Notifikasi *Network Intrusion Detection System* Menggunakan Media Aplikasi Telegram
(Studi Kasus: Kantor Imigrasi Tasikmalaya)

Fitri Nuraeni[1)], Indra Nurfajri[2)]
STMIK Tasikmalaya, Jl. RE Martadinata No.272 A Kota Tasikmalaya
Teknik Informatika
Email : nenk.ufit@gmail.com [1)], basircomp@gmail.com [2)]

Abstrak

Pada jaringan local area network sering terdapat keluhan seperti sering terjadinya gangguan pada server dimana gangguan tersebut bisa berasal dari pihak-pihak yang tidak bertanggungjawab / penyusup (intruder) dengan memanfaatkan kelemahan sistem keamanan jaringan local area network yang terhubung dengan server baik itu melalui media kabel maupun nirkabel. Untuk menanggulangi hal tersebut diperlukan Intrusion Detection System untuk mendeteksi adanya aktivitas jaringan yang mencurigakan dan mengirimkan notifikasi peringatan kepada administrator dengan cepat dan efektif melalui media yang popular saat ini seperti aplikasi Telegram Messenger yang digunakan pada smartphone. Oleh karena itu dibangun suatu sistem deteksi dan notifikasi dengan menggunakan metode Network Development Life Cycle (NDLC). Penelitian dilakukan dengan tahapan-tahapan analysis, design, simulation prototyping, implementation, monitoring dan management. Sistem ini menggunakan Snort sebagai sensor IDS dengan database MySQL, Acidbase sebagai web front-end untuk mengelola data alerting yang dideteksi oleh snort, kemudian menggunakan account Bot API Telegram sebagai media notifikasi kepada administrator. Dengan diimplementasikannya telegram sebagai media notifikasi Intrusion Detection System ini, diharapkan administrator dapat mengetahui ada atau tidak adanya aktivitas mencurigakan yang mengancam keamanan jaringan, sehingga administrator dapat melakukan pemulihan sistem jaringan dengan cepat.

***Kata Kunci** : IDS (Intrusion Detection System), LAN, Server, Snort, Telegram BOT*

Abstract

On the network local area network often there are complaints such as frequent disruptions in the server where the disorder can come from parties who are not responsible / intruder (Intruder) by exploiting weaknesses in the network security system local area network connected to the server either through cable media and wireless. To overcome this required Intrusion Detection System to detect suspicious network activity and sends notifications to alert administrators to quickly and effectively through popular media today as Telegram Messenger applications used on smartphones. Therefore built a system of detection and notification using the Network Development Life Cycle (NDLC). The study was conducted with the stages of analysis, design, simulation prototyping, implementation, monitoring and management. The system uses Snort as an IDS sensor with MySQL database, Acidbase as a web front-end to manage data alerting detected by snort, then use the account Bot API Telegram as media notifications to the administrator. With the implementation of the telegram as media notification Intrusion Detection System, it is expected the administrator can determine the presence or absence of suspicious activity that threatens the security of the network, so administrators can perform system recovery network quickly.

***Keywords** : IDS (Intrusion Detection System), LAN, Server, Snort, Telegram BOT*

1. PENDAHULUAN

Local Area Network (LAN), merupakan jaringan yang bersifat internal dan biasanya milik pribadi di dalam sebuah perusahaan kecil atau menengah dan biasanya berukuran sampai beberapa kilometer. LAN seringkali digunakan untuk menghubungkan komputer-komputer pribadi dan *workstation* dalam kantor suatu perusahaan atau pabrik-pabrik untuk pemakaian sumber daya bersama (*resource*, baik *hardware* maupun *software*) serta sarana untuk saling bertukar informasi[1].

Salah satunya yang sudah menggunakan LAN yaitu Kantor Imigrasi Tasikmalaya sebagai unit pelaksana teknis yang menjalankan fungsi Direktorat Jenderal Imigrasi di wilayah Priangan Timur provinsi Jawa Barat. Secara administratif wilayah tersebut terdiri dari Kota Tasikmalaya, Kota Banjar, Kabupaten Tasikmalaya, Kabupaten Garut, Kabupaten Ciamis dan Kabupaten Pangandaran. Pelayanan keimigrasian yang dilaksanakan di Kantor Imigrasi Tasikmalaya meliputi pelayanan terhadap warga negara Indonesia dan pelayanan terhadap warga negara asing. Untuk mendukung pelayanan tersebut, saat ini Kantor Imigrasi Tasikmalaya sudah menggunakan sistem berbasis teknologi informasi, diantaranya adalah Sistem Informasi Manajemen Arsip Kantor Imigrasi Tasikmalaya, Sistem Informasi Pengarsipan Paspor, Sistem Informasi Manajemen Perpustakaan, dan Layanan SMS *Gateway Customer Service*. Sistem tersebut terpusat di PC Server pada *jaringan local area network* Kantor Imigrasi Tasikmalaya. Sistem-sistem tersebut bisa digunakan apabila infrastruktur fisik dan *logica*l jaringan *local area network* di Kantor Imigrasi Tasikmalaya sudah terintegrasi dengan benar. PC Server beserta perangkat lunak yang telah terpasang didalamnya adalah infrastruktur fisik dan *logical* yang vital apabila sistem tersebut sedang digunakan.

Penggunaan *local area network* (LAN) sering mendapat keluhan, salah satunya sering terjadi gangguan pada Server untuk *Customer Service*. Secara umum server adalah sebuah komputer yang berisi program baik sistem operasi maupun program aplikasi yang menyediakan pelayanan kepada komputer atau program lain yang sama ataupun berbeda. Komputer server adalah komputer yang biasanya dikhususkan untuk penyimpanan data yang akan digunakan bersama, atau sebagai basis data[2]. Pengontrolan PC Server yang dilakukan oleh administrator jaringan hanya sebatas pengontrolan pada infrastruktur fisik saja, sementara fasilitas pengontrolan pada infrastruktur *logical* belum ada. Sehingga administrator kesulitan dalam mendiagnosa masalah yang terjadi pada infrastruktur *logical* jaringan jika suatu insiden yang mengancam keamanan jaringan terjadi, apalagi insiden tersebut mengakibatkan adanya gangguan terhadap jaringan, terutama gangguan tersebut mengarah ke PC Server sebagai pusat data dan sistem informasi.

Gangguan tersebut bisa berasal dari pihak-pihak yang tidak bertanggungjawab/ penyusup (*intruder*) dengan memanfaatkan kelemahan sistem keamanan jaringan *local area network* yang terhubung dengan PC Server baik itu melalui media kabel maupun nirkabel. Jika penyusupan terjadi, dikhawatirkan hal-hal yang tidak diinginkan oleh administrator menjadi kenyataan, seperti pencurian data / informasi keimigrasian yang bersifat rahasia, penyalahgunaan hak akses, dan jaringan mengalami malfungsi sehingga pelayanan di kantor menjadi terganggu.

Oleh karena itu dibutuhkan suatu cara mendeteksi adanya aktivitas jaringan yang mencurigakan dan mengirimkan notifikasi kepada administrator secara cepat dan efektif. *Intrusion Detection System* (IDS) adalah sistem pendeteksi gangguan yaitu sebuah aplikasi perangkat lunak atau perangkat keras yang bekerja secara otomatis untuk memonitor kejadian pada jaringan komputer dan menganalisis masalah keamanan jaringan[3]. Jika ditemukan kegiatan-kegiatan yang mencurigakan berhubungan dengan *traffic* jaringan maka IDS akan memberikan peringatan kepada sistem atau administrator jaringan[4].

Suatu *Intrusion Detection System* (IDS) melakukan identifikasi, memberikan laporan terhadap aktivitas jaringan komputer menggunakan Snort[5]. Snort merupakan suatu perangkat lunak untuk mendeteksi penyusup dan mampu menganalisis paket yang melintasi jaringan secara *real time traffic* dan *logging* ke dalam database serta mampu mendeteksi berbagai serangan yang berasal dari luar jaringan. Snort tersedia di internet, www.snort.org. Snort bisa digunakan pada *platform* sistem operasi Linux, BSD, solaris, dan Windows. Snort bisa dioperasikan dengan tiga mode[6]:

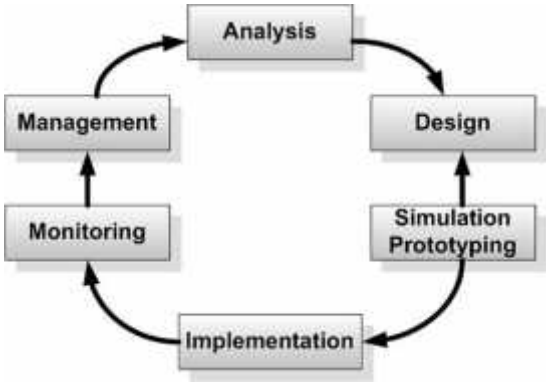
1) *Packet sniffer* : untuk melihat paket yang dilewati di jaringan;
2) *Packet logger* : untuk mencatat semua paket yang lewat di jaringan untuk dianalisis di kemudian hari;
3) NIDS, deteksi penyusup pada *network*: pada metode ini snort akan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer. Untuk menggunakan mode IDS ini diperlukan setiap dari berbagai *rules* atau aturan yang akan membedakan sebuah paket normal dengan paket yang membawa serangan;

Jika IDS hanya dapat memantau dan mendeteksi aktifitas jaringan, maka untuk kebutuhan informasi administrator dibutuhkan adanya pemberitahuan atau notifikasi. Media notifikasi ini dapat berupa pengiriman pesan teks yang dikirim ke alat yang dimiliki admin. Melihat maraknya penggunaan smartphone dan banyakya aplikasi messenger pada smartphone, maka hal ini dapat dijadikan kesempatan untuk memanfaatkannya pada sistem IDS. Sehingga pada penelitian ini, digunakan aplikasi telegram messenger sebagai media penyampaian notifikasi jika IDS mendeteksi adanya aktifitas tidak wajar di jaringan.

Telegram *Messenger* adalah aplikasi perpesanan gratis yang fokus pada kecepatan dan keamanan pengiriman maupun penerimaan pesan melalui jaringan internet. Telegram dapat digunakan hampir disemua *platform*, yaitu iOS, Android, Windows *Phone*, Telegram's Web *Version* ataupun dapat dipasang sebagai aplikasi *desktop* pada sistem operasi Windows, OS X dan Linux. Selain itu API (*Application Programming Interface*) telegram bersifat *free* dan *open*. Telegram memiliki API Bot, yaitu *platform* untuk para *developer* program aplikasi untuk membuat aplikasi yang berkaitan dengan telegram.

2. METODE PENELITIAN

Dalam pengembangan sistem ini digunakan metode *Network Development Life Cycle*. Penelitian dilakukan dengan tahapan-tahapan seperti pada gambar 1. Tahap awal yang dilakukan yaitu analisa kebutuhan, analisa permasalahan yang muncul, analisa keinginan *user*, dan analisa topologi / jaringan yang sudah ada saat ini.

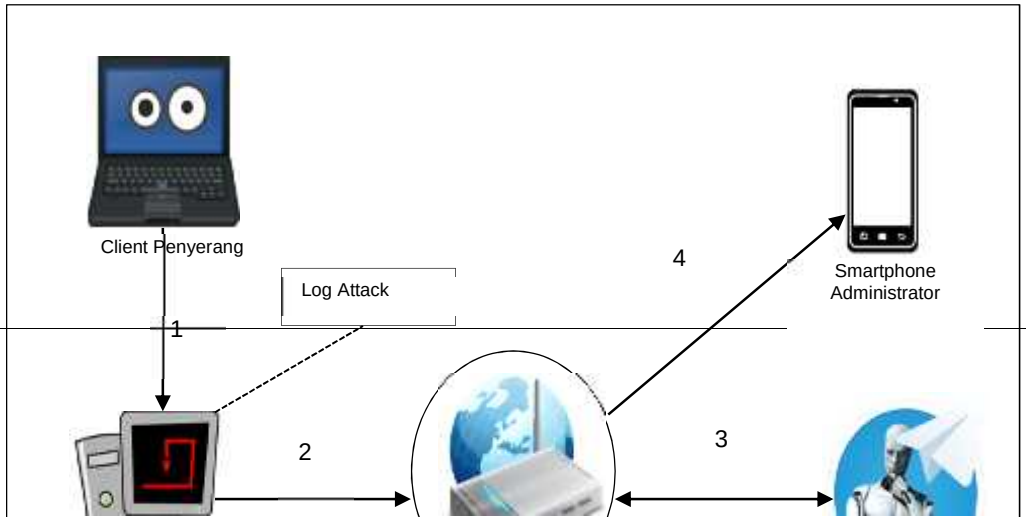


Gambar 1. NDLC (Network Development Life Cycle)[7]

Dari data-data yang didapatkan sebelumnya, tahap *Design* ini akan membuat gambar *design topology* jaringan interkoneksi yang akan dibangun, diharapkan dengan gambar ini akan memberikan gambaran seutuhnya dari kebutuhan yang ada. *Design* bisa berupa *design* struktur *topology*, *design* akses data, *design* tata *layout* perkabelan, dan sebagainya yang akan memberikan gambaran jelas tentang *project* yang akan dibangun.

Secara sederhana sistem yang akan dibangun dapat diilustrasikan seperti pada Gambar 2, cara kerja sistem dapat dirincikan sebagai berikut :

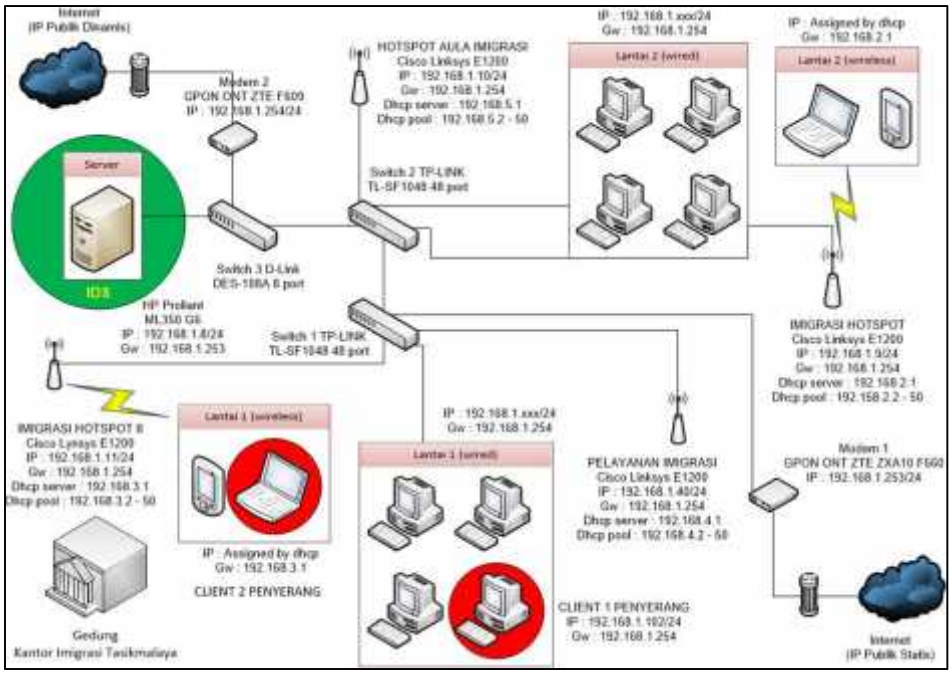
1) *Client* penyerang melakukan usaha-usaha penyusupan dengan cara mencari kelemahan sistem keamanan jaringan pada PC Server. IDS yang terpasang pada PC Server melakukan pengawasan terhadap kegiatan-kegiatan yang mencurigakan yang terjadi pada PC Server. Ketika *client* penyerang melakukan usaha-usaha penyusupan ke PC Server maka akan menembus terlebih dahulu aplikasi IDS yang terpasang yaitu Snort. Setelah Snort mendeteksi usaha-usaha penyusupan, Snort akan membuat suatu *log file* hasil *capture* paket penyusupan tersebut.
2) *Log* tersebut dikirim kepada akun Telegram BOT melalui https *request* lewat koneksi internet.
3) Telegram BOT menerima *log*, kemudian *log* dikirim kepada akun Telegram Messenger administrator melalui koneksi internet
4) Akun Telegram Messenger administrator menerima *log attack* yang terdeteksi oleh IDS melalui ponsel pintar miliknya yang terkoneksi dengan internet



Gambar 2. Desain Sistem

Rancangan Arsitektur Jaringan Komputer dari Sistem yang diajukan adalah seperti pada gambar 3. Dari arsitektur gambar 3, IDS yang dikembangkan berjenis HIDS (*Host Intrusion Detection System*), karena bekerja pada *host* yang akan dilindungi untuk melakukan pengawasan terhadap *traffic* yang menuju dan berasal dari PC Server. Rincian keterangan dari gambar topologi jaringan komputer di atas adalah sebagai berikut :

1) Jenis topologi jaringan yang diterapkan adalah *Star*
2) Seluruh IP *Address* yang digunakan adalah kelas C



Gambar 3. Arsitektur jaringan komputer yang diajukan

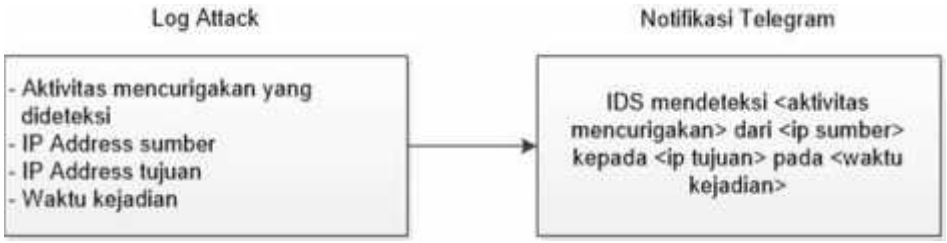
Pada topologi tersebut jenis kabel yang digunakan untuk menghubungkan perangkat-perangkat jaringan adalah *straight* dan *cross.*

Tabel 1. Rincian Koneksi

No	Sumber	Tujuan	Koneksi
1	*Client* 2 Penyerang (*wireless connection*)	*Access Point*	*Wireless*
2	Access Point	*Switch* 1	Kabel *Straight*
3	*Client* 1 Penyerang (*wired connection)*	*Switch* 1	Kabel *Straight*

4	*Switch* 1	*Switch* 2	Kabel *Cross*
5	*Switch* 2	*Switch* 3	Kabel *Cross*
6	*Switch* 3	PC Server	Kabel *Straight*
7	*Switch* 3	Modem	Kabel *Straight*

Untuk memenuhi kebutuhan administrator mengenai pengadaan media notifikasi yang cepat dan efektif, maka penulis perlu merancang pesan notifikasi yang akan diterima oleh administrator melalui Telegram Messenger. Perancangan pesan notikasi menggambarkan informasi yang akan diterima administrator dari sistem. Input dari sistem berupa *log attack* dan output yang diterima oleh administrator berupa informasi yang terdeteksi oleh sistem seperti pada gambar 4.



Gambar 4. Perancangan Pesan Notifikasi

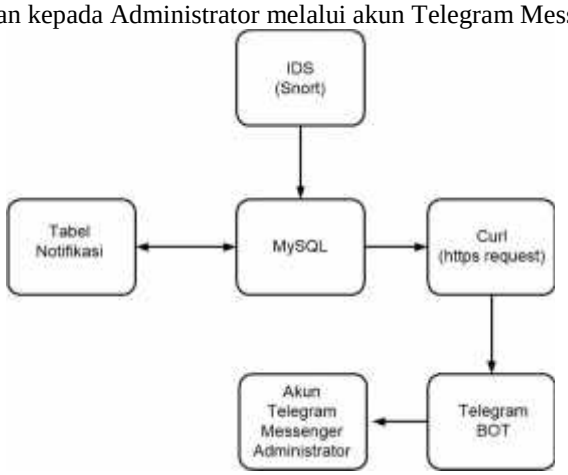
Pada tahap ini dispesifikasikan seluruh komponen software IDS yang dibutuhkan yang dapat dilihat pada Tabel 2.

Tabel 2. Spesifikasi Software

Device	Komponen	Keterangan
PC Server	1. Snort 2. Mysql 3. Acidbase 4. Curl	PC Server akan mendeteksi aktivitas gangguan dengan membuat *log alert* (Snort), *log alert* kemudian disimpan pada tabel database dan dilakukan penyaringan untuk dikirim pada tabel notifikasi (Mysql), *alert* dapat ditampilkan oleh acidbase sebagai *web front-end*, *alert* secara otomatis dikirim kepada akun telegram messenger administrator melalui akun telegram bot yang diberi perintah https *request* (curl)
Client Penyerang	1. Putty (ssh *access*) 2. Nmap 3. Nessus	Client penyerang untuk menguji fungsionalitas sistem IDS dan media notifikasi Telegram.
Smartphone Administrator	1. Telegram Messenger	Menerima notifikasi *alert* yang dikirim oleh akun Telegram BOT

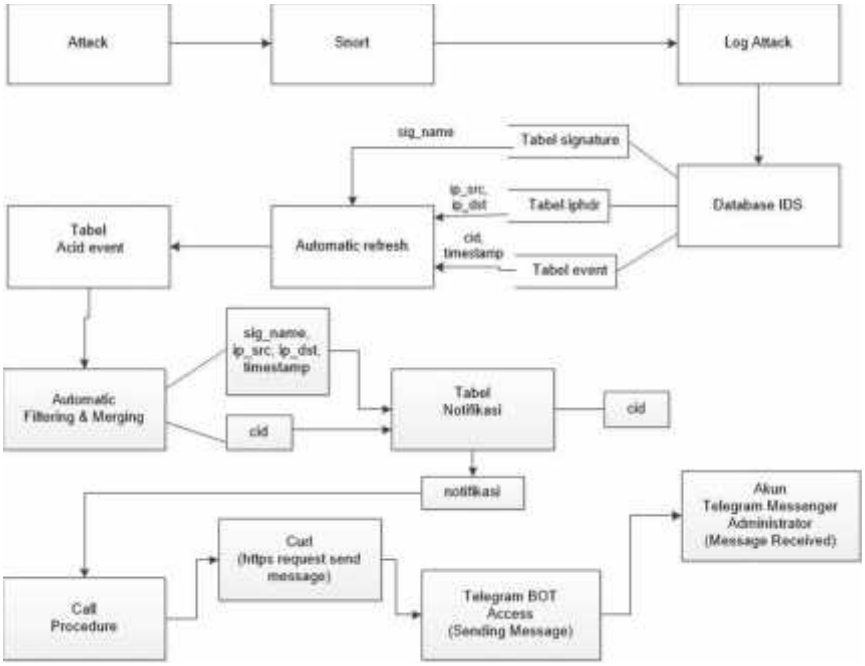
Garis besar kerja IDS pada penelitian ini terlihat pada gambar 5, bahwa data yang ditangkap akan diolah oleh Snort sesuai dengan *rule* yang telah ditentukan. *Alert* yang dibangkitkan oleh snort akan disimpan di dalam database dengan menggunakan server database MySQL. Kemudian data *alert* tersebut akan difilter, hasil *filter* ini ditampung ke dalam tabel notifikasi, sehingga diharapkan tidak terjadi

pengulangan pengiriman *alert* yang sama dalam waktu yang sama pula. Setelah terjadi pembaharuan pada tabel notifikasi maka curl akan mengeksekusi https *request* untuk mengirimkan perintah send message yang berisi data dari tabel notifikasi pada akun Telegram BOT yang sudah ditentukan. Sehingga pesan *alert* dari IDS dapat disampaikan kepada Administrator melalui akun Telegram Messenger miliknya.



Gambar 5. Alur Konfigurasi Software

Pada tahap ini dibuat pemodelan bagaimana data *log attack* bisa diterima oleh akun Telegram Messenger administrator dengan mendefinisikan elemen *database* yang menjadi sumber data media notifikasi.



Gambar 6. Alur Data Notifikasi

Networker's pada penelitian ini membuat dalam bentuk simulasi dengan bantuan *Tools* khusus di bidang network seperti PACKET TRACERT, hal ini dimaksudkan untuk melihat kinerja awal dari *network* yang akan dibangun dan sebagai presentasi dan *sharing* dengan *team work* lainnya.

Tahapan selanjutnya yaitu implementasi yang akan memakan waktu lebih lama dari tahapan sebelumnya. Dalam implementasi *networker's* menerapkan semua yang telah direncanakan dan didesign sebelumnya. Implementasi merupakan tahapan yang sangat menentukan dari berhasil / gagalnya *project* yang dibangun dan ditahap inilah *Team Work* diuji dilapangan untuk menyelesaikan masalah teknis dan non teknis.

Setelah implementasi tahapan *monitoring* merupakan tahapan yang penting, agar jaringan komputer dan komunikasi dapat berjalan sesuai dengan keinginan dan tujuan awal dari user pada tahap awal analisis, maka perlu dilakukan kegiatan *monitoring*. Manajemen atau pengaturan, salah satu yang menjadi perhatian khusus adalah masalah *Policy*, kebijakan perlu dibuat untuk membuat / mengatur agar sistem yang telah dibangun dan berjalan dengan baik dapat berlangsung lama dan unsur *Reliability* terjaga. *Policy* akan sangat tergantung dengan kebijakan level *management* dan strategi bisnis perusahaan tersebut. IT sebisa mungkin harus dapat mendukung atau *alignment* dengan strategi bisnis perusahaan.

3. HASIL DAN PEMBAHASAN

3.1. Implementasi

Tahap implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta perangkat lunak pada keadaan sesungguhnya.

3.1.1. Implementasi *Intrusion Detection System*

IDS atau sistem pendeteksi intrusi penulis bangun dengan menggunakan beberapa komponen utama, yaitu : Snort dan Acidbase. IDS dibangun pada PC Server dengan menggunakan sistem operasi linux Ubuntu 12.04. Berikut langkah-langkah yang dilakukan :

1) Membuat user dan *database* untuk snort;
2) Instalasi snort;
3) Konfigurasi *rules* snort;
4) Konfigurasi *file* snort.conf;
5) Instalasi Acidbase;
6) Konfigurasi Acidbase.

3.1.2. Implementasi Telegram Messenger sebagai Media Notifikasi IDS

Untuk mengimplementasikan Telegram Messenger sebagai media notifikasi IDS secara *real time*, penulis menggunakan beberapa komponen *software* yaitu phpmyadmin, *script* shell, php cli, *plugin* libmysqludf_sys, curl, fitur trigger dan *stored procedure* pada mysql.

1) Membuat tabel notifikasi

Gambar 6. Tabel Notifikasi

2) Membuat *Script* otomatis.sh

```
#!/bin/bash
while  (sleep 5 && php /usr/share/acidbase/base_main.php)&
do
  wait $!
Done
```

Fungsi dari *script* otomatis.sh adalah untuk menjalankan *file* base_main.php setiap 5 detik sekali. *File* base_main.php adalah *file* untuk memperbarui tabel acid_event yang menampung data *log attack* yang terdeteksi oleh snort

3) Instalasi plugin lib_mysqludf_sys

Plugin lib_mysqludf_sys digunakan agar mysql dapat melakukan perintah eksternal, dalam hal ini untuk mengeksekusi aplikasi curl. Plugin dapat diunduh secara gratis pada laman berikut https://github.com/mysqludf/lib_mysqludf_sys/archive/master.zip

4) Instalasi Curl

Curl merupakan sebuah fungsi yang berguna untuk meng-grab / mengambil konten dari sebuah halaman website. API Telegram BOT berbasis HTTP API. Penulis menginstal curl karena dibutuhkan untuk memberi perintah kepada akun Telegram BOT melalui https *request*

5) Membuat Trigger update_notifikasi

Trigger update_notifikasi berfungsi untuk menghindari duplikasi data *log attack* pada tabel notifikasi, menggabungkan field sig_name, ip_src, ip_dst dan timestamp menjadi satu field kemudian memasukkan data tersebut ke dalam tabel notifikasi.

```
DELIMITER $$
CREATE TRIGGER `update_notifikasi` AFTER INSERT ON `acid_event`
 FOR EACH ROW begin
if (select count(*) from acid_event where timestamp=New.timestamp) = 1 then
insert into notifikasi (cid,notifikasi) values (New.cid,
(select concat('IDS mendeteksi ',sig_name,' dari ',
cast(inet_ntoa(ip_src) as char),' kepada ',cast(inet_ntoa(ip_dst) as char),
' pada ',timestamp) from acid_event where cid=New.cid));
end if;
END$$
DELIMITER;
```

6) Membuat *Stored Procedure* notifikasitelegram

Maksud membuat *stored procedure* notifikasi telegram adalah untuk memberi perintah kepada akun Telegram BOT melalui plugin libmysqludf_sys guna mengeksekusi https_request pada curl.

```
DELIMITER $$
CREATE DEFINER=`ids`@`localhost` PROCEDURE `notifikasitelegram`(IN `pesan`
    VARCHAR(255))
BEGIN
 DECLARE cmd char(255);
 DECLARE result varchar(255);
set cmd =concat('ALERT !! : ',pesan);
SET result = sys_exec(concat('/usr/bin/curl -s -X POST
    "https://api.telegram.org/token_telegram_bot/sendMessage?chat_id=id_telegram_pe
    nerima&&text=',cmd,""));
END$$
```

7) Membuat Trigger Telegram

Maksud membuat trigger dengan nama **triggertelegram** adalah untuk memanggil *stored procedure* notifikasi telegram supaya akun Telegram BOT segera mengirim *data log attack* yang baru masuk ke dalam tabel notifikasi

```
DELIMITER $$
CREATE TRIGGER `triggertelegram` AFTER INSERT ON `notifikasi`
 FOR EACH ROW begin
CALL notifikasitelegram(New.notifikasi);
END$$
DELIMITER;
```

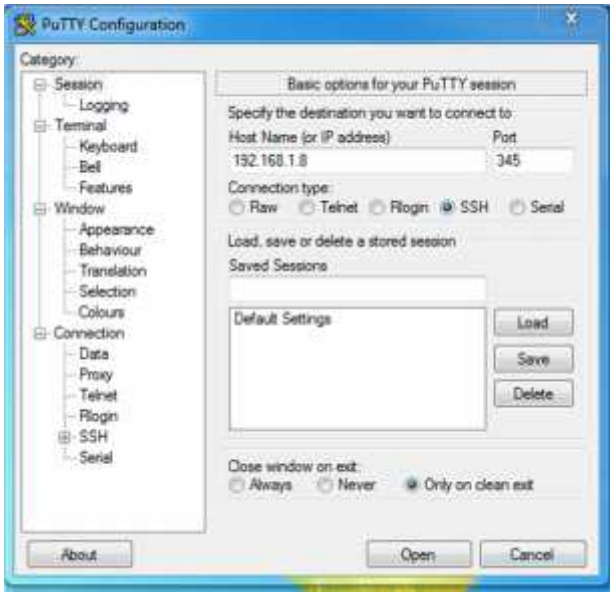
3.2. Pengujian Sistem

Pengujian Sistem adalah untuk mengetahui apakah Sistem yang telah dibuat sesuai dengan perancangannya atau tidak. Selain itu juga untuk mengetahui detail jalannya sistem serta kesalahan yang ada untuk pengembangan dan perbaikan lebih lanjut.

3.2.1. Pengujian SSH Remote Access

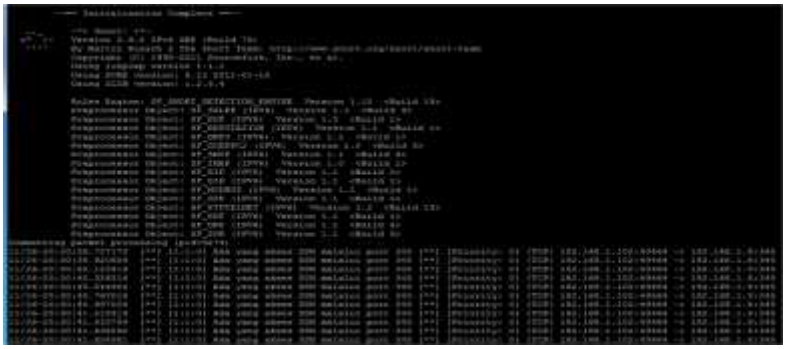
Penulis melakukan pengujian apakah sistem sudah mampu mendeteksi bahwa ada yang meremote server melalui SSH (*Secure Shell*).

1) Client Mengakses SSH Melalui Putty



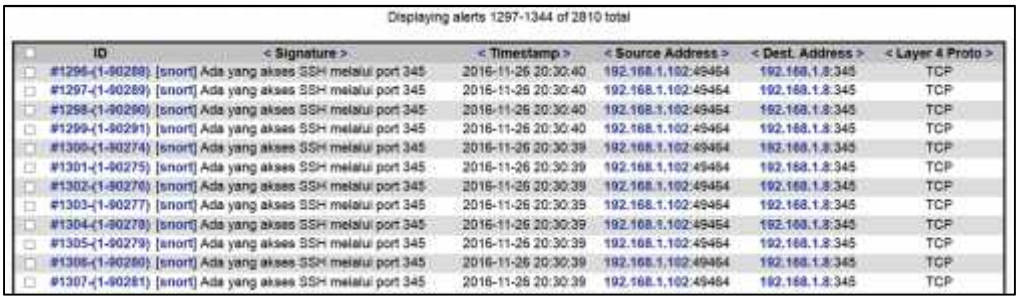
Gambar 6. Client mencoba akses ssh

2) Mengecek fungsionalitas Snort menggunakan perintah *snort –c /etc/snort/snort.conf –A console –K ascii*



Gambar 7. Snort Mendeteksi SSH Access

3) Mengecek fungsionalitas Acidbase menggunakan browser dengan mengakses http://192.168.1.8/acidbase

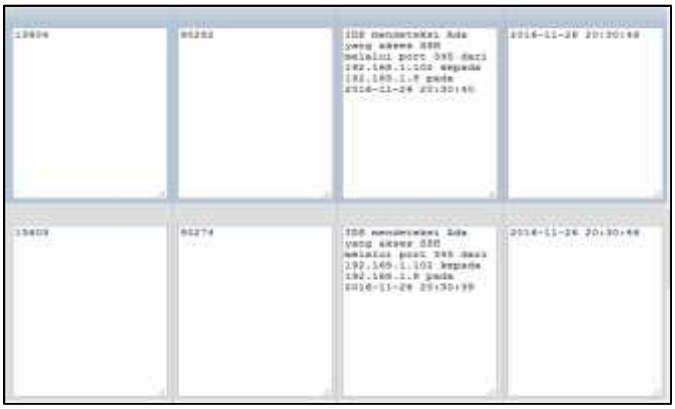


Displaying alerts 1297-1344 of 2810 total

ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#1296-(1-90288)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:40	192.168.1.102:49464	192.168.1.8:345	TCP
#1297-(1-90289)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:40	192.168.1.102:49464	192.168.1.8:345	TCP
#1298-(1-90290)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:40	192.168.1.102:49464	192.168.1.8:345	TCP
#1299-(1-90291)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:40	192.168.1.102:49464	192.168.1.8:345	TCP
#1300-(1-90274)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1301-(1-90275)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1302-(1-90276)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1303-(1-90277)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1304-(1-90278)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1305-(1-90279)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1306-(1-90280)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP
#1307-(1-90281)	[snort] Ada yang akses SSH melalui port 345	2016-11-26 20:30:39	192.168.1.102:49464	192.168.1.8:345	TCP

Gambar 8. Acidbase Menampilkan alert SSH Access

4) Mengecek fungsionalitas tabel notifikasi melalui phpmyadmin



Gambar 9. Tabel Notifikasi menerima alert SSH Access

5) Mengecek fungsionalitas Notifikasi Telegram

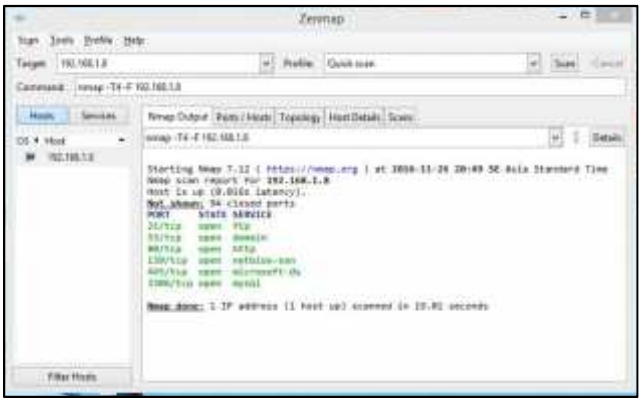
Gambar 10. Notifikasi ssh access telegram messenger

Gambar di atas menunjukan bahwa data dari tabel notifikasi sudah diterima dari akun Telegram BOT IDSkanimtasik kepada akun Telegram Messenger Administraotr. Terlihat waktu penerimaan adalah pada pukul 20:30. Artinya notifikasi dapat diterima dalam waktu satuan detik.

3.2.2. Pengujian Nmap *Port Scanning*

Pada kasus ini penulis akan menguji apakah sistem sudah dapat mendeteksi aktivitas *port scanning* yang mengarah kepada PC Server atau belum.

1) *Client* melakukan *port scanning* menggunakan Nmap



Gambar 11. Client melakukan port scanning

2) Mengecek fungsionalitas Notifikasi Telegram

Gambar 12. Notifikasi port scanning telegram messenger

Gambar di atas menunjukan bahwa pesan notifikasi *port scanning* sudah diterima dari akun Telegram BOT IDSkanimtasik kepada akun Telegram Messenger Administrator.

4. KESIMPULAN

Dari hasil penelitian yang dilakukan dapat diambil kesimpulan sebagai berikut:

a. Sistem pendeteksi aktivitas gangguan jaringan yang dipasang pada PC Server dapat dilengkapi fasilitas notifikasi yang dikirim langsung ke smartphone admin jaringan.
b. Diterapkannya Snort sebagai IDS dan Acidbase sebagai web *front-end* nya, administrator dapat dengan mudah mendiagnosa masalah yang terjadi pada infrastruktur *logical* jaringan PC Server
c. Diimplementasikannya Telegram Messenger sebagai media notifikasi pesan gangguan yang terdeteksi oleh snort, administrator dapat mengetahui gangguan yang terjadi pada PC Server dengan cepat dan efektif
d. Administrator dapat menerima notifikasi adanya gangguan yang terdeteksi dalam kisaran waktu 3 - 7 detik. Sehingga dapat langsung melakukan hal preventif yang diperlukan untuk mengatasi gangguan yang terjadi

DAFTAR PUSTAKA

[1] D. Sopandi, *Instalasi dan Konfigurasi Jaringan Komputer. 2nd*. Bandung: Infomatika, 2010.
[2] D. Ariyus, *Intrusion Detection System*, Sigit Suya. Yogyakarta: ANDI, 2007.
[3] T. Ariyadi, Y. N. Kunang, R. Santi, J. Jenderal, A. Yani, and N. Palembang, “Implementasi Intrusion Prevention System (IPS) Pada Jaringan Komputer Kampus B UNIVERSITAS BINA

DARMA," *J. Ilm. Tek. Inform. Ilmu Komput.*, vol. 14, no. 2, pp. 1–14, 2012.
[4] R. Arief, "Penggunaan Sistem IDS (Intrution Detection System) untuk Pengamanan Jaringan dan Komputer," *STMIK AMIKOM Yogyakarta*, 2013.
[5] L. Putri, "Implementasi Intrusion Detection System (Ids) Menggunakan Snort Pada Jaringan Wireless (Studi Kasus : Smk Triguna Ciputat)," UIN SyarifHidayatullah, 2011.
[6] E. J. C. Suhari, "APLIKASI HIERARCHICAL CLUSTERING PADA INTRUSION DETECTION SYSTEM BERBASIS SNORT," *EEPIS Final Proj.*, 2011.
[7] D. Setiawan, *network development cycles*. Palembang: Universitas Sriwijaya. [Online], 2009.