi Sistem

Pengujian pada tool DoS Defender, akan menggunakan sebuah tool dari DoS

Attacker buatan penulis sendiri yang Bernama PeInfo DoS Attack Tool yang dibuat menggunakan Bahasa pemorgraman Python 3. Untuk lebih jelas penulis akan memberikan repository tool tersebut untuk dapat digunakan sebagia alat testing untuk menyerang tool defender tersebut. Sebelum melakukan cloning aplikasi, pastikan terlebih dahulu komputer telah diinstal python3.



Gambar 21. Repository DoS Attack Tool

Kloning Repository tersebut dengan perintah git clone https://github.com/craftgirlsss/DoS-Attack-Tool

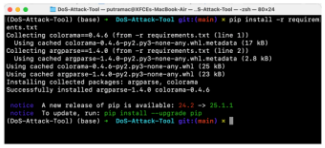
Gambar 22. Cloning DoS Attack Tool

Jika berhasil, maka akan muncul tampilan seperti pada gambar 21.

Langkah kedua yaitu mengaktifkan lingkungan environment python3 dengan perintah sebagai berikut. cd DoS-Attack-Tool && python3 -m venv ./
Setelah itu langkah selanjutnya melakukan aktivasi lingkungan environment python3 yang telah diinisialisasi oleh perintah sebelumnya dengan perintah sebagai berikut, cd bin && source ./activate && cd ..

Gambar 22. Aktivasi Environment

Jika berhasil maka akan tampil seperti pada gambar 22. Langkah selanjutnya yaitu instalasi package PIP yang dibutuhkan oleh tool tersebut dengan perintah sebagai berikut, pip install -r requirements.txt && chmod +x peinfo.py



Gambar 23. Install Package menggunakan PIP

Jika berhasil, maka akan tampil seperti pada gambar 22.

Gambar 24. Testing PeInfo App

Pada gambar 23, merupakan tampilan awal ketika kita mengeksekusi tool tersebut dengan perinta ./peinfo.py -h , jika berhasil maka akan muncul tampilan tersebut. Sistem yang digunakan menggunakan waktu nyata seperti hping3, slowloris, dan3 ping flood memantau kinerja CPU dan server, mengevaluasi

kecepatan perbaikan dan efisiensi pemblokiran, dan memiliki fitur sprint untuk memperbaiki DoS secara waktu nyata dan mengurangi kinerja server tanpa perubahan yang signifikan.

3.7 Scrum Harian dan Iterasi

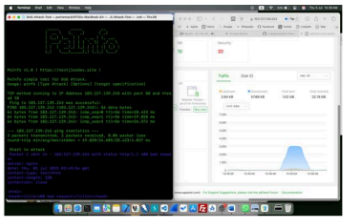
Selama fase ini, sistem diuji untuk menyelesaikan masalah teknis, merencanakan sprint mendatang, dan menyediakan scrum harian yang efektif untuk mempertahankan produktivitas dan mencegah stagnasi.

3.8 Tinjauan dan Retrospeksi Sprint

Tinjauan dilakukan selama setiap sprint untuk menunjukkan tugas yang diselesaikan seperti pemodelan dan pengujian unit. Evaluasi retrospektif digunakan untuk mengevaluasi proses kerja dan mengidentifikasi area yang perlu ditingkatkan. Kriteria evaluasi meliputi ancaman dengan ambang batas tinggi dan log SQLite yang tertunda.

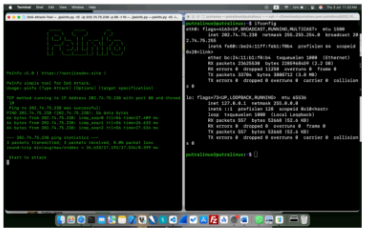
Metodologi Agile Scrum bekerja dengan sangat baik untuk membuat proses pengembangan lebih mudah dan lebih mudah diakses. Sistem tinjauan retrospektif berhasil dievaluasi untuk memperhitungkan waktu keseluruhan struktur sprint guna memastikan bahwa proses tersebut memenuhi tujuan-tujuan ini, konsisten dan berkelanjutan, sehingga memastikan sistem yang terkini dan adaptif.

4. HASIL



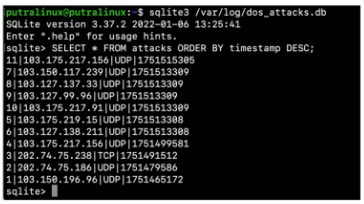
Gambar 25. Hasil Testing Attacking Server 1

Pada gambar tersebut merupakan hasil dari DoS Attacking menggunakan tool PeInfo yang dibuat oleh penulis dengan sasaran server tanpa dilindungi oleh DoS Attack Defender Tool yang dibuat oleh penulis. Terlihat bahwasanya request yang terjadi saat tool PeInfo dijalankan dengan target tujuan IP Server tersebut, request flood hingga 3000 requst per second. Ini bertanda bahwasanya tool PeInfo mampu melakukan serangan kepada server target yang dapat membanjiri bandwith server korban.



Gambar 25. Hasil Testing Attacking Server 2

Pada gambar 25, terlihat bahwasanya untuk terminal sisi kiri merupakan host local sebagai attacker menggunakan tool PeInfo, sedangkan pada sisi kanan merupakan terminal pada server 2 yang penulis remote menggunakan SSH. Pada server 2 telah diinstall tool DoS Defender Tool pada Langkah diatas sebelumnya, terlihat bahwsanya PeInfo tidak mampu melakukan serangan, terbukti Ketika dijalankan, tidak tampak hasil serangan seperti pada gambar 24 yang diperlihatkan hasil serangan. Ini terbukti bahwasanya tool tersebut sudah bekerja dengan baik. Namun untuk memastikan hal tersebut sudah bekerja dengan semestinya, mari kita baca database SQLite yang mana telah difungsikan untuk menyimpan daftar IP Address yang diidentifikasi sebagai kejahatan cyber dengan teknik DoS. Berikut perintah yang digunakan untuk membaca hasil Log Serangan



Gambar 26. Hasil Log Serangan pada Server 2

Hasil penelitian ini menerapkan sistem mitigasi Denial of Service (DoS) real-time pada server VPS yang menunjukkan bahwa pendekatan ini dapat secara efektif melindungi terhadap serangan yang membahayakan server. Sistem ini menggunakan bahasa pemrograman C dan dirancang untuk bekerja sebagai daemon (proses latar belakang) pada sistem operasi Linux [14].

Pada tahap awal, sistem terhubung ke basis data SQLite lokal untuk menganalisis log alamat IP yang diidentifikasi sebagai sumber daya keamanan [15]. Sistem juga mengaktifkan pustaka libpcap untuk menangkap paket yang masuk dari antarmuka, yang menyediakan transfer data berkinerja tinggi.

Respons sistem akan mencakup pelacakan aktivitas dalam basis data SQLite, memblokir alamat IP yang masuk dengan iptables, dan mengirimkan pemberitahuan log ke sistem untuk akses administrator yang lebih baik. Pendekatan ini dapat membantu mencegah serangan real-time seperti HTTP Flood dan ICMP Flood. Implementasinya menunjukkan bahwa sistem dapat mendeteksi dan memblokir alamat IP dalam waktu singkat, sehingga mengurangi kebutuhan untuk pemodelan atau analisis yang rumit. Sistem ini juga memiliki kinerja tinggi karena penggunaan bahasa C tingkat rendah, penggunaan memori rendah, dan CPU kecil [7].

Keterbatasan sistem ini antara lain deteksi dan mitigasi real-time dalam waktu singkat, tidak melibatkan administrator secara manual, pengoperasian yang efisien, modularitas dan persaudaraan, serta respons efektif berdasarkan waktu paket [16]. Sistem ini juga tidak memiliki kemampuan pemantauan visual, hanya menyediakan file log dan data dasar untuk audit, serta tidak mengintegrasikan notifikasi real-time kepada administrator melalui email atau Telegram [17].

Implementasi program ini menunjukkan bahwa penggunaan bahasa pemrograman C dapat secara efektif melindungi terhadap serangan DoS pada server VPS [17], [18]. Sistem dalam penelitian ini memungkinkan pemrosesan secara real-time dan oportunistik, yang memungkinkan sistem untuk digunakan pada server kecil dan besar tanpa biaya dan kompleksitas yang tinggi.Kesimpulan dan Pekerjaan Masa Depan [19].

Bagian ini berisi simpulan penelitian [15]. Usahakan untuk menyertakan data yang memperkuat simpulan yang dibuat di bagian simpulan ini. Hal ini akan menunjukkan kekuatan dan kelemahan penelitian. Anda juga dapat menambahkan saran untuk pengembangan penelitian di masa mendatang di bagian ini. Simpulan ditulis dalam bentuk paragraf naratif dan tidak ditulis dalam bentuk poin [20].

5. KESIMPULAN

Penelitian ini berfokus pada penerapan deteksi serangan DoS secara real-time dan mitigasi DoS secara real-time pada server VPS berbasis C. Sistem ini dirancang untuk digunakan sebagai daemon yang berjalan pada mesin host, yang dapat mendeteksi paket masuk untuk setiap alamat IP dalam jangka waktu tertentu. Sistem ini menggunakan ambang batas dinamis untuk memblokir alamat IP menggunakan iptables dan mencatat aktivitasnya dalam basis data log SQLite lokal.

Proses pengembangannya meliputi beberapa sprint, termasuk menyiapkan lingkungan pengembangan, mengintegrasikan libpcap,

menganalisis paket, melakukan debugging, pemblokiran baru, dan memantau sistem untuk simulasi. Hasilnya menunjukkan bahwa sistem ini memberikan manfaat signifikan dalam memperbaiki dan memblokir serangan DoS seperti *HTTP Flood dan ICMP Flood* dalam waktu 0,5 detik. Sistem ini juga menggunakan CPU dan jejak memori yang kecil, sehingga cocok untuk lingkungan VPS dengan memori terbatas.

Penggunaan bahasa C juga memegang peranan penting dalam penerapan sistem ini. Bahasa C memberikan kontrol yang lebih baik terhadap proses JavaScript berkinerja tinggi dan menawarkan kinerja yang optimal dibandingkan dengan bahasa seperti Python atau Java yang membutuhkan lebih banyak memori dan waktu eksekusi. Mekanisme pemblokiran otomatis menggunakan iptables dapat dilihat sebagai proses yang menyederhanakan proses dengan menghilangkan kebutuhan intervensi manual, mengurangi waktu henti, dan mencegah potensi ancaman keamanan.

6. SARAN

Penelitian ini menyarankan beberapa perbaikan untuk meningkatkan efisiensi dan respons terhadap serangan. Penelitian ini merekomendasikan integrasi yang *real-time*, seperti melalui Telegram, email, atau dasbor berbasis web, untuk memungkinkan administrator mendeteksi dan merespons serangan dengan cepat. Dasbor visual dapat dikembangkan untuk memantau status server secara intuitif, menyediakan informasi waktu nyata tentang grafik lalu lintas jaringan, IP yang diblokir, dan log aktivitas. Algoritma *Machine Learning* seperti *Random Forest*, *Isolation Forest*, atau KNN dapat digunakan untuk deteksi anomali dan meningkatkan kemampuan prediksi dan deteksi serangan lapisan aplikasi. Ambang batas dinamis dapat digunakan untuk mengurangi positif palsu dan negatif palsu. Sistem dapat mendeteksi serangan pada beberapa port dan protokol secara bersamaan, terutama jika VPS digunakan untuk beberapa layanan dalam satu sistem. Menerapkan daftar putih IP dapat mencegah pemblokiran IP sah yang sering mengakses sistem. Sistem dapat diuji dalam lingkungan skala besar dan produksi untuk memastikan skalabilitas, keandalan, dan toleransi terhadap beban tinggi. Penelitian lebih lanjut harus difokuskan pada penyusunan modul pustaka untuk penggunaan kembali dan integrasi yang lebih mudah oleh pengguna dengan kebutuhan yang berbeda.

DAFTAR PUSTAKA

[1] R. R. Nuiaa, S. Manickam, and A. H. Alsaeedi, “Distributed reflection denial of service attack: A critical review,” Dec. 01, 2021, *Institute of Advanced Engineering and Science*. doi: 11591/ijece.v11i6.pp5327-5341.

[2] P. Simarmata, N. F. Saragih, I. K. Jaya, and H. Artikel, “Deteksi Serangan DDOS Pada VPS Menggunaka Metode Deep Neural Network,” 2023. [Online]. Available: http://ojs.fikom-ethodist.net/index.php/methotika

[3] R. Sommese *et al.*, “Investigating the impact of DDoS attacks on DNS infrastructure,” in *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, Association for Computing Machinery, Oct. 2022, pp. 51–64. doi: 10.1145/3517745.3561458.

[4] U. Kingsley and J. Sonia, “Analysis of Linux Kernel Iptables for Mitigating DDOS Attacks; A Component-Based Approach,” *International Journal of Computer Science and Mathematical ory*, 2021, doi: 10.56201/ijcsmt.v9.no4.2023.pg12.22.

[5] L. D. Garcia, “REAL-TIME NETWORK SIMULATIONS FOR ML/DL DDOS DETECTION USING DOCKER,” 2024.

[6] D. Lincopinis, B. B. Chavez, T. Angelo, J. Gitalan, and D. R. Lincopinis, “C Programming Language: A Review,” *Journal of Universal Computer Science*, vol. 27, no. 1, 2021, doi: 10.3897/jucs.

[7] A. Alsabeh, J. Khoury, E. Kfoury, J. Crichigno, and E. Bou-Harb, “A Survey on

Security Applications of P4 Programmable Switches and a STRIDE-based Vulnerability Assessment," 2022. [Online]. Available: https://www.elsevier.com/open-access/userlicense/1.0/

[8] W. Chrisdianto and S. A. Putri, "PENGEMBANGAN SISTEM MANAJEMEN TEMA WEBSITE BERBASIS METODE AGILE SCRUM."

[9] A. C. Sassa, I. Alves De Almeida, T. Nakagomi, F. Pereira, and M. Silva De Oliveira, "Scrum: A Systematic Literature Review." [Online]. Available: w.ijacsa.thesai.org

[10] D. J. K. Putra and P. F. Tanaem, "Perancangan Aplikasi Pembukuan Menggunakan Metode Agile Scrum," *nal Teknik Informatika dan Sistem Informasi*, vol. 8, no. 3, Dec. 2022, doi: .28932/jutisi.v8i3.5060.

[11] C. Resmi Rachmawati, Deyana Kusuma Wardani, Wifda Muna Fatihia, Arna Fariza, and Hestiasari Rante, "Implementing Agile Scrum Methodology in The Development of SICITRA Mobile Application," *Jurnal ESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 1, pp. 41–50, Feb. 2023, doi: 10.29207/resti.v7i1.4688.

[12] N. Rizky *et al*., "Analisis Kebutuhan Sistem Informasi Manajemen Kegiatan Kemahasiswaan STIKI Malang Menggunakan Agile Requirements Engineering," *J-Intech : Journal of Information and Technology*, vol. 10, no. 1, pp. 21–29, 2022, doi: 10.32664/j-tech.v10i01.

[13] N. Edrina Christine *et al*., "PENERAPAN METODE AGILE SCRUM PADA SISTEM E-POSYANDU BERBASIS WEB," 2024.

[14] S. Hartono and K. Khotimah, "DETEKSI DAN MITIGASI SERANGAN BACKDOOR MENGGUNAKAN PYTHON WATCHDOG."

[15] P. Szynkiewicz, "Signature-Based Detection of Botnet DDoS Attacks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13300 LNCS, Springer Science and Business Media Deutschland GmbH, 2022, pp. 120–135. doi: 10.1007/978-3-031-04036-8_6.

[16] D. Said, "Quantum Computing and Machine Learning for Cybersecurity: Distributed Denial of Service (DDoS) Attack Detection on Smart Micro-Grid," *Energies (Basel)*, vol. 16, no. 8, Apr. 2023, doi: 10.3390/en16083572.

[17] "Naskah Publikasi_L200180128".

[18] M. Hamidouche, B. F. Demissie, and B. Cherif, "Real-time Threat Detection Strategies for Resource-constrained Devices," Mar. 2024, [Online]. Available: http://arxiv.org/abs/2403.15078

[19] F. Panduardi, H. Yuliandoko, and A. Priyo Utomo, "Network Security Using Honeypot and Attack Detection with Android Application," *Indonesian Journal of EngineeringResearch*, vol. 2, no. 2, pp. 53–60, 2021, doi: 10.11594/ijer.02.02.04.

[20] H. Setiawan, W. Sulistyo, F. Teknologi Informasi, and U. Kristen Satya Wacana, "SIEM (Security Information Event Management) Model for Malware Attack Detection Using Suricata and Evebox," *International Journal of Engineering*, vol. 5, no. 2, 2023.

final-journal-saputra.docx

ORIGINALITY REPORT

10%	8%	7%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to LL Dikti IX Turnitin Consortium Student Paper	3%
2	eprints.umm.ac.id Internet Source	1%
3	Submitted to Aotearoa Career & Management Institute Student Paper	1%
4	repository.stiki.ac.id Internet Source	<1%
5	Submitted to University of Technology Student Paper	<1%
6	ejournal.uin-suska.ac.id Internet Source	<1%
7	journal.uii.ac.id Internet Source	<1%
8	Ade Bastian, Sarmidi, Dadan Zaliluddin, Mochammad Bagasnanda Firmansyah. "Development of Augmented Reality Programming Language using Agile Scrum Methodology", Jurnal Online Informatika, 2023 Publication	<1%
9	Muhammad Reza Maulana. "EVALUASI METODOLOGI WATERFALL DAN AGILE: STUDI LITERATUR PADA SISTEM PERPUSTAKAAN",	<1%

	Jurnal Informatika dan Teknik Elektro Terapan, 2025 Publication	
10	Submitted to St. Petersburg College Student Paper	<1 %
11	journal.universitasbumigora.ac.id Internet Source	<1 %
12	Submitted to Purdue University Student Paper	<1 %
13	id.scribd.com Internet Source	<1 %
14	Said Ridho. "Analisis Preferensi Konsumen dalam Memilih Produk Hortikultura Menggunakan Metode Algoritma C45 dan Naive Bayes", Emitor: Jurnal Teknik Elektro, 2024 Publication	<1 %
15	caranyaitu.blogspot.com Internet Source	<1 %
16	es.scribd.com Internet Source	<1 %
17	jurnal.stmik-yadika.ac.id Internet Source	<1 %
18	www.coursehero.com Internet Source	<1 %
19	Muhammad Firdaus Yusuf, Ira Rosianal Hikmah, Amiruddin, Septia Ulfa Sunaringtyas. "Security Testing of XYZ Website Application Using ISSAF and OWASP WSTG v4.2 Methods", Teknika, 2025 Publication	<1 %
20	Submitted to RMIT University Student Paper	

		<1%
21	Ridwan Setiawan, Deni Heryanto, Faizal Rifaldy. "Optimisasi Monitoring Tugas Akhir Mahasiswa Dengan Integrasi Formasi Metode Agile Framework Scrum dan Notifikasi WhatsApp di Institut Teknologi Garut", Teknika, 2024 Publication	<1%
22	Rodney Quaye. "Event-Database Architecture for Computer Games - Volume 1, Software Architecture and the Software Production Process", CRC Press, 2025 Publication	<1%
23	diahasnas.blogspot.com Internet Source	<1%
24	eksplorasi.org Internet Source	<1%
25	journal.ubm.ac.id Internet Source	<1%
26	jurnal.umko.ac.id Internet Source	<1%
27	repositorium.sdum.uminho.pt Internet Source	<1%
28	toffeedev.com Internet Source	<1%
29	usv.ro Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		