Jurnal Sains Informatika Terapan (JSIT)
E-ISSN: 2828-1659, Vol. 2, No. 2, Bulan Juni, Tahun 2023
Hal: 74 - 80, Available online at: https://rcf-indonesia.org/home/

IMPLEMENTASI INTRUSION DETECTION PREVENTION SYSTEM SEBAGAI SISTEM KEAMANAN JARINGAN KOMPUTER KEJAKSAAN NEGERI PARIAMAN MENGGUNKAN SNORT DAN IPTABLES BERBASIS LINUX

Hasri Awal[1)], Aggy Pramana Gusman[2)]
[12]Universitas Putra Indonesia YPTK Padang
Corresponding Author: [2] hasriawal@upiyptk.ac.id

Article Info

Article history:

Received: April 10, 2023
Revised: May 05, 2023
Accepted: June 02, 2023

Keywords:

Keamana jaringan
IDS
Snort
Linux
DoS
Port Scanning

ABSTRACT

Perkembangan jaringan komputer terus berlanjut, dalam skalabilitas, jumlah node, dan teknologi. Komputer yang terhubung ke jaringan berpotensi mengalami gangguan atau serangan. Maka dari itu keamanan jaringan sangat penting dalam sebuah sistem jaringan komputer untuk menghindari serangan dan melindungi jaringan komputer. *Intrusion Detection System* (IDS) dengan *Snort* yang diimplementasikan pada sistem operasi *linux* dapat melakukan pemantauan serangan DoS (*Denial of Service*) dan *Port Scanning*. *Snort mode* IDS akan memberi *alert* secara *real-time* sesuai dengan *rules Snort* yang diatur dalam *local.rules*. *IPTables* sebagai *tools* IPS akan menghentikan serangan/gangguan tersebut dengan *rules IPTables* yang diterapkan. Dalam penelitian ini dilakukan pengujian sistem *Snort* IDS, *IPTables* dan pengujian kualitas layanan *server*. Hasil pengujian Snort IDS dapat memberikan alert bahwa adanya serangan secara *real-time*. Hasil pengujian IPS dapat mengatasi serangan/gangguan yang masuk dengan memblokir alamat IP intruder. Pengujian kualitas layanan server setelah diterapkan IDPS nilai index yang diperoleh adalah 3,75 yang sebelumnya kualitas layanan *server* memiliki nilai index 2. Yang artinya IDPS mampu mengatasi serangan/gangguan yang masuk ke jaringan.

1. PENDAHULUAN

Perkembangan jaringan komputer terus berlanjut, dalam skalabilitas, jumlah node, dan teknologi.

Jaringan komputer merupakan interkoneksi dua atau lebih komputer dengan media transmisi *wired* atau *wireless*. Istilah *client-server* umum digunakan dalam jaringan komputer. Client adalah pihak yang meminta/menerima layanan, sedangkan server adalah pihak yang menyediakan/mengirim layanan[1]. Perangkat dengan transmisi wireless memungkinkan pengiriman informasi antar host tanpa kabel menggunakan gelombang elektromagnetik[2].

Kantor kejaksaan yang menjadi objek penelitian penulis merupakan lembaga pemerintah yang memiliki infrastruktur jaringan komputer dan *server*. Pada objek penelitian ini terdapat permasalahan yaitu tidak adanya pemantauan jaringan komputer. Ketika gangguan/serangan masuk ke *server* yang berada di kejaksaan, administrator tidak mengetahui jenis gangguan/serangan apa yang masuk ke *server*. Serangan/gangguan yang masuk tanpa sepengetahuan dan tidak segera dilakukan penanganan untuk menghentikan gangguan/serangan yang masuk maka dapat mengakibatkan kerusakan.

Serangan atau gangguan biasa terjadi pada komputer yang terhubung ke jaringan adalah serangan DoS dan *port scan*. DoS serangan yang berasal dari satu perangkat, sedangkan DDoS serangan yang lebih dari satu perangkat. DoS dan DDoS sama-sama serangan *traffic flooding* yang menggunakan paket data besar yang dapat membebani dan memblokir akses ke server. Serangan yang umumnya digunakan adalah UDP Flooding, SYN Flooding, dan *Ping of Death*[3]. *Port scan* serangan yang dilakukan dengan cara memindai *port* jaringan target, menganalisa *port* jaringan target kemudian mencari celah pada port target yang terbuka[4]. Maka dari itu keamanan jaringan sangat penting dalam sebuah sistem jaringan komputer untuk menghindari serangan dan melindungi jaringan komputer dari acaman jaringan luar maupun dalam.

Dari permasalahan tersebut, Alternatif pemecahan permasalahan adalah dengan menerapkan keamanan jaringan menggunakan metode IDPS menggunakan *Snort* dan *IPTables*. *Snort* IDS dapat mendeteksi serangan dan IPS *IPTables* dapat melakukan tindakan penyaringan dengan cara menginputkan alamat IP penyerang[5].

2. METOLOGI PENELITIAN

Metode penelitian yang dilakukan yaitu mengimplementasikan Intrusion Detection Prevention System IDPS menggunakan *Snort* dan *IPTables*.

Intrusion Detection System (IDS) merupakan sebuah aplikasi perangkat lunak atau perangkat keras yang mampu mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. Apabila aktivitas mencurigakan terdeteksi pada *traffic jaringan*, *Intrusion Detection System* (IDS) akan memperingati sistem atau administrator[6]. Sementara itu, fungsi utama *Intrusion Prevention System* (IPS) adalah menghentikan serangan yang sedang berlangsung[7].

Snort merupakan *tools* (alat) paket instalasi sistem *linux* yang dapat mendeteksi penyusup, menganalisa paket secara *real-time*, dan menyimpan *file log* ke *database*. *Snort* merupakan contoh IDS dalam kategori NIDS yang mendeteksi *intrusion* di sistem jaringan. *Snort* dapat bekerja sebagai *packet-logger* untuk mencatat *traffic* jaringan dan memberi peringatan, serta sebagai *packet sniffe*r untuk membaca *traffic* jaringan. Snort digunakan

sebagai alat pendeteksi dan pencegah terindikasi sebuah paket data di *traffic* jaringan sebagai *threats* atau ancaman. *Snort* juga memiliki aturan layaknya *firewall* sebagai pendeteksi ancaman dalam jaringan. Penerapan aplikasi *Snort* menggunakan *rules* set yang memungkinkan sistem *linux* untuk mendeteksi dan memberikan peringatan terhadap pola serangan dari *attacker*[8].

IPTables merupakan *tools* yang berfungsi sebagai penyaringan atau pengatur lalu lintas data di sistem operasi *linux*. *IPTables* memiliki tiga jenis aturan dalam tabel *filter*, yaitu *firewall chain*. Terdapat tiga *chain* yaitu *INPUT*, *OUTPUT*, dan *FORWARD*. *IPTables* memiliki tiga tabel yaitu, *NAT, MANGLE*, dan *FILTER*. *Filter* berfungsi sebagai penyaring paket data, seperti *DROP, LOG, ACCEPT* atau *REJECT*. NAT berfungsi sebagai pengganti alamat asal atau tujuan dari paket data. Mangle berfungsi pemghalusan paket data seperti TTL, TOS, dan MARK. RAW digunakan untuk konfigurasi pengecualian dari connection tracking dengan NOTRACK[7].

Linux adalah sistem operasi *Open Source* berbasis GNU/Linux dengan berbagai varian seperti Slackware, Linux Mint, Debian, Open Suse, Archlinux, Redhat, dan perangkat lunak Open Source lainnya. Banyak varian GNU/Linux hanya menyediakan aplikasi tertentu yang mungkin kurang bermanfaat bagi pengguna. Ini mengakibatkan banyak pengguna melakukan remastering untuk memenuhi kebutuhannya[9]. Pemasangan Linux dilakukan di dalam aplikasi virtualbox untuk meminimalisir resiko kegagalan. VirtualBox itu sendiri merupakan sebuah program untuk virtualisasi komputer pada komputer desktop, server, dan laptop. Dapat memvirtualisasikan sistem operasi 32-bit dan 64-bit pada komputer dengan prosesor Intel dan AMD baik di perangkat lunak maupun perangkat keras. Virtualbox merupakan perangkat lunak virtualisasi gratis dan open source yang menyediakan kemudahan dan kemampuan untuk membuat mesin virtual secara native[10].

3. HASIL DAN PEMBAHASAN

Implementasi dilakukan dengan melakukan instalasi pada *server* IDPS menggunakan *Linux Mint*, pemasangan perangkat lunak *Snort*, konfigurasi *Snort* dan *rules Snort*. Untuk mengatasi serangan/gangguan konfigurasi aturan IPTables yang merupakan *tools* IPS. Setelah sistem berbasil terpasang secara keseluruhan, selanjutnya melakukan pengujian, bertujuan untuk membuktikan bahwa sistem yang diterapkan dapat bekerja dengan baik. Pengujian yang akan penulis coba lakukan untuk menguji sistem keamanan adalah melakukan serangan *SYN Flood* dan *Scan Port.*

A. Pengujian IDS Dengan Serangan *SYN Flood*

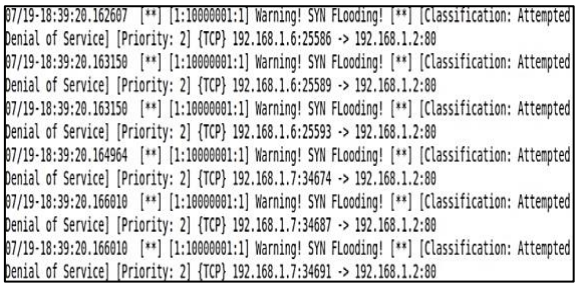
Pada tahapan ini penulis mencoba melakukan penyerangan *SYN Flood* melalui terminal *Backtrack 5*. *SYN Flood* ini merupakan salah satu serangan *Denial of Service* (DoS) maupun *Distributed Denial 0f Service* (DDoS) dimana penyerangan ini bertujuan mengkonsumsi sumber daya dari *server* sehingga *server* tidak dapat melayani lalu lintas jaringan yang memang benar-benar sah.

Berikut ini merupakan tampilan serangan/gangguan yang dilakukan menggunakan *Backtrack 5*:

Gambar 1 Serangan *SYN Flood* dari *Backtrack* 5 (1)

Gambar 2 Serangan SYN Flood dari Backtrack 5 (2)

Pada gambar 1 dan 2 serangan *Syn Flood* dilakukan dengan perintah *hping3 -i u1 -S -p 80 192.168.1.2*, yang dilancarkan secara bersamaan melalui dua sistem operasi yang berjalan pada mesin virtual. Gambar 3 merupakan *alert Snort* IDS yang sudah diatur di dalam *local.rules*.



Gambar 3 Alert Serangan/gangguan *SYN Flood*

Dari gambar di atas dapat dilihat bahwa *Snort* IDS yang diterapkan dapat berjalan dengan baik. *Snort* IDS yang dipasang pada *server* dapat mendeteksi gangguan yang masuk ke *server* yang sudah si terapkan *Snort* IDS. Dari hasil *capture* tersebut menunjukan informasi bahwa *IP* 192.168.1.6 dan IP 192.168.1.7 yang merupakan *IP intruder* melakukan *Syn flood* terhadap *IP* 192.168.1.2 yang merupakan *IP server* IDPS dengan peringatan "*Warning! SYN Flooding!*". Lengkap dengan informasi waktu, tanggal kejadian dan klasifikasi serangan/gangguan.

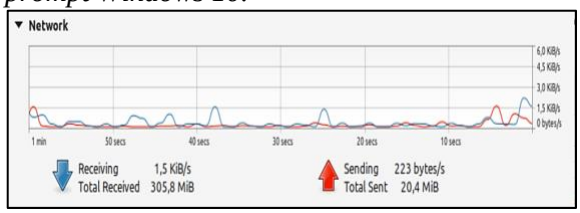
Gambar 4 Serangan *SYN Flood* dari *Backtrack 5* (1) Terblokir

Gambar 5 Serangan *SYN Flood* dari *Backtrack 5* (2) Terblokir

Hasil yang ditunjukkan pada gambar 4 dan 5 menyatakan bahwa kedua *intruder* tidak dapat melakukan serangan *SYN Flood* ke alamat *IP* 192.168.1.2 yang merupakan *IP server*, yang artinya *rules IPTables* yang telah diterapkan berhasil menghentikan penyerangan.

B. Pengujian IDS dan *IPTables* Dengan Serangan *Ping of Death*

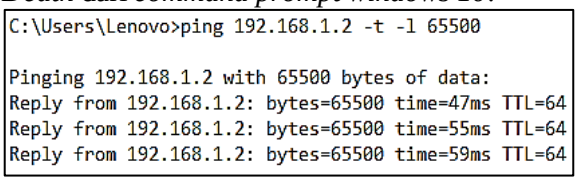
Pengujian selanjutnya melakukan penyerangan dari *Windows 10* ke *server* dengan metode permintaan reply dari *server* berulang kali bermaksud membuat mesin server sibuk menanggapi permintaan dari *intruder*. Upaya penyerangan ini dilakukan melalui *command prompt Windows 10.*



Gambar 6 Tampilan *Resource Server* Sebelum Mengalami Gangguan

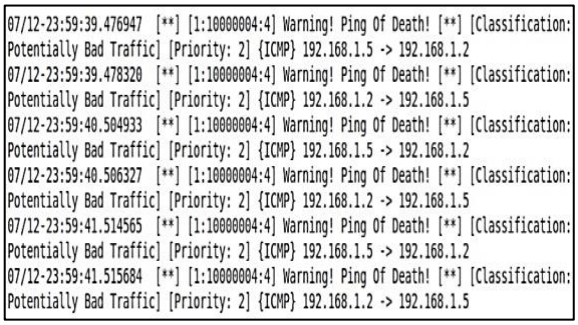
Sebelum adanya serangan *Ping of Death*, tampilan grafik *network* yang ditampilkan *resource* IDPS *server* masih terlihat normal, belum ada kenaikan grafik yang signifikan pada grafik *network*.

Berikut tampilan gambar serangan *Ping of Death* dari *command prompt windows 10*:



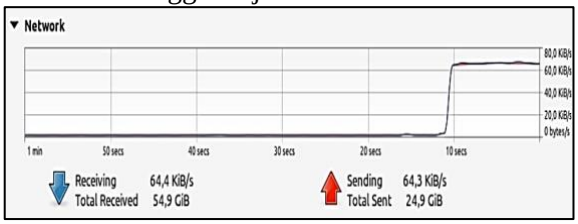
Gambar 7 Tampilan Serangan *Ping of Death*

Serangan icmp dilakukan dengan jumlah beban 65500 yang dikirim oleh intruder ke alamat IP *server* secara terus-menerus.



Gambar 8 Tampilan *Alert* Serangan *Ping of Death*

Dari hasil *capture* tersebut menunjukkan informasi bahwa IP 192.168.1.5 yang merupakan IP intruder melakukan Ping of Death terhadap IP 192.168.1.2 yang merupakan IP server IDPS dengan peringatan "*Warning! Ping of Death!*". Lengkap dengan informasi waktu dan tanggal kejadian.



Gambar 9 Tampilan *Resource Server* Setelah Mengalami Gangguan

Pada gambar 10 adanya kenaikan yang signifikan pada grafik *network*, paket yang

diterima dan dikirim naik menjadi 64,4 KiB dan 64,3 KiB per detik dari alamat IP yang sama.

```
C:\Users\Lenovo>ping 192.168.1.2 -t -l 65500

Pinging 192.168.1.2 with 65500 bytes of data:
Reply from 192.168.1.2: bytes=65500 time=44ms TTL=64
Reply from 192.168.1.2: bytes=65500 time=59ms TTL=64
Reply from 192.168.1.2: bytes=65500 time=43ms TTL=64
Request timed out.
Request timed out.
Request timed out.
```

Gambar 10 Tampilan Serangan Setelah Diterapkan Aturan IPTables

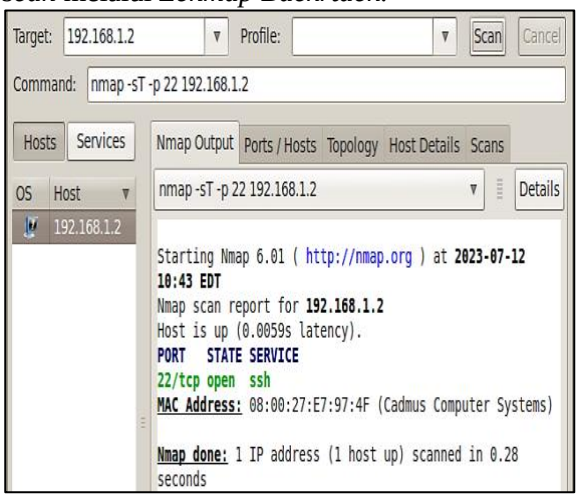
Hasil yang ditunjukkan pada gambar 10 merupakan hasil serangan *Ping of Death* setelah menerapkan aturan IPS *Iptables*. Hasil yang diperoleh menyatakan bahwa blokir IP dengan menggunakan *rule IPTables* berhasil. Dapat dilihat pada gambar yang sedang melakukan *Ping of Death* ke alamat IP 192.168.1.2 yang merupakan alamat IP *server* IDPS yang tengah berjalan tiba-tiba mengalami *Request time out* (RTO) pada mesin *intruder* yang artinya mesin intruder tersebut tidak dapat melakukan *Ping of Death* ke alamat IP tujuan.

C. Pengujian IDS dan *IPTables* Dengan Serangan Nmap Port Scan

Pengujian yang lakukan selanjutnya adalah melakukan serangan/gangguan UDP *port scan* dan TCP *port scan* menggunakan *Zenmap* yang tersedia di *Backtrack*, untuk mencari *port* mana saja yang pada *server* IDPS.

a) TCP Port Scan

Berikut tampilan gambar serangan TCP *port scan* melalui *Zenmap Backrtack:*



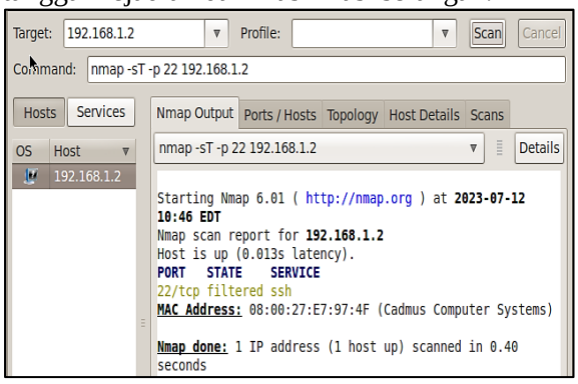
Gambar 11 Tampilan Pengujian *Nmap* Dengan Protokol TCP

Dapat dilihat pada gambar 11, penulis mencoba melakukan TCP *port scan* menggunakan *Zenmap* ke *IP address* 192.168.1.2. Hasil dari gambar tersebut menyatakan bahwa ada *port* dengan protokol TCP yang terbuka pada mesin *server* IDPS yaitu *port* 22.

```
root@tokkie:~/snort_src/snort-2.9.20# snort -A console -q -i enp0s3 -c /etc/snort/snort.conf
07/13-00:04:21.803328  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
07/13-00:04:21.808153  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
07/13-00:04:21.808153  [**] [1:10000002:2] Warning! NMAP TCP scan! [**] [Classification: Detection
of a Network Scan] [Priority: 3] {TCP} 192.168.1.6:53301 -> 192.168.1.2:22
```

Gambar 12 Tampilan *Alert* Serangan *Nmap* Dengan Protokol TCP

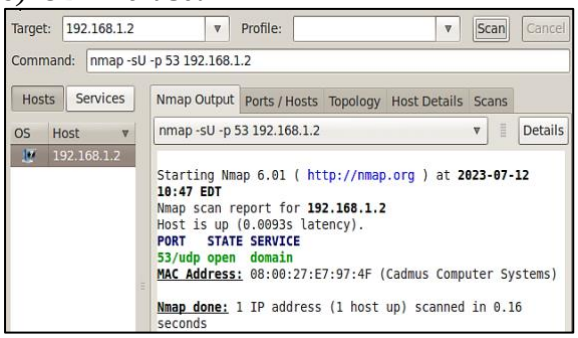
Gambar 12, merupakan tampilan *alert* dari *Snort* IDS. Dapat dilihat bahwa gangguan berasal dari IP 192.168.1.12 dengan protokol TCP ke alamat *IP* 192.168.1.2 yang merupakan mesin *server* IDPS. IDPS dengan peringatan "*Warning! NMAP TCP Scan!*". Pada hasil *capture Snort* IDS di atas juga terdapat waktu, tanggal kejadian dan klasifikasi serangan.



Gambar 13 Pengujian *Nmap* Protokol TCP yang Telah Terblokir

Pada gambar 13 menyatakan bahwa *intruder* tidak dapat melakukan *port scan* ke alamat IP 192.168.1.2, dapat dilihat dari tabel keterangan bahwa *port* 22 dengan protokol TCP telah ter-*filtered.*

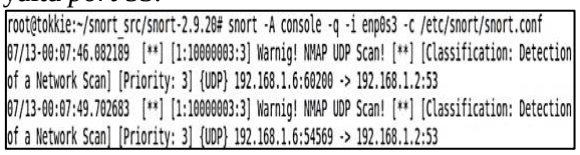
b) UDP Port Scan



Gambar 14 Tampilan Pengujian *Nmap* Dengan Protokol UDP

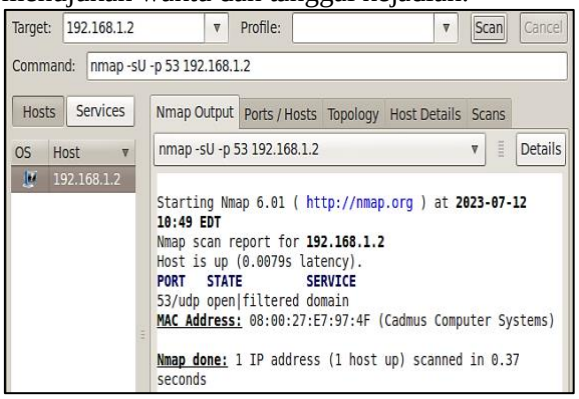
Dapat dilihat pada gambar 14, penulis mencoba melakukan UDP *port scan* yang juga menggunkan *Zenmap* ke *IP address* 192.168.1.2. Hasil dari gambar tersebut menyatakan bahwa ada *port* dengan protokol

UDP yang terbuka pada mesin *server* IDPS yaitu *port* 53.



Gambar 15 Tampilan *Alert* Serangan *Nmap* Dengan Protokol UDP

Pada gambar 15, tampilan *alert* dari *Snort* IDS terdapat gangguan yang berasal dari *IP address* yang sama yaitu 192.168.1.13 dengan protokol UDP ke IP *address* 192.168.1.2 dengan peringatan "Warning! *NMAP UDP Scan!*". Pada hasil *capture Snort* IDS diatas menujukan waktu dan tanggal kejadian.



Gambar 16 Pengujian *Nmap* Protokol UDP yang Telah Terblokir

Pada gambar 16 menjelaskakn *intruder* tidak dapat melakukan *port scan* ke alamat IP 192.168.1.2, dapat dilihat dari tabel keterangan bahwa *port* 53 dengan protokol UDP telah ter-*filtered.*

D. Pengujian Kualitas Layanan *Server*

Mengukur kualitas layanan *server* dalam penelitian ini penulis menggunakan tabel QoS. QoS (*Quality of Service*) merupakan metode pengukuran yang digunakan untuk menentukan kemampuan sebuah jaringan. QoS ini tentu sudah memiliki penilaian yang berstandarlisasi TIPHON. Pengujian dilakukan dengan menggunakan *iperf3*, *ping* dan *wireshark. Iperf3* digunakan untuk menguji kecepatan *upload* dan *download* sebuah *server*. *Ping* dilakukan untuk melihat berapa paket yang hilang. Kemudian *Wireshark* digunakan untuk menentukan *throughput, delay* dan *jitter.*

a) Pengujian Sebelum Ada Serangan Syn Flood

Upload	*Download*
94,0 Mbit	95,2 Mbit

Tabel 1 Kecepatan *Upload Download* Sebelum Ada Serangan

Upload download yang diuji dengan *iperf3 client* ke *server*, nilai yang di dapatkan adalah sebesar 94,0 Mbit untuk *upload* dan 95,2 Mbit untuk *download.*

Parameter QoS	Nilai Rata-Rata	Index	Kategori
Throughput	83 Mbit	4	Sangat Baik
Delay	0,50ms	4	Sangat Baik
Jitter	0,03ms	4	Sangat Baik
Packet Lost	0%	4	Sangat Baik
Index Rata-Rata		**4**	**Sangat Baik**

Tabel 2 Parameter QoS Sebelum Ada Serangan

Hasil yang diperoleh pada tabel 2 merupakan hasil pengujian *iperf3* dan *test ping* dari *client*, masing-masing diuji selama sepuluh detik. Hasil yang didapatkan sebelum masuknya gangguan, kualitas layanan *server* berada dalam kategori sangat baik. *Upload* dan *download* yang dimiliki bernilai besar. Sebab apa bila *Upload* dan *download* semakin besar nilainya maka semakin bagus.

b) Pengujian Saat Ada Serangan Syn Flood

Upload	*Download*
210 Kbit	1,46 Mbit

Tabel 3 Kecepatan *Upload Download* Saat Ada Serangan

Parameter QoS	Nilai Rata-Rata	Index	Kategiru
Throughput	10 Kbit	1	Buruk
Delay	236,56ms	3	Baik
Jitter	6,88ms	3	Baik
Packet Lost	70%	1	Buruk
Index Rata-Rata		**2**	**Sedang**

Tabel 4 Parameter QoS Saat Ada Serangan

Pengujian yang dilakukan saat adanya serangan, *server* berada di dalam kategori sedang. Karena rata-rata index yang diperoleh saat itu adalah 2, artinya kualitas layanan *server* menurun sebanyak 50%. Nilai *upload* dan *download* pun menurun drastis.

c) Pengujian Saat Serangan Sudah Syn Flood Diatasi

Upload	*Download*
86,0 Mbit	91,4 Mbit

Tabel 5 Kecepatan Upload Download Saat Serangan Sudah Diatasi

Parameter QoS	Nilai Rata-Rata	Index	Kategori
Throughput	68 Mbit	4	Sangat Baik
Delay	0,71ms	4	Sangat Baik
Jitter	1,13ms	3	Baik
Packet Lost	0%	4	Sangat Baik

Index Rata-Rata	**3,75**	**Baik**

Tabel 6 Parameter QoS Saat Serangan Sudah Diatasi

Pengujian yang dilakukan saat serangan telah berhasil diatasi dengan aturan *IPTables*, *server* kembali membaik, meskipun tidak 100%, akan tetapi kualitas layanan *server* berada dalam kategori baik.

4. KESIMPULAN

Penerapan IDS Snort pada *server*, secara efektif dapat bekerja sebagai keamanan jaringan komputer yang berbasis open source dalam mendeteksi sebuah serangan atau gangguan pada mesin server IDPS. IPTables dapat menjadi solusi untuk mengatasi gangguan atau serangan yang masuk ke dalam *server* IDPS. Dengan menerapkan metode ini dalam pengujian yang peneliti lakukan, dapat memulihkan kualitas layanan *server* dalam sebuah jaringan dengan index sebanyak 3,75 dari nilai index sebesar 4.

References

[1] K. Al Fikri and Djuniadi, "Keamanan Jaringan Menggunakan Switch Port Security," *InfoTekJar J. Nas. Inform. dan Teknol. Jar.*, vol. 5, no. 2, pp. 302–307, 2021, doi: https://doi.org/10.30743/infotekjar.v5i2.3501.

[2] R. W. Ismail and R. Pramudita, "Metode Penetration Testing pada Keamanan Jaringan Wireless Wardriving PT . Puma Makmur Aneka Engineering Bekasi," *J. Mhs. Bina Insa.*, vol. 5, no. 1, pp. 53–62, 2020, [Online]. Available: https://ejournal-binainsani.ac.id/index.php/JMBI/article/view/1373

[3] Wahyuni and P. Adytia, "Perbandingan Algoritma Machine Learning Dalam Mendeteksi Serangan DDOS," *Temat. J. Teknol. Inf. Komun.*, vol. 9, no. 2, pp. 161–166, 2022, doi: 10.38204/tematik.v9i2.1070.

[4] N. Nuryadi and E. C. Nainggolan, "Implementasi Intrusion Detection System Pada Local Area Network (Studi Kasus : Yayasan Pendidikan Tanah Tingal Tangerang)," *SITEKIN J. Sains, Teknol. dan Ind.*, vol. 19, no. 1, pp. 1–8, 2021, [Online]. Available: https://ejournal.uin-suska.ac.id/index.php/sitekin/article/view/11098

[5] G. Tambunan and I. Mantra, "Implementasi Keamanan Ids / Ips Dengan Snort Dan IP Tables pada Server," *Semin. Nas. Mhs. Ilmu Komput. dan Apl. Jakarta-Indonesia, 28 Januari 2020 IMPLEMENTASI*, pp. 10–16, 2020, [Online]. Available: https://conference.upnvj.ac.id/index.php/senamika/article/view/352

[6] D. Kusuma, U. Darussalam, and D. Hidayatullah, "Implementasi Monitoring Jaringan Melalui Aplikasi Sosial Media Telegram Dengan Snort," *J I M P - J. Inform. Merdeka Pasuruan*, vol. 5, no. 1, pp. 6–9, 2020, doi: 10.37438/jimp.v5i1.242.

[7] H. Alamsyah, Riska, and A. Al Akbar, "Analisa Keamanan Jaringan Menggunakan Network Intrusion Detection and Prevention System," *JOINTECS (Journal Inf. Technol. Comput. Sci.*, vol. 5, no. 1, p. 17, 2020, doi: 10.31328/jointecs.v5i1.1240.

[8] S. Khadafi, Y. D. Pratiwi, and E. Alfianto, "Keamanan Ftp Server Berbasiskan Ids Dan Ips Menggunakan Sistem Operasi Linux Ubuntu," *Netw. Eng. Res. Oper.*, vol. 6, no. 1, p. 11, 2021, doi: 10.21107/nero.v6i1.190.

[9] F. F. Phasa, J. D. Irawan, and S. A. Wibowo, "Sistem Autentifikasi Hostpot Menggunakan Ldap Server," *JATI (Jurnal Mhs. Tek. Inform.*, vol. 4, no. 2, pp. 120–127, 2020, doi: 10.36040/jati.v4i2.2703.

[10] A. Z. Mardiansyah, Y. M. Abdussyakur, and A. H. Jatmika, "OPTIMASI PORT KNOCKING DAN HONEYPOT MENGGUNAKAN IPTABLES SEBAGAI KEAMANAN JARINGAN PADA SERVER," vol. 3, no. 2, pp. 189–199, 2021, doi: https://doi.org/10.29303/jtika.v3i2.144.