IMPLEMENTASI MONITORING KEAMANAN JARINGAN PADA SERVER UBUNTU MENGGUNAKAN SNORT INTRUSION DETECTION PREVENTION SYSTEM (IDPS) DAN TELEGRAM BOT SEBAGAI MEDIA NOTIFIKASI DI PT SS UTAMA

Gugus Pradita, Anang Pramono*
Teknik Informatika, Universitas 17 Agustus 1945 Surabaya
Jalan Semolowaru 45 Surabaya, Indonesia
anangpramana@untag-sby.ac.id

ABSTRAK

Pertumbuhan pesat Teknologi Informasi (TI) telah secara signifikan mengubah kehidupan manusia, terutama dengan munculnya internet yang memudahkan akses ke berbagai informasi. Namun juga berpotensi munculnya ancaman baru terhadap keamanan data dan informasi. Keamanan komputer bukan hanya bagian integral dari sistem informasi tetapi juga memainkan peran krusial dalam memvalidasi, menjamin integritas data, dan memberikan layanan kepada pengguna. Penelitian ini dilakukan di PT SS Utama di Surabaya, Jawa Timur, bertujuan untuk meningkatkan keamanan jaringan pada *server* perusahaan dengan mengimplementasikan *Intrusion Detection and Prevention System* (IDPS). IDPS adalah sistem keamanan komputer yang dirancang untuk mendeteksi dan mencegah serangan pada sistem komputer. Pengujian penetrasi dibagi menjadi lima jenis seperti, *Port Scanning, Brute Force, DDoS, SQL Injection* dan *XSS Reflected Attack*. Hasil pengujian penetrasi dari kelima jenis serangan tersebut menunjukkan bahwa Snort berhasil mendeteksi pengujian tersebut. Perbedaan waktu antara deteksi Snort dan Telegram *Bot* setelah 25 kali percobaan pengiriman pesan adalah 3,52 detik untuk deteksi Snort dan 2,72 detik untuk Telegram *Bot*. Penerapan *prevention system* pemantauan ini dapat memblokir dan membuka IP *Address attacker* melalui Telegram *Bot* sehingga dapat meningkatkan efisiensi dan efektivitas dalam pengelolaan keamanan jaringan dan *server* serta kinerja sistem secara keseluruhan di PT SS Utama.

Kata kunci : *Keamanan Jaringan, Intrusion Detection Prevention System, Snort, Penetration Testing*

1. PENDAHULUAN

Perkembangan Teknologi Informasi (TI) yang terus meningkat secara signifikan telah mengubah paradigma kehidupan manusia, terutama dengan kehadiran internet yang mempermudah akses terhadap berbagai informasi. Kemampuan untuk menghubungkan jaringan komputer di seluruh dunia melalui internet membawa manfaat besar, namun juga menimbulkan tantangan baru terkait keamanan data dan informasi [1]. Keamanan jaringan komputer menjadi aspek kritis yang perlu diperhatikan, mengingat adanya potensi eksploitasi oleh pihak yang tidak bertanggung jawab [2].

Keamanan jaringan komputer tidak hanya menjadi bagian integral dari sistem informasi, tetapi juga berperan dalam menjaga validitas, integritas data, dan memastikan ketersediaan layanan bagi pengguna [3]. Sistem harus dijaga agar terlindungi dari berbagai jenis serangan dan penyusupan oleh pihak yang tidak memiliki otoritas untuk mengaksesnya [4].

PT SS Utama merupakan perusahaan manufaktur sepatu di Surabaya, Jawa Timur. Jaringan internet menjadi sarana penting untuk menjalankan berbagai aktivitas, termasuk pengolahan data dan pengawasan melalui CCTV. Saat ini PT SS Utama memiliki sistem jaringan dan *server* yang belum optimal. Komputer yang terhubung ke internet, seperti *server* sangat rentan terhadap pencurian data oleh pihak yang tidak sah dapat mengakibatkan lambatnya proses pertukaran data atau bahkan kerusakan sistem. Sistem keamanan jaringan komputer yang belum optimal dapat berakibat fatal bagi penyedia maupun pengguna sistem [4].

Kondisi ini menuntut perlunya solusi keamanan yang dapat melakukan pemantauan, analisis, dan tindakan terhadap aktivitas mencurigakan di dalam jaringan komputer [5]. Salah satu pendekatan yang diusulkan untuk mengatasi permasalahan keamanan *server* adalah penerapan *Intrusion Detection Prevention System* (IDPS). IDPS merupakan sistem keamanan komputer yang dirancang untuk mendeteksi dan mencegah akses yang tidak sah atau aktivitas yang mencurigakan pada jaringan atau sistem komputer [6]. Dalam implementasinya menggunakan Snort sebagai sistem deteksi dan *Iptables* sebagai pencegahan serangan, Snort adalah *tool open-source* yang mampu mendeteksi aktivitas penyusupan sesuai dengan aturan yang telah ditetapkan [7]. Tantangan lain dihadapi oleh *administrator* jaringan di PT SS Utama adalah keterbatasan keberadaan mereka di dekat *server*. Untuk mengatasi hal ini, solusi notifikasi *real-time* diperlukan. Telegram *Bot* diusulkan sebagai alat komunikasi untuk memberikan peringatan kepada *administrator* saat terjadi serangan.

Dengan menggabungkan Snort sebagai sistem deteksi serangan, *Iptables* sebagai pencegahan serangan, Telegram *Bot* sebagai aplikasi notifikasi dan kontrol *server*, serta pemanfaatkan *Basic Analysis and Security Engine (*BASE*)* untuk keperluan analisis forensik jaringan, diharapkan solusi ini dapat mempermudah *administrator* dalam memonitoring

keamanan jaringan secara efektif. Dengan demikian, upaya peningkatan keamanan jaringan di PT SS Utama dapat diimplementasikan dengan lebih efisien dan responsif.

2. TINJAUAN PUSTAKA

2.1. Keamanan Jaringan

Keamanan jaringan mencakup serangkaian strategi, kebijakan, teknologi, dan tindakan yang diterapkan untuk melindungi integritas, kerahasiaan, dan ketersediaan data dalam jaringan komputer [3]. Tujuan utama dari keamanan jaringan adalah untuk mencegah akses yang tidak sah, serangan *malware*, perusakan data, dan gangguan lainnya yang dapat membahayakan sistem. Hal ini bertujuan untuk memastikan bahwa sistem jaringan dapat beroperasi secara normal dan aman dari berbagai ancaman. [4].

2.2. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah sistem yang digunakan untuk memonitor lalu lintas jaringan dengan tujuan mendeteksi upaya intrusi atau aktivitas mencurigakan, kemudian melaporkannya melalui peringatan atau alert [8]. IDS berfungsi untuk meningkatkan keamanan dan kinerja jaringan dengan mendeteksi ancaman. IDS terbagi menjadi dua jenis: *Network-based Intrusion Detection System* (NIDS) dan *Host-based Intrusion Detection System* (HIDS) [9].

2.3. Intrusion Prevention System (IPS)

Intrusion Prevention System (IPS) adalah komponen keamanan jaringan yang bertujuan untuk mengidentifikasi dan mencegah upaya intrusi atau serangan keamanan pada jaringan komputer secara aktif. IPS tidak hanya mendeteksi seperti IDS, tetapi juga mengambil tindakan pencegahan untuk menghentikan atau membatasi serangan yang terdeteksi [9].

2.4. Snort

Snort adalah software *open-source* berperan dalam melakukan fungsi untuk mengenali serta menganalisis tindakan yang mencurigakan di dalam jaringan. Snort mampu menangkap dan mencatat paket data yang melewati jaringan secara langsung, kemudian menyimpannya ke dalam basis data *logging*. Snort termasuk dalam kategori *network-based intrusion detection system* (NIDS) [10]. Secara prinsip, snort memiliki tiga fungsi utama yaitu *packet sniffer, packet logger, Network Intrusion Detection System* (NIDS) [11].

2.5. Telegram Bot

Telegram *Bot* adalah program komputer yang dirancang untuk berinteraksi dengan pengguna melalui platform Telegram. *Bot* ini menggunakan *Aplication Programing Interface* (API) Telegram untuk menerima dan mengirim pesan, serta melakukan berbagai tindakan sesuai perintah pengguna [12].

2.6. Damn Vulnerable Web Application (DVWA)

Damn Vulnerable Web Application (DVWA) adalah aplikasi web yang dikembangkan menggunakan PHP dan MySQL untuk menguji kerentanan. DVWA menyediakan lingkungan yang aman bagi profesional keamanan untuk menguji keterampilan mereka dan membantu pengembang memahami cara mengamankan aplikasi web [13]. DVWA memungkinkan pengujian serangan seperti XSS (*Cross Site Scripting*) dan *SQL Injection.*

3. METODE PENELITIAN

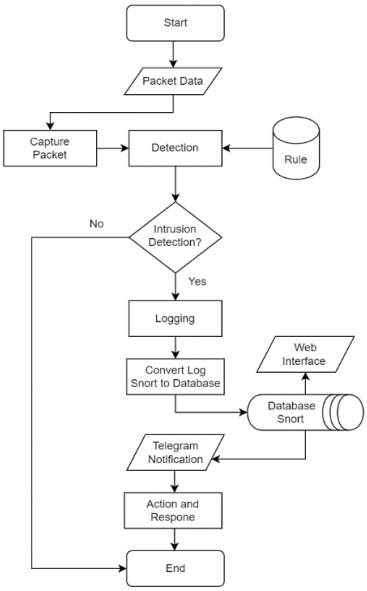
3.1. Metode Pengumpulan Data

Pengumpulan data adalah proses mengumpulkan informasi yang diperlukan untuk rekayasa. Ini merupakan langkah krusial dalam metode ilmiah, karena data yang dikumpulkan umumnya digunakan untuk menguji hipotesis yang telah dirumuskan. Dengan mengikuti prosedur yang sistematis dan standar yang diperlukan, diharapkan dapat ditemukan hubungan antara metode pengumpulan data dengan masalah yang akan diselesaikan. Pada penelitian ini, metode pengumpulan data dilakukan melalui observasi dan Studi literatur.

3.2. Metode Penelitian

Penelitian ini berfokus pada keamanan jaringan komputer di PT SS Utama. Metode atau model pengembangan sistem yang digunakan dalam penelitian ini adalah PPDIOO, yang merupakan metode perancangan jaringan yang bertujuan untuk mendukung pengembangan jaringan. PPDIOO terdiri dari beberapa tahap, yaitu *Prepare*, *Plan*, *Design*, *Implement*, *Operate*, dan *Optimize* [4]. Dengan meningkatnya kompleksitas layanan jaringan, diperlukan metodologi yang mendukung perancangan arsitektur dan desain jaringan.

3.3. Desain Sistem



Gambar 1. Flowchart intrusion detection system snort dan pencegahan telegram bot

Penelitian dilakukan berdasarkan struktur Snort dan integrasi antara Telegram *Messenger*. Desain Sistem ini akan memantau semua jenis paket data yang mencurigakan dan aliran data yang masuk ke *server* komputer melalui Snort. Berikut adalah skema *Intrusion Detection System* Snort dan pencegahan melalui Telegram *Bot* serta BASE sebagai antarmuka web.

Gambar 1 adalah *flowchart* perancangan *Intrusion Detection System* Snort dan pencegahan melalui Telegram *Bot* serta BASE sebagai antarmuka web. Konsol terminal membaca setiap paket jaringan dari target [5]. Paket tersebut kemudian diperiksa apakah paket tersebut merupakan paket berbahaya. Setiap hasil deteksi akan disimpan ke dalam *log*. Aplikasi pihak ketiga bernama barnyard2 akan mengubah *log* Snort menjadi database dan disimpan ke *administrator* sistem menggunakan antarmuka web atau notifikasi *database* Telegram *Messenger* [14]. Lalu *administrator* mengirimkan perintah ke Telegram *Bot* yang akan di eksekusi oleh *server* dan *server* akan merespon inputan yang diberikan oleh *administrator* melalui Telegram *Bot*.

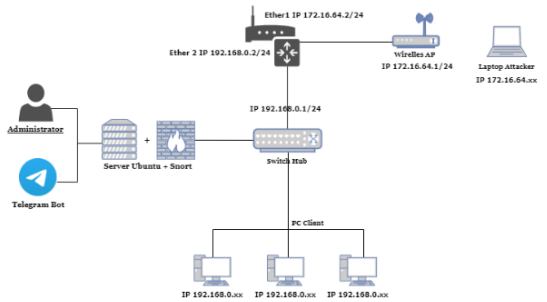
4. HASIL DAN PEMBAHASAN

4.1. Implementasi Sistem

Implementasi atau penerapan sistem keamanan jaringan komputer memerlukan detail rancangan yang akan digunakan sebagai panduan langkah demi langkah selama proses implementasi. Dengan demikian, sistem yang dibangun dapat sesuai dengan desain yang telah direncanakan.

4.2. Topologi jaringan

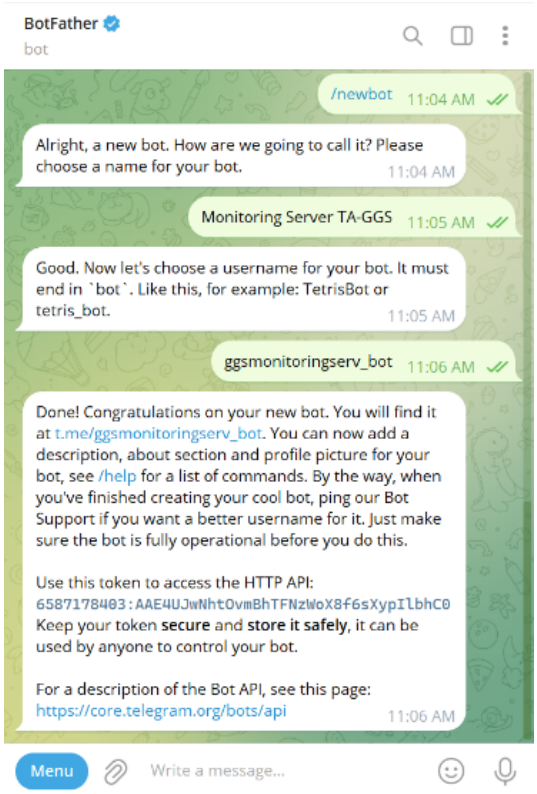
Dalam penerapan topologi jaringan, semua perangkat yang diperlukan, baik perangkat keras maupun perangkat lunak, dikumpulkan terlebih dahulu. Selanjutnya, perangkat-perangkat tersebut ditempatkan sesuai dengan topologi yang telah direncanakan sebelumnya. Setelah semua perangkat saling terhubung, langkah berikutnya adalah mengkonfigurasi setiap perangkat agar dapat berkomunikasi satu sama lain.



Gambar 2. Topologi Jaringan

4.3. Implementasi Telegram Bot

Pada tahap ini, implementasi dan konfigurasi Telegram *Bot* berfungsi sebagai media notifikasi *real-time* untuk *administrator* jaringan. Dengan demikian, jika ada *log* dari *database* snort yang masuk, *Bot* akan secara otomatis mengirimkan notifikasi secara *real-time*. Gambar 3 di bawah ini menunjukkan tampilan awal dalam proses pembuatan Telegram *Bot*.



Gambar 3. Tampilan pembuatan telegram bot

Pada tahap ini, implementasi dan konfigurasi Telegram *Bot* bertujuan untuk mengintegrasikan Telegram *Bot* dengan *server*. Telegram *Bot* dapat memberikan perintah untuk memblokir IP *Address attacker*, sehingga jika ada intrusi yang terdeteksi dan notifikasi dikirim melalui Telegram *Bot*, *administrator* jaringan dapat segera mengambil langkah pengamanan. Gambar 4 di bawah ini menunjukkan fungsi yang menghubungkan Telegram *Bot* dengan *server* agar dapat berkomunikasi satu sama lain. Fungsi ini memerlukan *API key* dan *Chat ID* dari Telegram *Bot*.



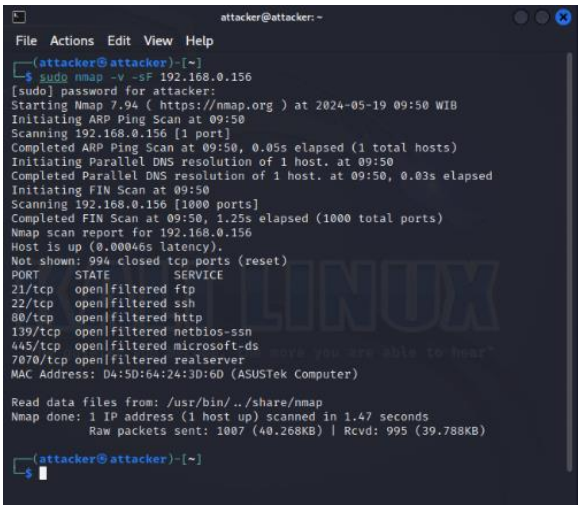
Gambar 4. Function penghubung telegram bot

4.4. Pengujian Sistem

Pengujian bersifat fungsionalitas dilakukan untuk memastikan bahwa sistem atau aplikasi dapat beroperasi sesuai dengan yang diharapkan untuk memverifikasi apakah sistem dapat menghasilkan output yang valid.

4.5. Penetration Testing

Pengujian penetrasi dilakukan pada jaringan lokal sesuai dengan rancangan yang telah dibuat. Tahapan pengujian ini bertujuan untuk mengetahui apakah aplikasi utama pada Snort berjalan dengan baik atau tidak dalam memonitor keamanan jaringan.



Gambar 5. Penetration test using Nmap port scanning

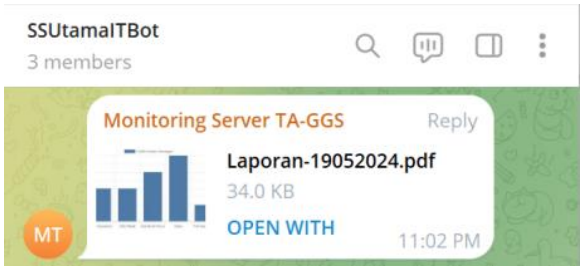
Pengujian penetrasi menggunakan *Nmap port scanning* dapat dilihat pada Gambar 5, yang merupakan jenis serangan dari komputer *attacker* dengan sistem operasi Kali Linux. *Attacker* mengetikkan perintah nmap diikuti dengan alamat IP korban. *Tool* tersebut akan memulai dan menunjukkan *port* apa saja yang terbuka dari komputer korban.

Uji penetrasi menggunakan *Nmap port scanning* berhasil diluncurkan, dan Telegram *Bot* akan mengirimkan informasi tentang paket data berbahaya yang berasal dari serangan tersebut, seperti yang ditunjukkan pada Gambar 6.



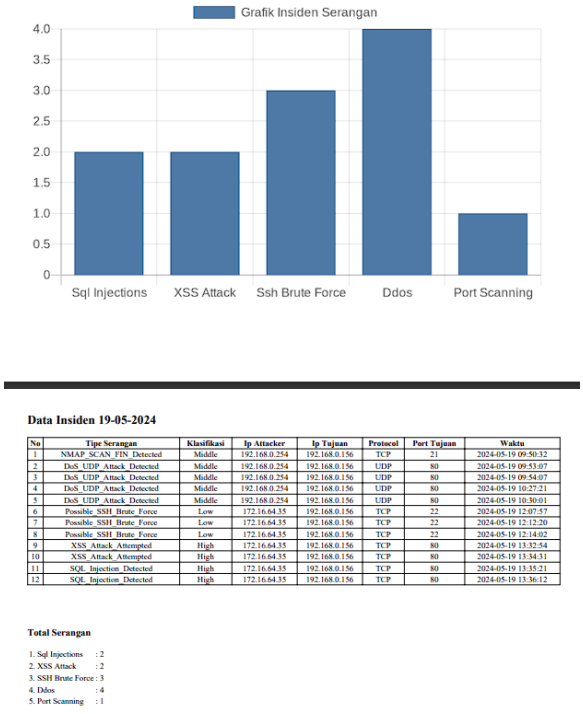
Gambar 6. Notifikasi dari bot telegram

Selain itu, *administrator* juga menerima laporan mengenai intrusi yang terdeteksi dalam bentuk dokumen digital berformat PDF yang dapat dilihat pada Gambar 7. Laporan tersebut menampilkan informasi insiden yang terjadi dalam satu hari.



Gambar 7. Laporan dokumen digital dari bot telegram

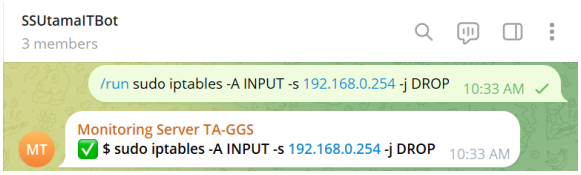
Laporan tersebut berisi grafik secara kuantitas dan informasi dalam bentuk tabel mengenai tipe serangan, klasifikasi serangan, ip *address attacker*, ip *address* tujuan, *protocol*, *port* tujuan dan waktu insiden terjadi yang dapat dilihat pada Gambar 8. Laporan digital tersebut dihasilkan dalam interval satu hari sebelum dikirimkan ke *administrator*.



Gambar 8. Informasi dari laporan digital

4.6. Prevention Testing

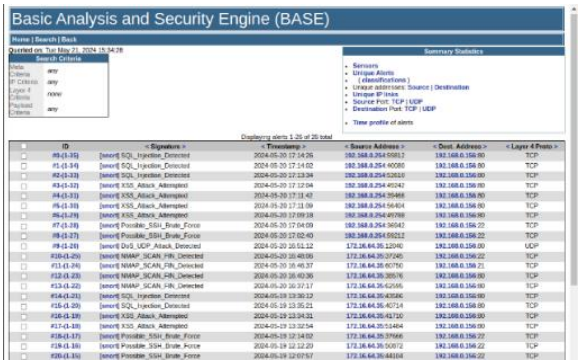
Prevention system merupakan mekanisme keamanan jaringan yang berfungsi untuk mencegah ancaman yang telah terdeteksi. Sistem ini terus memantau jaringan, mencari kemungkinan insiden berbahaya, dan menangkap informasi. *Administrator* dapat mengendalikan *prevention system* secara manual untuk memblokir alamat IP *Address attacker* melalui *shell-bot* yang ada di *server*.



Gambar 9. Fitur sistem pencegahan pada bot telegram

Gambar 9 menunjukkan bahwa Telegram *Bot* dapat digunakan untuk sistem pencegahan. Fitur ini menggabungkan *iptables* Linux untuk memblokir alamat IP klien yang mencurigakan. *Administrator* hanya perlu mengetikkan perintah run diikuti dengan perintah *iptables* dan alamat IP yang mencurigakan pada Telegram *Bot* untuk memblokir mereka dari sistem.

4.7. Snort Web Base interface



Gambar 10. Snort web basic analysis and security engine

Tabel *1*.

Snort memiliki banyak antarmuka berbasis web pihak ketiga, dan BASE adalah salah satu dari antarmuka tersebut. Pada Gambar 6 merupakan *Basic Analysis and Security Engine* (BASE) digunakan untuk melihat semua *log* Snort tanpa harus membuka terminal.

4.8. Hasil Akurasi Deteksi Intrusi

Untuk mengetahui efektivitas sistem ini, peneliti menghitung besarnya penundaan antara terjadinya serangan hingga saat notifikasi diterima oleh *administrator*. Dari aktivitas pengujian terhadap *server*, diperoleh waktu masing-masing aktivitas mulai dari awal serangan, deteksi, hingga pengiriman dan penerimaan notifikasi.

4.9. Tingkat Akurasi Waktu

Tingkat akurasi waktu dihitung mulai dari awal serangan terjadi hingga terdeteksi. Tingkat deteksi rata-rata bergantung pada faktor ini. Selain itu, tingkat notifikasi yang dikirim kepada *administrator* juga bergantung pada selisih waktu antara pengiriman dan penerimaan, seperti yang terlihat dalam

Tabel 1. Tingkat akurasi waktu

No	Tipe	Tingkat Akurasi Waktu (Timestamp)			
		Waktu Awal Serangan (m^a,s^a)	Waktu Snort Deteksi Serangan (m^b,s^b)	Waktu Pengiriman Telegram (m^c,s^c)	Waktu Penerimaan *Admin* (m^d,s^d)
1	*NMAP Port Scanning*	09:50:29	09:50:32	09:50:41	09:50:44
	NMAP Port Scanning	16:37:13	16:37:17	16:37:31	16:37:34
3	*NMAP Port Scanning*	16:40:32	16:40:36	16:40:41	16:40:43
4	*NMAP Port Scanning*	16:46:34	16:46:37	16:40:41	16:40:43
5	*NMAP Port Scanning*	16:48:04	16:48:08	16:48:11	16:48:16
6	*Brute Force SSH Attack*	12:07:50	12:07:57	12:08:01	12:08:05
7	*Brute Force SSH Attack*	12:12:12	12:12:20	12:12:22	12:12:26
8	*Brute Force SSH Attack*	12:13:57	12:14:02	12:14:11	12:14:14
9	*Brute Force SSH Attack*	17:02:34	17:02:40	17:02:41	17:02:43
10	*Brute Force SSH Attack*	17:04:02	17:04:09	17:04:11	17:04:13
11	*DDoS Attack*	09:53:04	09:53:07	09:53:11	09:53:15
12	*DDoS Attack*	09:54:03	09:54:07	09:54:11	09:54:14
13	*DDoS Attack*	10:27:18	10:27:21	10:27:21	10:27:23
14	*DDoS Attack*	10:29:58	10:30:01	10:30:11	10:30:14
15	*DDoS Attack*	16:51:08	16:51:12	16:51:21	16:51:24
16	*SQL Injection*	13:32:52	13:32:54	13:33:01	13:33:03
17	*SQL Injection*	13:34:29	13:34:31	13:34:42	13:34:44
18	*SQL Injection*	17:09:15	17:09:18	17:09:21	17:09:24
19	*SQL Injection*	17:11:07	17:11:09	17:11:11	17:11:13
20	*SQL Injection*	17:12:02	17:12:04	17:12:11	17:12:13
21	*XSS Reflected Attack*	13:35:19	13:35:21	13:35:31	13:35:34
22	*XSS Reflected Attack*	13:36:10	13:36:12	13:36:21	13:36:24
23	*XSS Reflected Attack*	17:13:33	17:13:34	17:13:41	17:13:43
24	*XSS Reflected Attack*	17:14:00	17:14:02	17:14:11	17:14:13
25	*XSS Reflected Attack*	17:14:24	17:14:26	17:14:31	17:14:33

Berdasarkan

Tabel *1*, tingkat akurasi pengukuran waktu mengacu pada lima jenis serangan: *Nmap Port Scanning, Brute Force SSH Attack, DDoS Attack, SQL Injection, dan XSS Reflected Attack*. *Timestamp* mencakup waktu awal serangan (m^a, s^a), waktu Snort deteksi serangan (m^b, s^b), waktu pengiriman Telegram (m^c, s^c), dan waktu penerimaan oleh *administrator* (m^d, s^d). Waktu awal serangan adalah stempel waktu ketika *attacker* mulai menyerang sistem. Waktu Snort deteksi serangan adalah stempel waktu ketika Snort mendeteksi paket data berbahaya. Waktu pengiriman Telegram adalah stempel waktu ketika Telegram *Bot* mengirimkan pesan kepada *administrator* jaringan. Waktu penerimaan admin adalah stempel waktu *administrator* jaringan menerima semua pesan informasi data paket berbahaya.

Tabel *2*.

4.10. Perbedaan Waktu

Perbedaan waktu antara lima jenis penetrasi *NMAP Port Scanning, Brute Force SSH Attack, DDoS Attack, SQL Injection dan XSS Reflected Attack* dapat ditunjukkan pada Tabel 2 Perbedaan waktu dicatat dalam satuan waktu detik. Perbedaan waktu antara menyerang dan mendeteksi diperoleh dari stempel waktu awal serangan (m^a, s^a) dan waktu Snort deteksi serangan (m^b, s^b). Perbedaan waktu antara dikirim dan diterima diperoleh dari stempel waktu pengiriman Telegram (m^c, s^c) dan waktu penerimaan admin (m^d, s^d). Berikut adalah hasil perbedaan waktu yang ditunjukkan pada

Tabel 2. Perbedaan waktu

No	Tipe Serangan	Perbedaan Waktu (Detik)	
		Selisih awal serangan dan terdeteksi (S^x)	**Selisih dikirim dan diterima (S^y)**
1	*NMAP Port Scanning*	3	3
2	*NMAP Port Scanning*	4	3
3	*NMAP Port Scanning*	4	2
4	*NMAP Port Scanning*	3	2
5	*NMAP Port Scanning*	4	5
6	*Brute Force SSH Attack*	7	4
7	*Brute Force SSH Attack*	8	4
8	*Brute Force SSH Attack*	5	3
9	*Brute Force SSH Attack*	6	2
10	*Brute Force SSH Attack*	7	2
11	*DDoS Attack*	3	4
12	*DDoS Attack*	4	3
13	*DDoS Attack*	3	2
14	*DDoS Attack*	3	3
15	*DDoS Attack*	4	3
16	*SQL Injection*	2	2
17	*SQL Injection*	2	2
18	*SQL Injection*	3	3
19	*SQL Injection*	2	2
20	*SQL Injection*	2	2
21	*XSS Reflected Attack*	2	3
22	*XSS Reflected Attack*	2	3
23	*XSS Reflected Attack*	1	2
24	*XSS Reflected Attack*	2	2
25	*XSS Reflected Attack*	2	2
Total		88	69
Rata - rata		3,52	2,72

Berdasarkan

Tabel *2*, nilai tertinggi perbedaan waktu antara awal serangan dan deteksi adalah 8 detik pada *Brute Force SSH Attack*, sedangkan nilai terendah adalah 1 detik pada tipe *XSS Reflected Attack*. Nilai tertinggi perbedaan waktu antara pengiriman dan penerimaan adalah 5 detik pada tipe *Nmap Port Scanning*. Waktu rata-rata dari 25 tes penetrasi dengan lima jenis serangan antara awal serangan dan Snort mendeteksi adalah 3,52 detik, sedangkan waktu rata-rata antara pengiriman dan penerimaan pesan Telegram adalah 2,72 detik.

Perbedaan waktu diperoleh dari hasil stempel waktu pada Tabel 1. Rumus untuk mendapatkan perbedaan waktu adalah dengan mengurangkan antara hasil stempel waktu pada Tabel 1. Berikut adalah rumus untuk mendapatkan perbedaan waktu antara awal serangan dan deteksi Snort.

$S^x = ((m^b \times 60 + s^b) - ((m^a \times 60) + s^a)$
$S^x = (60m^b + s^b) - (60m^a + s^a)$ (1)

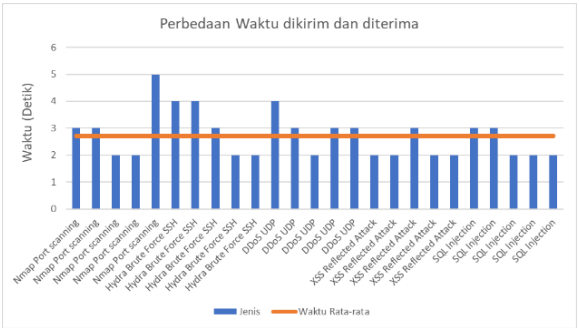
Penjelasan:
S^x = Perbedaan waktu dalam detik antara *attacker* dan Snort deteksi
m^a = Satuan menit waktu dari *attacker*.
s^a = Satuan detik waktu dari *attacker*.
m^b = Satuan menit waktu dari Snort deteksi.
s^b = Satuan detik waktu dari Snort deteksi.

Sedangkan untuk mendapatkan perbedaan waktu antara pengiriman dan penerimaan pesan dapat menggunakan rumus berikut:

$S^y = ((m^d \times 60 + s^d) - ((m^c \times 60) + s^c)$
$S^y = (60m^d + s^d) - (60m^c + s^c)$ (2)

Penjelasan:
S^y = Perbedaan waktu dalam detik antara kirim dan terima.
m^c = Satuan menit waktu dari kirim.
s^c = Satuan detik waktu dari kirim.
m^d = Satuan menit waktu dari terima.
s^d = Satuan detik waktu dari terima.

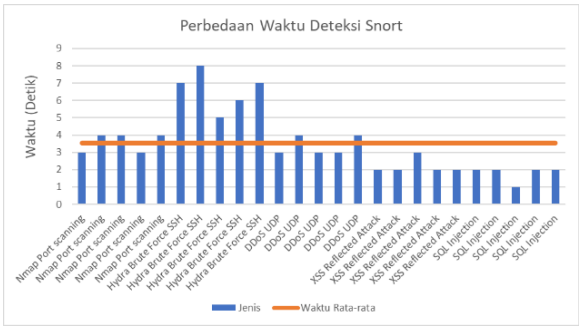
Analisis data dilakukan untuk memvisualisasikan grafik yang menunjukkan perbedaan waktu antara pengiriman dan penerimaan pesan. Gambar 11 menampilkan aliran tingkat transformasi pada aplikasi Telegram *Messenger* dalam proses pengiriman pesan informasi.



Gambar 11. Grafik perbedaan waktu dikirim dan diterima

Gambar 1 menunjukkan analitik perbedaan waktu antara pengiriman dan penerimaan pesan melalui Telegram *Bot*. Grafik tersebut memperlihatkan variasi waktu pengiriman pesan selama simulasi berlangsung

Data analitik juga digunakan untuk melihat grafik data perbedaan waktu antara deteksi oleh Snort dan waktu awal serangan. Gambar 12 menunjukkan aliran tingkat deteksi Snort untuk mendeteksi paket data berbahaya.



Gambar 12. Grafik perbedaan waktu deteksi snort

Gambar 12 menampilkan analitik perbedaan waktu antara awal serangan dan deteksi serangan oleh Snort, berdasarkan data paket berbahaya yang terdeteksi. Grafik tersebut menunjukkan adanya variasi waktu di tengah simulasi.

Hasil pengujian sistem disajikan dalam bentuk tabel untuk menilai apakah sistem berfungsi dengan baik atau tidak. Secara rinci, hasil pengujian dapat dilihat pada Tabel .

Tabel 3. Hasil pengujian sistem

No	Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
1	NMAP FIN Scan	Terdeteksi	Terdeteksi	Berhasil
2	*Brute Force* SSH	Terdeteksi	Terdeteksi	Berhasil
3	DDoS UDP	Terdeteksi	Terdeteksi	Berhasil
4	XSS Attack	Terdeteksi	Terdeteksi	Berhasil
5	SQL *Injection*	Terdeteksi	Terdeteksi	Berhasil

Berdasarkan Tabel , seluruh pengujian mendapatkan hasil yang sesuai dengan yang diharapkan. Sistem berhasil mendeteksi berbagai jenis serangan yang dilakukan oleh *attacker*, seperti *Port Scanning, Brute Force SSH, DDoS, XSS Reflected*, dan *SQL Injection*. Sistem yang dibangun telah diuji dan terbukti mampu mendeteksi intrusi.

Tabel 4. Hasil pengujian telegram bot

No	Uji Coba	Hasil Pengujian	Kesimpulan
1	Notifikasi alert Snort ke Telegram *Bot* secara *real time*	Terdeteksi	Berhasil
2	Notifikasi laporan insiden grafik harian	Terdeteksi	Berhasil
3	Memblock IP *Address attacker* dengan *iptables*	Terdeteksi	Berhasil
4	Membuka IP *Address attacker* dengan *iptables*	Terdeteksi	Berhasil

Hasil pengujian Telegram *Bot* disajikan pada Tabel . Seluruh pengujian mendapatkan hasil yang sesuai dengan yang diharapkan. Notifikasi *alert* secara *real-time* dan notifikasi laporan insiden grafik harian berhasil sesuai yang diharapkan. Selain itu, aplikasi *instant messaging* Telegram mampu memblokir dan membuka IP *address attacker* melalui Telegram *Bot* menggunakan perintah *iptables*, yang merupakan *firewall* dari sistem operasi berbasis GNU/Linux.

5. KESIMPULAN DAN SARAN

Penerapan Snort dan Telegram *Bot* menjadi kolaborasi dalam menjaga keamanan jaringan dan *server*. Dari pengujian lima jenis penetrasi selama 25 kali pengujian berhasil mendeteksi paket data yang mencurigakan. Terdapat keterlambatan pengiriman pesan dari Telegram dengan nilai 2,76 detik. *Prevention system* dalam pemantauan ini dapat memblokir dan membuka IP *Address attacker* melalui Telegram *Bot* menggunakan perintah *iptables* linux. Sehingga dapat meningkatkan efisiensi dan efektivitas dalam pengelolaan keamanan jaringan dan *server*.

DAFTAR PUSTAKA

[1] N. Dwipoyono, Khairil, and A. Sudarsono, "Penerapan Firewall Pada Sistem Keamanan Jaringan Komputer Di Sekolah SMK Negeri 5 Seluma," *Jurnal Media Infotama*, vol. 19, no. 2, p. 454, 2023.

[2] D. T. Yuwono, "Analysis Performance Intrusion Detection System in Detecting Cyber-Attack on Apache Web Server," *IT Journal Research and Development*, pp. 169–178, Feb. 2022, doi: 10.25299/itjrd.2022.7853.

[3] Y. Arta, A. Syukur, and R. Kharisma, "Simulasi Implementasi Intrusion Prevention System (IPS) Pada Router Mikrotik," *IT Journal Research and Development*, vol. 3, no. 1, pp. 104–114, Aug. 2018, doi: 10.25299/itjrd.2018.vol3(1).1346.

[4] W. W. Purba and R. Efendi, "Perancangan dan analisis sistem keamanan jaringan komputer menggunakan SNORT," *AITI: Jurnal Teknologi Informasi*, vol. 17, no. Agustus, pp. 143–158, 2020.

[5] I. P. G. A. Sudiatmika, I. P. Y. A. Ariwanta, and I. G. A. S. Melati, "Mengoptimalkan Keamanan Jaringan Komputer Menggunakan Snort dan Telegram Bot yang Terintegrasi dengan Mikrotik," *Journal of Computer System and Informatics (JoSYC)*, vol. 3, no. 4, Aug. 2022, doi: 10.47065/josyc.v3i4.2037.

[6] A. Krishna *et al.*, "Intrusion Detection and Prevention System Using Deep Learning," pp. 273–278, 2020, doi: 10.1109/ICESC48915.2020.9155711.

[7] L. Feronika Nainggolan, N. F. Saragih, F. G. N. Larosa, and H. Artikel, "Monitoring Keamanan Jaringan Pada Server Ubuntu Dari Serangan DDoS Menggunakan Snort IDS," 2022. [Online]. Available: http://ojs.fikom-methodist.net/index.php/METHOTIKA

[8] J. Lirama Junior Pandari, W. Sulistyo, and U. Kristen Satya Wacana, "Implementasi Intrusion Detection System (IDS) Untuk Mendeteksi Serangan Metasploit Exploit Menggunakan Snort dan Wireshark," *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, vol. 6, no. 1, pp. 41–50, 2023.

[9] D. Santoso, A. Noertjahyana, and J. Andjarwirawan, "Implementasi dan Analisa Snort dan Suricata Sebagai IDS dan IPS Untuk Mencegah Serangan DOS dan DDOS," 2022.

[10] S. Khadafi, Y. Dian Pratiwi, E. Alfianto, P. Studi Sistem Komputer, F. Teknik Elektro dan Teknologi Informasi, and T. Adhi Tama Surabaya, "Keamanan FTP Server Berbasiskan IDS dan IPS Menggunakan Sistem Operasi Linux Ubuntu," *Jurnal Ilmiah NERO*, vol. 6, no. 1, pp. 11–24, 2021.

[11] The Snort Project, "SNORT R Users Manual 2.9.16 The Snort Project," 2020.

[12] J. F. Fahana and F. Ridho, "Pemanfaatan Telegram Sebagai Notifikasi Serangan Untuk Keperluan Forensik Jaringan," 2017.

[13] S. Tyagi and K. Kumar, "Evaluation of Static Web Vulnerability Analysis Tools," *Institute of Electrical and Electronics Engineers*, pp. 1–6, 2018, doi: 10.1109/PDGC.2018.8745996.

[14]A. Erlansari, F. Farady Coastera, and A. Husamudin, "Early Intrusion Detection System (IDS) using Snort and Telegram approach," *SISFORMA Journal of Information Systems (e-Journal)*, vol. 7, no. 1, 2020, doi: 10.24167/Sisforma.