IMPLEMENTASI *INTRUSION DETECTION SYSTEM* (IDS) PADA SISTEM KEAMANAN JARINGAN MENGGUNAKAN TELEGRAM SEBAGAI MEDIA NOTIFIKASI

Niras Furqan[1], Ipan Suandi[2], Muhammad[3]
[1,2,3)]Prodi Teknologi Rekayasa Jaringan Telekomunikasi
Jurusan Teknik Elektro Politeknik Negeri Lhokseumawe
Email: nirasf00@gmail.com, ipan@pnl.ac.id, cekm4d@gmail.com

ABSTRAK

Server menjadi hal yang perlu mendapat perhatian lebih mengenai tingkat keamanannya. *Server* yang memiliki celah kelemahan dapat dimanfaatkan oleh pihak yang tidak bertanggung jawab. Data-data yang seharusnya bersifat pribadi bisa saja disalahgunakan oleh pihak yang tidak bertanggung jawab. Administrator harus memastikan bahwa sistem benar-benar aman. Salah satu cara menjaga keamanan *server* yaitu dengan pendeteksian intrusi yang dianggap berbahaya menggunakan *Intrusion Detection System* (IDS). Sistem pendeteksian intrusi dibangun berdasarkan aturan yang telah disimpan dalam sebuah *database* (*signature-based*). *Snort* merupakan salah satu perangkat lunak yang berfungsi untuk mengetahui adanya intrusi. Paket-paket data yang melalui lalu lintas jaringan akan dianalisa terlebih dahulu. Paket-paket data yang terdeteksi sebagai intrusi akan memicu sebuah *alert* yang kemudian disimpan dalam *file log*. Dengan begitu, administrator dapat mengetahui intrusi yang terjadi pada *server*. Namun, administrator perlu menganalisa pada komputer *server* tersebut. Adanya aplikasi *instant messaging* dapat membantu administrator untuk memperoleh pemberitahuan secara *real time*. Salah satunya menggunakan Telegram dimana administrator mendapatkan informasi singkat dan laporan adanya intrusi pada *server*. Informasi yang didapatkan berupa waktu, tanggal kejadian, Jenis serangan, IP Sumber dan IP Target serangan.

Kata-kata kunci: Intrusion Detection System, Snort, Telegram

I. PENDAHULUAN

Keamanan merupakan salah satu masalah terbesar bagi pengguna Internet terutama penyedia sebuah *server* maupun sistem jaringan komputer. Masalah tersebut menimbulkan kecenderungan besar untuk memiliki *Intrussion Detection System* (IDS) pada setiap jaringan. IDS merupakan perangkat lunak atau perangkat keras sistem yang secara otomatis melakukan proses pemantauan (*monitoring*) insiden yang terjadi dalam sistem komputer atau jaringan serta menganalisis tanda-tanda adanya masalah terhadap keamanan sistem [1].

Sistem yang tidak aman akan berdampak negatif bagi penyedia maupun pengguna sistem. Oleh karena itu, perlu adanya monitoring keamanan jaringan dengan tujuan untuk meminimalisir terjadinya penyusupan. Salah satu aplikasi yang digunakan sebagai IDS adalah *Snort*. Aplikasi *opensource* tersebut memiliki kemampuan untuk mendeteksi adanya penyusupan terhadap sistem sesuai dengan aturan yang telah ditetapkan. Hasil deteksi tersebut akan direkam dan disimpan pada database. Peringatan deteksi dapat memanfaatkan aplikasi instant messaging sebagai media untuk memberitahu kepada administrator jaringan mengenai indikasi penyusupan ke *server*.

Aplikasi *instant messaging* saat ini popular digunakan oleh berbagai kalangan. Salah satu aplikasi tersebut yang memiliki berbagai fitur yaitu Telegram. Aplikasi tersebut selain untuk *chatting*, terdapat fitur pertukaran dokumen. Fitur tersebut dapat dimanfaatkan untuk memberikan laporan keamanan sistem dalam bentuk dokumen digital.

II. TINJAUAN PUSTAKA

A. Keamanan Jaringan

Sebuah jaringan dengan desain dasar yang aman memerlukan beberapa penerapan aturan dasar yang sederhana yang perlu disesuaikan dengan kontek yang berbeda, seperti berikut :

- Menutup portIP yang tidak terpakai
- Memutuskan titik control traffic
- Periksa semua jalur akses dan desain multilayer dalam kasus jaringan yang kompleks

Adapun upaya meningkatkan keamanan jaringan sebuah sistem harus memenuhi beberapa unsur, antara lain:

1. *Confidentiality* (kerahasiaan). Pembatasan akses hanya kepada user yang berhak atas suatu data atau informasi, dan mencegah akses dari user yang tidak memiliki hak.
2. *Integrity* (integritas). Keaslian data atau informasi yang dikirm melalui jaringan dari sumber ke penerima secara lengkap, tanpa ada modifikasi atau manipulasi oleh pihak yang tidak berwenang.

3. *Availabillity* (ketersediaan). Ketersediaan data atau informasi ketika dibutuhkan saat itu juga

.

B. *Intrusion Detection System* (IDS)

Intrusion Detection System (IDS) adalah salah satu langkah penting untuk mengurangi komputer gangguan jaringan/host. IDS tidak hanya fokus pada deteksi aktivitas abnormal pada jaringan komputer, tetapi juga menentukan apakah kegiatan tersebut berbahaya atau tidak. Pada dasarnya ada 2 (dua) jenis IDS, yaitu *host-based* IDS (HIDS) dan *Network Intrusion Detection* System (NIDS). HIDS berkonsentrasi pada kegiatan dalam sebuah host tanpa mempertimbangkan kegiatan di jaringan komputer. Disisi lain, NIDS menempatkan fokus pada jaringan komputer tanpa memeriksa kegiatan host. metodologi *Intrusion Detection* dapat diklasifikasikan sebagai deteksi berbasis Signature, deteksi berdasarkan Anomaly dan analisis Stateful protokol deteksi. [2]

C. Snort

Snort menggunakan *Rules* disimpan dalam file text yang dapat dimodifikasi oleh editor text. *Rules* disimpan dalam satu file, kemudian dimasukkan dalam file konfigurasi utama disebut *snort.conf*. Snort membaca aturan ini pada saat start-up dan membangun struktur data internal atau rantai untuk menerapkan rules untuk capture packet. Menemukan packet yang sama dengan rules adalah proses yang rumit dan harus dilakukan secara real time. Maka dari itu snort mempunyai beberapa komponen yang saling bekerja sama untuk mendeteksi serangan-serangan dan menghasilkan format yang dibutuhkan oleh sistem. Komponen-komponen yang ada dalam snort diantaranya:

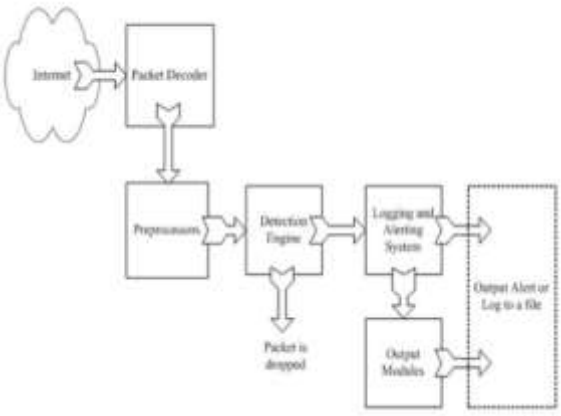
1. *Packet Decoder*
   Paket decoder mengambil paket dari berbagai jenis interface jaringan, bekerja di layer 2 (Data Link) dan memisahkan interface tersebut apakah masuk melalui ethernet atau wireless yang selanjutnya akan dikirim ke tahap preprocessor.

2. *Preprocessor*
   Preprocessor adalah komponen yang dapat digunakan *Snort* untuk mengatur atau memodifikasi paket data sebelum masuk ke tahap detection engine.

3. *Detection Engine*
   Detection Engine adalah komponen yang paling penting dari snort, packet yang datang dari tahap sebelumnya akan dibandingkan dengan rules yang telah ditetapkan apabila packet sama dengan rules yang ditetapkan maka itu diasumsikan sebagai serangan.

4. *Logging and Alerting System*
   Komponen ini menghasilkan packet apa yang ditemukan pada detection engine, packet digunakan untuk log aktifitas atau menghasilkan peringatan.

5. *Output Modules*
   Pada tahapan ini menyimpan output yang dihasilkan oleh sistem snort, Output yang dihasilkan bervasiasi, seperti teks (*ASCII*), *XML*, *syslog*, *tcpdump*, *binary format*, atau *Database* (*MySQL*, *MsSQL*, *PostgreSQL*, dan sebagainya [3].



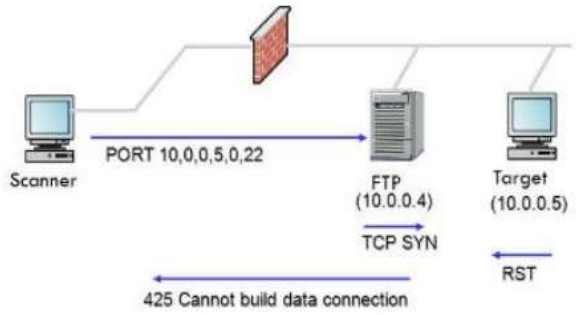
Gbr 1 Alur Kerja Snort

D. *Instant Messaging Telegram*

Telegram secara definisi menurut telegram.org merupakan alternatif layanan aplikasi perpesanan untuk ponsel (*mobile*) maupun dekstop yang berbasis cloud dengan tingkat keamanan tinggi serta kecepatan aksesnya. Aplikasi *intant messaging* tersebut tersedia untuk berbagai device seperti ponsel yang memiliki sistem operasi Android, iOS, Windows Phone, Ubuntu Touch. Tidak hanya dapat digunakan pada perangkat mobile, tetapi juga dapat berjalan pada sistem desktop seperti Windows, OS X, dan Linux. Meskipun aplikasi tersebut terlihat sederhana, tetapi gratis dan memiliki fitur yang lebih unggul dibandingkan dengan aplikasi *instant messaging* lainnya. Telegram diklaim sebagai aplikasi yang aman karena salah satunya memiliki fitur dimana menyediakan pilihan pesan end to end yang dienkripsi serta dapat hancur dengan sendirinya dalam jangka waktu tertentu.[4]

E. *Port Scanning*

Port scanning yaitu teknik penyerangan yang dilakukan untuk memperoleh informasi berupa port-port yang terbuka pada server, sistem operasi dan versi sistem operasi dari server, sehingga penyusup dapat memeroleh informasi penting mengenai kondisi sumber daya dan security yang dimiliki oleh target serangan. Adapun bentuk serangan Port Scanning dapat dilihat pada gambar 2.



Gbr 2 Serangan *Port Scanning*

F. *Denial of Service*

Denial of service (DoS) adalah serangan/ancaman besar bagi ketersediaan layanan internet. Tujuan dari serangan DoS adalah untuk benar-benar mengikat sumber daya dari server, yang mencegah pengguna yang sah dari mengakses layanan atau menyediakan layanan yang sah.

G. *Ping of Death*

Ping of death adalah salah satu jenis serangan DoS. Pesan ICMP ECHO_REQUEST kirim ke tuan rumah sistem untuk memeriksa konektivitas dan mengharapkan ECHO_REPLY. Dalam ping of death banyak sistem digunakan untuk mengirim beberapa permintaan ke sistem target. Ping of death menyerang upaya untuk jenuh jaringan dengan mengirimkan serangkaian terus menerus dari permintaan ICMP echo (ping) melalui sambungan bandwidth tinggi ke host target pada koneksi bandwidth rendah untuk menyebabkan itu untuk mengirim kembali ICMP echoreply untuk setiap permintaan. serangan ping of death dapat memperlambat jaringan atau bahkan menonaktifkan konektivitas jaringan.

H. *SYN Flooding Attack*

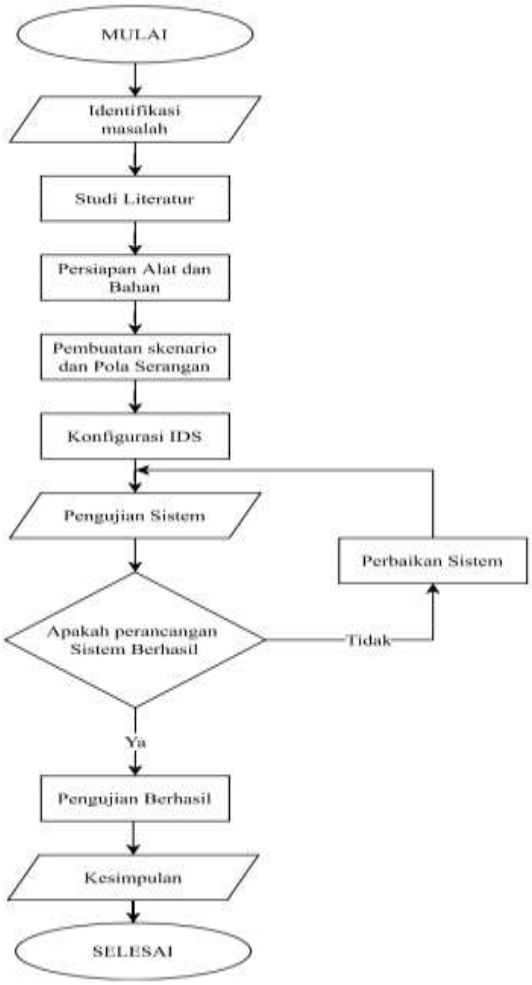
SYN flooding attack adalah serangan yang dirancang khusus, yang mempekerjakan banjir paket *SYN* untuk mengkonsumsi semua koneksi jaringan baru yang tersedia pada host yang ditargetkan, sehingga penundaan menanggapi permintaan koneksi jaringan yang sah dan tersendat-sendat akhirnya penyedia layanan. Secara teoritis, serangan ini berlaku untuk semua koneksi TCP, seperti WWW, Telnet, e-mail, dan sebagainya. Dalam kebanyakan sistem UNIX, beberapa struktur memori perlu dialokasikan untuk setiap pembentukan koneksi TCP. Mengambil sistem BSD sebagai contoh, struktur socket digunakan untuk menyimpan informasi komunikasi, seperti protokol yang digunakan, informasi alamat, antrian koneksi, *buffer* dan *flags*. Tujuan dari *SYN flooding* atau (DOS) bukan untuk mendapatkan akses tidak sah ke komputer atau data, tapi untuk mencegah pengguna yang sah dari layanan dari menggunakannya. Tujuan penyerang sebagai berikut:

1. Banjir jaringan dengan lalu lintas, sehingga mencegah lalu lintas jaringan yang sah.
2. Mengganggu koneksi antara dua mesin, sehingga mencegah akses ke layanan.
3. Mencegah individu tertentu dari mengakses layanan.

III. METODOLOGI

A. Diagram Alir Penelitian

Penelitian yang dilakukan melalui beberapa tahap yang dapat dilihat pada Gambar 3.

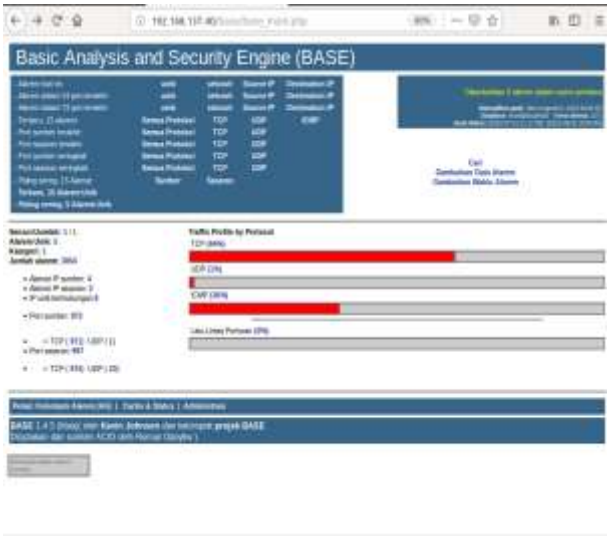


Gbr 3 Diagram Alur Penelitian

A. Desain Antarmuka

Laporan mengenai deteksi intrusi dapat diakses melalui web-based. Dengan adanya laporan berbasis web tersebut, administrator dapat melihat maupun

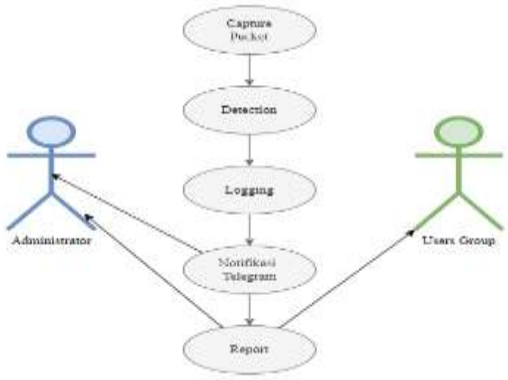
melakukan pengaturan terhadap akun Telegram yang digunakan. Akan terdapat dua web yang dapat digunakan, yaitu web yang menampilkan laporan singkat beserta akun Telegram dengan memanfaatkan framework Code Igniter dan web untuk menganalisa hasil deteksi *Snort* yaitu *Basic Analysis and Security Engine* (BASE). Laporan berbasis web dibuat dengan antarmuka (*interface*) sederhana, tetapi dapat dimanfaatkan sebaik mungkin oleh Administrator. Pada BASE, Administrator akan disajikan informasi lengkap megenai insiden yang terdeteksi. Informasi khusus untuk menganalisa insiden yang terjadi disajikan dalam BASE berbasis web yang memiliki tampilan sederhana dan mencakup semua log hasil deteksi *Snort*. Tampilan antarmuka BASE dapat dilihat pada gambar 4.



Gbr 4 Interface *BASE*

B. Proses Kerja Sistem

Perancangan dibuat sebagai visualisasi dari alur kerja sistem dimana dapat menjadi sebuah standar atau disebut juga dengan *Unified Modelling Language* (UML). Diagram *Use case* merupakan salah satu jenis diagram UML yang menggambarkan interaksi antara sistem dan aktor. Cara kerja sistem yang akan dibangun dapat mudah dipahami oleh pengguna. Adapun perancangan sistem pada penelitian ini yang dapat dilihat pada Gambar 5.



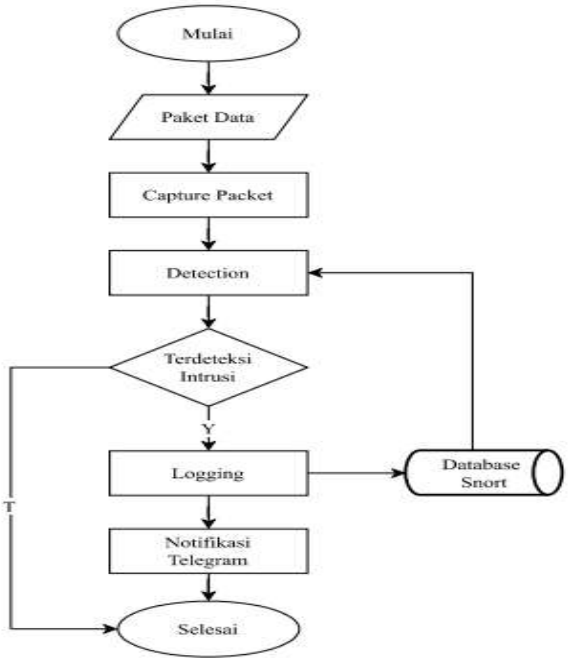
Gbr 5 Diagram Use Case Sistem

Pada Gambar 5 menunjukkan bagaimana gambaran cara kerja *Snort* sebagai *Intrusion Detection System*. *Snort* akan menangkap paket dalam lalu lintas jaringan menggunakan packet *capture-decode engine*. Kemudian akan dilakukan pendeteksian terhadap paket dengan membadingkan aturan yang ada. Jika teridentifikasi sebagai intrusi, maka akan dilakukan pencatatan yang nantinya dihasilkan sebuah peringatan secara real time maupun dalam bentuk dokumen digital.

Use case tersebut mempresentasikan sebuah interaksi antara aktor dengan sistem. Terdapat dua aktor yaitu admininstrator dan users group dimana memiliki peran yang sama untuk mendapatkan notifikasi mengenai deteksi terhadap intrusi. Namun, hal yang membedakan yaitu jenis notifikasai yang dikirimkan oleh sistem kepada kedua aktor. Selain mendapatkan notifikasi serangan secara real time.

C. Alur Deteksi Serangan

Pada Gambar 6 menggambarkan alur bagaimana sistem bekerja dalam mendeteksi intrusi dan memberikan peringatan secara *real time*.

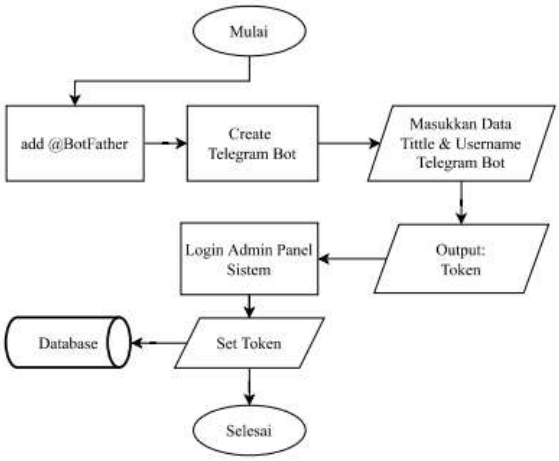


Gbr 6 Flowchart Deteksi Serangan

Pertama, paket yang masuk akan ditangkap dan dianalisa oleh Snort berdasarkan aturan yang telah ditetapkan. Jika paket tersebut tidak terdeteksi sebagai sebuah intrusi atau serangan, maka paket tersebut akan dibuang dan proses berakhir. Namun, ketika paket tersebut terdeteksi sebagai sebuah intrusi, selanjutnya akan dilakukan pencatatan pada *file log* maupun *database*. Setelah dicatat dan disimpan dalam database maka akan terjadi sebuah *trigger* untuk mengeksekusi *file* php yang berfungsi untuk mengirimkan notifikasi Telegram terhadap Administrator.

D. Telegram *Bot token* dan ID Pengguna

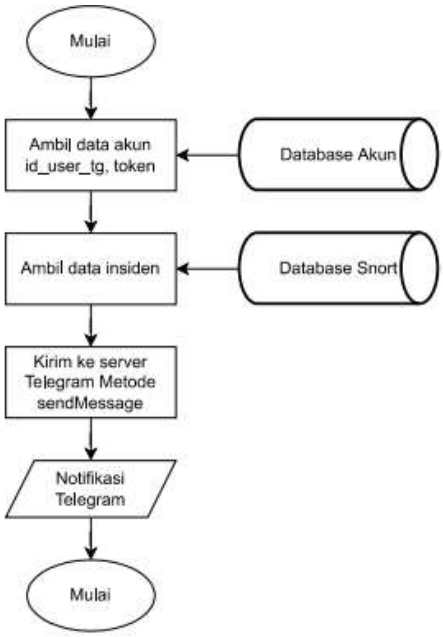
Untuk pemberitahuan peringatan intrusi dilakukan dengan memanfaatkan media Telegram dengan menggunakan fitur Telegram *bot.* Hal yang utama dalam penggunaan fitur tersebut yaitu dengan memiliki token atau kode otentikasi yang dapat mengatur tindakan Telegram *bot.* Alur mendapatkan bot serta menyimpannya dalam *database* sistem dapat dilihat pada Gambar 7.



Gbr 7 *Flowchart* Set Telegram *Bot Token*

E. Alur kirim Notifikasi

Paket yang telah terdeteksi sebagai intrusi akan disimpan dalam *database* yang kemudian akan memicu proses pengiriman notifikasi kepada administrator.



Gbr 8 *Flowchart* alur Kirim Notifikasi

Metode yang digunakan oleh Telegram *bot* untuk mengirimkan pesan yaitu *sendMessage*. Perintah dasar untuk mengirimkan pesan yaitu dengan mengakses alamat API Telegram https://api.telegram.org/bot/send Message? chat_id=&text=. ID Pengguna dan token diambil dari *database* akun, sedangkan informasi insiden diambil dari database Snort. Parameter yang digunakan untuk mengirimkan isi pesan yaitu *text*. Format pesan notifikasi yang dikirimkan kepada administrator yaitu informasi mengenai tipe serangan, klasifikasi, *IP Address attacker*, dan waktu insiden terjadi.

F. Metode Analisis

Setelah melakukan pengujian serangan berupa *Ping of Death, SYN Flooding Atatck* dan *Scanning Port* selesai dilakukan, selanjutnya data hasil pengujian dilakukan analisa. Metode analisa yang di gunakan yaitu deskriptif kualitatif, yaitu mengamati perubahan sebelum dan sesudah terjadinya serangan dan menampilkannya dalam bentuk tabulasi dan grafik.

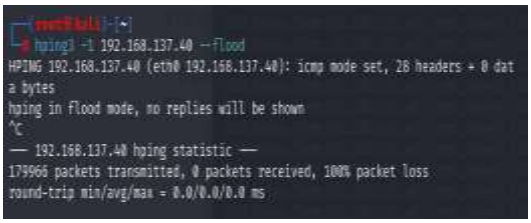
IV. HASIL DAN PEMBAHASAN

A. Hasil Pengujian Serangan

Pengujian serangan dilakukan pada jaringan lokal sesuai dengan rancangan yang telah dibuat.

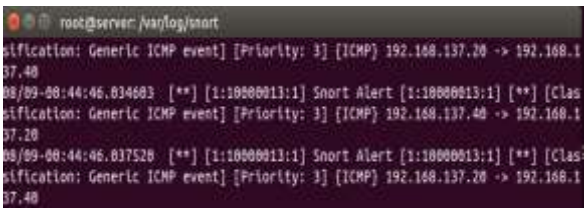
1. *Ping of Death*

Pengujian ini untuk mengirimkan ping dalam format yang salah atau berbahaya ke komputer target sehingga menyebabkan komputer target lambat, pengujian ini menggunakan *tools hping3* pada kemputer *Attacker.*



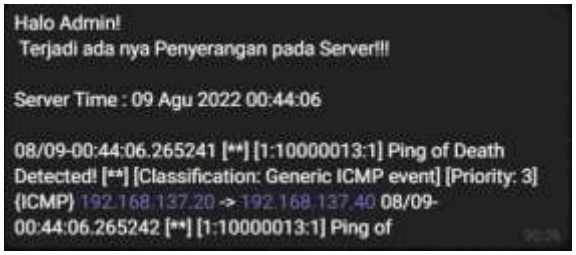
Gbr 9 Pengujia *Ping of Death*

Intrusi yang terdeteksi oleh IDS pada tanggal 09/08 pukul 00:44:46 dengan SID 10000013 berasal dari IP *Address* 192.168.137.20 dengan target 192.168.137.40. Hasil deteksi dapat dilihat pada Gambar 10.



Gbr 10 Deteksi *Ping of Death*

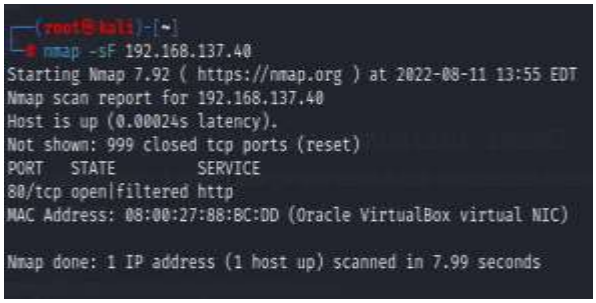
Notifikasi akan dikirimkan ke Administrator setelah intrusi tersebut tercatat oleh IDS. Notifikasi dikirimkan dengan pesan "*Ping of Death Detected*!". Pemberitahuan tersebut dapat dilihat pada Gambar 11.



Gbr 11 Notifikasi Telegram *Ping of Death*

2. *Port Scanning*

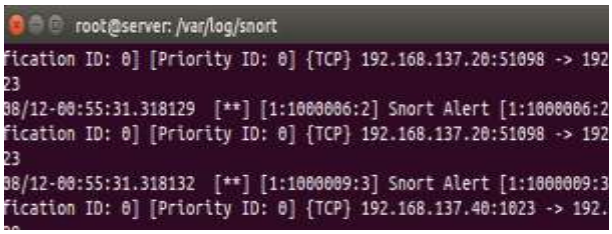
Pengujian *port scanning* bertujuan untuk mendapatkan informasi mengenai *port* yang terbuka pada *server*. Pengujian menggunakan aplikasi nmap dapat dilihat pada Gambar 12.



Gbr 12 Pengujian *Port Scanning*

Pada pengujian *port scanning* menggunakan *FIN scan* untuk mengetahui *port* yang terbuka. Pengujian ini dilakukan oleh *attacker* dengan *IP Address* 192.168.137.20 pada jam 13:55 EDT tanggal 2022-08-11. Pengujian berhasil mendapatkan 1 (satu) *port* yang terbuka yaitu *port* 80/HTTP.

Di waktu yang sama, IDS telah aktif dan mendeteksi adanya intrusi dengan SID 10000006 yang dilakukan oleh *IP Address* 192.168.137.20 ke *IP Address* 192.168.137.40 *port* 80. Deteksi intrusi dapat dilihat pada Gambar 13.



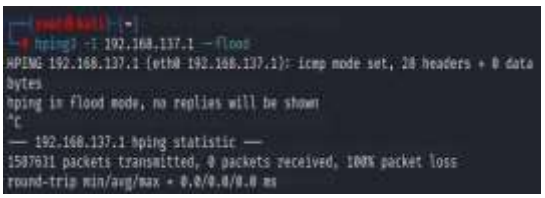
Gbr 13 Deteksi *Port Scanning*

Setelah terdeteksi sebuah intrusi, administrator mendapatkan notifikasi melalui aplikasi *instant messasing* Telegram. Notifkasi tersebut dapat dilihat pada Gambar 14

Gbr14 Notifikasi *Port Scanning*

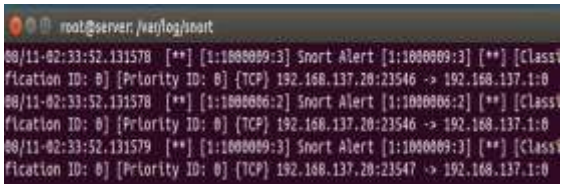
3. *SYN Flooding Attack*

Pengujian *SYN Flood Attack* ini menggunakan aplikasi *hping3* pada computer *attacker*. Target pengujian yaitu *IP Address* 192.168.137.1 dengan protocol TCP. Pengujian dapat dilihat pada gambar 15.



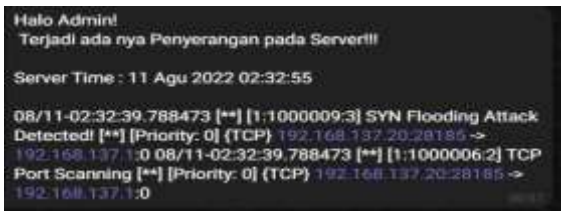
Gbr 15 Pengujian *SYN Flooding Attack*

Di sisi lain, IDS telah aktif dan mendeteksi adanya intrusi dengan waktu 02:33:05 tanggal 08/11 dan SID 1000009 yang dilakukan oleh *IP Address* 192.168.137.20 ke *IP Address* 192.168.137.1. Deteksi intrusi dapat dilihat pada Gambar 16.



Gbr 16 Deteksi *SYN Flooding Attack*

Beberapa saat setelah intrusi terdeteksi, terjadi sebuah *trigger* yang mengirimkan notifikasi kepada administrator sesuai *rule* yang telah dibuat. Notifikasi dapat dilihat pada Gambar 17.

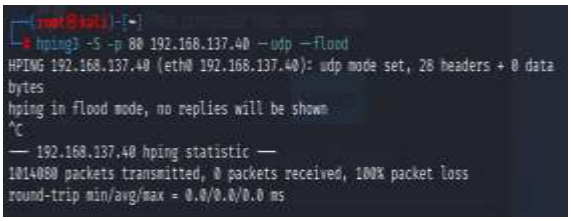


Gbr 17 Notifikasi *SYN Flooding Attack*

4. Hasil Pengujian *Ddos Attack*

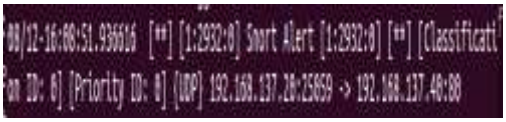
Pengujian Ddos menggunakan aplikasi hping3 pada komputer *attacker*. Target pengujian yaitu *IP Address* 192.168.137.40 dengan port 80 dan protokol UDP. Pengujian dapat dilihat pada Gambar 18.

.



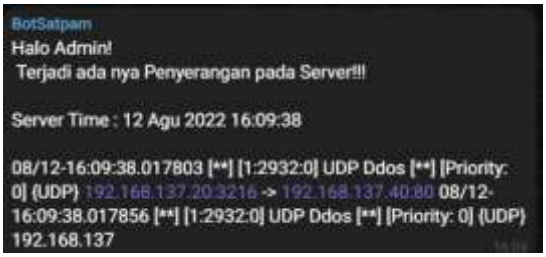
Gbr 18 Pengujian *Ddos Attack*

Intrusi yang terdeteksi oleh IDS pada tanggal 08/12 pukul 16:08:51 dengan SID 2932 berasal dari *IP Address* 192.168.137.20. Hasil deteksi dari IDS dapat dilihat pada Gambar 19.



Gbr 19 Deteksi *Dddos Attack*

Notifikasi akan dikirimkan ke administrator setelah intrusi tersebut tercatat oleh IDS. Notifikasi dikirimkan dengan pesan "Ddos udp" sesuai dengan *signature* yang terdeteksi oleh IDS. Pemberitahuan tersebut dapat dilihat pada Gambar 20.



Gbr 20 Notifikasi *Ddos Attack*

B. Hasil Deteksi Serangan

Dari serangkaian pengujian didapatkan informasi mengenai insiden yang terjadi. IDS mendeteksi terdapat *attacker* dengan *IP Address* 192.168.2137.20 melakukan serangan terhadap server yang dijaga oleh IDS. *Attacker* melalukan uji coba serangan pada *server*. Tabel 1 menunjukkan informasi serangan yang terdeteksi.

TABEL I
Informasi Serangan Terdeteksi

No	Waktu	Asal Serangan	Tipe Serangan	
			Port	Layanan
1	09-08-2022 00:43:51	192.168.137.20	80	Server
2	09:08-2022 00:55:31	192.168.137.20	80	Server
3	12-08-2022 02:32:55	192.168.137.20	80	Server
4	12-08-2022 16:08:49	192.168.137.20	80	Server

V. KESIMPULAN

Berdasarkan implementasi snort sebagai *Intrusion Detection System* dan menggunakan Telegram sebagai media notifikasi, maka diperoleh kesimpulan sebagai berikut:

1. Proses pendeteksian dan pengirimian Notifikasi juga di pengaruhi oleh kecepatan pada jaringan yang digunakan, seperti pada serangan *Ping of Death* dan *Ddos Attack* memiliki jeda waktu selama 2 hinggaa 15 detik sebelum serangan tersebut terdeteksi.
2. Dengan menerapkan IDS, Administrator jaringan akan sangat terbantu dalam *monitoring* karena pada sangat salah satu server diserang, administrator akan langsung mengetahuinya secara *real time*.
3. Dengan menggunakan snort sebagai IDS, administrator dapat mengetahui kapan terjadinya serangan, dan dari mana siapa penyerang tersebut melalui *IP* nya.
4. IDS yang dibangun memberikan notifikasi secara *realtime* kepada administrator melalui aplikasi *instant messaging* telegram.

REFERENSI

[1] A. Anitha, "**Network Security Using Linux Intrusion Detection System**," *International Journal of Research in Computer Science*, p. 33, 2011.

[2] R. U. Rehman, **Intrusion Detection Systems with Snort: Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID**, New Jersey: Prentice Hall PTR, 2003.

[3] S. A. D. V. Bhagayan, "**A brief study and comparison of, Open Source Intrusion Detection System Tools**," *International Journal* , pp. 23-73, 2013.

[4] J. D. Vico, "**Telegram Bypassing The Authentication Protocol**," 2014.