JURNAL MEDIA AKADEMIK (JMA)
Vol.2, No.12 Desember 2024
e-ISSN: 3031-5220; DOI: 10.62281, Hal XX-XX
PT. Media Akademik Publisher
AHU-084213.AH.01.30.Tahun 2023

PENGEMBANGAN *FIREWALL* SEDERHANA BERBASIS *PYTHON* DENGAN *SCIPY*

Oleh:

Linna Oktaviana Sari[1]
Ridho Zikril Hidayatullah[2]

Universitas Riau
Alamat: Kampus Bina Widya KM. 12,5, Simpang Baru, Kec. Tampan, Kota Pekanbaru, Riau (28293).
Korespondensi Penulis: linnaosari@lecturer.unri.ac.id

Abstract. *Network security is a crucial aspect in protecting systems from external threats. A firewall serves as a barrier to filter and block unwanted data packets from outside the network. This article develops a simple firewall using the Python programming language and the Scapy library. The system utilizes Scapy to capture packets passing through the network and then filters packets based on specified rules, such as blocking packets from a specific source IP address. Additionally, the firewall logs blocked packets in a log file for further analysis. In the implementation of this firewall, the rule used is to block all packets coming from a certain IP address, such as 192.168.1.100. Testing is performed by sending packets from two devices with different IP addresses. The test results show that the firewall successfully blocks packets from the IP address 192.168.1.100 and allows packets from the IP address 192.168.1.101. These results indicate that the developed firewall can effectively block packets from the specified IP address and allow packets that meet the rules. This system shows potential for use in securing personal networks and can be further developed by adding filters based on ports or specific protocols.*

Keywords: *Network, Firewall, Python, IP Address.*

Abstrak. Keamanan jaringan merupakan aspek penting dalam melindungi sistem dari ancaman luar. *Firewall* berfungsi sebagai penghalang untuk memfilter dan memblokir

Received November 26, 2024; Revised December 05, 2024; December 09, 2024
*Corresponding author: linnaosari@lecturer.unri.ac.id

paket data yang tidak diinginkan dari luar jaringan. Artikel ini mengembangkan *firewall* sederhana menggunakan bahasa pemrograman *Python* dan pustaka *Scapy*. Sistem ini memanfaatkan *Scapy* untuk menangkap paket-paket yang melewati jaringan, kemudian memfilter paket berdasarkan aturan yang ditentukan, seperti pemblokiran paket berdasarkan alamat IP sumber. Selain itu, *firewall* juga mencatat paket yang diblokir dalam *file log* untuk analisis lebih lanjut. Pada implementasi *firewall* ini, aturan yang digunakan adalah untuk memblokir semua paket yang datang dari alamat IP tertentu, seperti 192.168.1.100. Pengujian dilakukan dengan mengirimkan paket dari dua perangkat dengan alamat IP yang berbeda. Hasil pengujian menunjukkan bahwa *firewall* berhasil memblokir paket dari alamat IP 192.168.1.100 dan meneruskan paket dari alamat IP 192.168.1.101. Hasil ini menunjukkan bahwa *firewall* yang dikembangkan dapat secara efektif memblokir paket dari alamat IP yang telah ditentukan dan meneruskan paket yang sesuai dengan aturan. Sistem ini menunjukkan potensi untuk digunakan dalam pengamanan jaringan pribadi, dan dapat dikembangkan lebih lanjut dengan menambahkan fitur filter berdasarkan *port* atau protokol tertentu.

Kata Kunci: Jaringan, *Firewall*, *Python*, Alamat IP**.**

LATAR BELAKANG

Di era digital, ancaman terhadap keamanan jaringan semakin kompleks seiring dengan meningkatnya penggunaan internet untuk kebutuhan pribadi, bisnis, dan pemerintahan. Peretas sering memanfaatkan celah untuk mengakses data sensitif atau merusak sistem, sehingga *firewall* menjadi kebutuhan penting. *Firewall* berfungsi menyaring lalu lintas data untuk memastikan hanya paket yang aman yang dapat melewati jaringan. Namun, *firewall* tradisional seringkali memerlukan perangkat keras khusus dan biaya tinggi, membuat solusi ini kurang terjangkau bagi pengguna jaringan kecil atau pribadi.

Sebagai alternatif, pengembangan *firewall* berbasis perangkat lunak menggunakan *Python* dan pustaka seperti *Scapy* menawarkan solusi yang lebih fleksibel dan ekonomis. *Scapy* memungkinkan penangkapan, pemfilteran, dan pengiriman ulang paket jaringan, menjadikannya alat yang efektif untuk membangun *firewall* sederhana. Dalam penelitian ini, dikembangkan sistem *firewall* yang dapat memblokir paket

berdasarkan alamat IP tertentu, menyediakan solusi pengamanan jaringan yang mudah diterapkan dan dapat dikembangkan lebih lanjut untuk kebutuhan yang lebih besar.

KAJIAN TEORITIS

Muqorobin, M., Hisyam, Z., Mashuri, M., & Hanafi, H. (Tahun). Penerapan Network Intrusion Detection System (NIDS) untuk Keamanan *Cloud Computing*. Penelitian ini bertujuan untuk meningkatkan keamanan pada sistem cloud computing dengan menggunakan *Network Intrusion Detection System* (NIDS). Sistem ini dirancang untuk mendeteksi ancaman dari luar maupun dalam cloud, dengan memanfaatkan metode mirroring traffic pada switch untuk menganalisis seluruh lalu lintas jaringan. Hasil penelitian menunjukkan bahwa NIDS mampu mendeteksi serangan dan memberikan peringatan (*alert*) terhadap traffic yang mencurigakan baik dari luar maupun antar virtual machine dalam sistem cloud. Penelitian ini relevan untuk mendukung pengembangan firewall berbasis perangkat lunak sebagai bagian dari pengamanan jaringan.

Dharmesta, P. A., Suarjaya, I. M. A. D., & Raharja, I. M. S. Aplikasi Pembelajaran Lalu Lintas Protokol pada Jaringan Komputer dengan *Scapy* dan *Natural Language*. Penelitian ini mengembangkan sebuah aplikasi berbasis Scapy untuk pembelajaran lalu lintas protokol jaringan komputer, dengan menggunakan natural language untuk menyampaikan hasil translasi proses *sniffing*. Hasil penelitian menunjukkan bahwa penggunaan natural language dalam menampilkan hasil *sniffing* meningkatkan pemahaman pengguna tentang proses kerja paket jaringan. Berdasarkan skala Likert, hasil translasi dengan *natural language* mendapatkan skor pemahaman sebesar 73%, dibandingkan dengan hasil asli *sniffing* yang hanya sebesar 37%. Penelitian ini membuktikan efektivitas pendekatan natural language dalam meningkatkan pemahaman konsep jaringan, khususnya untuk kalangan siswa jurusan Teknik Komputer dan Jaringan. Studi ini relevan untuk mendukung pengembangan perangkat lunak yang memanfaatkan Scapy, khususnya dalam aplikasi yang berfokus pada analisis paket jaringan dan peningkatan pemahaman pengguna.

Menurut Ramadhan, R. A., Tira, A. T., & Fadhilah, M. R. Implementasi *Network Forensic Generic Process* dalam Penanganan Serangan *Address Resolution Protocol Poisoning*. Penelitian ini bertujuan untuk memahami dan menangani serangan *Address Resolution Protocol* (ARP) Poisoning yang dapat mencuri data, memodifikasi *traffic*, dan

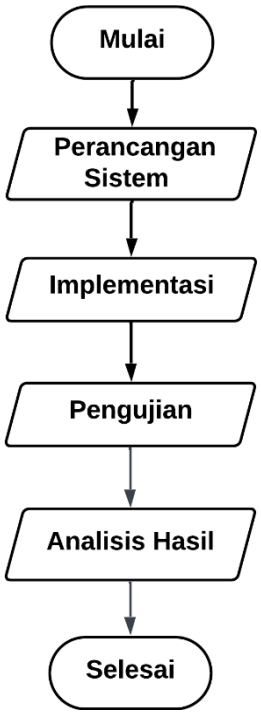
menghentikan komunikasi jaringan. Menggunakan model *Network Forensic Generic Process*, penelitian ini menganalisis dampak serangan terhadap *quality of service* (QoS) jaringan dan membandingkan parameter QoS sebelum dan selama serangan. *Tools* seperti Wireshark, XArp, dan Snort digunakan untuk mengidentifikasi, menganalisis, dan mengumpulkan bukti forensik dari serangan yang terjadi. Hasil penelitian menunjukkan adanya perubahan signifikan pada QoS selama serangan, dan proses forensik berhasil mendapatkan bukti yang autentik. Temuan ini memberikan wawasan tentang ancaman keamanan jaringan serta langkah-langkah yang dapat diambil untuk mencegah dan merespons serangan di masa mendatang.

Hibatul Wafi, (2016). Penerapan *Honeypot* dengan *Modern Honey Network* untuk Meningkatkan Keamanan Jaringan di PUSTIKNAS. Penelitian ini membahas penerapan sistem *Honeypot* untuk meningkatkan keamanan jaringan di PUSTIKNAS, yang sering menjadi target serangan dari pihak luar. Dengan menggunakan *Honeypot* yang dikelola melalui *Modern Honey Network* (MHN), seperti kippo, glastopf, dan dionaea, penelitian ini bertujuan untuk mendeteksi dan memantau usaha penyusupan yang dilakukan oleh attacker. Pengujian sistem dilakukan dengan berbagai skenario serangan seperti port scanning, DoS, DDoS, dan brute-force. Hasil penelitian menunjukkan bahwa *Honeypot* berhasil mengecoh *attacker* dengan membuka *port-port* tipuan pada server, yang akhirnya dapat mendeteksi aktivitas mencurigakan dan menganalisis serangan yang datang. Sistem ini memberikan kontribusi signifikan terhadap pengembangan sistem keamanan jaringan yang lebih efektif dan dapat diterapkan pada instansi pemerintah atau organisasi lainnya.

Menurut Raj, R.. Manipulasi Paket Interaktif dengan *Scapy*: Penggunaan, Kekuatan, dan Kelemahan dalam Jaringan Komputer. Makalah ini membahas alat Scapy, yang ditulis dalam bahasa Python, untuk manipulasi paket interaktif di jaringan komputer. Penelitian ini merinci perintah-perintah penting dalam Scapy, serta kekuatan dan kelemahannya dalam memanipulasi dan memonitor lalu lintas jaringan. Selain itu, penulis juga menjelaskan dukungan Scapy terhadap berbagai protokol dan cara penggunaan alat ini melalui snapshot, serta integrasi masa depan dengan protokol yang lebih baru. Penelitian ini menawarkan panduan praktis bagi pengguna yang ingin memahami penggunaan Scapy untuk pelacakan paket yang dikirim dan diterima, serta cara menganalisis data menggunakan Wireshark.

METODE PENELITIAN

Metode yang digunakan dalam pengembangan *firewall* sederhana ini adalah metode pengembangan perangkat lunak dan pemrograman jaringan yang terdiri dari beberapa tahap yang mencakup tahapan perancangan, pengembangan, implementasi, dan pengujian sistem *firewall* menggunakan *Python* dan pustaka *Scapy*, tahap-tahap tersebut adalah sebagai berikut:



Gambar 1. Tahap Penelitian

1. Perancangan Sistem :
    - Sistem *firewall* dirancang untuk menyaring dan memblokir paket jaringan berdasarkan alamat IP sumber yang telah ditentukan. Aturan filter yang diterapkan berupa pemblokiran paket yang datang dari alamat IP tertentu, seperti 192.168.1.100, dan memungkinkan paket lainnya untuk diteruskan.
    - Desain sistem mengandalkan pustaka *Python Scapy* untuk menangkap dan memodifikasi paket data yang ada dalam jaringan.
2. Implementasi :
    - Sistem *firewall* diimplementasikan menggunakan *Python* dan pustaka *Scapy*. *Scapy* digunakan untuk menangkap paket (*sniffing*)

yang melewati jaringan, serta untuk memodifikasi dan memfilter paket berdasarkan kriteria yang sudah ditentukan.

- Kode *Python* dikembangkan untuk menangkap paket, memeriksa apakah alamat IP sumber sesuai dengan aturan pemblokiran, dan memblokir atau meneruskan paket tersebut.
- *File log* digunakan untuk mencatat paket yang diblokir, yang kemudian bisa dianalisis untuk mendeteksi potensi ancaman atau aktivitas jaringan yang mencurigakan.

3. Pengujian :
    - Pengujian dilakukan dengan mengirimkan paket dari dua perangkat yang memiliki alamat IP yang berbeda. Salah satu perangkat menggunakan alamat IP yang diblokir (192.168.1.100), sementara perangkat lainnya menggunakan alamat IP yang diizinkan (192.168.1.101).
    - Tujuan pengujian adalah untuk memastikan bahwa *firewall* berfungsi sesuai dengan aturan yang telah ditentukan, yaitu memblokir paket dari IP yang tidak diinginkan dan meneruskan paket yang sesuai.
    - Hasil pengujian dievaluasi berdasarkan *output* konsol dan *file log* untuk memverifikasi keberhasilan pemblokiran dan penerusan paket.
4. Analisis Hasil :
    - Hasil pengujian dievaluasi untuk memastikan bahwa sistem berfungsi dengan baik dalam memblokir paket yang tidak diinginkan dan meneruskan paket yang diizinkan.
    - Analisis dilakukan dengan memeriksa log file dan hasil output console untuk memverifikasi apakah aturan firewall diterapkan dengan benar.
    - Keberhasilan sistem diukur berdasarkan apakah firewall berhasil memblokir paket yang sesuai dan mencatatnya dalam file log.

HASIL DAN PEMBAHASAN

Pada bagian ini, akan dibahas mengenai hasil pengujian dari sistem *firewall* yang telah dikembangkan dan implementasi fungsionalitasnya. Sistem *firewall* sederhana berbasis *Python* dan *Scapy* ini dirancang untuk memblokir paket dari alamat IP tertentu dan meneruskan paket lainnya. Berikut adalah rincian hasil yang diperoleh dari pengujian dan pembahasan terkait efektivitas sistem.

Hasil Pengujian

Pengujian dilakukan dengan mengirimkan paket data melalui dua perangkat dengan alamat IP yang berbeda. Perangkat pertama menggunakan alamat IP yang diblokir (misalnya 192.168.1.100), sementara perangkat kedua menggunakan alamat IP yang diizinkan (misalnya 192.168.1.101). Pengujian bertujuan untuk memastikan bahwa *firewall* berfungsi sesuai dengan aturan yang telah ditentukan, yaitu memblokir paket dari alamat IP yang tidak diinginkan dan meneruskan paket lainnya.

Output Console : Pada saat pengujian dilakukan, *output console* menunjukkan hasil berikut:

Paket yang Diblokir:

```
Blocked packet from 192.168.1.100 to 192.168.1.1
Blocked packet from 192.168.1.100 to 192.168.1.2
```

Gambar 2. Output Console Paket Yang Diblokir

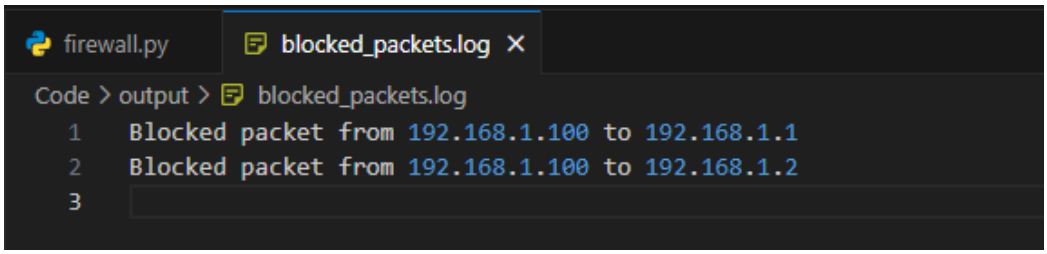
Paket yang Diteruskan:

```
Allowed packet from 192.168.1.101 to 192.168.1.1
Allowed packet from 192.168.1.101 to 192.168.1.2
```

Gambar 3. Output Console Paket Yang Diteruskan

Dari hasil pengujian ini, dapat disimpulkan bahwa *firewall* berhasil memblokir paket yang berasal dari alamat IP 192.168.1.100 dan membiarkan paket dari alamat IP 192.168.1.101 untuk melewati sistem tanpa gangguan

File Log : Selain *Output Console*, *Firewall* juga mencatat paket yang diblokir dalam sebuah *file log* (blocked_packets.log). Berikut adalah contoh entri dalam *log*:



Gambar 4. Hasil File Log Program

Hasil ini mengindikasikan bahwa *firewall* bekerja dengan baik dalam mencatat setiap paket yang diblokir, yang memungkinkan pengguna untuk menganalisis aktivitas jaringan yang mencurigakan.

Pembahasan

Sistem *firewall* yang dikembangkan menggunakan *Python* dan *Scapy* menunjukkan hasil yang memadai dalam menyaring paket jaringan sesuai dengan aturan yang ditentukan. Pada pengujian pertama, *firewall* dapat dengan efektif memblokir paket yang datang dari alamat IP yang telah diblokir (192.168.1.100), sementara paket dari alamat IP lainnya, seperti 192.168.1.101, diteruskan tanpa masalah. Hal ini menunjukkan bahwa mekanisme pemfilteran berbasis alamat IP dalam *firewall* berfungsi dengan baik.

Salah satu aspek penting yang dicatat selama pengujian adalah bahwa *firewall* tidak hanya memblokir paket yang tidak diinginkan, tetapi juga mencatat aktivitas tersebut dalam *file log*. Fitur pencatatan ini sangat penting untuk analisis keamanan lebih lanjut, karena memungkinkan administrator jaringan untuk melacak potensi ancaman atau serangan yang mungkin terjadi. Selain itu, meskipun sistem *firewall* ini sudah berjalan dengan baik untuk aturan dasar pemblokiran berdasarkan alamat IP, sistem ini masih dapat dikembangkan lebih lanjut dengan menambahkan fitur-fitur tambahan, seperti:

Filter berdasarkan *Port* atau Protokol: Menambah kemampuan untuk memfilter paket berdasarkan *port* atau jenis protokol (misalnya TCP/UDP) untuk meningkatkan kontrol atas lalu lintas jaringan. Peningkatan Antarmuka Pengguna: Mengembangkan antarmuka berbasis grafis (GUI) untuk memudahkan konfigurasi dan *monitoring firewall* bagi pengguna yang tidak terbiasa dengan baris perintah. Keamanan dan *Log* Pengelolaan: Menambahkan mekanisme enkripsi pada *log* yang dicatat dan memberikan opsi pengelolaan *log* yang lebih canggih untuk analisis lebih lanjut. Secara keseluruhan, *firewall* yang dikembangkan ini menunjukkan potensi untuk digunakan dalam

pengamanan jaringan pribadi atau jaringan kecil, dan dapat lebih ditingkatkan untuk memenuhi kebutuhan pengamanan yang lebih besar dan lebih kompleks.

Keterbatasan dan Pengembangan Lebih Lanjut

Meskipun *firewall* ini dapat menjalankan fungsi dasar pemblokiran paket berdasarkan alamat IP dengan baik, ada beberapa keterbatasan yang perlu diperhatikan: Ketergantungan pada Alamat IP: *Firewall* ini hanya memblokir paket berdasarkan alamat IP, yang dapat di *bypass* dengan menggunakan teknik seperti *spoofing* alamat IP. Kinerja pada Jaringan Besar: Pada jaringan yang lebih besar, dengan volume paket yang tinggi, efisiensi sistem ini bisa berkurang, terutama jika dikembangkan lebih lanjut tanpa optimalisasi.

Penambahan Fitur Lanjutan: Pengembangan lebih lanjut, seperti kemampuan untuk memblokir atau memperbolehkan paket berdasarkan *port* atau protokol tertentu, akan sangat berguna untuk meningkatkan kontrol dan fleksibilitas *firewall*. Ke depan, sistem ini dapat dikembangkan dengan menambahkan fitur keamanan tambahan, seperti pencegahan serangan DoS/DDoS atau fitur untuk menangani paket yang mencurigakan berdasarkan pola trafik, yang akan membuat *firewall* lebih canggih dan lebih efektif.

KESIMPULAN DAN SARAN

Kesimpulan

Penelitian ini berhasil mengembangkan *firewall* sederhana menggunakan *Python* dan *Scapy* yang dapat memblokir paket berdasarkan alamat IP sumber. Sistem ini berhasil memblokir paket dari alamat IP yang tidak diinginkan dan meneruskan paket dari IP lainnya. Selain itu, *firewall* juga mencatat paket yang diblokir dalam *file log* untuk analisis lebih lanjut. Meskipun berhasil, sistem ini memiliki keterbatasan seperti ketergantungan pada alamat IP sebagai satu-satunya kriteria pemblokiran. Kedepannya, *firewall* ini dapat dikembangkan dengan menambahkan fitur filter berdasarkan *port* atau protokol dan peningkatan kinerja untuk jaringan yang lebih besar. Secara keseluruhan, *firewall* berbasis *Python* dan *Scapy* ini efektif untuk pengamanan jaringan kecil dan memiliki potensi untuk pengembangan lebih lanjut.

Saran

Berdasarkan hasil dan pembahasan pada penelitian "Pengembangan *Firewall* Sederhana Berbasis *Python* dengan *Scapy*" Terdapat beberapa saran yang akan menjadi pengembangan lagi utk sistem ini berikutnya :

1. **Pengembangan Fitur Lanjutan**
   Firewall ini dapat dikembangkan lebih lanjut dengan menambahkan filter berdasarkan port, protokol, atau pola lalu lintas untuk memberikan kontrol yang lebih mendetail terhadap lalu lintas jaringan.
2. **Peningkatan Kinerja**
   Untuk penggunaan pada jaringan dengan volume lalu lintas yang tinggi, diperlukan optimisasi agar firewall dapat memproses paket secara lebih efisien tanpa mengurangi kecepatan jaringan.
3. **Antarmuka Pengguna (GUI)**
   Menambahkan antarmuka pengguna berbasis grafis akan memudahkan pengguna yang tidak terbiasa dengan baris perintah untuk mengatur dan memonitor firewall ini.
4. **Keamanan Log dan Data**
   Meningkatkan keamanan log dengan menambahkan fitur enkripsi atau proteksi akses akan membantu menjaga kerahasiaan data terkait aktivitas jaringan yang dicatat oleh firewall.

DAFTAR REFERENSI

A. Zein, "Pendeteksian Kantuk Secara Real Time Menggunakan Pustaka OPENCV dan DLIB PYTHON Real Time Sleepiness Detection Using OPENCV Library and PYTHON DLIB," *Sainstech*, vol. 28, no. 2, pp. 22–26, 2018.

Asiva Noor Rachmayani, *Meningkatkan Keamanan Data Pada Attendance System Berbasis Face Recognation*. 2015.

I. K. Astuti, "Fakultas Komputer INDAH KUSUMA ASTUTI Section 01," *Jar. Komput.*, p. 8, 2018, [Online]. Available: https://id.scribd.com/document/503304719/jaringan-komputer

M. Muqorobin, Z. Hisyam, M. Mashuri, H. Hanafi, and Y. Setiyantara, "Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open

Cloud Computing," *Maj. Ilm. Bahari Jogja*, vol. 17, no. 2, pp. 1–9, 2019, doi: 10.33489/mibj.v17i2.205.

Muhammad Romzi and B. Kurniawan, "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma," *JTIM J. Tek. Inform. Mahakarya*, vol. 03, no. 2, pp. 37–44, 2020.

P. A. Dharmesta *et al.*, "Terakreditasi SINTA Peringkat 2 Efektivitas Sniffer Menggunakan Natural Language dalam Pembelajaran Lalu Lintas Jaringan Komputer," *Masa Berlaku Mulai*, vol. 1hs, no. 3, pp. 392–403, 2017.

R. A. Ramadhan, A. T. Tira, and M. R. Fadhilah, "Network Forensic: Analysis of Client Attack and Quality of Service Measurement by ARP Poisoning using Network Forensic Generic Process (NFGP) Model," *Sistemasi*, vol. 13, no. 2, p. 713, 2024, doi: 10.32520/stmsi.v13i2.3804.

S. Rohith Raj, R. Rohith, M. Moharir, and G. Shobha, "SCAPY-a powerful interactive packet manipulation program," *2018 Int. Conf. Networking, Embed. Wirel. Syst. ICNEWS 2018 - Proc.*, pp. 1–5, 2018, doi: 10.1109/ICNEWS.2018.8903954.