Analisis Efektivitas Integrasi Intrusion Detection Prevention System (IDPS) dan Software-Defined Networking dalam Keamanan Jaringan

Septya Handayani
Fakultas Informatika
Direktorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
septyahandayani@student.telkomuniversity.ac.id

Wahyu Adi Prabowo
Fakultas Informatika
Direktorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
wahyup@telkomuniversity.ac.id

Alon Jala Tirta Segara
Fakultas Informatika
Direktorat Kampus Universitas Telkom Purwokerto
Purwokerto, Indonesia
alonhs@telkomuniversity.ac.id

***Abstrak* — Serangan Distributed Denial of Service (DDoS) seperti *ICMP Flood* dan *SYN Flood* merupakan tantangan dalam menjaga ketersediaan dan kestabilan jaringan. *Software-Defined Network* (SDN) melakukan kontrol jaringan terpusat dan fleksibel, namun belum dilengkapi mekanisme keamanan untuk mendeteksi dan memblokir serangan secara otomatis. Untuk mengatasinya, diperlukan integrasi dengan sistem yang memantau trafik dan melakukan mitigasi ketika ancaman terdeteksi. Penelitian ini merancang sistem keamanan jaringan dengan mengintegrasikan SDN menggunakan *OpenDayLight* sebagai *controller*, Suricata sebagai sistem deteksi dan mitigasi, serta *OpenvSwitch* untuk pengelolaan trafik. Sistem diuji dan dievaluasi menggunakan parameter QoS meliputi *throughput*, *packet loss, delay* dan *jitter*. Hasil menunjukkan sistem berhasil mendeteksi dan memitigasi serangan DDoS otomatis. Pada serangan *ICMP Flood*, *throughput* turun dari 234.254 bit/s menjadi 117.743 bit/s, *delay* meningkat dari 1.746 ms menjadi 4.904 ms, *jitter* dari 1.158 ms menjadi 3.298 ms, dan *packet loss* tetap 0%. Pada serangan *SYN Flood*, *throughput* turun dari 410.315 bit/s menjadi 165.172 bit/s, *delay* dari 1.159 ms menjadi 3.298 ms, *jitter* dari 1.158 ms menjadi 3.298 ms, dan *packet loss* dari 0,00262% menjadi 0%. Sistem terbukti menjaga kestabilan jaringan dan mempertahankan QoS selama serangan berlangsung.**

***Kata kunci*— IDPS, SDN, DDoS, Quality of Service, keamanan jaringan**

I. PENDAHULUAN

Kemajuan teknologi komputer dan informasi telah mengubah cara komunikasi massa dengan dampak sosial dan politik yang signifikan. Di era *modern*, orang bergantung pada perangkat digital yang memungkinkan siaran terpadu berbasis teks, audio, dan video melalui berbagai aplikasi. Ketergantungan ini menimbulkan risiko baru bagi keamanan siber individu, kelompok, dan negara [1].

Berdasarkan data dari Badan Siber dan Sandi Negara (BSSN) trafik anomali serangan siber Indonesia tahun 2023 mencapai 403.990.813 anomali. Trafik anomali dapat mengakibatkan penurunan kinerja perangkat dan jaringan, pencurian informasi sensitif, serta merusak reputasi dan kepercayaan terhadap suatu organisasi [2].

Keamanan siber harus diprioritaskan karena ketergantungan manusia pada teknologi, khususnya internet, menciptakan celah baru bagi ancaman keamanan negara[3]. Kejahatan siber adalah serangkaian kejahatan terorganisir yang menyerang dunia maya dan keamanan siber. Kejahatan siber mencakup aktivitas kriminal yang dilakukan menggunakan komputer dan Internet, seperti mengakses komputer secara ilegal, mengakses sistem komputer yang mengirimkan data komputer, atau mengakses komputer melalui cara lain yang memungkinkan orang mengakses data komputer[3].

Contoh kasus *Distributed Denial of Service* (DDoS) yang pernah terjadi yaitu adalah serangan Mirai Botnet yang terjadi pada tahun 2016[4]. Kemudian serangan DDoS juga melakukan penyerangan kepada *GitHub* pada tahun 2018 dengan lalu lintas serangan mencapai 1,35 terabit per detik (Tbps)[5]. Serangan terbesar lainnya terjadi pada tahun 2020, di mana *Amazon Web Services* (AWS) mengalami serangan DDoS dengan puncak mencapai 2,3 Tbps, menjadikannya serangan terbesar dalam sejarah layanan AWS. Serangan DDoS yang dialami oleh AWS menggunakan *server web Connection-less Lightweight Directory Access Protocol* (CLDAP) [6].

Berdasarkan masalah keamanan siber diatas, diperlukan *pemanfaatan integrasi antara Software Defined Network dan Intrusion Detection Prevention System (IDPS)*. SDN mengatur jaringan melalui pengontrol terpusat yang berada di bidang kontrol, sementara bidang data yang terdiri dari *switch* akan meneruskan paket ke titik keluar yang ditentukan [7]. Untuk memastikan keamanan jaringan yang optimal, IDPS bertugas untuk memantau, mendeteksi, dan menghentikan ancaman serangan secara real-time serta memberikan peringatan kepada administrator tentang potensi serangan [8].

Salah satu ancaman siber yang umum terjadi yaitu *Distributed Denial of Service* (DDoS) yaitu serangan yang menghentikan layanan server secara terus-menerus secara permanen maupun sementara pada aplikasi atau *website* dengan cara membanjiri permintaan melalui *botnet* secara bersama yang dapat merusak sistem. Serangan ini menyebabkan layanan tidak dapat diakses oleh pengguna dan menimbulkan kerugian. Dua bentuk umum dari serangan ini adalah *ICMP Flood* dan *SYN Flood* yang menyerang pada *network layer* dan *transport layer*. Oleh karena itu, kedua

jenis serangan ini dipilih dalam penelitian sebagai skenario pengujian untuk dianalisis efektivitas integrasi SDN dan IDPS dalam mendeteksi serta mitigasi serangan[9].

Penelitian ini berfokus pada analisis efektivitas integrasi SDN dan IDPS dalam meningkatkan keamanan jaringan terhadap serangan DDoS. Dengan melakukan simulasi serangan *ICMP Flood* dan *SYN Flood* menggunakan *OpenDayLight* sebagai SDN *Controller d*an Suricata sebagai IDPS, penelitian ini bertujuan untuk mengevaluasi sejauh mana sistem dapat mendeteksi dan mencegah serangan yang terjadi. Hasil dari penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan strategi mitigasi serangan DDoS yang lebih optimal.

II. KAJIAN TEORI

A. *Software Defined Network* (SDN)

Software Defined Network (SDN) adalah paradigma baru dalam jaringan komputer yang memisahkan kontrol jaringan (*control plane*) dari perangkat keras yang menjalankan fungsi penerusan data (*data plane*). SDN memungkinkan konfigurasi dan manajemen jaringan yang lebih dinamis dan fleksibel [10].

B. *Intrusion Detection Prevention System* (IDPS)

Intrusion Detection Prevention System (IDPS) merupakan sistem yang berfungsi untuk mendeteksi dan mencegah aktivitas yang mencurigakan pada jaringan komputer ataupun pada sistem perangkat keras [11].

C. *Distributed Denial of Service* (DDoS)

Distributed Denial of Service (DDoS) merupakan upaya yang dilakukan untuk membuat *server* atau sistem jaringan mengalami gangguan hingga tidak dapat berfungsi dengan baik. Serangan ini dilakukan dengan membanjiri *server* atau target menggunakan paket data yang banyak atau permintaan secara terus-menerus sehingga *server* tidak dapat menangani lalu lintas jaringan dengan baik hingga *server* melambat atau hingga berhenti beroperasi [12].

Jenis-jenis serangan DDoS antara lain:

1. *UDP Flood*
   Serangan *UDP Flood* merupakan proses serangan DDoS yang menghancurkan sumber daya *host*, sehingga menyebabkan halaman *web* tidak dapat diakses [13].
2. *SYN Flood*
   Serangan *SYN Flood* yaitu serangan yang menggunakan permintaan palsu pada *server* dan menerima paket ACK dari *server.* Namun sambungan diterima akan diarahkan ke *timeout* daripada untuk menyelesaikan permintaan sehingga sumber daya *server* berkurang [13].
3. *ICMP Flood*
   Serangan *ICMP Flood* merupakan serangan yang bertujuan untuk membanjiri target dengan *request ICMP* secara cepat tanpa menunggu respons dari target. Jenis serangan ini akan menyebabkan sistem menjadi lambat pada target penyerangan [13].
4. *Ping of Death*
   Ping of Death merupakan serangan yang dikirim dengan menggunakan perintah *ping* tetapi mengandung sesuatu yang berbahaya ke komputer.

D. *Quality of Service* (QoS)

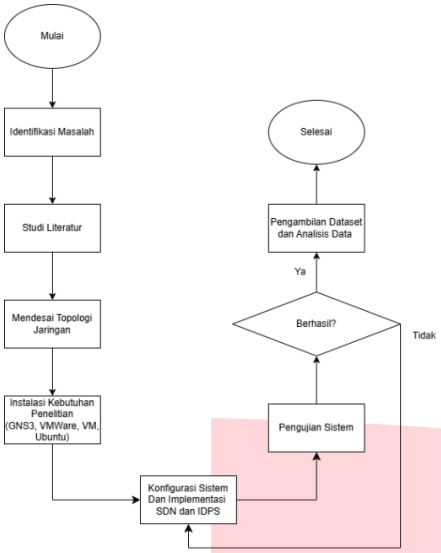
Quality of Service (QoS) merupakan penilaian kinerja layanan untuk mengukur tingkat kepuasan pengguna dalam sebuah layanan jaringan. Terdapat 4 kategori penilaian pada QoS untuk menilai sebuah layanan jaringan antara lain[14]:

1. *Throughput*
   Throughput merupakan kecepatan *transfer* data efektif yang dukur dalam satuan bps. Jumlah paket yang berhasil diamati pada tujuan dalam interval waktu tertentu dibagi dengan durasi interval waktu tersebut [14]. Penilaian *throughput* dapat dilakukan dengan persamaan berikut[14]:
   $$Throughput = \frac{Jumlah\ bit\ yang\ dikirim}{Total\ waktu\ pengiriman\ paket} \quad (1)$$
2. *Packet Loss*
   Packet loss merupakan jumlah data yang hilang karena paket yang hilang, yang biasanya disebabkan oleh kemacetan jaringan [14]. Penilaian *packet loss* dapat dilakukan dengan persamaan berikut[14]:
   $$Packet\ Loss = \frac{Paket\ Terkirim - Paket\ Diterima}{Paket\ Terkirim}\% \quad (2)$$
3. *Delay*
   Delay merupakan waktu yang dibutuhkan oleh paket untuk mencapai tujuan akhirnya Hal ini sering dipengaruhi oleh penundaan antrean, yang terjadi ketika banyak paket menunggu untuk dikirim karena kemacetan. Penilaian *delay* dapat dilakukan dengan persamaan berikut[14]:
   $$Delay = \frac{Total\ delay}{Total\ paket\ yang\ diterima}\ ms \quad (3)$$
4. *Jitter*
   Jitter merupakan kemacetan dapat menyebabkan paket datang terlambat atau tidak berurutan pada jaringan, yang dapat menyebabkan distorsi atau kesenjangan dalam transmisi. . Penilaian *delay* dapat dilakukan dengan persamaan berikut[14]:
   $$Jitter = \frac{Total\ variasi\ delay}{Total\ paket\ yang\ diterima} \quad (4)$$
   $$Total\ variasi\ delay = delay - (rata\ delay) \quad (5)$$

III. METODE

Penelitian ini bertujuan untuk mengetahui efektifivitas dari integrasi antara IDPS yang menggunakan *Suricata* dan SDN yang menggunakan *OpenDayLight* dalam menghadapi serangan DDoS yang meliputi *ICMP Flood* dan *SYN Flood*. Kemudian akan dianalisis dengan menggunakan parameter *QoS* yang meliputi *throughput, packet loss, delay* dan *jitter.*

A. Alur Penelitian



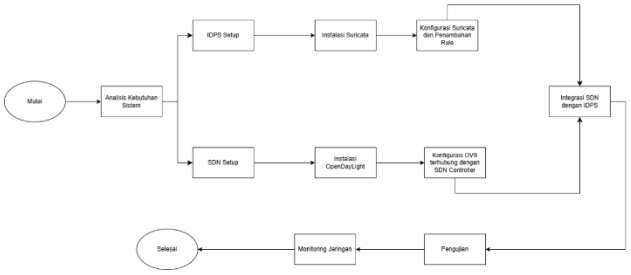
GAMBAR 1
(DIAGRAM ALIR PENELITIAN)

Penelitian ini diawali dengan melakukan identifikasi permasalahan dan studi literatur terkait SDN, IDPS serta mengenai keamanan jaringan. Selanjutnya berdasarkan dari identifikasi masalah, topologi jaringan dirancang yang mengintegrasikan *OpenDayLight* (ODL) sebagai *SDN controller, Suricata* sebagai sistem deteksi dan mitigasi atau sebagai IDPS serta *Open vSwitch* sebagai pengelola lalu lintas jaringan. Simulasi dilakukan dalam lingkungan virtual dengan menggunakan *GNS3*, *VMWare* dan *Ubuntu* yang terdiri dari empat komponen yaitu *attacker, server, controller* dan *client*.

Setelah dilakukan instalasi dan konfigurasi pada perangkat lunak, kemudian merupakan proses integrasi antara IDPS dan SDN. *Suricata* dikonfigurasi untuk melakukan pendeteksian pola serangan dan mengirim perintah ke *controller* agar dapat memblokir lalu lintas jaringan yang berbahaya melalui *OpenFlow*. Selanjutnya, sistem dilakukan pengujian dengan menggunakan dua skenario yaitu tanpa perlindungan IDPS dan dengan perlindungan IDPS terhadap serangan *ICMP Flood* dan *SYN Flood*.

Kemudian data akan dikumpulkan selama proses pengujian dan akan dianalisis dengan menggunakan parameter *Quality of Service* (QoS) yaitu *throughput, packet loss, delay* dan *jitter* untuk mengetahui efektivitas sistem dalam menjaga performa jaringan selama dilakukan pengujian dengan menggunakan serangan.

B. Konfigurasi Sistem



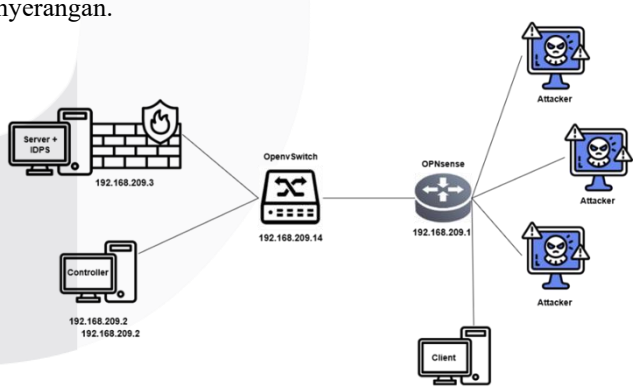
GAMBAR 2
(FLOWCHART KONFIGURASI SISTEM)

Proses konfigurasi dimulai dengan analisis kebutuhan yang mencakup perangkat lunak dan perangkat keras yang diperlukan. Kemudian sistem dibangun dibagi menjadi dua komponen utama yaitu IDPS dan SDN.

Pada *setup* IDPS dilakukan instalasi *suricata* sebagai sistem deteksi dan pencegahan dari lalu lintas jaringan yang berbahaya. Kemudian setelah dilakukan instalasi, *suricata* dikonfigurasi dengan aturan (*rules)* yang sesuai dengan deteksi serangan yang relevan yakni *ICMP Flood* dan *SYN Flood*.

Selanjutnya pada *setup SDN* dilakukan instalasi *OpenDayLight* sebagai *controller* dan melakukan konfigurasi *Open vSwitch* (OVS) agar jaringan dapat dikendalikan oleh *controller*. Setelah kedua sistem dikonfigurasi secara terpisah, kemudian dilakukan tahap integrasi agar *suricata* dapat mengirim *flow* ke *controller* untuk melakukan mitigasi secara otomatis ketika terdeteksi ancaman.

Setelah integrasi telah berhasil dikonfigurasi, sistem akan diuji dengan skenario serangan untuk memastikan deteksi dan mitigasi serangan berfungsi dengan benar. Pengujian ini dilanjutkan dengan proses monitoring jaringan secara *real-time* untuk mengevaluasi sistem. Tahapan ini bertujuan untuk memastikan bahwa sistem dapat menjaga kinerja jaringan secara berkelanjutan meskipun sedang mengalami penyerangan.



GAMBAR 3
(TOPOLOGI JARINGAN)

Gambar 3 merupakan topologi jaringan yang digunakan dalam penelitian ini. Topologi jaringan terdiri dari atas *controller* SDN dengan menggunakan *OpenDayLight*, *server* berfungsi sebagai sistem dan pencegahan dari aktivitas jaringan yang berbahaya dengan menggunakan *Suricata,* serta *Open vSwitch* digunakan sebagai pengelola lalu lintas jaringan. Kemudian terdapat *router* digunakan untuk mengatur konektivitas antarsegment jaringan. Selanjutnya terdapat empat komputer yang digunakan sebagai *attacker* dan *client*. Tiga komputer *attacker* digunakan untuk

melakukan penyerangan dengan menggunakan DDoS *ICMP Flood* dan *SYN Flood* untuk menguji ketahanan sistem. Sedangkan pada satu komputer *client* digunakan sebagai pengguna normal yang menggunakan layanan jaringan. Arsitektur ini dirancang agar mampu mendeteksi aktivitas jaringan yang berbahaya dengan memblokir secara otomatis aktivitas yang berbahaya secara *real-time* serta memblokit lalu lintas jaringan. Selain itu, penelitian ini dilakukan untuk mengetahui efektivitas dari integrasi antara SDN dan IDPS dalam menghadapi serangan *ICMP Flood* dan *SYN Flood*.

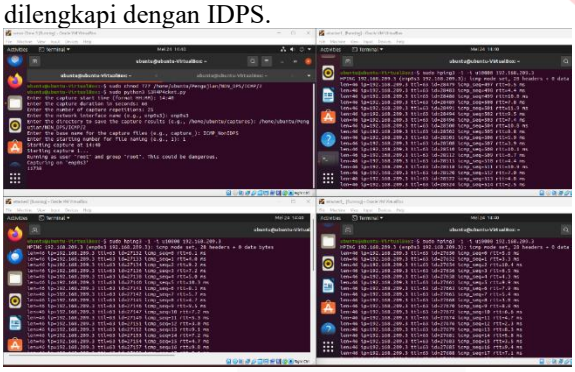
IV. HASIL DAN PEMBAHASAN

A. Skenario Penyerangan dan Pengujian Sistem

Pengujian yang dilakukan dalam penelitian yaitu dengan melakukan percobaan penyerangan yang dilakukan oleh tiga komputer *attacker* dengan target komputer *server*. Selain itu, pengujian dilakukan dengan empat kondisi dalam waktu masing-masing 25 menit dan kemudian akan dianalisis dampaknya dengan menggunakan parameter QoS.

1. Serangan *ICMP Flood* Tanpa Dilindungi IDPS

Pengujian yang dilakukan simulasi serangan *ICMP Flood* dari *attacker* ke *server* tanpa dilengkapi dengan IDPS.



GAMBAR 4 Pengujian *ICMP Flood* Tanpa Dilindungi IDPS

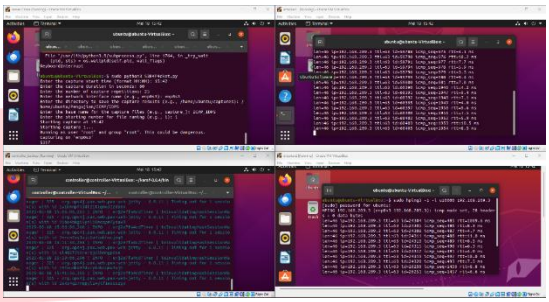
Gambar 4 menunjukkan proses pengujian serangan DDoS *ICMP Flood*. Terdapat tiga komputer *attacker* menghasilkan lalu lintas serangan secara terus-menerus ke target penyerangan yakni komputer *server*. Sedangkan pada komputer *server* melakukan dokumentasi menyimpan paket-paket jaringan secara otomatis dengan format *.pcap* sehingga *file* tersebut dapat digunakan untuk analisis lebih lanjut.

GAMBAR 5 *CAPTURE WIRESHARK* SERANGAN *ICMP FLOOD* TANPA MENGGUNAKAN IDPS

Gambar 5 menunjukkan hasil *capture* lalu lintas jaringan dengan menggunakan *wireshark* ketika pengujian dengan serangan *ICMP Flood* tanpa dilindungi IDPS. Terlihat bahwa terdapat IP *attacker* 192.168.89.2 dab 192.168.89.3 mengirim *ICMP Echo Request* secara terus-menerus ke *server* dengan IP 192.168.209.3. Setiap permintaan dibalas oleh *server* dengan *"Echo Reply"* seperti yang terlihat pada gambar 5. Pola trafik dengan interval sangat rapat ini menjadi indikasi kuat serangan *ICMP Flood* yang membebani sistem. Tanpa IDPS, serangan tidak terdeteksi maupun diblokir sehingga seluruh proses respons ditangani langsung oleh *server*.

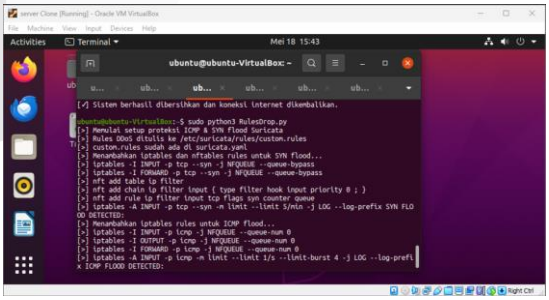
2. Serangan *ICMP Flood* Dilindungi IDPS

Pengujian yang dilakukan simulasi serangan *ICMP Flood* dari *attacker* ke *server* dilindungi dengan IDPS.



GAMBAR 6 PENGUJIAN *SERANGAN ICMP FLOOD* DILINDUNGI IDPS

Gambar 6 menujukan pengujian serangan *ICMP Flood* dalam lingkungan virtualisasi beberapa terminal dan *virtual machine* yang memiliki peran yang berbeda. Pengujian ini bertujuan untuk mengamati proses serangan yang dilakukan oleh *attacker,* respons sistem pada *OpenDayLight* serta perlindungan oleh *suricata* sebagai IDPS yang diletakan dalam *server*. Tampak proses pengambilan data trafik lalu lintas jaringan dalam format *.pcap* menggunakan *script python.* pada sisi *attacker*, perintah *hping3* dijalankan terus-menerus untuk mengirim serangan *ICMP Flood* ke target dengan IP 192.168.209.3 dengan *output* yang menampilkan *round-trip time* pada setiap paket. Pada gambar 6 juga menampilkan terminal *karaf* pada *controller OpenDayLight* yang menampilkan aktivitas pengelolaan sesi *web* yang menandakan *OpenDayLightaktif* dan siap menerima perintah mitigasi.

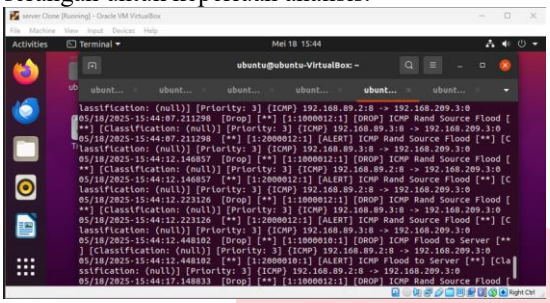


GAMBAR 7 EKSEKUSI *SCRIPT RULESDROP.PY* UNTUK PROTEKSI *DDOS ICMP FLOOD*

Gambar 7 memperlihatkan proses eksekusi *script Python RulesDrop.py* yang dijalankan pada *server* untuk mengaktifkan sistem proteksi terhadap serangan *DDoS ICMP Flood* dan *SYN Flood*. *Script* ini mengatur beberapa aturan *iptables* secara otomatis, seperti pengalihan paket *TCP* dan

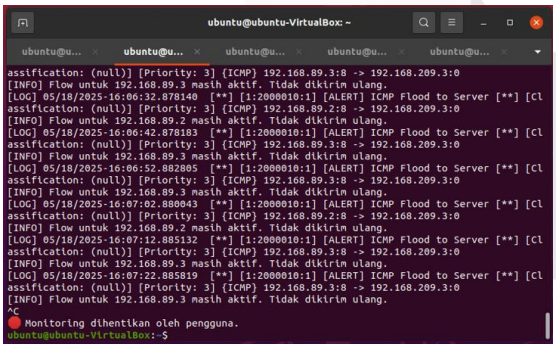
ICMP mencurigakan ke antrean *NFQUEUE*, yang kemudian dianalisis oleh *Suricata*.

Selain itu, terdapat pula konfigurasi batas laju (*rate limit*) untuk trafik *ICMP* menggunakan *opsi* --*limit* dan --*limit-burst*, serta pencatatan log dengan prefiks tertentu. Dengan konfigurasi ini, sistem mampu mendeteksi dan *drop* lalu lintas berbahaya secara otomatis, sekaligus mencatat aktivitas serangan untuk keperluan analisis.



GAMBAR 8 *FASTLOG* KETIKA SERANGAN *ICMP FLOOD* DENGAN IDPS

Log pada gambar 8 merupakan *output* dari *file fast.log* dari *suricata* ketika sistem menjalakan deteksi dan pencegahan dari serangan *ICMP Flood. Suricata* berhasil mendeteksi lalu lintas dari *ICMP Flood* yang mencurigakan dan mengklasifikasinya sebagai serangan *ICMP Rand Source Flood* yang berasal dari IP 192.168.89.2 dan 192.168.89.3 menuju *server* dengan IP 192.168.209.3. Log menunjukkan paket serangan berhasil diidentifikasi dan diblokir ditandai dengan status *alert* dan *drop* pada pesan yang berada dalam *fastlog*. Proses deteksi berlangsung secara terus-menerus dalam interval waktu yang rapat, mengindikasikan pola serangan *flood* yang konsisten. Sehingga berdasarkan gambar 8 tersebut membuktikan IDPS berjalan sesuai fungi dan mampu mengenali serangan, mencatat detail kejadian dan melakukan mitigasi secara otomatis.

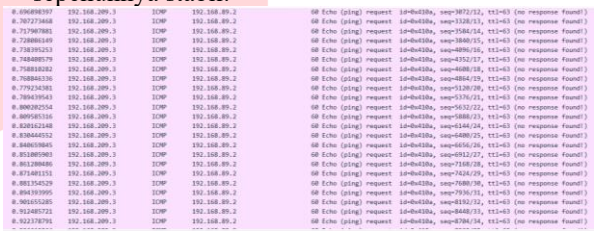


GAMBAR 9 DETEKSI *ICMP FLOOD* DAN PENGIRIMAN *FLOW DROP* KE *ODL*

Gambar 9 menampilkan hasil monitoring terminal saat sistem mendeteksi *serangan ICMP Flood* dan mencoba mengirimkan perintah *DROP* ke *controller OpenDaylight* (ODL). Log menunjukkan bahwa *Suricata* berhasil mengenali serangan yang berasal dari IP 192.168.89.3 dan 192.168.89.2 yang ditujukan ke *server* 192.168.209.3. Serangan diklasifikasikan sebagai "*ICMP Rand Source Flood*", dan sistem memberikan respons dengan mengirimkan aksi *[ACTION] DROP* ke *ODL*.

Namun, terlihat bahwa beberapa upaya pengiriman perintah *DROP* melalui *REST API* mengalami kegagalan, ditandai dengan *Error Connection timed out* ke IP 192.168.209.2 pada *port* 8181. Hal ini dapat terjadi akibat kepadatan trafik saat serangan berlangsung, yang menyebabkan terganggunya koneksi ke ODL. Meski demikian, terdapat indikasi bahwa sebagian perintah berhasil dikirim dan tereksekusi, ditandai dengan log *flow* berhasil dikirim ke ODL.

Log ini menunjukkan bahwa sistem IDPS telah mampu mendeteksi serangan *ICMP Flood* dan secara otomatis mencoba melakukan respons aktif melalui pengiriman *flow DROP* ke *controller SDN*, meskipun dalam kondisi jaringan yang tidak sepenuhnya stabil.

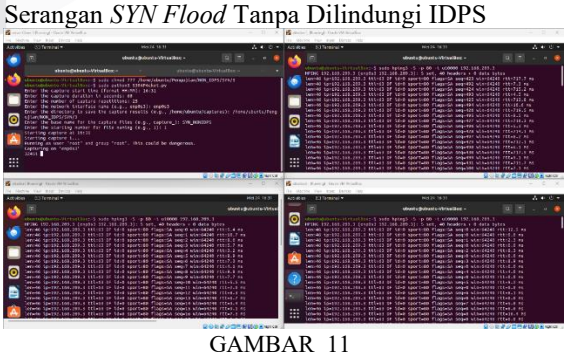


GAMBAR 10 *CAPTURE WIRESHARK* SERANGAN *ICMP FLOOD* DENGAN IDPS

Pada gambar 10 menunjukkan hasil *capture trafik ICMP* dengan menggunakan *wireshark* setelah sistem perlindungan IDPS diaktifkan. Dalam gambar tersebut, terlihat bahwa *attacker* dengan IP 192.168.89.2 secara terus-menerus mengirimkan *ICMP Echo Request* ke *server* target 192.168.209.3. Namun, tidak satu pun dari permintaan tersebut mendapatkan balasan, sebagaimana ditunjukkan oleh pesan "*no response found*" pada kolom paling kanan.

Setiap paket yang dikirim memiliki ukuran 60 *byte* dan TTL 63, dengan interval waktu yang sangat rapat, yang mengindikasikan bahwa serangan masih berlangsung dalam bentuk *flood*. Namun karena tidak ada respons dari sisi *server*, dapat disimpulkan bahwa sistem IDPS telah berhasil memblokir lalu lintas *ICMP* tersebut secara efektif, sehingga paket yang dikirim tidak mencapai atau tidak direspons oleh target.

3. Serangan *SYN Flood* Tanpa Dilindungi IDPS



GAMBAR 11
PENGUJIAN *SERANGAN SYN FLOOD* TANPA DILINDINGI IDPS

Gambar 11 menunjukkan pengujian yang dilakukan simulasi serangan *SYN Flood* dari *attacker* ke *server* tanpa dilengkapi dengan IDPS.

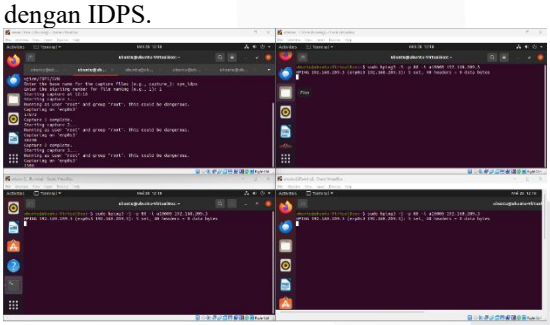
GAMBAR 12 *CAPTURE WIRESHARK* SERANGAN *SYN FLOOD* TIDAK MENGGUNAKAN IDPS

Gambar 12 merupakan hasil *capture wireshark* ketik dilakukan simulasi pengujian penyerangan *SYN Flood* terhadap *server* dengan alamat IP 192.168.209.3. Dalam *capture* terlihat adanya paket TCP yang tidak normal yang berasal IP 192.168.89.2 dan 192.168.209.3 sebagai *attacker*. Paket *SYN* tersebut tidak diikuti oleh proses *handshake* yang lengkap. Sebaliknya, muncul paket *RST (Reset)* yang mengindikasikan koneksi diputus secara sepihak. Pola ini menunjukkan bahwa *attacker* mencoba membanjiri *server* dengan koneksi palsu dalam jumlah besar.

Karena sistem IDPS belum diaktifkan, semua paket diterima tanpa penyaringan. Kondisi ini memperlihatkan bahwa server dalam keadaan tidak terlindungi dan rentan terhadap serangan *flood*, yang berpotensi menyebabkan gangguan atau penurunan performa layanan.

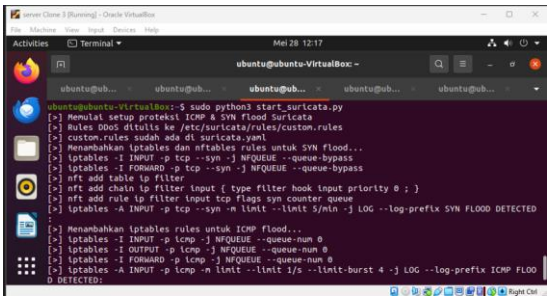
4. Serangan *SYN Flood* Dilindungi IDPS

Pengujian yang dilakukan simulasi serangan *SYN Flood* dari *attacker* ke *server* dilindungi dengan IDPS.



GAMBAR 13 PENGUJIAN SERANGAN *SYN FLOOD* DILINDUNGI IDPS

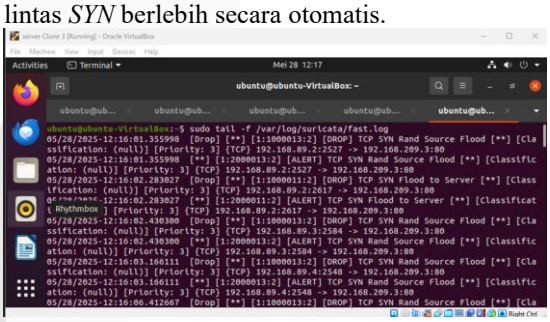
Gambar 13 menunjukkan pengujian serangan *SYN Flood* dengan IDPS aktif dengan menggunakan *Suricata* melalui *NFQUEUE* dan *iptables* untuk melakukan memfilterkan paket yang masuk. Pada komputer *server*, *script python* dijalankan untuk merekam semua aktivitas jaringan. Kemudian untuk tiga terminal lainnya, terlihat *attacker* sedang menjalankan perintah *hping3* secara bersamaan untuk mengirim paket *TCP* dengan *flag SYN* ke *server* dengan IP 192.168.209.3 dengan interval yang tinggi menggunakan *-i u10000*. Pengujian ini menyimulasikan serangan DDoS *SYN Flood*.



GAMBAR 14 EKSEKUSI *SCRIPT START_SURICATA.PY* UNTUK PROTEKSI DDOS *ICMP FLOOD*

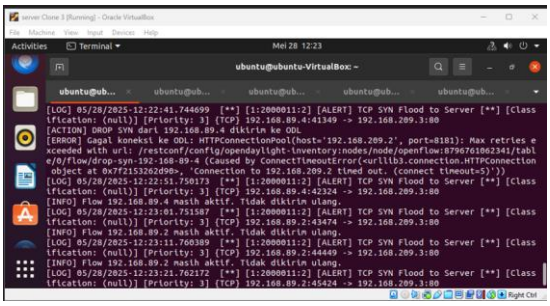
Gambar 14 memperlihatkan bagian dari proses eksekusi *script python* start_suricata.py yang secara otomatis mengatur sistem untuk mendeteksi dan memblokir serangan *SYN Flood*. Dalam bagian ini, *script* menggunakan kombinasi dari *iptables*, *NFQUEUE* dan *nftables*. *Rules* pertama menggunakan *iptables* untuk mengarahkan semua paket *TCP* dengan *flag SYN* ke antrean *NFQUEUE*, agar dapat dianalisis oleh *suricata*. Opsi *--queue-bypass* digunakan agar sistem tetap berjalan meski *Suricata* tidak aktif. Selanjutnya, konfigurasi *nftables* dilakukan dengan membuat tabel dan rantai baru untuk memantau lalu lintas *TCP SYN*. Paket yang sesuai akan dicatat menggunakan fitur *counter* sebagai dasar pemantauan intensitas koneksi.

Sebagai perlindungan tambahan, diterapkan juga *rate limiting* melalui *iptables*, yaitu maksimal lima koneksi *SYN* per menit. Jika melebihi, sistem akan mencatatnya ke log dengan label "*SYN FLOOD DETECTED*". Konfigurasi ini memastikan sistem mampu mendeteksi, mencatat, dan membatasi lalu lintas *SYN* berlebih secara otomatis.



GAMBAR 15 *FASTLOG* KETIKA SERANGAN *SYN FLOOD* DENGAN IDPS

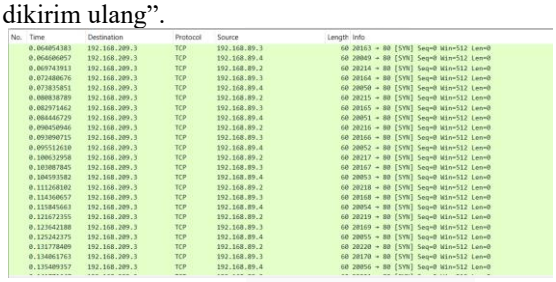
Gambar 15 menampilkan *fast.log* dari *suricata* yang dijalankan selama pengujian serangan *SYN Flood*. Dari tampilan log terlihat bahwa suricata berhasil mendeteksi pola *serangan TCP SYN Rand Source Flood* yang berasal dari IP attacker 192.168.89.2 dan 192.168.89.3, dengan target *server* 192.168.209.3 pada *port* 80. Setiap entri log menunjukkan bahwa paket-paket tersebut tidak hanya dikenali sebagai serangan, tetapi juga secara aktif diblokir, ditandai dengan label *[Drop]*. Suricata mencatat dan memberikan informasi jenis ancaman melalui label seperti *"TCP SYN Flood to Server"* dan *"TCP SYN Rand Source Flood"*.



GAMBAR 16 DETEKSI *SYN FLOOD* DAN PENGIRIMAN *FLOW DROP* KE ODL

Gambar 16 menunjukkan proses deteksi respons terhadap *TCP SYN Flood* oleh *suricata* yang terintegrasi dengan *controller OpenDayLight* (ODL). Dari log terlihat bahwa *suricata* mendeteksi serangan *SYN* dari IP 192.168.89.4 ke *server* dengan alamat IP 192.168.209.3 dan langsung mencatatnya sebagai *[ALERT] TCP SYN Flood to Server*.

Sistem kemudian mencoba mengirim *flow DROP* ke *ODL* melalui *REST API* agar serangan dapat diblokir langsung pada level jaringan. Namun, terlihat beberapa kegagalan koneksi ke ODL (*Connection timed out*) karena keterlambatan respons *REST API*. Tetapi log menunjukkan bahwa ada juga *flow* yang berhasil dikirim, ditandai dengan pesan "*Flow* 192.168.89.x masih aktif. Tidak dikirim ulang".

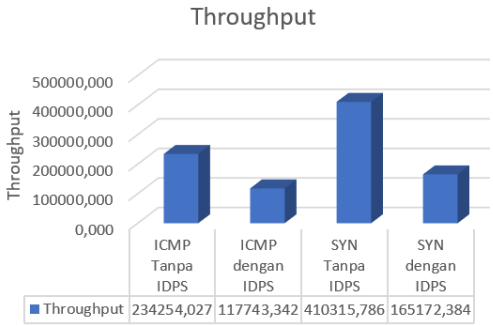


GAMBAR 17 *CAPTURE WIRESHARK* SERANGAN *SYN FLOOD* DENGAN IDPS

Gambar 17 menunjukkan hasil *capture wireshark* saat serangan *SYN Flood* dilakukan ke server 192.168.209.3 oleh *attacker* 192.168.89.3 dan 192.168.89.4, setelah IDPS yaitu *suricata* telah aktif. Terlihat bahwa *attacker* terus mengirimkan paket *TCP* dengan *flag [SYN]* dalam interval waktu yang sangat rapat. Namun tidak terlihat adanya respons dari *server*, seperti *ACK* atau *RST*, yang mengindikasikan bahwa paket-paket tersebut tidak diterima atau diblokir sebelum mencapai target penyerangan.

B. Pengukuran *Quality of Service* (QoS)

1. *Througput*



GAMBAR 18 PERBANDINGAN *THROUGHPUT* PADA SETIAP PENGUJIAN

Grafik yang ditunjukkan pada gambar 18 menunjukkan perbandingan nilai *throughput* dari dua jenis serangan yaitu *ICMP Flood* dan *SYN Flood* dalam dua kondisi yaitu dengan dilindungi oleh IDPS dan dengan tidak dilindungi oleh IDPS. Secara umum, terlihat nilai *throughput* memiliki nilai yang lebih tinggi ketika kondisi tidak dilindungi oleh IDPS dibandingkan ketika dalam kondisi dilindungi oleh IDPS dalam kondisi serangan *ICMP Flood* maupun serangan *SYN Flood*. Hal ini menunjukkan bahwa ketika tidak ada sistem perlindungan dan menyaring lalu lintas jaringan, maka seluruh lalu lintas jaringan termasuk dengan paket yang berbahaya dapat melewati jaringan tanpa hambatan, sehingga nilai *throughput* menjadi tinggi.

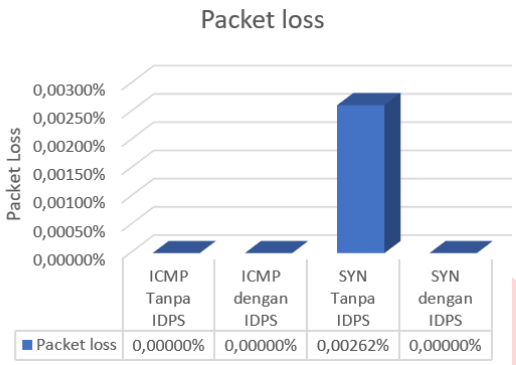
Pada serangan *ICMP Flood*, nilai rata-rata *throughput* dengan tidak ada perlindungan IDPS mencapai 234.254,027 bit/s. Sedangkan ketika serangan *ICMP Flood* dengan dilindungi oleh IDPS, nilai rata-rata *throughput* turun menjadi 117.743,342 bit/s. Penurunan ini menggambarkan bahwa IDPS bekerja efektif menyaring sebagian besar paket *ICMP* sehingga hanya sebagian yang diteruskan ke target penyerangan. Dengan kata lain, penurunan *throughput* pada kondisi dengan menggunakan IDPS terjadi karena sebagian besar lalu lintas yang berbahaya telah di drop sistem keamanan sehingga hanya paket uang dianggap sah yang diteruskan oleh jaringan.

Sementara itu, pada serangan *SYN Flood*, *throughput* tanpa menggunakan perlindungan dari IDPS tercatat jauh lebih tinggi yaitu dengan nilai rata-rata *throughput* 410.315,786 bit/s dan menurun menjadi 165.172.384 ketika perlindungan IDPS diaktifkan. Perbedaan yang sangat signifikan ini menunjukkan bahwa *SYN Flood* menghasilkan volume lalu lintas yang jauh lebih besar dibanding *ICMP Flood*, serta lebih berdampak terhadap sistem jika tidak difilter. Namun, kembali terlihat bahwa IDPS mampu menahan sebagian besar lalu lintas *SYN* tersebut, sehingga nilai *throughput* menjadi lebih terkendali.

Sehingga dari grafik pada gambar 18 dapat disimpulkan bahwa dengan mengaktifkan IDPS

dapat menyebabkan penurunan nilai *throughput* secara konsisten untuk kedua jenis serangan yang diujikan. Penurunan nilai *throughput* mencerminkan keberhasilan sistem dalam menyaring lalu lintas jaringan dan menjaga stabilitas jaringan.

2. *Packet Loss*



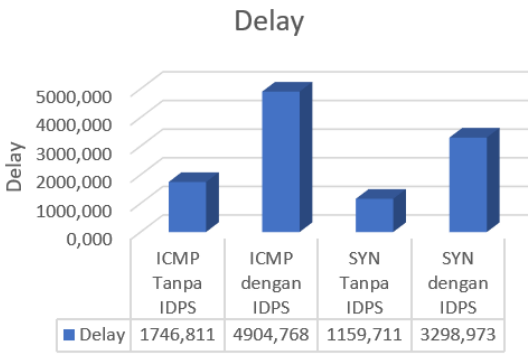
GAMBAR 19 PERBANDINGAN *PACKET LOSS* PADA SETIAP PENGUJIAN

Pada gambar 19 menunjukkan perbandingan nilai *packet loss* selama pengujian dalam empat kondisi yang berbeda yaitu ketika serangan *ICMP Flood* tanpa adanya perlindungan IDPS, *ICMP Flood* dengan adanya perlindungan IDPS, *SYN Flood* tanpa adanya perlindungan IDPS dan *SYN Flood* dengan adanya perlindungan IDPS. Dari grafik dapat dilihat bahwa nilai *packet loss* sebesar 0,00262% hanya muncul pada skenario serangan *SYN Flood* tanpa IDPS, sedangkan pada ketiga kondisi lainnya, termasuk kedua skenario *ICMP Flood* dan *SYN Flood* dengan IDPS, nilai *packet loss* tercatat 0% secara konsisten.

Pada kondisi serangan *ICMP Flood* yang menggunakan perlindungan IDPS maupun yang tidak menggunakan IDPS menghasilkan nilai *packet loss* 0%. Hal ini terjadi karena jenis serangan *ICMP* tidak menyebabkan penumpukan koneksi. Meskipun serangan ini dilakukan secara terus-menerus, *server* dan jaringan masih mampu menampung lalu lintas ICMP tanpa kehilangan paket.

Sedangkan pada kondisi serangan *SYN Flood* tanpa adanya perlindungan dengan IDPS terjadi *packet loss* karena serangan tersebut berhasil membanjiri *server* dengan koneksi *SYN* sehingga sumber daya sistem penuh. Namun ketika IDPS diaktifkan pada skenario *SYN Flood, packet loss* kembali ke 0%. Hal ini menunjukkan bahwa IDPS berhasil mencegah sebagian besar lalu lintas *SYN* berbahaya masuk ke sistem, sehingga beban terhadap server berkurang drastis, dan tidak terjadi penolakan koneksi akibat kehabisan sumber daya.

3. *Delay*



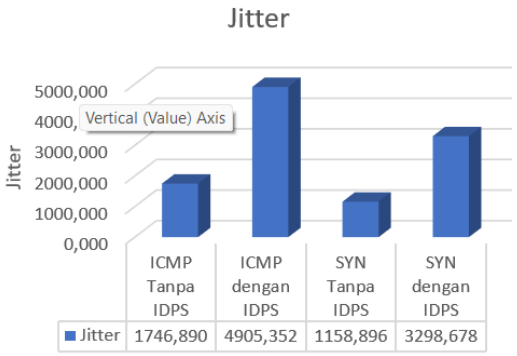
GAMBAR 20 PERBANDINGAN *DELAY* PADA SETIAP PENGUJIAN

Gambar 20 merupakan perbandingan nilai rata-rata *delay* dari empat skenario pengujian yaitu pada kondisi penyerangan dengan *ICMP Flood* yang sistem IDPS tidak aktif, *ICMP Flood* yang sistem IDPS aktif, *SYN Flood* yang sistem IDPS tidak aktif serta *SYN Flood* yang sistem IDPS aktif. Dari hasil yang ditampilkan nilai Dari hasil yang ditampilkan, terlihat bahwa nilai *delay* tertinggi terjadi pada skenario *ICMP Flood* dengan IDPS, yaitu sebesar 4904,768 ms, disusul oleh *SYN Flood* dengan IDPS sebesar 3298,973 ms. Sedangkan *delay* paling rendah terdapat pada *SYN Flood* tanpa IDPS, yaitu 1159,711 ms, dan *ICMP Flood* tanpa IDPS sebesar 1746,811 ms.

Nilai *delay* yang lebih tinggi pada kedua skenario dengan IDPS menunjukkan bahwa mengaktifkan IDPS memang berdampak langsung terhadap peningkatan waktu tunda. Hal ini terjadi karena IDPS melakukan pengecekan mendalam terhadap setiap paket data, sehingga menambahkan proses sebelum paket diteruskan ke tujuan. Proses filtering ini meningkatkan *delay* sebagai konsekuensi dari perlindungan sistem terhadap lalu lintas berbahaya.

Selain itu, dapat diamati bahwa serangan ICMP cenderung menghasilkan *delay* yang lebih tinggi dibanding *SYN*, baik dengan maupun tanpa IDPS. Hal ini disebabkan oleh karakteristik *ICMP Flood* yang mengirimkan paket secara cepat dan terus-menerus, yang menekan *resource* sistem dan menyebabkan antrean lebih panjang, sehingga waktu pemrosesan meningkat. Sebaliknya, meskipun *SYN Flood* menyerang koneksi *TCP*, server masih dapat mengelola permintaan tersebut dengan relatif efisien selama belum mencapai batas koneksi maksimum.

4. *Jitter*



GAMBAR 21 PERBANDINGAN *JITTER* PADA SETIAP PENGUJIAN

Gambar 21 merupakan perbandingan nilai rata-rata *jitter* dari empat skenario pengujian yaitu pada kondisi penyerangan dengan *ICMP Flood* yang sistem IDPS tidak aktif, *ICMP Flood* yang sistem IDPS aktif, *SYN Flood* yang sistem IDPS tidak aktif serta *SYN Flood* yang sistem IDPS aktif. Berdasarkan grafik, dapat dilihat bahwa rata-rata nilai *jitter* tertinggi terjadi pada kondisi *ICMP Flood* dengan IDPS, yaitu sebesar 4905,352 ms. Sebaliknya, nilai *jitter* terendah ditemukan pada *SYN Flood* tanpa IDPS, yakni 1158,896 ms. Sementara itu, kondisi *ICMP Flood* tanpa IDPS mencatat nilai *jitter* sebesar 1746,890 ms, dan *SYN Flood* dengan IDPS sebesar 3298,678 ms.

Secara umum, dapat disimpulkan bahwa aktifnya IDPS menyebabkan kenaikan nilai *jitter* pada kedua jenis serangan. Peningkatan ini terjadi karena setiap paket yang masuk harus melalui proses inspeksi terlebih dahulu sebelum diteruskan ke tujuan. Proses ini menambah waktu antar kedatangan paket, sehingga menghasilkan *jitter* yang lebih tinggi.

Sebaliknya, pada kondisi tanpa IDPS, meskipun lalu lintas serangan masuk langsung ke server tanpa penyaringan, nilai *jitter* justru lebih rendah. Hal ini menunjukkan bahwa lalu lintas mengalir secara langsung tanpa hambatan tambahan, sehingga waktu antar pengiriman paket cenderung lebih konsisten.

Selain itu, perbandingan antara *ICMP* dan *SYN Flood* juga menunjukkan bahwa serangan *ICMP* cenderung menghasilkan *jitter* yang lebih tinggi dibanding *SYN Flood*. Hal ini dapat dijelaskan oleh karakteristik serangan *ICMP Flood* yang lebih padat dan konstan, sehingga berpotensi menimbulkan ketidakteraturan waktu pengiriman lebih besar dibanding serangan *SYN Flood*.

V. KESIMPULAN

Integrasi antara SDN dengan IDPS yang dibangun berhasil mendeteksi dan memblokir serangan secara otomatis sesuai dengan *rules* yang ditentukan dan hasil penelitian menujukan bahwa penilaian *QoS* terjaga. Pada serangan *ICMP Flood*, sistem IDPS–SDN menyebabkan *throughput* menurun dari 234.254 bit/s menjadi 117.743 bit/s karena sebagian besar trafik serangan berhasil diblokir. *Delay* dan *jitter* meningkat dari sekitar 1.746 ms menjadi 4.904 ms akibat pemrosesan lalu lintas serta pengiriman instruksi pemblokiran ke sistem SDN. *Packet loss* tetap 0%, menunjukkan tidak ada kehilangan paket selama proses proteksi. Pada serangan *SYN Flood*, *throughput* turun dari 410.315 bit/s menjadi 165.172 bit/s, *delay* meningkat dari 1.159 ms menjadi 3.298 ms, dan jitter dari 1.158 ms menjadi 3.298 ms. *Packet loss* yang awalnya 0,00262% berhasil ditekan menjadi 0%, menandakan sistem mampu menjaga kestabilan jaringan secara optimal selama serangan berlangsung. Hasil ini menunjukkan bahwa sistem mampu merespons serangan secara efektif, menjaga kestabilan jaringan, dan mempertahankan *QoS* selama serangan berlangsung.

REFERENSI

[1] K. Surya, N. Kepala, S.-B. Evaluasi, M. Diklat, P. Bahasa, and B. Kemhan, "Keamanan dan Ancaman Cyber Bagi Sektor Privat dan Industry Militer Di Era 4.0 (Security and Cyber Threats for the Private Sector and Military Industry in the Era 4.0)," *Jurnal Diplomasi Pertahanan*, vol. 7, no. 1, p. 2021.

[2] "LANSKAP KEAMANAN SIBER INDONESIA," Jakarta, 2023. Accessed: Apr. 27, 2024. [Online]. Available: https://www.bssn.go.id/wp-content/uploads/2024/03/Lanskap-Keamanan-Siber-Indonesia-2023.pdf

[3] Sunil. C. Pawar, R. S. Mente, and Bapu. D. Chendage, "Cyber Crime, Cyber Space and Effects of Cyber Crime," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 210–214, Feb. 2021, doi: 10.32628/cseit217139.

[4] G. Bastos *et al.*, "Identifying and Characterizing Bashlite and Mirai CC Servers," *Proc IEEE Symp Comput Commun*, vol. 2019-June, 2019, doi: 10.1109/ISCC47284.2019.8969728.

[5] H. A. Sidharta, "GitHub mengalami serangan DDOS 1.35 terabits per second," Binus University. Accessed: May 16, 2024. [Online]. Available: https://binus.ac.id/malang/2018/07/github-mengalami-serangan-ddos-1-35-terabits-per-second/

[6] "Famous DDoS attacks: The largest DDoS attacks of all time," cloudfare.com. Accessed: May 16, 2024. [Online]. Available: https://www.cloudflare.com/en-gb/learning/ddos/famous-ddos-attacks/

[7] E. Tan, Y. W. Chong, and M. F. R. Anbar, "Flow management mechanism in software-defined network," *Computers, Materials and Continua*, vol. 70, no. 1, pp. 1437–1459, 2021, doi: 10.32604/cmc.2022.019516.

[8] S. Thapa and A. Mailewa, "EasyChair Preprint The Role of Intrusion Detection/Prevention Systems in Modern Computer Networks: A Review," 2020.

[9] S. Parulian, D. A. Pratiwi, and M. Cahya Yustina, "Ancaman dan Solusi Serangan Siber di Indonesia," *Telecommunications, Networks, Electronics, and Computer Technologies (TELNECT)*, vol. 1, no. 2, pp. 85–92, 2021, [Online]. Available: http://ejournal.upi.edu/index.php/TELNECT/

[10] D. Kreutz, F. M. V. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig,

"Software-defined networking: A comprehensive survey," in *Proceedings of the IEEE*, Institute of Electrical and Electronics Engineers Inc., Jan. 2015, pp. 14–76. doi: 10.1109/JPROC.2014.2371999.

[11] H. Azam *et al.*, "Defending the Digital Frontier: IDPS and the Battle Against Cyber Threat," *International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence*, vol. 2, no. 1, 2023, doi: 10.54938/ijemdcsai.2023.02.1.253.

[12] M. Zidane, "Klasifikasi Serangan Distributed Denial-of-Service (DDoS) menggunakan Metode Data Mining Naïve Bayes," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 1, pp. 172–180, 2022, [Online]. Available: http://j-ptiik.ub.ac.id

[13] Jagoan Hosting Team, "Apa itu DDoS Attack? Jenis, Ciri, dan Cara Mencegahnya," Jagoan Hosting. Accessed: Jun. 06, 2024. [Online]. Available: https://www.jagoanhosting.com/blog/ddos-adalah/

[14] N. Khaerani Hamzidah *et al.*, "SISTEMASI: Jurnal Sistem Informasi Studi Komparatif QoS pada Aplikasi Video Meeting Tool dalam Jaringan 4G LTE Menggunakan Wireshark Comparative Study of QoS on Video Meeting Tool Application in 4G LTE Network Using Wireshark." [Online]. Available: http://sistemasi.ftik.unisi.ac.id