ANALISIS SISTEM MONITORING KEAMANAN *SERVER* DENGAN SMS *ALERT* BERBASIS SNORT

I Komang Krisna Ade Marta[1], I Nyoman Buda Hartawan[2], I Kadek Susila Satwika[3]

[1] Prodi Sistem Komputer STIMIK STIKOM INDONESIA, Jl. Tukad Pakerisan No.97, Panjer, Kec. Denpasar Sel., Kota Denpasar, Bali 80225 INDONESIA
[2] Prodi Sistem Komputer STIMIK STIKOM INDONESIA, Jl. Tukad Pakerisan No.97, Panjer, Kec. Denpasar Sel., Kota Denpasar, Bali 80225 INDONESIA
[3] Prodi Sistem Komputer STIMIK STIKOM INDONESIA, Jl. Tukad Pakerisan No.97, Panjer, Kec. Denpasar Sel., Kota Denpasar, Bali 80225 INDONESIA

Abstrak

Keamanan *server* merupakan hal penting yang perlu diberikan perhatian lebih saat melakukan konfigurasi *server*. Pada umumnya serangan yang terjadi pada *server* diketahui setelah terjadinya kegagalan pada *server* dalam memberikan layanan. Pada penelitian ini, dibangun sebuah sistem keamanan *server* yang dapat melakukan monitoring pada sebuah *server* ketika terdeteksi adanya aktivitas yang tidak wajar. Pemberitahuan akan dikirimkan melalui SMS (*Short Message Service*) ke *handphone Administrator* jaringan. Sistem yang dibangun melakukan pendeteksian intrusi pada *server* secara *realtime* menggunakan SNORT. Ketika terjadi akses yang tidak wajar pada *server*, maka SNORT akan mendeteksi dan mengirimkan informasi terjadinya aktivitas yang tidak wajar ke *Administrator* jaringan. Sistem ini diujikan dengan lima jenis serangan yakni *PING Attack*, *DoS/DDoS Attack*, *Port Scanning*, *Telnet Access* dan *FTP Access*. Parameter yang diamati pada penelitian ini adalah beban aktivitas yang terjadi pada sumber daya *server* meliputi CPU, *Memory* (RAM) dan beban jaringan. Hasil penelitian menunjukkan bahwa saat terjadi upaya serangan terhadap *server*, SNORT dapat menghasilkan *alert* yang akan disimpan pada *log* sekaligus dikirimkan ke *handphone Administrator* melalui SMS.

Kata Kunci:
Keamanan *Server*, Monitoring *Server*, IDS, SNORT, SMS *Gateway*.

Abstract

Server security is an important thing that needs to be given more attention when configuring a server. In general, attacks that occur on the server are known after a failure on the server in providing services. In this study, a server security system was built that could monitor a server when an unusual activity was detected. Notifications will be sent via SMS (Short Message Service) to the network Administrator's smartphone. The system is built to detect intrusions on the server in real time using SNORT. When improper access occurs on the server, SNORT will detect and send information about the occurrence of unusual activity to the network Administrator. This system is tested with five types of attacks namely PING Attack, DoS / DDoS Attack, Port Scanning, Telnet Access and FTP Access. The parameters observed in this study are the activity load that occurs on server resources including CPU, Memory (RAM) and network load. The results showed that when an attempt was made to attack the server, SNORT could produce alerts that would be stored in a log as well as sent to the Administrator's smartphone via SMS.

Keywords:
Server Security, Server Monitoring, IDS, SNORT, SMS Gateway.

*Korespondensi
E-mail: krisnaademarta@yahoo.co.id, buda.hartawan@gmail.com, susila.satwika@gmail.com

1. PENDAHULUAN

Seiring dengan berkembangnya teknologi informasi, jaringan komputer dan *server* sudah umum digunakan dalam sebuah institusi baik itu di sebuah perusahaan, pemerintahan, organisasi dan lain sebagainya. Pemanfaatan teknologi saat ini khususnya dibidang pendidikan memungkinkan untuk melaksanakan kegiatan pembelajaran melalui jarak jauh. Praktikuam adalah salah satu kegiatan pembelajaran yang umumnya dilakukan secara tatap muka di ruang laboratorium. Namun, pemanfaatan *server* memungkinkan kegiatan laboratorium khusunya tentang bidang teknologi informasi melalui laboratorium virtual(Hartawan & Satwika, 2016). *Server* adalah komputer yang memberikan layanan kepada klien sehingga perlu dilengkapi dengan spesifikasi yang tinggi dan media penyimpanan yang besar (Hartawan & Iswara, 2015). Namun selain itu semakin berkembangnya penggunaan *server* yang ada saat ini, faktor keamanan/*security* merupakan faktor vital dan perlu diperhitungkan secara serius. Keamanan *server* yang dimaksud dapat berupa upaya memonitoring dan mencegah penggunaan *server* yang tidak sah ataupun serangan dari pihak eksternal maupun internal. Tujuannya yaitu untuk mencegah terjadinya kerusakan yang fatal pada layanan *server* apabila terjadi serangan. Saat ini perkembangan teknologi memungkinkan perangkat elektronik dikontrol melalui jaringan internet (Hartawan & Sudiarsa, 2019). Perangkat-perangkat elektronik saat ini sudah mulai diintegrasikan dengan aplikasi teknologi komputer (Ekayana, Hartawan, Desnanjaya, & Joni, 2020)(Desnanjaya, Iswara, Ekayana, Santika, & Hartawan, 2020). Hal ini juga dapat menimbulkan potensi terjadinya serangan, yang menyebabkan alih kendali terhadap perangkat elektronik yang digunakan.

Metode yang paling umum digunakan untuk mengamankan sebuah *server* adalah dengan menggunakan *firewall*. Sistem keamanan *firewall* tidaklah cukup untuk meminimalkan dampak serangan yang terjadi pada *server*. Pada umumnya serangan yang terjadi pada *server* diketahui setelah terjadinya kegagalan *server* dalam memberikan layanan. *Administrator* membutuhkan waktu yang cukup lama untuk menganalisa kerusakan dan proses pemulihan terhadap *server* tersebut. Hal ini tentu sangat merugikan baik pihak *Administrator* maupun pengguna *server* itu sendiri.

Oleh karena itu, untuk mengatasi permasalahan diatas, perlu dibangun sebuah sistem keamanan *Server* yang dapat memonitoring aktivitas sebuah *Server* ketika terdeteksi aktivitas yang tidak wajar dengan mengirimkan notifikasi melalui SMS (*Short Message Service*) ke *handphone Administrator* jaringan secara *realtime*. Dengan kata lain *Administrator* dapat langsung menerima pemberitahuan bahwa sedang terjadi serangan/aktifitas yang tidak wajar pada *server* tersebut, sehingga dapat langsung mengambil tindakan secepatnya, seperti dengan mengendalikan *server* secara jarak jauh untuk mencegah penyerangan yang berakibat lebih buruk terhadap kondisi kestabilan dan keamanan *server*.

Pada penelitian ini dibuat rancang bangun sistem monitoring keamanan *server* dengan SMS *alert* berbasis SNORT. Sistem ini mampu mengirimkan pemberitahuan melalui *SMS Gateway*. Sistem ini diujikan dengan lima jenis serangan yakni *PING Attack*, *DoS/DDoS Attack*, *Port Scanning*, *Telnet Access* dan *FTP Access*. Parameter yang diamati pada penelitian ini adapa CPU, *Memory* (RAM) dan beban jaringan.

2. METODE

Snort merupakan bagian dari IDS dan merupakan sebuah perangkat lunak open soure. Snort mampu melakukan analisa *realtime traffic* dan *packet logger* pada jaringan IP dan dapat menganalisa *protocol* dan melakukan pendeteksian variasi penyerangan. Snort juga memiliki kemampuan *realtime alert*, dimana mekanisme pemaksukan *alert* dapat berupa *user* syslog, *file*, uni *socket* ataupun melalui *Server* (Rehman, 2003). Dalam mengoprasikan snort mempunyai tiga buah mode, yaitu:

A. Sniffer Mode

Sniffer Mode ini berfungsi untuk melihat paket yang lewat di jaringan, maka untuk menjalankan snort pada *sniffer mode* tidak terlalu susah.

B. *Packet logger* mode

Packet logger mode berfungsi untuk mencatat semua paket yang lewat dijaringan yang kemudian akan dianalisa. Bahkan dapat menyimpan paket dalam disk. Sehingga perlu diinisialisasikan terlebih dahulu *logging* direktorinya pada *file* configurasi snort.

C. Network Intrusion Detection System (NIDS) mode

Dengan menggunakan *network Intrusion Detection System* (NIDS) tidak diperlukan lagi untuk menyimpan seluruh paket yang datang pada sebuah jaringan. karena pada mode ini data yang disimpan

atau ditampilkan adalah paket – paket yang berbahaya dengan cara mengkonfigurasi snort.conf terlebih dahulu.

3. SMS *GATEWAY*

SMS *Gateway* adalah sebuah perangkat yang menawarkan layanan transit SMS, mentransfor-masikan pesan ke jaringan selular dari media lain, atau sebaliknya, sehingga memungkinkan pengiriman atau penerimaan pesan SMS dengan atau tanpa menggunakan ponsel (Tarigan, 2012).

Pada hakekatnya SMS *Gateway* sama dengan ketika kita menggunakan *handphone* seperti biasa. Hanya saja pada beberapa aplikasi sms *gateway* yang telah dikembangkan, ada perbedaan untuk *interface*-nya. Bila menggunakan *handphone* maka *interface*-nya berupa keypad dan layar *handphone* tersebut, sedangkan bila menggunakan sms *gateway* maka *interface*-nya keyboard komputer dan layar monitor. Selain *interface*, ada perbedaan pula pada cara kerjanya. Pada beberapa aplikasi ada fitur autoreply. Fitur ini memungkinkan sistem tersebut membalas otomatis sms dari user secara langsung.

Sebagaimana penjelasan diatas, SMS *Gateway* dapat terhubung ke media lain seperti perangkat SMSC dan *Server* milik *Content Provider* melalui link IP untuk memproses suatu layanan SMS.

4. BARNYARD DAN BASE (BASIC ANALYSIS AND SECURITY ENGINE)

Barnyard2 adalah *tool open source* sebagai penerjemah *alert unified* dan *log* dari Snort. Barnyard2 dapat meningkatkan efisiensi Snort dengan cara mengurangi beban pada sensor deteksi. Barnyard2 bekerja dengan membaca *Snort's unified logging output files* dan memasukannya kedalam *database* (Tarigan, 2012). Jika *database* tidak tersedia maka Barnyard2 akan memasukan semua data ketika *database* tersedia kembali sehingga tidak ada *alert* atau *log* yang hilang. Sedangkan, Base adalah sebuah *interface* web untuk melakukan analisis dari intrusi yang snort telah deteksi pada jaringan. BASE ditulis oleh kevin johnson adalah program analisis sitem jaringan berbasis PHP yang mencari dan memproses database dari *security* event yang dihasilkan oleh berbagai program monitoring jaringan, firewall, atau sensor IDS.

5. LOIC (LOW ORBIT ION CANNON)

LOIC Adalah *Low Orbit Ion Cannon* atau bisa disebut LOIC berfungsi untuk melumpuhkan *server* sebuah situs website. ini adalah *software* DDOS yang paling ampuh, terbukti komunitas hacker sekelas '*Anonymous*' menggunakan tool loic ini untuk melancarkan aksinya. *Software* Loic ini juga pernah melumpuhkan *server* facebook yang memiliki 60 *server* yang tersebar luas di seluruh dunia walau hanya beberapa menit (Tarigan, 2012).

6. METODE

A. Teknik Pengumpulan Data

Dalam penelitian ini, penulis menggunakan teknik pengumpulan data sekunder. Data Sekunder adalah studi yang mempelajari, meneliti, dan mencari berbagai sumber dari buku-buku, jurnal ilmiah, situs-situs resmi di *internet* serta bacaan-bacaan yang berkaitan dengan penelitian ini. Penulis mempelajari hal-hal yang berkaitan dengan jaringan komputer, *Server*, keamanan jaringan dan *Server*, *firewall*, IDS, Snort, SMS *Gateway* dan lain sebagainya.

B. Analisis Kebutuhan Data

Adapun analisis kebutuhan sistem fungsional dari sistem monitoring kemanaan *Server* dengan pemberitahuan melalui SMS berbasis Snort yaitu:

1. Sistem dapat bekerja sesuai dengan *rule* keamanan yang ditentukan.
2. Sistem dapat menyimpan data serangan kedalam File *Alert Database*
3. Sistem dapat mengirim pemberitahuan serangan melalui SMS ke *Handphone Administrator*.

Adapun analisis kebutuhan sistem non-fungsional dari sistem monitoring kemanaan *Server* dengan pemberitahuan melalui SMS ini terdiri dari Kebutuhan *Hardware* dan Kebutuhan *Software*.

a. Kebutuhan Hardware

Berikut perangkat keras (*Hardware*) yang penulis gunakan dalam membangun sistem monitoring kemanaan *Server* dengan pemberitahuan melalui SMS berbasis SNORT:

Tabel 1. Daftar Kebutuhan *Hardware*

No	Nama	Jumlah
1	*Server*	1
2	*PC Attacker*	1
3	Kabel LAN	1
4	*Switch*	1
5	*Modem*	1
6	*Handphone*	1

b. Kebutuhan Software

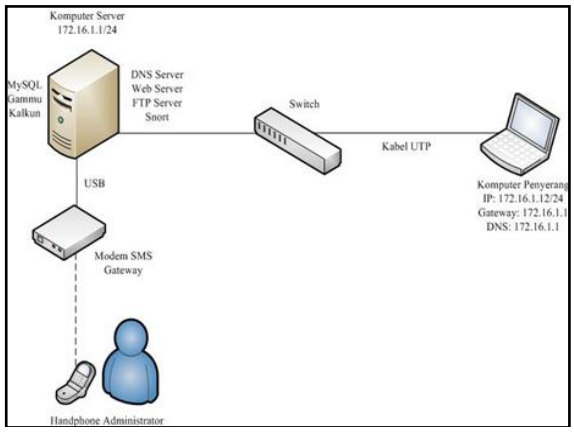
Berikut perangkat keras (*Hardware*) yang penulis gunakan dalam membangun sistem monitoring kemanaan *Server* dengan pemberitahuan melalui SMS berbasis Snort:

Tabel 2. Daftar Kebutuhan *Software*

No	Nama
1	*Sistem Operasi Linux Ubuntu 14.04 LTS*
2	*Command Line*
3	SNORT 2.9
4	*Banyard2 dan BASE*
5	*LAMP*
6	*GAMMU*
7	*Sistem Operasi Windows 10 Professional-64 Bit*
8	*LOIC*
9	*Zmap-Zenmap*

C. Topologi Jaringan

Setelah memahami *flowchart* (alur kerja) dari sistem monitoring kemanaan *Server* dengan pemberitahuan melalui SMS berbasis Snort, selanjutnya akan dijelaskan tentang Topologi Jaringan dari sistem ini. Pada Topologi Jaringan, Komputer *Server* dan Komputer Penyerang terhubung dengan media Kabel UTP dengan switch sebagai konsentratornya. Di Komputer *Server* terpasang Modem SMS *Gateway* dengan media USB. Modem SMS *Gateway* ini akan mengirimkan SMS *Alert* ke *Handphone Administrator*. Topologi jaringan ini meggunakan skema IP kelas C dengan network 172.16.1.0/24.



Gambar 1. Topologi Jaringan Sistem

D. Skenario Pengujian Sistem

Pengujian dilakukan dengan cara melakukan berbagai macam serangan terhadap komputer *server*. Yang bertindak sebagai penyerang dalam skenario pengujian ini adalah komputer penyerang yang dengan media kabel LAN dan menggunakan penghubung switch. Serangan yang diujikan pada penelitian ini adalah *PING Attack*, *DoS/DDoS Attack*, *Port Scanning*, *Telnet Access* dan *FTP Access*. Pada setiap pengujian serangan, parameter Komputer *Server* yang berupa CPU, *Memory* dan Beban Jaringan akan di monitor untuk perbandingan sebelum serangan dan pada saat terjadi serangan.

a. PING Attack (ICMP Flooding)

PING Attack adalah penyerangan dengan mengeksploitasi *system* agar dapat membuat suatu target menjadi *crash*. Terjadinya *system* dikarenakan oleh pengiriman sejumlah paket yang besar bahkan sangat besar kearah target. Tidak semua perintah *PING* terhadap Komputer *Server* dapat dikatakan sebagai *PING Attack*, ada beberapa kondisi *PING* dimana Snort akan tetap mencatat *alert* tersebut namun tidak dikirimkan ke *Handphone Administrator*. Untuk lebih memahaminya dapat dilihat pada Tabel 3

Tabel 3. Protokol Kerja SMS *Alert* Terhadap *PING Attack*

No	Nama	*Alert* Dikirim
1	*PING normal 4 paket (ping 172.16.1.1)*	*Tidak*
2	*PING terus menerus (ping 172.16.1.1 –t)*	*Tidak*
3	*PING dengan lebih dari 10 paket dalam satu detik.*	*Ya*

Berdasarkan Tabel 3, SMS *Alert* hanya akan bekerja pada kondisi dimana terjadi *PING* dengan banyak paket lebih dari 10 dalam satu detik kepada Komputer *Server*. Hal ini dibuat untuk menghindari banyaknya SMS yang akan diterima oleh *Administrator* serta menimbulkan *false alarm*/ alarm yang salah.

b. DoD/DdoS Attack

Untuk pengujian *DoS/DDoS Attack* digunakan tool *freeware* LOIC (*Low Orbit Ion Cannon*), sama seperti *PING Attack*, ada beberapa kondisi *DoS/DDoS* dimana Snort akan tetap mencatat *alert* tersebut namun tidak dikirimkan ke *Handphone Administrator*. Untuk lebih memahaminya dapat dilihat pada Tabel 4.

Tabel 4. Protokol Kerja SMS *Alert* Terhadap *DoS/DDoS Attack*

No	Kondisi	*Alert* Dikirim
1	*DoS/DDoS dengan jumlah thread dibawah 100*	*Tidak*
2	*DoS/DDoS dengan speed 'slower'*	*Tidak*
3	*DoS/DDoS dengan thread diatas 100, speed faster, (jumlah paket DoS yang masuk >1000 dalam 3 detik)*	*Ya*

Berdasarkan Tabel 4, SMS *Alert* hanya akan bekerja pada kondisi dimana terjadi *DoS/DDoS* dengan banyak paket lebih dari 1000 dalam tiga detik kepada Komputer *Server*. Hal ini dibuat untuk menghindari banyaknya SMS yang akan diterima oleh *Administrator* serta menimbulkan *false alarm /* alarm yang salah.

c. Port Scanning

Untuk pengujian *Port Scanning* digunakan tool NMap-ZenMap. Untuk Port *Scanning*, setiap aksi dari Port *Scanning* selama bertujuan untuk memindai port TCP akan dikirimkan ke *Handphone Administrator*. Untuk lebih memahaminya dapat dilihat pada Tabel 5

Tabel 5. Protokol Kerja SMS *Alert* Terhadap Port *Scanning*

No	Kondisi	*Alert* Dikirim
1	*Port Scanning UDP*	*Tidak*
2	*Port Scanning TCP*	*Ya*

Berdasarkan Tabel 5, SMS *Alert* hanya akan bekerja pada kondisi dimana terjadi Port *Scanning* terhadap Port TCP.

d. Telnet Access

Untuk pengujian *Telnet* Akses *tools* yang digunakan adalah *Command prompt* yang tersedia pada *Windows.* Untuk *Telnet* Access, setiap aksi dari *Telnet* Access dikirimkan ke *Handphone Administrator*. Untuk lebih memahaminya dapat dilihat pada Tabel 6.

Tabel 6. Protokol Kerja SMS *Alert* Terhadap *Telnet* Akses

No	Kondisi	*Alert* Dikirim
1	*Telnet Akses terhadap Server (telnet 172.16.1.1)*	*Ya*

Berdasarkan Tabel 6, SMS *Alert* akan bekerja pada kondisi dimana terjadi *Telnet* Akses. Hal ini dikarenakan aktivitas *Telnet* hanya digunakan oleh *Administrator* jaringan sendiri. Apabila ada user yang mencoba untuk melakukan aktivitas *Telnet* maka akan ada SMS *Alert*.

e. FTP Access

Untuk pengujian FTP Akses *tools* yang digunakan adalah *Command prompt* yang tersedia pada *Windows*. Untuk FTP Access, apabila ada yang gagal untuk login ke FTP *Server* akan ada SMS *Alert* yang dikirimkan ke *Handphone Administrator*. Untuk lebih memahaminya dapat dilihat pada Tabel 7.

Tabel 7. Protokol Kerja SMS *Alert* Terhadap FTP Akses

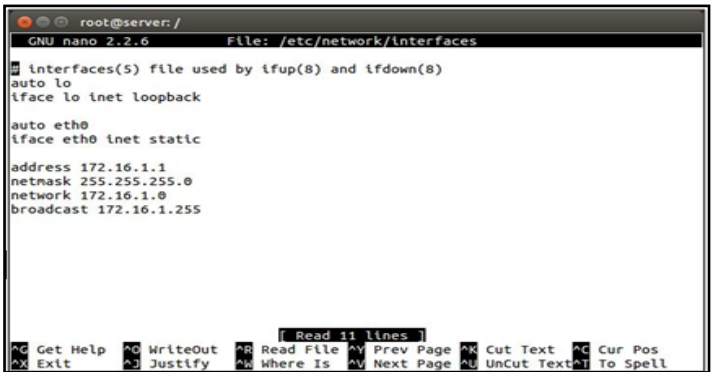
No	Kondisi	*Alert* Dikirim
1	*Login FTP Akses terhadap Server berhasil*	*Tidak*
2	*Login FTP Akses terhadap Server gagal*	*Ya*

Berdasarkan Tabel 7, SMS *Alert* akan bekerja pada kondisi dimana ada yang gagal untuk login ke FTP.

7. IMPLEMENTASI DAN PEMBAHASAN

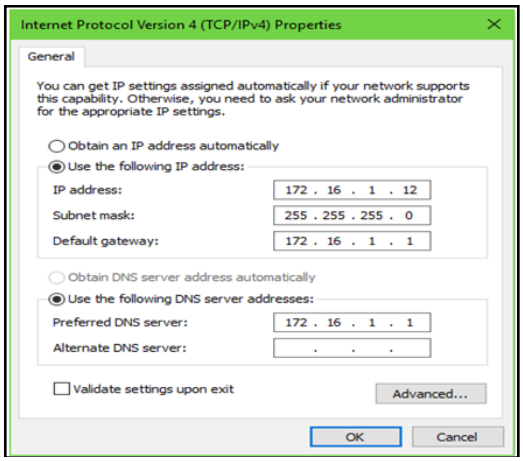
A. Konfigurasi IP Address Jaringan

Sebelum melakukan pengujian serangan terhadap sistem, perlu dilakukan persiapan terlebih dahulu. Persiapan pertama adalah konfigurasi IP Address pada masing-masing komputer agar dapat terhubung. Untuk Konfigurasi IP Address pada Komputer *Server* dapat dilihat pada Gambar 2



Gambar 2. IP Address Komputer *Server*

Untuk Konfigurasi IP Address pada Komputer Penyerang menggunakan IP *Address* 172.16.1.12 yakni satu *network* dengan Komputer *Server*. Selain itu Komputer Penyerang menggunakan IP *Default Gateway* dan DNS 172.16.1.1 seperti yang dapat dilihat pada Gambar 3.



Gambar 3. IP Address Komputer Penyerang

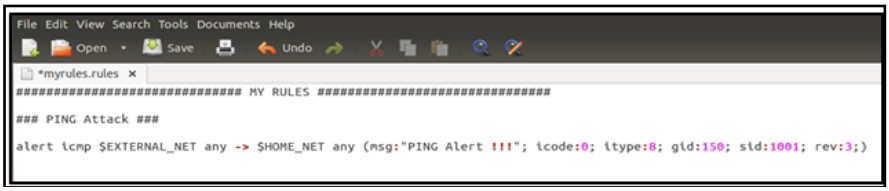
B. Pengujian Sistem

Apabila semua langkah persiapan selesai dilakukan, selanjutnya beralih ke tahap pengujian masing-masing serangan. Dalam pengujian ini penulis mengujikan serangan yang berupa:

a. PING Attack (ICMP Flooding)

Untuk pengujian *PING Attack* dengan cara sebagai berikut:

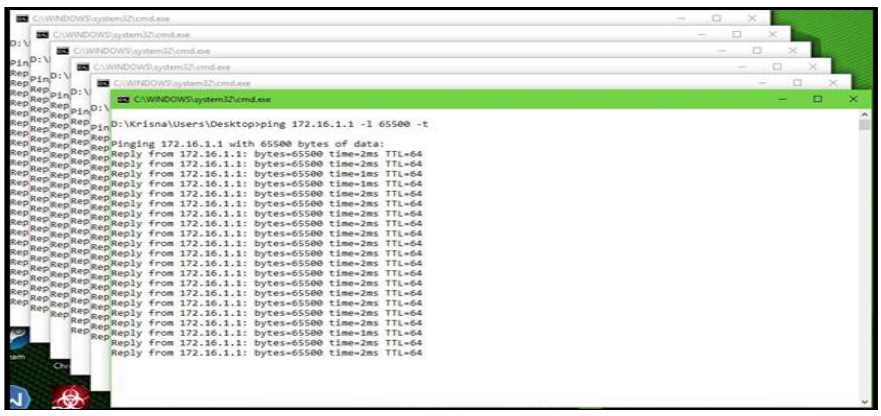
a) Tambahkan rules pada *file* myrules.rules untuk memberikan *alert* apabila ada yang mengirim paket ICMP (*PING*) ke Komputer *Server*.



Gambar 4. Rules Snort untuk *PING Attack*

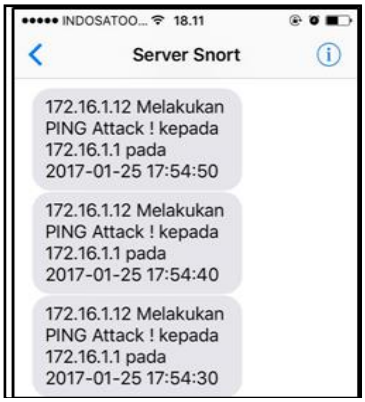
Rules yang digunakan untuk mendeteksi *PING Attack* adalah memberikan *alert* apabila ada ICMP dari $EXTERNAL_NET ke $HOME_NET dengan port apapun. *Alert* ini dikenali dengan pesan "*PING Alert* !!!" icode:0; dan itype:8; adalah kode dari sebuah aksi *PING*, gid: 150 berarti Generator ID nya adalah 150 dan sid;1001 berarti Snort *Alert* ID nya 1001 serta revisi nomor 3.

b) Lakukan *PING Attack* melalui Komputer Penyerang dengan membuka file *PING Attack* .bat yang ada di Windows sebanyak 25 kali. Lebih jelasnya dapat dilihat pada Gambar 5.



Gambar 5. *Ping Attack* Terhadap Komputer Sever

c) Lihat hasil SMS *Alert* yang dikirim oleh SMS *Gateway*.

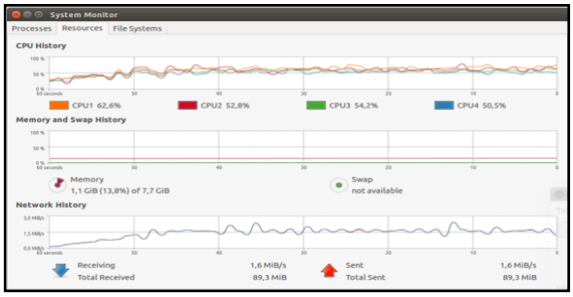


Gambar 6. SMS *Alert* pada *Handphone* Administrator

Pada Gambar 6 terlihat isi pesan SMS yang diterima oleh *Handphone Administrator*, waktu yang diperlukan sejak serangan terdeteksi sampai pesan diterima oleh *Handphone Administrator* adalah 20-30 detik.

d) Amati perbandingan hasil monitoring pengujian (CPU, *Memory*, Beban Jaringan) sebelum terjadi serangan dan pada saat terjadi serangan.

Gambar 7. Hasil Monitoring Sebelum Terjadi *PING Attack*



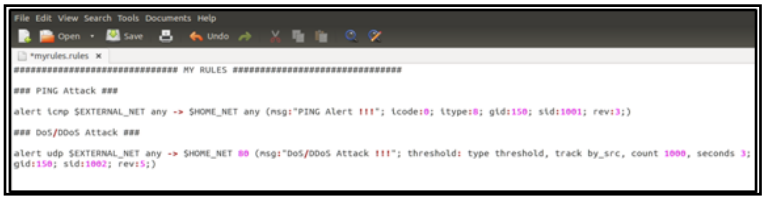
Gambar 8. Hasil Monitoring Pada Saat Terjadi *PING*

Pada Gambar 7 menunjukkan hasil monitoring sebelum terjadi *PING Attack*, rata-rata penggunaan CPU adalah 17,5%, penggunaan RAM sebesar 637 MB (8,1%) dan yang terakhir beban jaringan masih 0 *bytes*/s untuk Sent dan Receivingnya. Sedangkan pada saat terjadi serangan *PING Attack*, rata-rata penggunaan CPU naik menjadi 54,5%, penggunaan RAM sebesar 1,1 GB (13.8%) dan yang terakhir beban jaringan naik untuk Receiving menjadi 1,6 MB/s dan Sent sama yaitu 1,6 MB/s. Untuk lebih detail, dapat dilihat pada Gambar 8. Dari pengujian ini dapat dikatakan bahwa sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort ini berhasil memberikan peringatan berupa SMS *Alert* ke *Handphone* Adminstrator apabila terjadi *PING Attack*.

b. DoS/DDoS Attack

Untuk pengujian *DoS/DDoS Attack* digunakan tool *freeware* LOIC (*Low Orbit Ion Cannon*) dengan cara sebagai berikut:

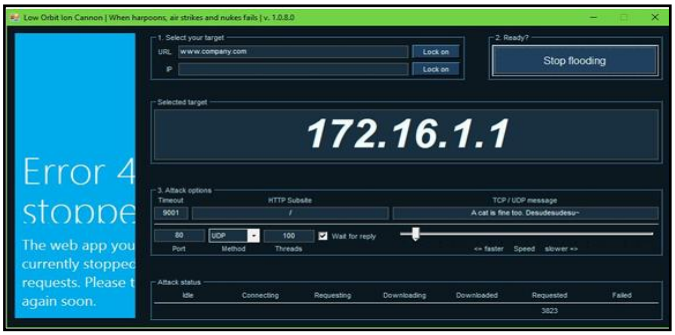
a) Tambahkan rules pada file myrules.rules untuk memberikan *alert* apabila ada yang melakukan *DoS/DDoS Attack* ke Komputer *Server* dengan kriteria *alert* apabila ada 1000 paket dalam 3 detik ke Komputer *Server*.



Gambar 9. Rules Snort untuk *DoS/DDoS Attack*

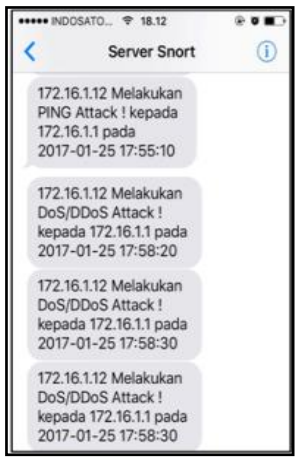
Pada Gambar 9, rules yang digunakan untuk mendeteksi *DoS/DDoS Attack* adalah memberikan *alert* apabila ada UDP dari $EXTERNAL_NET ke $HOME_NET dengan port 80. *Alert* ini dikenali dengan pesan "*DoS/DDoS Attack* !!!" dengan kriteria *alert* apabila ada 1000 paket dalam 3 detik berarti sebuah aksi *DoS/DDoS*, gid: 150 berarti Generator ID nya adalah 150 dan sid;1002 berarti Snort *Alert* ID nya 1002 serta nomor revisi 5.

b) Lakukan *DoS/DDoS Attack* melalui Komputer Penyerang dengan membuka Program LOIC dan mengatur target serangan yaitu 172.16.1.1 dan mengatur tipe serangan menjadi UDP jumlah threadnya 100 dan jalankan serangan. Untuk tingkat kecepatannya gunakan faster paling kiri. Lebih jelasnya dapat dilihat pada Gambar 10



Gambar 10. Aplikasi LOIC Melakukan *DoS/DDoS Attack*

c) Lihat hasil SMS *Alert* yang dikirim oleh SMS *Gateway*.

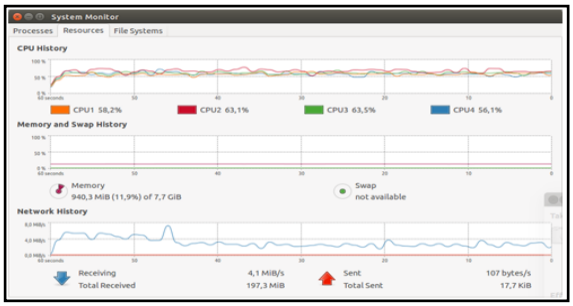


Gambar 11. SMS *Alert* pada *Handphone* Administrator

Pada Gambar 11 isi pesan SMS yang diterima oleh *Handphone Administrator* terlihat semula ada pesan "172.16.1.12 Melakukan *PING Attack* ! kepada 172.16.1.1" yakni pesan *PING Attack* dari pengujian sebelumnya kemudian diikuti dengan pesan "172.16.1.12 Melakukan *DoS/DDoS Attack* ! kepada 172.16.1.1" secara bertahap. Waktu yang diperlukan sejak serangan terdeteksi sampai pesan diterima oleh *Handphone Administrator* adalah 20-30 detik.

d) Amati perbandingan hasil monitoring pengujian (CPU, *Memory*, Beban Jaringan) sebelum terjadi serangan dan pada saat terjadi serangan.

Gambar 12. Hasil Monitoring Sebelum Terjadi *DoS/DDoS Attack*



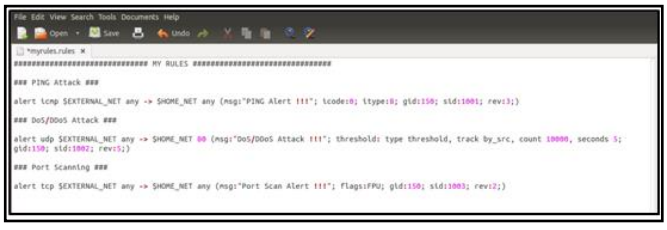
Gambar 13. Hasil Monitoring Pada Saat Terjadi *DoS/DDoS Attack*

Pada Gambar 12 menunjukkan hasil monitoring parameter sebelum terjadi *DoS/DDoS Attack*, rata-rata penggunaan CPU adalah 18%, penggunaan RAM sebesar 635 MB (8,1%) dan yang terakhir beban jaringan masih 0 *byte*s/s untuk Sent dan Receivingnya. Sedangkan pada saat terjadi serangan *DoS/DDoS Attack*, rata-rata penggunaan CPU naik menjadi 60%, penggunaan RAM sebesar 940 MB (11,9%) dan yang terakhir beban jaringan naik untuk Receiving menjadi 4,1 MB/s dan Sent yaitu 107 *byte*s. Untuk lebih detailnya dapat dilihat pada Gambar 13. Dari pengujian ini dapat dikatakan bahwa sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort ini berhasil memberikan peringatan berupa SMS *Alert* ke *Handphone* Adminstrator apabila terjadi *DoS/DDoS Attack*.

c. Port Scanning

Untuk pengujian *Port Scanning* digunakan tool NMap-ZenMap dengan cara sebagai berikut:

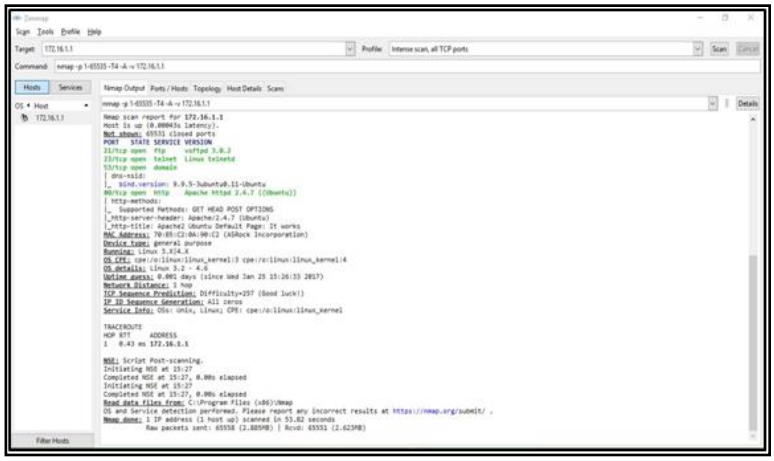
a) Tambahkan rules pada file myrules.rules untuk memberikan *alert* apabila ada yang melakukan Port *Scanning* ke Komputer *Server*.



Gambar 14. Rules Snort untuk Port *Scanning*

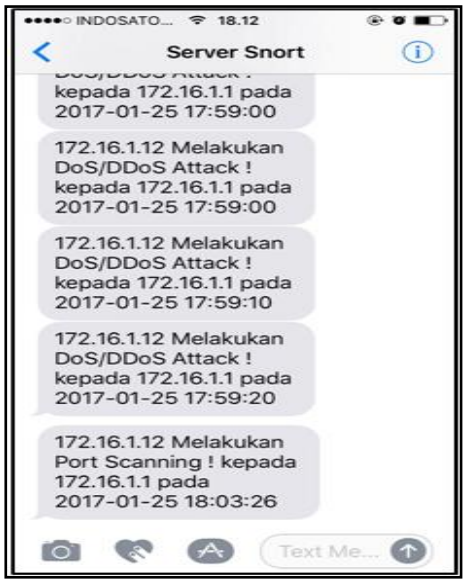
Pada Gambar 14, rules yang digunakan untuk mendeteksi *Port Scanning* adalah memberikan *alert* apabila ada TCP dari $EXTERNAL_NET ke $HOME_NET dengan port apapun. *Alert* ini dikenali dengan pesan "Port Scan *Alert* !!!" flags:FPU adalah flag yang dimiliki oleh nmap, gid: 150 berarti Generator ID nya adalah 150 dan sid;1003 berarti Snort *Alert* ID nya 1003 serta nomor revisi 2.

b) Lakukan Port *Scanning* dari Komputer Penyerang dengan membuka Program NMap - ZenMap dan menetapkan target IP menjadi 172.16.1.1 dengan setting profile "Intense scans, All TCP Ports". Apabila sudah siap, klik Start untuk mulai melakukan Port *Scanning*, proses ini memerlukan waktu sekitar satu menit. Untuk lebih jelasnya dapat dilihat pada Gambar 15.



Gambar 15. Aplikasi Zenmap melakukan Port *Scanning*

c) Lihat hasil SMS *Alert* yang dikirim oleh SMS *Gateway*.

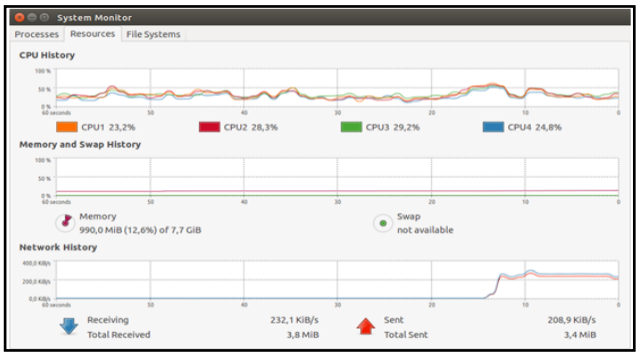


Gambar 16. SMS *Alert* pada *Handphone*

Pada Gambar 16 isi pesan SMS yang diterima oleh *Handphone* Administrator terlihat semula ada pesan "172.16.1.12 Melakukan *DoS/DDoS Attack* ! kepada 172.16.1.1" yakni pesan DoS *Attack* dari pengujian sebelumnya kemudian diikuti dengan pesan "172.16.1.12 Melakukan Port *Scanning* ! kepada 172.16.1.1". Waktu yang diperlukan sejak serangan terdeteksi sampai pesan diterima oleh *Handphone* Administrator adalah 20-30 detik.

d) Amati perbandingan hasil monitoring pengujian (CPU, *Memory*, Beban Jaringan) sebelum terjadi serangan dan pada saat terjadi serangan.

Gambar 17. Hasil Monitoring Sebelum Terjadi Port *Scanning*



Gambar 18. Hasil Monitoring Pada Saat Terjadi Port *Scanning*

Pada Gambar 17 menunjukkan hasil monitoring sebelum terjadi *Port Scanning*, rata-rata penggunaan CPU adalah 19%, penggunaan RAM sebesar 648 MB (8,2%) dan yang terakhir beban jaringan masih 0 *bytes*/s untuk Sent dan Receivingnya. Sedangkan pada saat terjadi serangan *Port Scanning*, rata-rata penggunaan CPU naik menjadi 26%, penggunaan RAM sebesar 990 MB (12,6%) dan yang terakhir beban jaringan naik untuk Receiving menjadi 232 KB/s dan Sent sama yaitu 208 KB/s. Untuk lebih detailnya dapat dilihat pada Gambar 18. Dari pengujian ini dapat dikatakan bahwa sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort ini berhasil memberikan peringatan berupa SMS *Alert* ke *Handphone* Adminstrator apabila terjadi Port *Scanning*

d. Telnet Access

Untuk pengujian *Telnet* Akses tools yang digunakan adalah Command prompt yang tersedia pada Windows, dengan cara sebagai berikut:

a) Tambahkan rules pada file myrules.rules untuk memberikan *alert* apabila ada yang mengakses *server* via *Telnet*.

Gambar 19. Rules Snort untuk *Telnet* Access

Pada Gambar 19, rules yang digunakan untuk mendeteksi *Telnet Access* adalah memberikan *alert* apabila ada TCP dari $HOME_NET dengan port 23 ke $EXTERNAL_NET. *Alert* ini dikenali dengan pesan "*Telnet Access* !!!" flow: from_*server* adalah aliran data berasal dari *Server*, content:

SERVER adalah isi konten yang dikirim, gid: 150 berarti Generator ID nya adalah 150 dan sid;1004 berarti Snort *Alert* ID nya 1004 serta nomor revisi 7.

b) Lakukan *Telnet* akses dari Komputer Penyerang dengan membuka Program Command prompt dan mengakses *server* dengan *telnet* 172.16.1.1 seperti pada gamabr 20.

Gambar 20. Tools Command Prompt melakukan *Telnet* Access

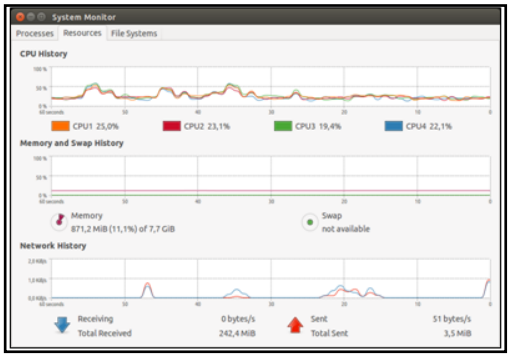
c) Lihat hasil SMS *Alert* yang dikirim oleh SMS *Gateway*.

Gambar 21. SMS *Alert* pada *Handphone* Administrator

Pada Gambar 21 isi pesan SMS yang diterima oleh *Handphone Administrator* terlihat semula ada pesan "172.16.1.12 Melakukan *Port Scanning* ! kepada 172.16.1.1" yakni pesan *Port Scanning* dari pengujian sebelumnya kemudian diikuti dengan dua pesan "172.16.1.12 Melakukan *Telnet* Akses ! kepada 172.16.1.1". Waktu yang diperlukan sejak serangan terdeteksi sampai pesan diterima oleh *Handphone Administrator* adalah 20-30 detik.

d) Amati perbandingan hasil monitoring pengujian (CPU, *Memory*, Beban Jaringan) sebelum terjadi serangan dan pada saat terjadi serangan.

Gambar 22. Hasil Monitoring Sebelum Terjadi *Telnet* Access



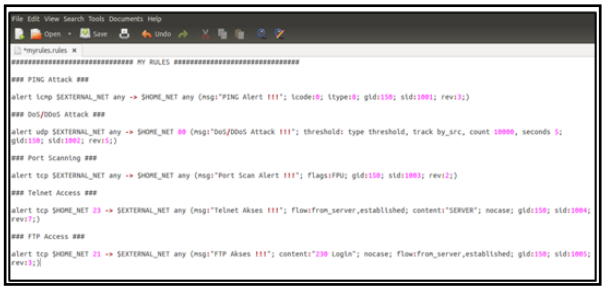
Gambar 23. Hasil Monitoring Pada Saat Terjadi *Telnet* Access

Pada Gambar 23 menunjukkan hasil monitoring parameter sebelum terjadi *Telnet* Akses, rata-rata penggunaan CPU adalah 15%, penggunaan RAM sebesar 712 MB (9,0%) dan yang terakhir beban jaringan masih 0 *bytes*/s untuk Sent dan Receivingnya. Sedangkan pada saat terjadi *Telnet Access*, rata-rata penggunaan CPU naik menjadi 22,5%, penggunaan RAM sebesar 871 MB (11,1%) dan yang terakhir beban jaringan tetap sama untuk Receiving yaitu 0 *bytes*/s dan Sent naik menjadi 51 *bytes*. Terlihat tidak ada perubahan yang signifikan terhadap penggunaan sumber daya pada saat terjadi *Telnet Access*. Hal ini dikarenakan *telnet* akses tidak bertujuan untuk melumpuhkan *server* secara langsung, namun untuk mendapatkan akses kendali *server*. Dari pengujian ini dapat dikatakan bahwa sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort ini berhasil memberikan peringatan berupa SMS *Alert* ke *Handphone* Adminstrator apabila terjadi *Telnet* Akses.

e. FTP Access (File Transfer Potocol)

Untuk pengujian FTP Akses *tools* yang digunakan adalah *Command prompt* yang tersedia pada *Windows*, dengan cara sebagai berikut:

a) Tambahkan rules pada file myrules.rules untuk memberikan *alert* apabila ada yang mengakses FTP *Server*.



Gambar 24. Rules Snort untuk FTP Access

Pada Gambar 24, rules yang digunakan untuk mendeteksi FTP *Access* adalah memberikan *alert* apabila ada TCP dari $HOME_NET dengan port 21 ke $EXTERNAL_NET. *Alert* ini dikenali dengan pesan "FTP *Access* !!!" flow: from_*server* adalah aliran data berasal dari *Server*, content: 230 *log*in adalah isi konten yang dikirim, gid: 150 berarti Generator ID nya adalah 150 dan sid;1005 berarti Snort *Alert* ID nya 1005 serta nomor revisi 3.

b) Lakukan FTP akses dari Komputer Penyerang dengan membuka Program Command prompt dan mengakses *server* dengan ftp 172.16.1.1.

Gambar 25. Tools Command Prompt melakukan FTP Access

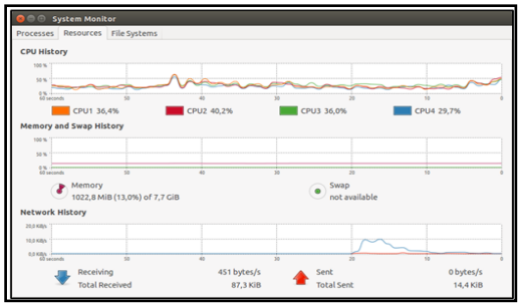
c) Lihat hasil SMS *Alert* yang dikirim oleh SMS *Gateway*.

Gambar 26. SMS *Alert* pada *Handphone*

Pada Gambar 26 isi pesan SMS yang diterima oleh *Handphone Administrator* terlihat semula ada pesan "172.16.1.12 Melakukan *Telnet* Akses ! kepada 172.16.1.1" yakni pesan *Telnet* Akses dari pengujian sebelumnya kemudian diikuti dengan pesan "172.16.1.12 Melakukan FTP Akses ! kepada 172.16.1.1". Waktu yang diperlukan sejak serangan terdeteksi sampai pesan diterima oleh *Handphone Administrator* adalah 20-30 detik.

d) Amati perbandingan hasil monitoring pengujian (CPU, *Memory*, Beban Jaringan) sebelum terjadi serangan dan pada saat terjadi serangan.

Gambar 27. Hasil Monitoring Sebelum Terjadi FTP Access



Gambar 28. Hasil Monitoring Pada Saat Terjadi FTP Access

Terlihat tidak ada perubahan yang signifikan terhadap penggunaan sumber daya pada saat terjadi FTP *Access*. Hal ini dikarenakan *telnet* akses tidak bertujuan untuk melumpuhkan *server* secara langsung, namun untuk mendapatkan akses *file* di *server*. Untuk lebih jelasnya dapat dilihat pada Gambar 28. Dari pengujian ini dapat dikatakan bahwa sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort ini berhasil memberikan peringatan berupa SMS *Alert* ke *Handphone* Adminstrator apabila terjadi FTP Akses.

8. PENUTUP

Berdasarkan pengujian dan pembahasan yang telah dilakukan maka dapat diambil kesimpulan terhadap sistem monitoring kemanaan *server* melalui pemberitahuan SMS berbasis Snort sebagai berikut:

1) Sistem monitoring kemanaan *server* yang diterapkan, menggukanan Snort sebagai mesin pendeteksi utama, Barnyard2 sebagai pembaca hasil dari keluaran Snort dan menyimpannya ke dalam *database*, BASE sebagai tampilan *database* dalam bentuk *web*, serta Gammu sebagai SMS *Gateway* untuk mengirim *alert* ke *Handphone Administrator*.
2) Sistem monitoring kemanaan *server* yang diterapkan telah berhasil dibangun dan diujikan. Secara keseluruhan, sistem ini dapat bekerja dengan baik sebagai pemberi peringatan dini adanya upaya serangan terhadap *server*.
3) Dari hasil pengujian masing-masing serangan, dapat dilihat pada saat terjadi serangan, penggunaan sumber daya untuk serangan *PING Attack*, *DoS/DDoS Attack* dan *Port Scanning* meningkat dalam hal penggunaan CPU, *Memory* dan Beban Jaringan. Namun untuk *Telnet* dan FTP Akses terlihat tidak ada perubahan yang signifikan terhadap penggunaan CPU, *Memory* dan Beban Jaringan.

Daftar Pustaka

Desnanjaya, I. G. M. N., Iswara, I. B. A. I., Ekayana, A. A. G., Santika, P. P., & Hartawan, I. N. B. (2020). Automatic high speed photography based microcontroller. *Journal of Physics: Conference Series*. https://doi.org/10.1088/1742-6596/1469/1/012096

Ekayana, A. A. G., Hartawan, I. N. B., Desnanjaya, I. G. M. N., & Joni, I. D. M. A. B. (2020). Body mass index measurement system as a desktop-based nutrition monitor. *J. Phys. Ser.*, *1469*.

Hartawan, I. N. B., & Iswara, I. B. A. I. (2015). Analisis Penerapan AoE dan LVM sebagai Teknologi Berbagi Media Penyimpanan pada Multi *Server*. *S@CIES*. https://doi.org/10.31598/sacies.v5i2.61

Hartawan, I. N. B., & Satwika, I. K. S. (2016). Rancang Bangun Laboratorium Virtual Berbasis Cloud Computing Di Stmik Stikom Indonesia. *S@CIES*. https://doi.org/10.31598/sacies.v7i1.117

Hartawan, I. N. B., & Sudiarsa, I. W. (2019). ANALISIS KINERJA INTERNET OF THINGS BERBASIS FIREBASE REAL-TIME DATABASE. *Jurnal RESISTOR (Rekayasa Sistem Komputer)*. https://doi.org/10.31598/jurnalresistor.v2i1.371

Rehman, R. U. (2003). *Intrusion Detection Systems with Snort*. New Jersey: Prentice Hall.

Tarigan, D. E. (2012). *Membangun SMS Gateway Berbasis Web dengan Codelgniter.* Jakarta: Lokomedia.