ISSN 2477-0043 Jurnal Teknologi Terpadu
e-ISSN 2460-7908 Vol. 5, No. 2, Desember 2019

Sistem *Reporting* Keamanan pada Jaringan *Cloud Computing* Melalui *bot* Telegram dengan Menggunakan Teknik *Intrussion Detection and Prevention System*

Eddy Prasetyo Nugroho[1], Eki Nugraha[2], Mokhamad Nizar Zulfikar [3]

Departemen Pendidikan Ilmu Komputer, FPMIPA,

Universitas Pendidikan Indonesia
Bandung, Indonesia
[1]eddypn@upi.edu, [2]ekinugraha@upi.edu, [3]zayzitations@student.upi.edu

ABSTRAK

Serangan Siber merupakan ancaman yang serius bagi kemanan jaringan, terutama pada jaringan yang bersifat publik sehingga bisa diakses oleh siapapun dari seluruh dunia. Akibat dari serangan siber sangat berdampak besar jika sampai berhasil mengganggu suatu kinerja jaringan bahkan hingga bisa menguasainya, terutama pada jaringan yang menyediakan layanan bagi public. Seorang administrator jaringan harus siap tanggap dalam menangani setiap serangan pada server yang dikelolanya. Langkah pertama dalam mencegah serangan yang bisa mengancam suatu jaringan yaitu dengan merancang suatu sistem untuk mendeteksi dan memberikan peringatan dini akan adanya suatu serangan yang dinamakan Intruision Detection System (IDS). Aplikasi yang digunakan sebagai IDS yaitu Snort yang berfungsi untuk mendeteksi serangan berdasarkan rules yang akan dicocokan dengan signature dari serangan tersebut, dan akan disimpan ke database untuk diteruskan kepada administrator melalui aplikasi instant messaging Telegram. Telegram digunakan sebagai media untuk menyempaikan peringatan dini jika terjadi upaya serangan, sehingga administrator dapat melakukan upaya penanggulangan terhadap serangan tersebut. Untuk menanggulangi serangan yang terjadi, digunakan aplikasi Fail2Ban dan Port Scan Attack Detector (PSAD) untuk menutup akses dari IP penyerang. Hasil dari penggunaan IDS dengan notifikasi melalui bot Telegram menggunakan bahasa pemrograman PHP mampu mendeteksi serangan DoS, Port Scanning, dan SSH Bruteforce berdasar rules yang dikonfigurasi pada Snort. Berdasarkan hasil analisis respon waktu pengiriman notifikasi didapatkan hasil yaitu SSH Bruteforce 18 detik, Port Scanning 20 detik, dan DoS yaitu 30 detik.

Keyword**:** *kemanan jaringan, cloud computing, intrusion detection and prevention system (IDSP), snort,* telegram, bot

I. PENDAHULUAN

Perkembangan teknologi semakin hari semakin berkembang dengan pesat, begitu pula dengan ancaman yang menghantui sisi kemanan dari teknologi tersebut. Internet merupakan teknologi yang memiliki perkembangan sangat pesat dalam kurun waktu 40 tahun terakhir. Ancaman terhadap keamanan jaringan internet selalu menjadi momok menakutkan terutama bagi perusahaan yang bergerak dalam bidang teknologi khususnya yang memanfaatkan teknologi *cloud computing*. Terlebih lagi lebih dari 1 dekade terakhir masyarakat kita sudah bergantung pada teknologi. Orang bergantung pada jaringan komputer untuk mendapatkan berita, berkomunikasi, belanja, hingga untuk keperluan penyimpanan file yang bersifat pribadi [1].

Sistem *cloudcomputing* dapat dengan mudah terancam oleh berbagai serangan cyber, karena sebagian besar sistem *cloud* memberikan layanan kepada begitu banyak orang yang tidak terbukti dapat dipercaya. Oleh karena itu, sistem *cloud* memerlukankomponen keamananyaitu berupa *Intrussion Detection and Prevention System* (IDPS) untuk melindungi setiap mesin virtual yang terdapat pada server cloud terhadap berbagai jenis ancaman [2].

Kemanan suatu sistem sangat berpengaruh baik bagi penyedia layanan jasa pada bidang teknologi mau pun pengguna pribadi, sehingga dibutuhkan sebuah sistem untuk me-*monitoring* keamanan dari suatu jaringan. Oleh karena itu peran administrator sangat berpengaruh terutama dalam memantau dan

memastikan bahwa server yang dikelolanya selalu aman dari berbagai potensi ancaman, sehingga disaat ada ancaman yang masuk server admin harus mengetahui ancaman yang sedang dihadapinya dan cara mengatasinya. Seorang server admin tidak sepenuhnya selalu berada di dekat server yang menjadi tanggung jawabnya, maka dibutuhkan cara cepat untuk dapat berkomunikasi dengan server.

Telegram merupakan layanan *instant messaging* yang berbasis *open-source* yang kini tengah populer, terlebihdengan adanya fitur *bot*dengan memanfaatkan berbagai API yang sudah disediakan oleh Telegram. Sehingga *bot* Telegram bisa mengirimkan pesan mau pun perintah secara otomatis sesuai dengan fungsi dari *bot* tersebut. Pemilihan Telegram sebagai media untuk notifikasi pada penelitian ini dikarenakan jumlah notifikasi dari suatu sistem keamanan bisa sangat banyak. Pada Telegram semua file mau pun pesan yang masuk disimpan di *cloud* sehingga tidak akan menguras memori dari *smartphone* administrator serta proses pengiriman jauh lebih cepat.

Pemanfaatan *Intrusion Detection and Prevention System* (IDPS) selain berfungsi untuk mendeteksi serangan yang terjadi pada jaringan, juga untuk mencatat setiap aktifitas intrusi yang terjadi. Data intrusi yang dicatat ke dalam *database* yang memuat informasi mengenai sumber serangan, waktu kejadian, jenis serangan, serta dampak yang dihasilkan bisa digunakan sebagai keperluan analisis forensik jaringan yang biasanya dilakuakn oleh administrator jaringan.

II. PENELITIAN TERKAIT

Perancangan *Intrusion Detection and Prevetion System* (IDPS) dibuat untuk mengatasi serangan yang terjadi pada server yang mana diperiksa terlebih dahulu pada pc-router yang sudah terpasang aplikasi PSADbeserta konfigurasi *firewall* untuk memeriksa tiap paket yang masuk untuk kemudian paket yang lolos akan diberikan akses ke server. PSAD akan memblokir akses IP yang terdeteksi melakukan intrusi sehingga tidak bisa mengakses server *resource*.Serangan yang diuji cobakan pada penelitian ini yaitu *port scanning* dan DoS ICMP *Flood*[3].

Pada penelitian [4] notifikasi intrusi dirancang dengan menggunakan aplikasi Gammu sebagai media untuk menyampaikan notifikasi berupa *Short Message Service* (SMS) dari server yang dipantau. Notifikasi yang disampaikan kepada admin yaitu berupa pesan yang berisi teks yang menginformasikan mengenai jenis serangan dan waktu kejadiannya, sementara untuk sumber dari serangan tersebut tidak dicantumkan dan administrator harus melakukan tindak lanjut secara manual terhadap paket-paket yang mencurigakan melalui perangkat desktop server.

Serangan yang menjadi fokus penelitian ini yaitu *ICMPFlooding*, *DDoSattack*, dan *port scanning*.

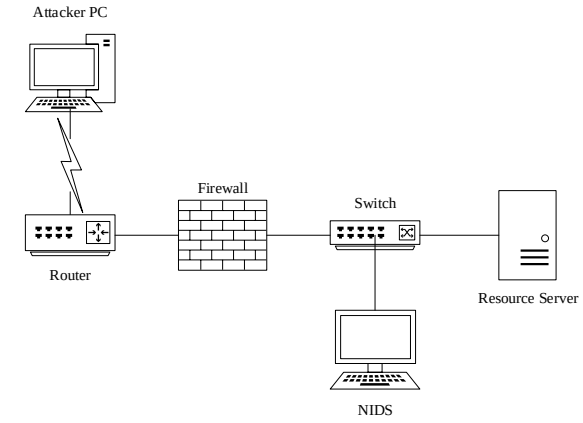
Penelitian mengenai keamanan jaringan dengan menggunakan Teknik *Intrusion Detection System* (IDS) juga dilakukan pada penelitian [5]. Sistem tersebut memiliki kemampuan untuk mengirimkan notifikasi kepada administrator melalui media media SMS dengan menggunakan bahasa pemrograman Java sebagai antar muka dari aplikasi tersebut. Pada penelitian [6] perangkat yang ke sebuah jaringan lokal, yang mana jaringan lokal tersebut sudah dipasangi sistem *monitoring* oleh administrator untuk memantau perangkat apa saja yang sedang terhubung. Penelitian ini menggunakan metode *Signature based Intrusion Detection System*, sehingga setiap aktifitas lalu lintas paket pada jaringan akan dipantau dan dicocokan dengan *signature* dari Snort yang selalu diupdate untuk menghindari adanya celah keamanan yang sewaktu-waktu bisa ditembus oleh penyusup.

III. USULAN METODE

Dalam penelitian ini, menggunakan Teknik *Intrusion Detection and Prevention System* (IDPS), yang merupakan gabungan dari Teknik IDS dan IPS yang berfungsi sebagai pencegahan dan peringatan dini akan adanya suatu intrusi. IDS merupakan aktifitas pemantauan peristiwa yang terjadi dalam suatu sistem jaringan komputer kemudian dianalisis untuk menemukan kemungkinan adanya interfensi asing kedalam sistem, yang merupakan pelanggaran atau ancaman terhadap kebijakan keamanan komputer, atau praktek standar keamanan[7]. Sedangkan IPS merupakan sistem perangkat lunak yang memiliki semua kemampuan pendeteksian intrusi yang memiliki kemampuan untuk menghentikan kemungkinan ancaman lebih lanjut dari serangan yang sedang berlangsung, dengan melakukan pemblokiran terhadap alamat yang dianggap tidak aman.

Pada penelitian yang dilakukan, menggunakan jenis IDS berdasarkan penempatannya, yaitu *Network Intrusion Detection System* (NIDS). Jenis penempatan IDS ini menempatkan sistem IDS sebagai pintu gerbang menuju jaringan yang dilindungi nya. Setiap akses yang menuju ke jaringan server terlebih dahulu akan melewati sistem IDS untuk dilakukan proses pemeriksaan dan pencocokan dengan *rules* aplikasi IDS (dalam hal ini Snort) terhadap paket-paket yang

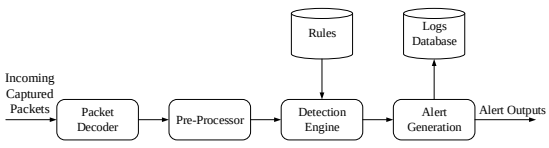
masuk. Berikut arsitektur jaringan pada NIDS dapat dilihat pada gambar 1.



Gambar 1Arsitektur Network Intrusion Detection System

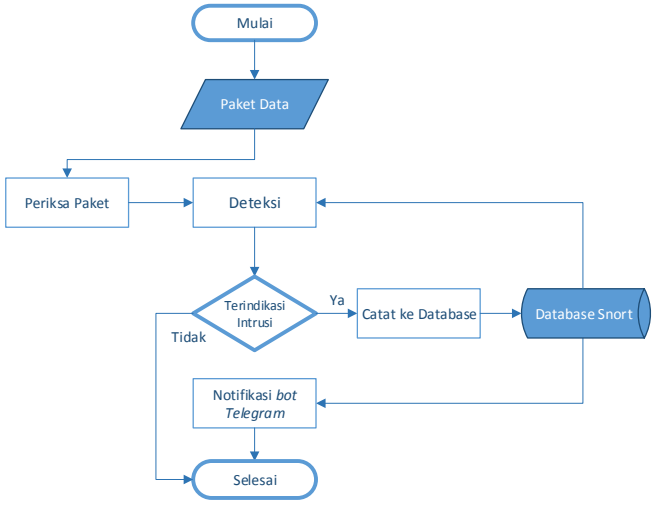
Berdasarkan gambaran arsitektur jaringan pada gambar 1, membuktikan bahwa IDS memiliki peranan penting dalam me-*monitoring* arus lalu lintas jaringan yang keluar masuk dari jaringan tersebut. Tiap paket yang masuk ke jaringan yang memiliki pertahanan IDS akan melewati proses pemeriksaan.

Dimulai dengan memisahkan paket yang melalui Ethernet card untuk diperiksa oleh snot. Untuk selanjutnya akan memasuki proses *packet decoder*, pada proses tersebut akan dilakukan pemisahan data *header* dari paket yang masuk untuk diambil data mengenai paket tersebut sehingga snort bisa memiliki informasi mengenai protocol dari paket yang sedang diperiksa. Pada proses *pre-processor* paket akan dimanipulasi dengan ditandai atau dikelompokan berdasarkan jenis paketnya. Setelah paket dikelompokan, maka akan dilakukan pencocokan dengan *rules* snort yang disimpan pada database rules, apakah paket tersebut memiliki kecocokan dengan *signature* yang mengindikasikan bahwa paket tersebut merupakan serangan yang ditujukan ke server atau tidak. Setiap paket yang terindikasi memiliki kecocokan dengan *signature* dari suatu serangan, snort akan menghasilkan *output* berupa pencatatan aktifitas intrusi yang sudah berhasil dideteksi. Sebagaimana yang terlihat pada gambar 2.



Gambar 2 Arsitektur Snort

Berdasarkan skema pada gambar 2, hasil dari pencatatan intrusi yang dilakukan snort bisa berupa format teks atau yang dikenal dengan, XML, binary format,tcpdump, syslog, atau Database. Pada penelitian ini, output dari snort masih dalam format biner, kemudian dengan menggunakan aplikasi Barnyard2 output tersebut diolah dan dimasukan ke database untuk menampungcatatan alert agar lebih mudah untuk dipahami dan kemudian bisa diolah untuk berbagai keperuan analisis forensik jaringan. Untuk alur pendeteksian pada penelitian ini memiliki kesamaan dengan arsitektur snort. Sebagaimana yang terlihat pada gambar 3.

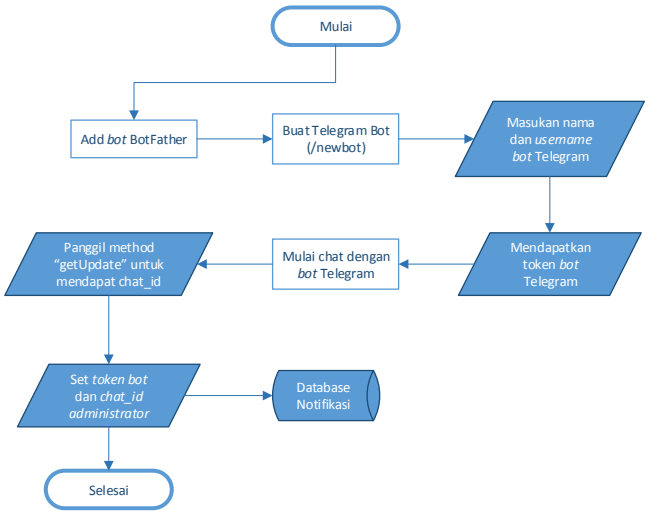


Gambar 3 Flowchart pendeteksian serangan

Diagram pada gambar 3 menjelaskan mengenai alur sebuah paket yang masuk ke dalam lalu lintas jaringan yang mana akan terlebih dahulu diperiksa. Dalam hal ini snort seperti polisi lalu lintas sedang melakukan razia, dan rules snort merupakan aturan perundang-undangan lalu lintas nya. Pada tahap pertama, paket-paket data yang masuk akan diperiksa dan dianalisa dengan melakukan pencocokan terhadap rules yang sudah didefinisikan. Jika cocok dan terbukti berisi ancaman, paket tersebut akan dicatat ke dalam format biner.

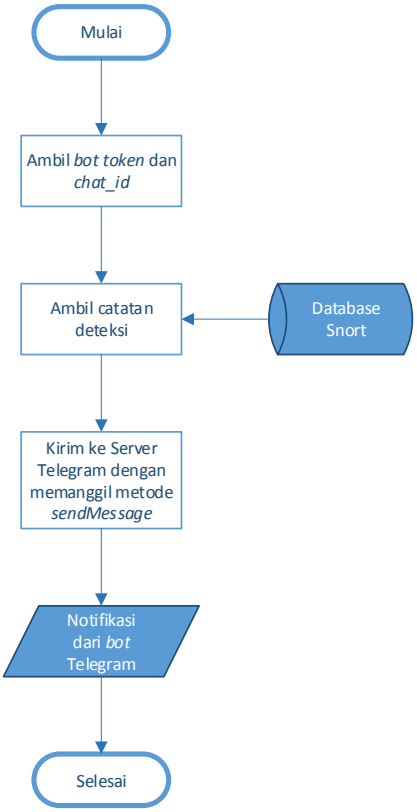
Pada proses selanjutnya yaitu pembuatan *bot* Telegram. Proses pembuatan *bot* dimulai dengan meminta pada akun bot resmi Telegram yaitu @BotFather. Dengan memasukan perintah untuk meminta *bot*baru hingga mendapatkan token dari *bot* yang akan digunakan sebagai media untuk mengirimkan notifikasi kepada administrator. Selain

token, *chat_id* dari akun Telegram administrator dibutuhkan agar notifikasi diterima oleh orang yang tepat. Untuk mendaptkan *chat_id*, tinggal memanggil kode API Telegram *getUpdates* dengan memasukan token dari *bot* yang dimaksud terlebih dahulu.Data notifikasi yang dikirimkan oleh *bot* Telegram sepenuhnya berasal dari 1 database yang sama dengan yang digunakan untuk menampung *alert* dari Snort. yang mana diagram arus dari proses tersebut seperti terlihat pada gambar 4 di bawah ini.



Gambar 4 Flowchart pembuatan bot Telegram

Seperti yang sudah dijelaskan pada proses sebelumnya, data notifikasi yang dikirimkan berasal dari 1*database* yang sama dengan yang digunakan oleh snort untuk menyimpan hasil *allert* nya. Token dari *bot* berfungsi untuk mengakses API Telegram, sehingga tidak sembarangan orang bisa mengakses API Telegram jika tidak memiliki token dari *bot* yang akan dijadikan sebagai media perantara pengiriman notifikasi. Berikut *flowchart* dari alur pengiriman notifikasi, sebagaimana yang tertera pada gambar 5.

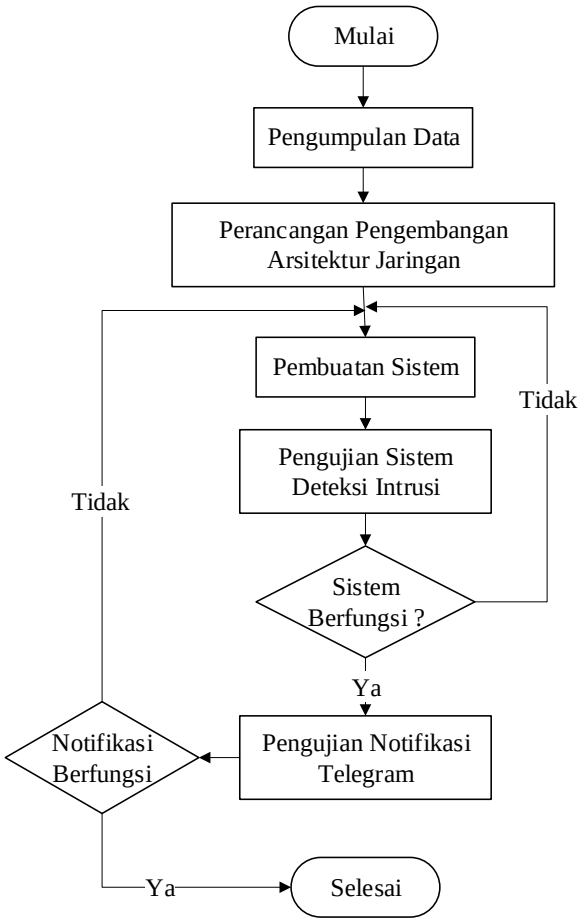


Gambar 5 Flowchart pengiriman notifikasi

IV. *ROADMAP* IMPLEMENTASI

Pada penelitian yang dilakukan ini, hasil akhirnya yaitu berupa notifikasi setiap aktifitas intrusi yang berhasil dideteksi oleh Snort berdasarkan pencocokan dengan *rules*yang sudah diatur. Sistem notifikasi monitoring kemanan jaringan tentunya memerlukan waktu respon yang cepat agar bisa diterima dengan segera oleh administrator. Hal tersebut dibutuhkan agar administrator mengetahui langkah apa yang harus dilakukan dalam mengatasi serangan yang sedang terjadi. Jika berbahaya maka dibutuhkan penanganan lebih lanjut agar server tidak mengalami gangguan yang signifikan.

Dalam proses penelitian yang dilakukan ini, diperlukan beberapa tahapan proses untuk bisa membangun sebuah sistem IDS dengan hasil akhir berupa notifikasi yang sampai ke tangan administrator melalui media Telegram. Berikut proses yang dilakukan dalam membangun sistem ini, sebagaimana yang tertera pada gambar 6 dibawah ini.



Gambar 6 Flowchart penelitian

Langkah dari penelitian ini yaitu sebagai berikut:

1. Pengumpulan Data

Pengumpulan data yang dilakukan adalah untukmemperoleh informasi mengenai analisis kebutuhan sistem, analisis jenis intrusi yang akan diuji coba, analisis kebutuhan jaringan, analisis fungsional dan proseskerja snort.

2. Perancangan Pengembangan Arsitektur Jaringan

Perancangan arsitektur jaringan yang akan dipakai sebagai infrastruktur dalam membangun sistem IDS. Mencakup penentuan peletakan IDS pada server berdasarkan jenis IDS yang digunakan, yang mana dalam penelitian ini menggunakan jenis *Network Intrusion Detection System* (NIDS).

3. Pembuatan Sistem

Tahap ini merupakan awal dilakukannya instalasi berbagai kebutuhan perangkat yang akan digunakan pada sistem IDS. Diawali dengan instalasi dan konfigurasi LAMP (Linux, Apache, MySQL, PHP) sebagai modal dasar dalam membangun computer server. Dilanjutkan dengan proses instalasi *library* pendukung diantaranya seperti *Data Acquisition* (DAQ), AdoDB, Libpcap untuk sistem IDS yaitu Snort, Barnyard2, dan BASE, kemudian dilanjutkan dengan instalasi ketiga aplikasi tersebut. Langkah berikutnya yaitu konfigurasi Snort, termasuk di dalamnya konfigurasi *rules* snort. Dilanjutkan dengan konfigurasi Barnyard2 dan BASE, serta pembuatan *bot* Telegram melalui akun @BotFather. Dan dilanjutkan dengan pembuatan *engine bot* serta sistem Web UI Snort untuk monitoring melalui *website*. Serta penambahan *library* terpenting pada sistem ini, yaitu MySQL UDF untuk mengeksekusi aplikasi di luar MySQL melalui *trigger*.

4. Pengujian Sistem Deteksi Intrusi

Pada proses pengujian deteksi intrusi ini bertujuan untuk mengetahui apakah Snort sudah bisa mendeteksi serangan yang masuk berdasarkan *rules* yang sudah dibuat. Sehingga serangan yang masuk bisa dimasukan ke *database* untuk kemudian digunakan sebagai sumber data untuk notifikasi yang akan dikirimkan kepada administrator.

5. Pengujian Notifikasi Telegram

Tahap ini menjadi patokan apakah sistem yang dibangun sudah sesuai dengan yang diharapkan. Pada proses ini akan dilakukan uji coba serangan intrusi terhadap server, sehingga ketika ada intrusi yang berhasil dideteksi oleh Snort, memicu *trigger* yang akan memanggil *engine bot* untuk mengirimkan notifikasi kepada administrator melalui *bot* Telegram.

Penghitungan responsibilitas sistem IDS baik responsibilitas dari lamanya sistem untuk mendeteksi serangan hingga berhasil mengirimkan notifikasi kepada administrator. Hasil dari pengujian tingkat responsibilitas tersebut digunakan untuk mengukur

seberapa efektif kah sistem IDS yang dibangun beserta penggunaan Telegram sebagai media untuk menyampaikan notifikasi intrusi.

Untuk mendapatkan waktu kecepatan deteksi dihitung berdasarkan rata-rata yang diperoleh dari selisih waktu dimulainya serangan dengan waktu serangan tersebut dideteksi oleh sistem IDS. Selain itu bisa didapatkan waktu kecepatan notifikasi yang diperoleh dari selisih waktu serengan tersebut dideteksi dengan waktu notifikasi sampai ke tangan administrator melalui bot Telegram.

$$\boldsymbol{RK} = \frac{(\boldsymbol{Wk\,1 - Wd\,1}) + \cdots + (\boldsymbol{Wk\,n - Wd\,n})}{\boldsymbol{n}} \qquad (1)$$

Keterangan:

RK = Rerata Kirim

Wn = Waktu Notifikasi

Wk = Waktu Kirim

n = banyaknya percobaan

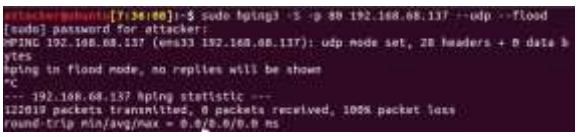
Untuk menghitung tingkat responsibilitas dari sistem IDS yang dibuat, baik tingkat responsibilitas deteksi mau pun notifikasi, dapat diperoleh dari rata-rata waktu respon deteksi tiap jenis serangan untuk responsibilitas deteksi. Dan untuk mendapatkan responsibilitas kirim diperoleh dari rata-rata respon kirim teiap jenis serangan.

V. PENGUJIAN DAN HASIL

Pada penelitian ini, dilakukan pengujian simulasi serangan dengan menggunakan 3 jenis serangan, yaitu DoS, *Port Scanning*, dan *SSH Bruteforce.*

1. Pengujian Serangan DoS

Serangan DDoS dilakukan dengan menggunakan bantuan aplikasi hping3 yang penulis instal pada komputer penyerang. Target dari pengujian ini tentunya komputer server yang memiliki alamat IP 192.168.68.137 melalui port 80 dengan menggunakan protokol UDP yang dilakukan pada pukul 7:36:08.



Gambar 7 Simulasi serangan DDoS

Serangan yang dilakukan oleh penulis diteruskan ke akun telegram administrator melalui pesan singkat yang dikirim oleh bot seperti terlihat pada gambar 8, dengan keterangan sebagai berikut : jenis serangan yang dilakukan terdeteksi sebagai serangan DDoS UDP yang sesuai dengan signature pada database Snort yang memiliki SID 1:10000008:1. Notifikasi serangan berhasil diterima oleh administrator pada pukul 7:36:28 seperti pada gambar 9.

Gambar 8 Notifikai Serangan DDoS

Gambar 9 Alert Serangan DDoS pada database

2. Pengujian Serangan *Port Scanning*

Pengujian serangan Port Scanning dilakukan dengan menggunakan bantuan aplikasi nmap pada perangkat penyerang. Jenis serangan yang dilakukan yaitu XMAS Scan, serangan dilakukan pada pukul 8:33:20 dengan IP tujuan yaitu 192.168.68.137 yang merupakan IP Server yang menjadi target serangan.

Gambar 10 Simulasi serangan Port Scanning

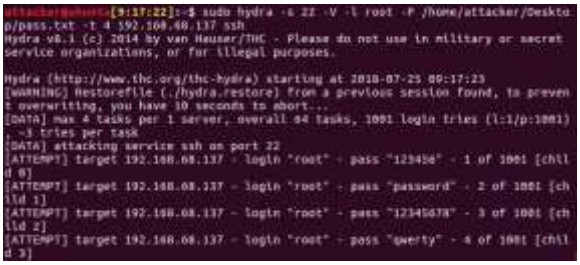
Administrator menerima peringatan akan terjadinya serangan pada server melalui pesan singkat Telegram yang dikirim oleh bot. Pesan yang diterima memberitahukan akan adanya jenis serangan NMAP Scan XMAS yang dikenal pada signature Snort dengan nomor SID 1:1228:7. Pesan notifikasi berhasil diterima oleh administrator pada pukul 8:33:22 seperti yang terlihat pada gambar.

Gambar 11 Notifikai Serangan DDoS

Gambar 12 Alert Serangan Port Scanning pada database

3. Pengujian Serangan *SSH Bruteforce*

Pengujian dengan melakukan serangan bruteforce ssh dilakukan dengan menggunakan aplikasi hydra. Sebagaimana nama dari jenis serangannya, serangan ini menargetkan protocol SSH melalui port 22. Penulis menggunakan kumpulan teks yang disimpan dalam file pass.txt



Gambar 13 Simulasi serangan SSH Bruteforce

Serangan yang sudah berlangsung diteruskan ke administrator melalui pesan singkat telegram. Jenis serangan SSH Bruteforce dikenali oleh signature snort dengan nomor SID 1:10000006:1 yang diterima oleh administrator pada pukul 9:17:33 sesuai dengan keterangan pada gambar.

Gambar 14 Notifikai Serangan DDoS

Gambar 15 Alert Serangan Port Scanning pada database

Setelah dilakukan pengujian terhadap system IDS dimulai dari iawal dilancarkan serangan hingga notifikasi sampai di tangan administrator, didapatkan catatan waktu untuk bisa menjadi tolak ukur kecepatan respon dari system notifikasi IDS yang dibangun. Hasil dari catatan waktu ini tentunya berdasarkan penmpatan servernya, yaitu di virtual machine, hasil waktu akan berbeda jika ditempatkan pada lingkungan server yang memiliki kemampuan pemrosesan yang lebih cepat.

Berdasarkan rumus perhitungan (1) dapat diperoleh hasil rerata deteksi serta dengan rumus perhitungan (2) diperoleh hasil rerata kirim. Berdasarkan penggunaan kedua rumus tersebut, dapat diperoleh hasil perhitungan yang diambil dari 10 kali percobaan dari masing-masing, sebagaimana yang terdapat pada table 1, table 2, dan table 3.

TABEL 1. DATA PENCATATAN WAKTU PERCOBAAN SSH BRUTEFORCE

No	SSH Bruteforce		
	Waktu Deteksi	Waktu Notifikasi	Selisih (detik)
1	8/15/2018 12:50	8/15/2018 12:50	0:00:24
2	8/15/2018 12:50	8/15/2018 12:50	0:00:22
3	8/15/2018 12:50	8/15/2018 12:50	0:00:20
4	8/15/2018 12:49	8/15/2018 12:49	0:00:17
5	8/15/2018 12:49	8/15/2018 12:49	0:00:16
6	8/15/2018 12:49	8/15/2018 12:49	0:00:14

TABEL 2. DATA PENCATATAN WAKTU PERCOBAAN DDOS

No	Denial of Service		
	Waktu Deteksi	Waktu Notifikasi	Selisih (detik)
1	8/14/2018 12:50	8/14/2018 12:50	0:00:11
2	8/14/2018 12:50	8/14/2018 12:50	0:00:09
3	8/14/2018 12:49	8/14/2018 12:50	0:00:19
4	8/14/2018 12:49	8/14/2018 12:50	0:00:07
5	8/14/2018 12:49	8/14/2018 12:49	0:00:06
6	8/14/2018 12:49	8/14/2018 12:49	0:00:12

TABEL 3. DATA PENCATATAN WAKTU PERCOBAAN PORTSCANNING

No	PortScanning		
	Waktu Deteksi	Waktu Notifikasi	Selisih (detik)
1	8/15/2018 15:06	8/15/2018 15:06	0:00:10
2	8/15/2018 14:52	8/15/2018 14:52	0:00:03
3	8/15/2018 14:51	8/15/2018 14:51	0:00:15
4	8/15/2018 14:45	8/15/2018 14:45	0:00:13
5	8/15/2018 14:43	8/15/2018 14:43	0:00:07
6	8/15/2018 14:42	8/15/2018 14:42	0:00:16

Dari hasil perhitungan diatas, diperoleh respon waktu pengiriman notifikasi untuk SSH Bruteforce yaitu 18 detik, Port Scanning 20 detik, dan DoS yaitu 30 detik.

TABEL 4. HASIL NETWORKING DAN PERFORMANCE TESTING

Sebelum Ada Perangkat Keamanan		**Sesudah Ada Perangkat Keamanan**	
Ping	***AVG Time (ms)***	***Ping***	***AVG Time (ms)***
www.upi.edu (Bandung, ID)	12.346	www.upi.edu (Bandung, ID)	12.25
www.telegram.org (Arizona, US)	211.11	www.telegram.org (Arizona, US)	212.17
www.vk.com (Moscow, RUS)	214.076	www.vk.com (Moscow, RUS)	213.662
www.tencent.com (Guang Dong, CN)	257.508	www.tencent.com (Guang Dong, CN)	236.004
www.ubuntu.com (Isle of Man, GB)	277.01	www.ubuntu.com (Isle of Man, GB)	276.551
Throughput	***AVG Speed (KB/s)***	***Throughput***	***AVG Speed (KB/s)***
Test DL Server: speedtest.tele2.net	2100 KB/s	Test DL Server: speedtest.tele2.net	2230 KB/s
Resource	***Load***	***Resource***	***Load***
CPU System	0.70%	CPU System	0.70%
Memory Used	63578 KB	Memory Used	271550 KB
Memory Cached	344457 KB	Memory Cached	499664 KB
Swap Used	12851 KB	Swap Used	12843 KB
Swap Cached	-	Swap Cached	-

Dari hasil testing diatas, didapatan kesimpulan bahwa perangkat kemanan jaringan, dalam hal ini untuk keperluan membangun sistem IDS tidak mengakibatkan efek yang begitu berarti terhadap kinerja server itu sendiri. Bahkan pada beberapa hasil tes di atas, didapatkan waktu proses komunikasi yang lebih cepat pada saat perangkat keamanan diaktifkan. Pada pengujian resource sistem, perangkat keamanan jaringan memakan kebutuhan memori yang tidak begitu besar, serta tidak mempengaruhi kinerja CPU ditunjukan dengan tetap samanya CPU Load baik saat perangkat keamanan aktif mau pun tidak aktif yaitu sebesar 0.7%.

Hasil pendeteksian Snort yang dimasukan ke *database* bisa diakses pada halaman *web*dan langsung ditampilkan pada halaman *dashboard*, seperti pada gambar 16. Unutk notifikasi yang berhasil dikirim akan dicatat ke *database*dan ditampilkan pada halaman *web uimonitoring*snort, sehingga bisa terlihat respon waktu yang diperlukan bagi sistem IDS untuk bisa mengirimkan notifikasi dihitung setelah aktifitas intrusi tersebut dideteksi, untuk hasil pencatatannya dapat dilihat pada gambar 17.

Gambar 17 Dashboard web monitoring Snort

Gambar 18 Hasil pencatatan notifiksi terkirim

VI. KESIMPULAN

Pemanfaatan sebuah *Intrusion Detection System* (IDS) sangat berpengaruh untuk memantau aktifitas lalu lintas pada jaringan. Berdasarkan hasil penelitian pada paper ini, sistem IDS berhasil mendeteksi serangan, yang mana dilakukan terlebih dahulu proses konfigurasi dan penambahan *rules* agar snort bisa mendeteksi serangan berdasarkan pencocokan dengan *signature* yang terdapat pada *rules* tersebut.

Fokus penelitian pada paper ini lebih kepada membangun sebuah sistem deteksi intrusi dengan meenghasilkan *output* tidak hanya berupa catatan aktifitas intrusi pada *database* tetapi juga notifikasi melalui *instant messaging* Telegram. Hasil pencatatan aktifitas intrusi pada *database*digunakan sebagai data untuk melakukan analisis forensik jaringan mengenai identitas dari sumber paket yang masuk, serta untuk menganalisa tingkat responsibilitas sistem IDS baik dalam mendeteksi serangan mau pun mengirimkan notifikasi kepada administrator.

Cepat lambatnya sistem mendeteksi sebuah serangan ditentukan dari jenis serangan tersebut serta pola *signature* dari *rules* snort, yang mana tiap serangan memiliki pengaturan konfigurasi *rules* yang berbeda.

Dan cepat lambatnya suatu sistem IDS mengirimkan notifikasi kepada administrator ditentukan dari kecepatan perangkat tersebut dalam merespon *trigger* MySQL, karena semakin banyak suatu serangan yang terdeteksi, akan menambah antrian proses pengiriman yang mana diproses atu persatu oleh *trigger* untuk menghasilkan *alarm*. Dan berdasarkan*Networking & Performance Testing* dapat diambil kesimpulan bahwa perangkat keamanan tidak begitu berpengaruh terhadap kinerja server.

REFERENSI

[1] Patel, A., Taghavi, M., Bakhtiyari, K., & JúNior, J. C. (2013). An intrusion detection and prevention system in cloud computing: A systematic review. Journal of network and computer applications, 36(1), 25-41.

[2] Lee, J. H., Park, M. W., Eom, J. H., & Chung, T. M. (2011, February). Multi-level intrusion detection system and log management in cloud computing. In Advanced Communication Technology (ICACT), 2011 13th International Conference on (pp. 552-555). IEEE.

[3] Utama, A.T.B. (2013). Perancangan dan implementasi keamanan jaringan untuk *cloudcomputing* menggunakan *firewall* berbasis GNU/Linux. Fakultas Teknik dan Ilmu Komputer, Universitas Komputer Indonesia,Bandung.

[4] Mutaqin, A. F. (2016). Rancang Bangun Sistem Monitoring Keamanan Jaringan Prodi Teknik Informatika Melalui SMS Alert dengan Snort. Jurnal Sistem dan Teknologi Informasi (JustIN), 4(1), 98-103.

[5] Nasution, M. I. P., & Tanjung, F. N. (2012). Implementasi Pemrograman Java Untuk Alert Intrusion Detection System.

[6] Nasution, D. M., Tulus, Sembiring, S. *IntrusionPreventionSystem* dengan Metode *SignatureBasedIntrusionDetection* pada Jaringan*Local Area Network* (LAN). Teknik Informarika. Sekolah Tinggi Teknik Harapan Medan.

[7] Scarfone, K., & Mell, P. (2007). Guide to intrusion detection and prevention system (IDPS), NIST special publication, 800(2007), 94.