Perancangan dan Analisis Kinerja Sistem Pencegahan Penyusupan Jaringan Menggunakan Snort IDS dan Honeyd

AGITA SYAIMI PUTRI UTAMI[1] , LITA LIDYAWATI[1] , ZUL RAMADHAN[2]

1. Jurusan Teknik Elektro Institut Teknologi Nasional Bandung
2. PT. TELKOM Research and Development Center Bandung

Email : agita.pu3@gmail.com

ABSTRAK

Kelemahan sistem keamanan jaringan akan dimanfaatkan oleh penyusup (intruder) untuk melakukan serangan dengan cara mencuri data dan merusak jaringan komputer. Pada penelitian ini dilakukannya pencegahan penyusupan menggunakan Snort IDS dan Honeyd. Snort IDS ini bekerja dengan cara mendeteksi serangan yang telah dilakukan oleh penyusup (intruder). Setelah serangan berhasil terdeteksi, maka serangan tersebut akan dibelokkan ke server palsu (Honeyd). Akibat dari serangan penyusup adalah terjadinya gangguan pada sisi sistem kinerja server. Begitupula pada pemakaian CPU history adanya peningkatan kapasitas server 94,1% ketika terjadi serangan. Akan tetapi setelah terjadinya pembelokan, kapasitas server menurun menjadi 47,4%. Setelah dilakukan proses pendeteksian dan pembelokan maka sistem sudah bekerja dengan baik untuk mengamankan suatu jaringan komputer.

Kata kunci : *CPU history, Honeyd, kinerja sistem, penyusup, Snort IDS.*

ABSTRACT

A disadvantage of network security system will be utilized by the intruders to carry out attacks in a way to steal data and destroy the computer network. In this research , it was conducted the prevention to detect attack using Snort IDS and Honeyd methods. The Snort IDS worked by using an attack detection that conducted by intruder. After attack detection, it would be moved to the fake server (Honeyd). The results of the intruder attack were the interference on the performance of server side system. Similarly, the CPU usage history would increase. In the event of attack, the state of the server would increase to 94.1%, but after bending state of server, it would decrease to 47.4%. After detection and deflection processes, the system would work back into the normal condition for securing the computer network.

Keywords *: CPU history, Honeyd, , Intuder, system performance, Snort IDS.*

1. PENDAHULUAN

Keamanan jaringan komputer sangat penting untuk menjaga *validitas* dan *integritas* data serta menjamin ketersediaan layanan bagi penggunanya. Agar sistem jaringan komputer tidak terganggu bahkan sampai rusak oleh serangan penyusup (*intruder*), maka diperlukan sistem keamanan jaringan yang dapat menanggulangi dan mencegah serangan penyusup (*intruder*) tersebut.

Serangan yang paling sering digunakan adalah *Port Scanning* dan DOS (*Denial Of Service*). *Port Scanning* adalah serangan yang bekerja untuk mencari *port* yang terbuka pada suatu jaringan komputer, dari hasil *port scanning* akan didapat letak kelemahan sistem jaringan komputer tersebut. DOS adalah serangan yang bekerja denga cara mengirimkan *request* ke server berulang kali untuk bertujuan membuat *server* menjadi sibuk menanggapi *request* dan *server* akan mengalami kerusakan atau *hang* (Renuka P, Dr Annamma , Suhas.V, Kundan Kumar, 2010).

IDS (*Intrusion Detection System*) adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. Snort IDS merupakan IDS yang *open source* yang secara *defacto* menjadi standar IDS di industri. SNORT berfungsi untuk mendeteksi serangan dengan cara menghasilkan *alert* secara *real time* (Kerry J & Christopher, 2004).

IPS (*Intrusion Prevevtion System*) adalah sebuah aplikasi yang bekerja untuk monitoring *traffic* jaringan, mendeteksi aktivitas yang mencurigakan, dan melakukan pencegahan dini terhadap serangan. IPS mengkombinasikan teknik *firewall* dan metoda *Intrusion Detection System* (IDS) dengan sangat baik.

Iptables adalah *firewall* bawaan Linux yang bekerja mengatur lalu lintas data dalam komputer, baik yang masuk ke komputer, keluar dari komputer, maupun sekedar melewati komputer. Honeyd adalah *server* palsu yang merupakan sebuah produk honeypot dengan interaksi rendah dengan berfungsi mensimulasikan tingkah laku sebuah komputer beserta sistem operasinya (Lance S, 2002).

Dalam pengerjaan penelitian ini telah dilakukan perancangan dan analisis kinerja Snort IDS (*Intrusion Detection System*) dan Honeyd yang dapat melindungi *server* dari serangan penyusup. Dengan adanya sistem keamanan jaringan ini dapat mempermudah administator melindungi *server* dari serangan penyusup (*intruder*).

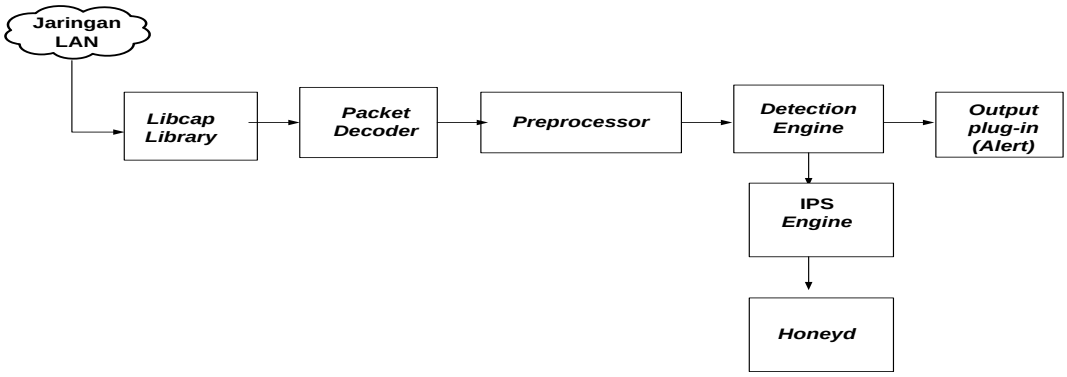
2. METODOLOGI PERANCANGAN PENCEGAHAN PENYUSUPAN

2.1 Perancangan Sistem Pencegahan Penyusupan Jaringan Komputer

Dalam perancangan ini akan dijelaskan tahap-tahap bagaimana sistem pencegahan penyusupan dirancang. Agar memenuhi kebutuhan fungsional, sistem pencegahan penyusupan pada jaringan menggunakan Snort IDS dan Honeyd dibutuhkan beberapa modul utama diantaranya : *libpcap library, packet decoder*, *preprocessor*, *detection engine*, *output modules*, IPS *engine* dan *Honeyd*.

2.2 Diagram Blok Sistem Pencegahan Penyusupan Menggunakan Snort IDS dan Honeyd

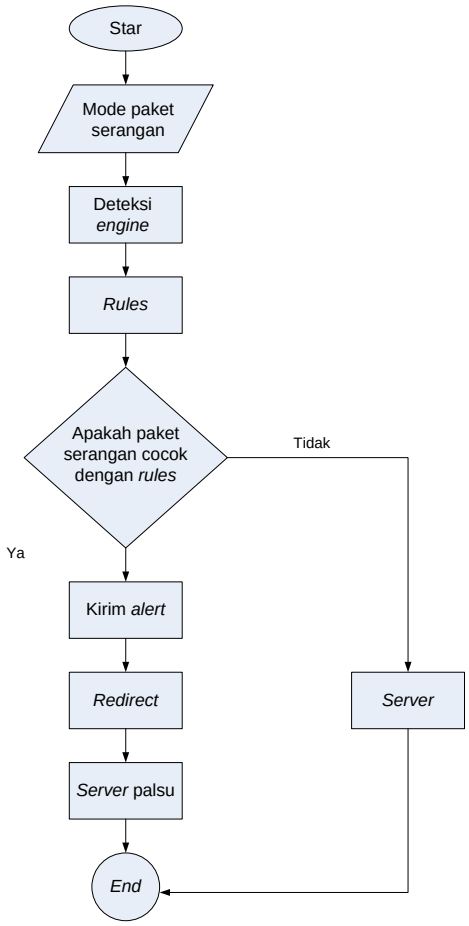
Gambar 1 merupakan diagram blok sistem pencegahan penyusupan yang dimana *libpcap library*, *packet decoder*, dan *preproccesor* bekerja untuk menangkap dan mengelompokan paket data yang ada dalam suatu jaringan. *Detection engine* bekerja menentukan apakah paket data tersebut terdeteksi serangan atau bukan. IPS *engine* bekerja membaca *alert* pada *database* lalu memerintahkan *iptables* membelokan (*redirect*) serangan. Honeyd bekerja sebagai *server* palsu dimana mensimulasikan tingkah laku sebuah komputer beserta sistem operasinya, sedangkan *output plug-in* bekerja menghasilkan *report* atau *alert*.



Gambar 1. Diagram Blok Sistem Pencegahan Penyusupan

2.3 *Flowchart* Sistem Pencegahan Penyusupan

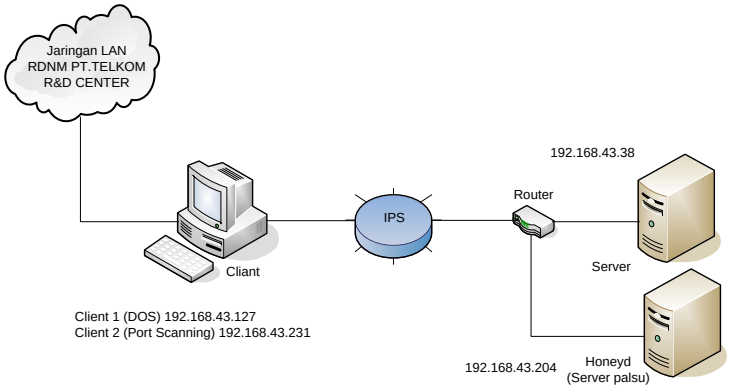
Untuk mengetahui prinsip kerja dari sistem yang akan dirancang, maka agar lebih mudah dalam pemahamannya dibuatlah terlebih dahulu diagram alir (*flowchart*) dari sistem tersebut. Gambar 2 menjelaskan apabila ada paket data yang masuk, sistem ini akan mulai bekerja yaitu dengan mengidentifikasi mode paket serangan apakah paket data tersebut. Setelah diketahui mode paketnya, pada *detection engine* akan membandingkan apakah sama dengan *rules* yang telah ada pada Snort, jika sama maka Snort akan mengeluarkan *alert* lalu membelokkan serangan tersebut ke Honeyd, jika tidak paket data tersebut langsung di kirim ke *server*.



Gambar 2. *Flowchart* Sistem Pencegahan Penyusupan

2.4 Topologi

Topologi yang dirancang dapat diilustrasikan pada Gambar 3 yaitu terdapat Jaringan Lan yang terdapat pada PT.Telkom, client berupa DOS dan Port Scanning, IPS yang terdiri dari Snort IDS dan blockit, *server*, dan Honeyd (*server* palsu)(fauzi, 2010).

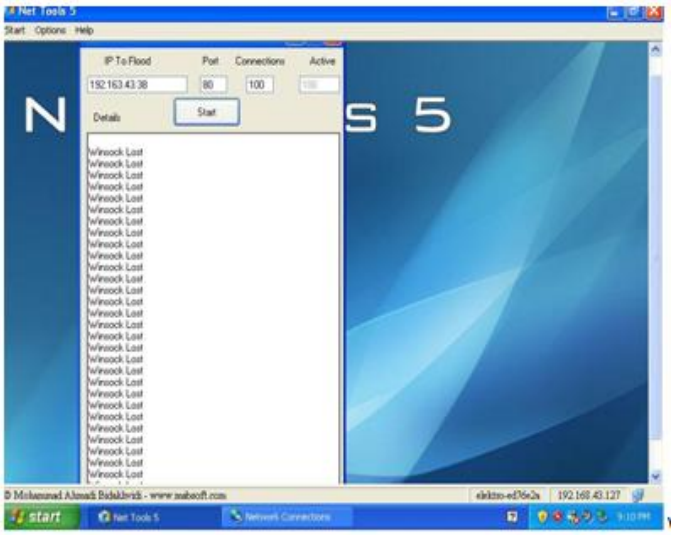


Gambar 3. Topologi Sistem Pencegahan Penyusupan

3. HASIL PENGUJIAN DAN ANALISIS SISTEM PENCEGAHAN PENYUSUPAN

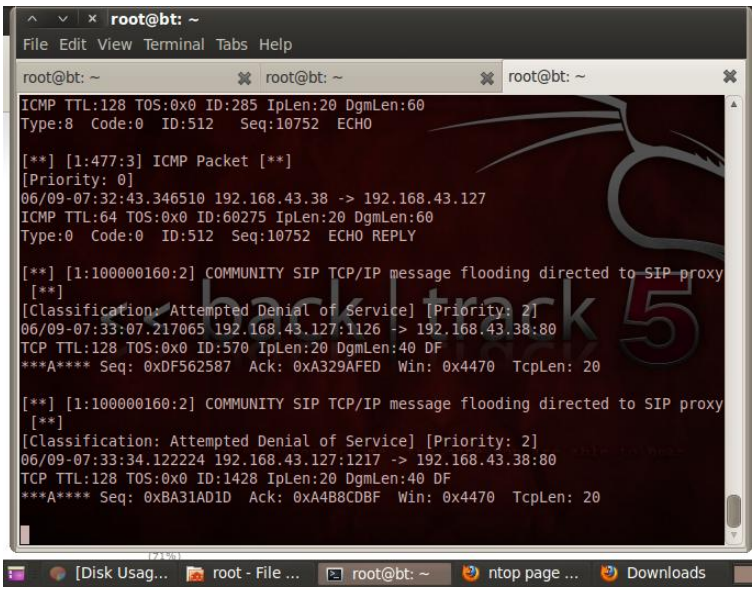
3.1 Pengujian Sistem Pencegahan Penyusupan Jaringan

Untuk menguji sistem pencegahan penyusupan yaitu dengan cara melancarkan paket serangan dari *client* ke *server*. Dalam perancangan ini, serangan yang digunakan adalah *Port Scanning* dan DOS.



Gambar 4. Paket Serangan DOS (*Denial Of Service*)

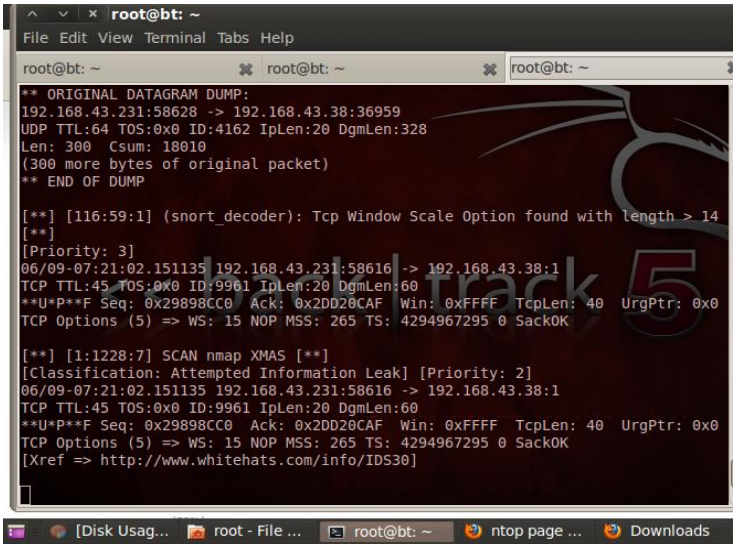
Serangan DOS dibangkitkan oleh *Net-Tools* pada *client* 1 dengan IP 192.168.43.127, pada Gambar 4 memperlihatkan bahwa *client 1* telah mengirimkan paket serangan. Setelah serangan dikirim, Snort di *server* mengeluarkan *alert* bahwa IP 192.168.43.127 melakukan serangan berupa DOS dapat dilihat pada Gambar 5 yang terdiri dari IP penyerang, IP yang diserang dan bentuk serangan (Rafiudin, 2010).



Gambar 5. *Alert* Snort berupa serangan DOS di *Server*

Gambar 6. Paket Serangan *Port Scanning* (nmap)

Serangan *Port Scanning* dibangkitkan oleh nmap berupa Zenmap pada *client* 2 dengan IP 192.168.43.231, pada Gambar 6 memperlihatkan bahwa *client* 2 telah mengirimkan paket serangan. Setelah serangan dikirim, Snort di *server* mengeluarkan alert bahwa IP 192.168.43.127 melakukan serangan berupa *Port Scanning* (nmap) yang dapat dilihat pada Gambar 7 yang terdiri dari IP penyerang, IP yang diserang dan bentuk serangan.

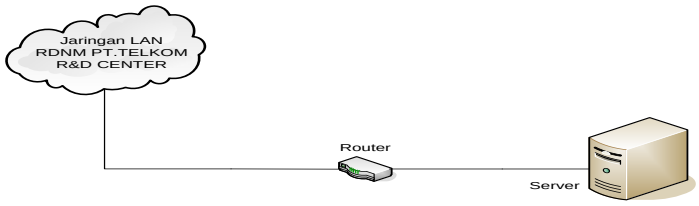


Gambar 7. *Alert* Snort berupa serangan *Port Scanning* di *Server*

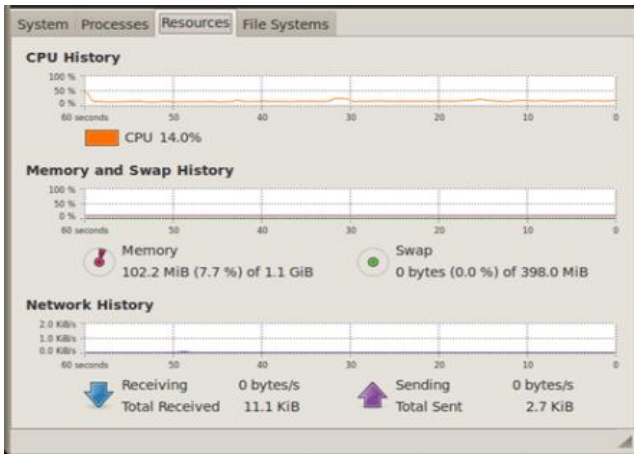
3.2 Hasil Pengujian

Pengujian dilakukan dengan beberapa tahap. Berikut ini tahap-tahap hasil pengujian yang dilakukan:

a. Hasil pengujian saat tidak ada serangan penyusup (*intruder*)
Pada pengujian ini, sistem pencegahan penyusupan dalam keadaan normal yaitu tidak ada paket data yang dikirim berupa serangan *port scanning* dan *DOS* dengan topologi pada Gambar 8. Hasil dari pengujian sebelum adanya serangan dapat dilihat pada Gambar 9 yaitu keadaan CPU menjadi 14% dan kapasitas memory 102,2 MB.

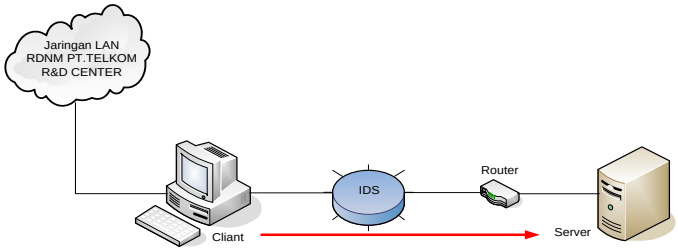


Gambar 8. Topologi pengujian saat tidak ada serangan penyusup (*intruder*)

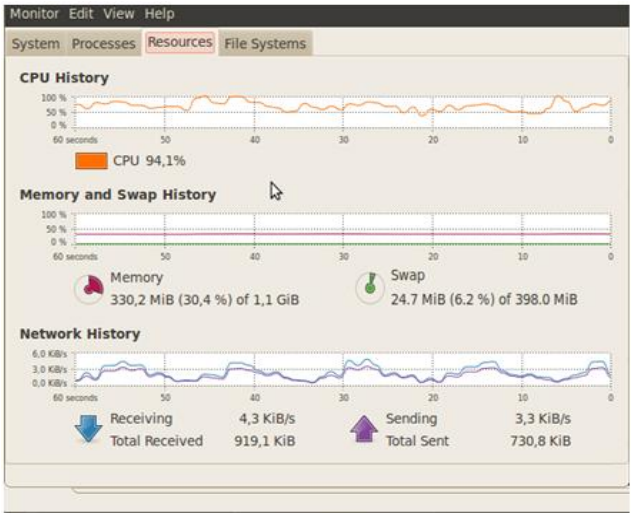


Gambar 9. Keadaan *Server* pada saat tidak ada serangan penyusupan (*intruder*)

b. Hasil pengujian saat ada serangan penyusup (*intruder*)
Pada pengujian ini, serangan berupa *port scanning* dan DOS langsung menuju *server* tanpa adanya pencegahan penyusupan dengan topologi pada Gambar 10. Hasil dari pengujian setelah adanya serangan dapat dilihat pada Gambar 11 yaitu keadaan CPU naik menjadi 94,1% dan kapasitas memory naik menjadi 330,2 MB.

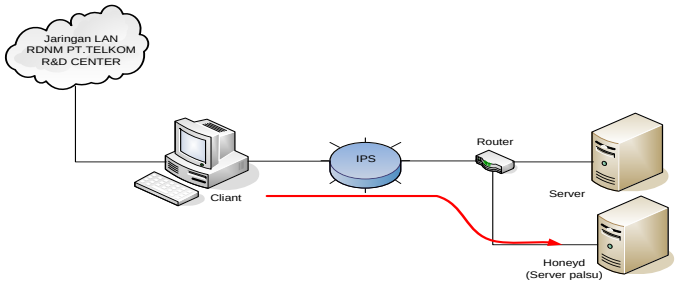


Gambar 10. Topologi pengujian saat terjadi serangan penyusup (*intruder*)

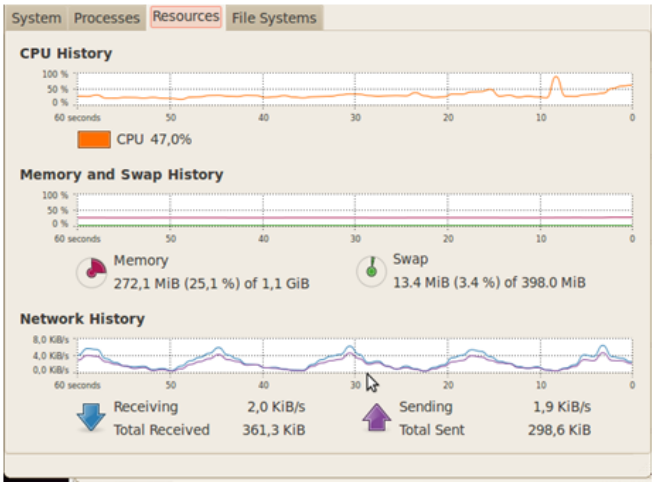


Gambar 11. Keadaan *Server* saat terjadi serangan penyusup (*intruder*)

c. Hasil pengujian pembelokan (*redirect*) serangan penyusup (*intruder*)
Pada pengujian ini, serangan yang berupa *port scanning* dan DOS akan dibelokan (*redirect*) menuju *server* palsu (Honeyd) dengan toplogi pada Gambar 12. Hasil dari pengujian setelah adanya pembelokan serangan penyusup dapat dilihat pada Gambar 13 yaitu keadaan CPU turun menjadi 47,0% dan kapasitas memory turun menjadi 272,1 MB.



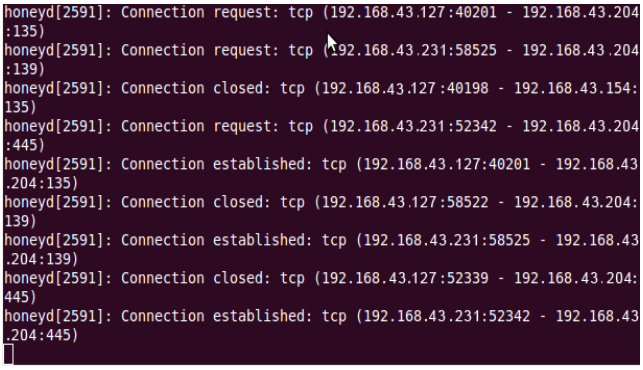
Gambar 12. Topologi pengujian saat terjadi pembelokan serangan penyusup



Gambar 13. Keadaan *Server* saat terjadi pembelokan serangan

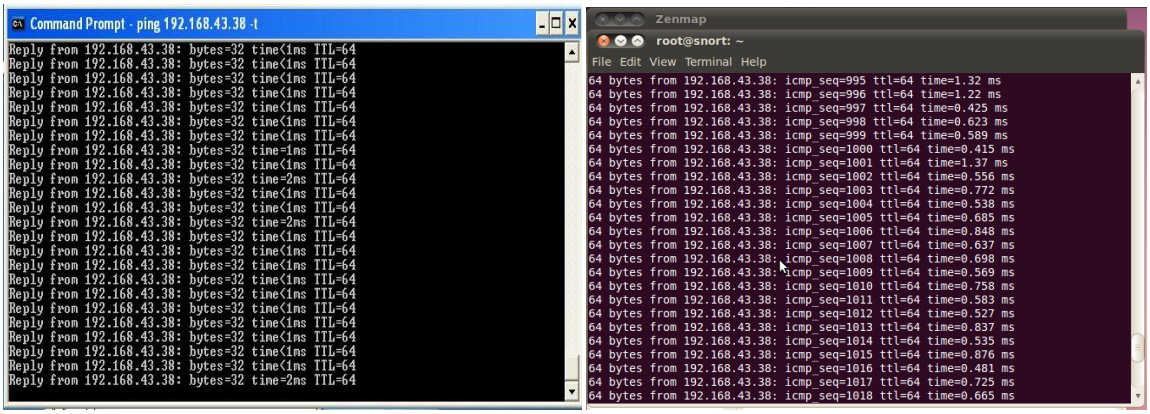
Setelah mendapatkan serangan penyusup (*intruder*) berupa *port scanning* dan *DOS*, Snort akan mengeluarkan *alert*. *Alert* yang keluar dari Snort akan dibaca oleh IPS *engine* berupa *blockit* (Isnan, 2011) yang bertugas untuk memerintahkan *iptables* untuk membelokan akses penyusupan ke *server* palsu (Honeyd) dengan sintaks seperti berikut:

```
tail –f /var/log/blockit/intruder
```



Gambar 14. Aktifitas Serangan Pada Honeyd

Pada Honeyd (*server* palsu) akan menampilkan aktifitas serangan yang terdiri IP serangan *port scanning* dan IP DOS seperti pada Gambar 14 (Utdirartatmo, 2005). Pembelokan serangan pada *server* palsu, tidak memberikan kecurigaan pada *intruder* (penyusup), karena penyusup (*intruder)* dengan IP 192.168.43.127 dan 192.168.43.231 masih bisa mengakses IP 192.168.43.38 dapat dilihat pada Gambar 15 (a) bukti bahwa DOS masih terhubung pada *server* dan (b) bukti bahwa *port scanning* masih terhubung pada *server*.



(a) (b)

Gambar 15. (a) Bukti Bahwa DOS (*Denial Of Service*) Masih Terhubung Ke *Server*. (b) Bukti Bahwa *Port Scanning* Masih Terhubung ke *Server*

3.3 Analisa Pengujian

Setelah dilakukan pengujian akan terlihat dampak yang mengganggu pada sistem kinerja *server* dapat dilihat pada Tabel 1 :

Tabel 1. Sistem Kinerja *Server*

No	***Performance System***	**Sebelum Serangan**	**Setelah Serangan**	**Setelah Pembelokan Serangan**
1	*CPU History*	14%	94,1%	47,4%
2	*Memory*	102,2 MB	330,2 MB	294,4 MB
3	*SWAP Memory*	0 bytes	24,7 MB	13,4 MB
4	*Receiving* Data	0 bytes/s	43,KBps	2,0 KBps
5	*Sending* Data	0 bytes/s	3,3 KBps	1,9 KBps
6	Total *Receive* Data	11,1 KB	919,1 KB	361,3 KB
7	Total *Send* Data	2,2 KB	730,8 KB	298,6 KB

Hasil dari beberapa pengujian pada Tabel 1 dapat dilihat kondisi setelah adanya serangan pada kapasitas CPU *History* dan *Memory* mengalami kenaikan yang berdampak pada sistem kinerja *server,* sedangkan pada kondisi setelah adanya pembelokan (*redirect*) kapasitas CPU *History* dan *Memory* akan berkurang sehingga sistem kinerja server tidak terganggu akan tetapi penurunan kapasitas tersebut tidak kembali pada kondisi sebelum adanya serangan. Hal ini dikarenakan *ping* ataupun *request* menuju *server* yang berkapasitas kecil tidak dianggap serangan sehingga tidak dibelokan ke Honeyd (*server* palsu). Dengan adanya perubahan kapasitas setelah adanya pembelokan (*redirect*) ke Honeyd (*server* palsu) dapat dilihat bahwa sistem pencegahan penyusupan yang telah dibuat bekerja dengan baik.

4. KESIMPULAN

Dari hasil pengujian sistem pencegahan penyusupan jaringan komputer dan analisis data dapat diambil kesimpulan bahwa :

1. *Rules* pada Snort berpengaruh penting dalam sistem pencegahan penyusupan, *rules* tersebut harus aktif melakukan *update*, agar semakin terlindungi dari gangguan atau serangan penyusup (*intruder*).
2. Paket serangan yang dikirim memberikan dampak terhadap sisi pada sistem kinerja *server*, dapat dilihat dari meningginya pada *processor, memory, swap* dan *traffic jaringan*. Jika keadaan ini terjadi cukup lama maka kondisi hadware akan rusak (*hang*).
3. Setelah penyusup (*intruder*) menyerang lalu terdeteksi oleh Snort maka *blockit* sebagai IPS *engine* akan membelokan serangan penyusup (*intruder*) ke server palsu (Honeyd).
4. Setelah adanya pembelokan serangan DOS dan *port scanning* ke Honeyd (*server* palsu), sistem kinerja *server* berjalan dengan baik.

DAFTAR RUJUKAN

Spitzner, L. (2002, September 13). Honeypots : Definitions and Value of Honeypots. Dipetik Juli 12, 2012, http://www.tracking-hackers.com

Kerry J. Cox & Christopher Gerg. (2004). "Managing Security With SNORT and IDS Tools", O'RIELLY.

Renuka Prasad.B, Dr Annamma Abraham, Suhas.V, Kundan Kumar. (2010). "DoS Attack Pattern Generator For Training The Neural Network Based Classifier To Dynamically Blacklist IP in HoneyPot Based NIDS/NIPS". International Conference on Contours of Computing ,Springerlink, ISBN -978-84-8489-988-2

Fauzi, S. R. (2010). *Implementasi Intrusion Detection And Prevention System Di PT. Telekomunikasi Indonesia*. Bandung: Politeknik Telkom.

Isnan, A. I. (2011). *Pengembangan Smart IPS untuk Mereduksi False Positive.* Bandung: Universitas Pendidikan Indonesia (UPI) Bandung.

Rafiudin, R. (2010). *Menggayang Hacker dengan Snort.* Yogyakarta: Andi.

Utdirartatmo, F. (2005). *Menjebak Hacker dengan Honeypot.* Yogyakarta: Andi.