IMPLEMENTASI SNORT SEBAGAI ALAT PENDETEKSI INTRUSI MENGGUNAKAN LINUX

Mohammad Affandi, Sigit Setyowibowo
Program Studi Teknik Informatika
STMIK PPKIA Pradnya Paramita Malang
Jl. Laksda Adi Sucipto No. 249-A Malang
Email: sigit@pradnya.ac.id

ABSTRACT

Network security system on the server is an important factor to ensure the stability , integrity and validity of the data . Implementation of Snort -based Intrusion Detection System can save the cost of procurement of software because it is free and quite reliable in detecting security attacks . Snort -based IDS systems can be implemented on the Linux operating system . Snort main settings and network settings , especially on existing Snort rule . An attack can be detected or not by SnortI IDS , depending on the presence or absence of an appropriate rule.
Testing the IDS system was done with several attack patterns to test the reliability of Snort to detect an attack against the security system . Based on the results of testing the system Snort IDS with ping , nmap port scanning , exploits , SQL Injection , accessing the database . Snort can provide warning of an attack against the security of a network system . The warning results can be used as a reference for determining the network security policy.

***Key words** : Linux , Intrusion Detection System , Snort.*

PENDAHULUAN

Keamanan pada server adalah salah satu faktor penting yang harus diperhatikan. Keamanan yang terjamin dapat menghindari kerugian yang disebabkan oleh serangan terhadap sistem keamanan jaringan tersebut.

Sistem keamanan jaringan merupakan faktor penting untuk menjamin stabilitas, integritas dan validitas data. Salah satu perangkat IDS yang sering digunakan pada sistem server adalah *Snort* . *Snort* adalah perangkat lunak berbasis IDS yang dibuat dan dikembangkan oleh Martin Roesch, kemudian menjadi sebuah proyek open source.

Implementasi Aplikasi pendeteksi intrusi / *Intrusion Detection System* berbasis *Snort* dapat menghemat biaya pengadaan software karena bersifat gratis dan cukup handal dalam mendeteksi serangan keamanan. Sistem IDS berbasis *Snort* dapat di-implementasikan pada sistem *Linux*.

Pengaturan utama *Snort* terutama pada pengaturan jaringan dan rule *Snort* yang ada. Sebuah serangan dapat terdeteksi atau tidak oleh *Snort* IDS, tergantung dari ada atau tidaknya rule yang sesuai. Pengujian pada sistem IDS dilakukan dengan beberapa pola serangan untuk menguji kehandalan *Snort* dalam mendeteksi sebuah serangan terhadap sistem keamanan. Berdasarkan hasil pengujian sistem *Snort* IDS dengan *ping, nmap*, *SQL Injection*, *eksploitasi*, dan pengaksesan database, *Snort* dapat memberikan peringatan adanya serangan keamanan terhadap sistem jaringan. Hasil peringatan tersebut dapat

digunakan sebagai acuan untuk menentukan kebijakan keamanan jaringan.

Salah satu cara untuk meningkatkan keamanan dalam jaringan adalah dengan mengimplementasikan *Intrusion Detection System* (IDS). *Intrusion Detection System* adalah sebuah sistem yang digunakan untuk melakukan deteksi adanya usaha-usaha penyusupan terhadap sebuah sistem dengan melakukan pengamatan trafik secara *real-time*.

Beberapa keunggulan *Snort* dibandingkan *software* IDS lain adalah kode sumber berukuran kecil, dapat digunakan pada banyak sistem operasi, cepat dan mampu mendeteksi serangan pada jaringan, mudah dikonfigurasi dan terutama *Snort* ini bersifat gratis.

Rumusan Masalah

Berdasarkan beberapa pertimbangan di atas, maka dapat dirumuskan rumusan masalah sebagai berikut :

Bagaimana hasil implementasi *Snort* sebagai *Intrusion Detection System* dapat digunakan untuk meningkatkan keamanan server pada Linux ?

Tujuan

Tujuan yang ingin dicapai dari penelitian ini adalah :

Meningkatkan keamanan server dengan *Snort* IDS melalui beberapa pengujian, sehingga didapat kesimpulan yaitu manfaat penggunaan *Snort* sebagai aplikasi *Intrusion Detection System.*

Manfaat

Manfaat yang didapat dari hasil penelitian ini adalah meningkatnya keamanan dalam jaringan, dapat digunakan pada banyak system operasi, cepat dan mampu mendeteksi serangan pada jaringan, mudah dikonfigurasi dan gratis.

LANDASAN TEORI

linux

linux adalah nama yang diberikan kepada sistem operasi komputer bertipe Unix. Linux merupakan salah satu contoh hasil pengembangan perangkat lunak bebas dan sumber terbuka utama. Seperti perangkat lunak bebas dan sumber terbuka lainnya pada umumnya, kode sumber Linux dapat dimodifikasi, digunakan dan didistribusikan kembali secara bebas oleh siapa saja.(www.linux.org)

Distribusi *linux* yang digunakan pada penelitian ini adalah Ubuntu 10.04. Ubuntu merupakan salah satu distribusi *Linux* yang berbasiskan Debian dan didistribusikan sebagai software bebas.

Intrusion Detection System

Intrusion Detection System adalah sebuah alarm keamanan yang dikonfigurasi untuk melakukan pengamatan terhadap *access point*. aktifitas *host*. dan kegiatan penyusupan. Cara paling sederhana untuk mendefinisikan IDS mungkin tergantung dari bagaimana mendeskripsikan IDS sehagai *tool* spesial yang dapat membaca dan mengintepretasikan isi dari *file-file* log dari *router*, *firewall serve,r* dan perangkat jaringan lainnya. Secara lebih

spesifik, *Intrusion Detection System* adalah sebuah sistem yang dapat mendeteksi adanya penggunaan tak ter-autorisasi (unauthorized use) pada sebuah sistem jaringan (Beale, 2003).

Dilihat dari cara kerja dalam menganalisa apakah paket data dianggap sebagai penyusupan atau bukan, IDS dibagi menjadi 2:

a. Knowledge-based.

Knowledge-based IDS dapat mengenali adanya penyusupan dengan cara menyadap paket data kemudian membandingkannya dengan

database rute IDS yang berisi signature-signature paket serangan. Berdasarkan pembandingan tersebut, jika paket data mempunyai pola yang sama dengan setidaknya salah satu pola di dalam database rule IDS, maka paket tersebut dapat dianggap sebagai serangan, dan demikian juga sebaliknya. Apabila paket data tersebut sama sekali tidak mempunyai pola yang sama dengan pola di database rule IDS. maka paket data tersebut dianggap bukan serangan.

b. Behavior based (anomaly)

IDS jenis ini dapat mendeteksi adanya penyusupan dengan mengamati adanya kejanggalan pada sistem, atau adanya penyimpangan-penyimpangan dari kondisi normal. Sebagai contoh, apabila terdapat penggunaan memori yang melonjak secara terus menerus atau terdapat koneksi paralel dari sebuah IP Address dalam jumlah yang banyak dalam waktu yang bersamaan, maka kondisi tersebut dapat dianggap sebagai sebuah kejanggalan, yang kemudian oleh IDS dianggap sebagai serangan.

Sedangkan dilihat dari kemampuan mendeteksi penyusupan pada jaringan. IDS dibagi menjadi 2. yaitu:

a. Host-based Intrusion Detection System.

Host-based mampu mendeteksi hanya pada host tempat implementasi IDS.

b. Network-based Intrusion Detection System.

Network-based IDS mampu mendeteksi seluruh host yang berada satu jaringan dengan host implementasi IDS tersebut.

Snort

Snort merupakan salah satu contoh program *Network-based Intrusion Detection System*, yaitu sebuah program yang dapat mendeteksi suatu usaha penyusupan pada suatu sistem jaringan komputer. *Snort* bersifat *open source* dengan lisensi GNU General Purpose License sehingga software ini dapat dipergunakan untuk mengamankan sistem server tanpa harus membayar biaya lisensi (Snort team 2009).

Suatu sistem IDS harus bersifat lintas platform, mempunyai sistem *footprinting* yang ringan, dan mudah dikonfigurasi oleh administrator sebuah sistem yang membutuhkan implementasi dari solusi kemananan dalam waktu yang singkat. Implementasi tersebut dapat berupa seperangkat software yang dapat diasosiasikan dalam melakukan aksi untuk merespon sistuasi keamanan tertentu. Selain itu. sebuah sistem IDS juga harus *powerfull* dan cukup fleksibel

untuk digunakan sebagai bagian permanen dari suatu sistem jaringan.

Snort memenuhi kriteria tersebut, yaitu dapat dikonfigurasi dan dibiarkan berjalan untuk periode yang lama tanpa meminta pengawasan atau perawatan bersifat administratif sebagai bagian dari sistem keamanan terpadu sebuah infrastruktur jaringan. Snort iuga dapat berjalan pada semua platform sistem operasi di mana *libpcap* dapat berjalan. Sampai saat ini, Snort telah teruji dapat berjalan pada sistem operasi RedHat Linux, Debian Linux, MkLinux, HP-UX, Solaris (x86 dan Sparc), x86 Free/Net/OpenBSD, Windows dan MacOS X.

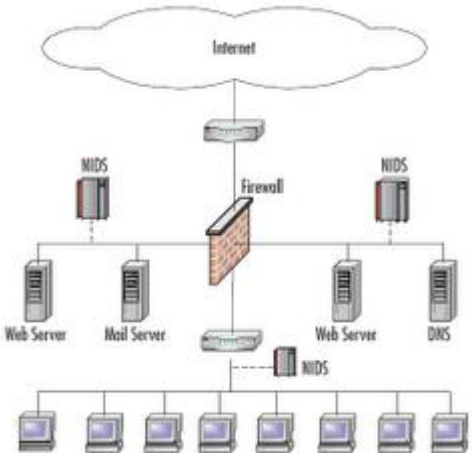
Snort dapat dioperasikan dalam 4 mode, yaitu:

a. ***Sniffer mode***. untuk menangkap dan melihat paket data yang lewat pada jaringan.
b. ***logger mode*** untuk mencatat semua paket data yang lewat pada jaringan untuk dianalisa di kemudian hari.
c. ***Intrusion Detection Mode***. pada mode ini Snort akan akan berfungsi untuk mendeteksi tindakan serangan yang dilakukan pada jaringan komputer. Untuk menggunakan mode ini diperlukan setup dari beberapa file atau aturan yang akan membedakan sebuah paket normal dengan paket yang membawa serangan.
d. ***Inline mode***, yaitu dengan membandingkan paket data dengan dengan rule iptables dan libpcap dan kemudian dapat menentukan iptables untuk melakukan drop atau allow paket tersebut berdasarkan rules Snort yang ditentukan secara spesifik.

Penempatan Snort sebagai IDS dalam jaringan dapat dilakukan dengan 3 konsep *Snort* IDS Placement, tergantung kebutuhan dan ketersediaan perangkat yang ada. Konsep penempatan Snort IDS dalam jaringan adalah sebagai berikut:

a. Snort Network Intrusion Detection System.

Sesuai namanya. Snort NIDS tidak hanya melakukan monitoring paket data pada perangkat jaringan rnesin Snort, namun juga terhadap seluruh trafik jaringan di dalam segmen jaringan dirnana Snort sensor berada. Skema jaringan Snort NIDS dijelaskan pada gambar 1



Gambar 1 Skema jaringan *Snort* NIDS

Pada Gambar 1 dapat dilihat hahwa jaringan menggunakan 3 unit NIDS. Pada implementasi nyata, ketiga unit Snort sensor tersebut dapat berupa 3 unit komputer yang menjalankan aplikasi Snort IDS ataupun 1 unit komputer yang menjalankan aplikasi Snort dengan 3 buah perangkat kartu jaringan (NIC). Ketiga sensor tersebut diletakkan pada sehmen

jaringan yang strategis dan mengawasi trafik data pada setiap perangkat jaringan dalam segmen tersebut.

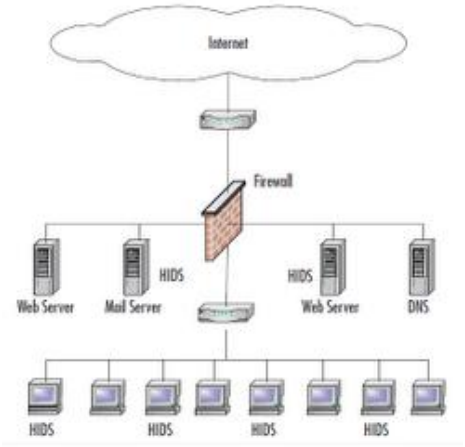
Keuntungan dari sistem ini adalah cakupan jaringan yang diawasi cukup luas, sehingga pengamanan sistem IDS mempunyai jangkauan yang lebih luas. Sedangkan kelemahannya adalah, komputer yang digunakan sebagai sensor IDS akan melakukan pemrosesan trafik data dengan jumlah yang sangat banyak, sehingga membutuhkan sumber daya yang besar.

Pertimbangan lain dari konsep sistem Network IDS ini yaitu kompleksitas manajemen rule yang digunakan oleh Snort memerlukan pertimbangan lebih lanjut dibandingkan IDS berbasis HIDS.

b. Snort Host-Based Intrusion Detection System.

Host Based IDS bersifat kebalikan dari NIDS. Sistem HIDS ini hanya melakukan pengamatan trafik dala pada perangkat jaringan di mana aplikasi Snort IDS ditempatkan. Implementasi nyata dari konsep sistem ini. Snort dijalankan pada mesin dengan aplikasi layanan lain seperti web server, *ITP server* maupun *SQL server*.

Keuntungan dari sistem IDS ini adalah pengelolaan rule yang lebih simpel. Misalnya Snort diimplementasikan pada web server dalam jaringan, maka rule yang diperlukan adalah rule unluk mendeteksi adanya eksploitasi terhadap web server tanpa perlu menggunakan rule untuk deteksi serangan terhadap aplikasi layanan yang tidak dijalankan oleh mesin tersebut. Konsekuensinya, maka rule yang digunakan lebih sedikit, sehingga dapat meningkatkan performa mesin komputer dan mengurungi beban pada processor karena data yang harus diperiksa oleh sensor IDS lebih sedikit. Skema jaringan sistem Snort HIDS digambarkan pada Gambar 2.



Gambar 2 Skerna Jaringan Sistem *Snort* HIDS

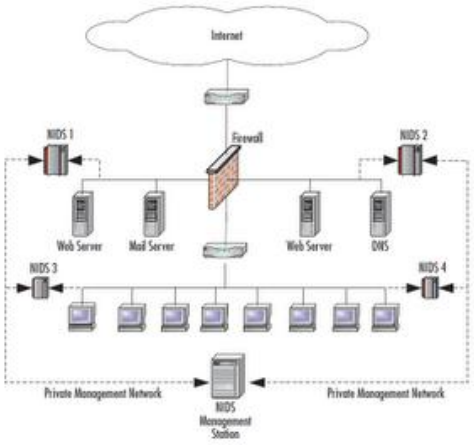
Pada Gambar 2 dicontohkan, Snort diimplementasikan pada mesin server dan juga pada host dalam jaringan internal. Masing-masing sensor IDS akan melakukan pengawasan terhadap paket data pada perangkai jaringan masing-masing mesin.

Seperti telah dinyalakan sebelumnya, rule yang digunakan oleh Snort pada mesin web server menggunakan rule yang dikhususkan untuk mendeteksi eksploitasi web server tanpa perlu menggunakan rule untuk deteksi terhadap usaha eksploitasi layanan lain yang tidak di jalankan oleh mesin tersebut.

c. Snort Distributed Intrusion Detection System.

Konsep sistem ini merupakan perpaduan dari konsep Network IDS dan Host-Based IDS. Masing-masing sensor yang diletakkan pada beberapa bagian dalam segmen jaringan

dikendalikan oleh sebuah mesin yang digunakan sebagai pusat manajemen sistem keamanan terpusat. Pusat menajemen sistem IDS akan menentukan rule yang digunakan oleh setiap sensor IDS dan mengumpulkan hasil peringatan dari setiap sensor dalam jaringan dalam sistem database terpusat. Skema sistem DIDS ini digambarkan pada gambar 3.

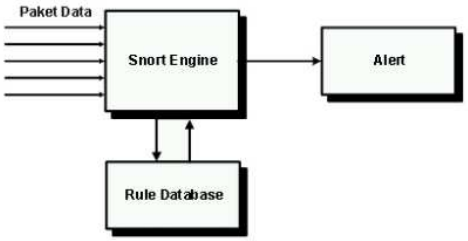


Gambar 3 Skerna jaringan sistem *Snort* DIDS

Pada gambar 3 diperlihatkan bahwa dalam jaringan terdapat 4 buah unit sensor dan sebuah pusat sistem manajemen IDS. Sensor NIDS I dan NIDS 2 melakukan pengamatan trafik data pada segmen jaringan server layanan publik. Sedangkan 2 sensor NIDS lain berada pada segmen jaringan internal dengan tingkat keamanan yang lebih ketat. Masing-masing sensor juga dapat berupa NIDS ataupun HIDS. disesuaikan berdasarkan kebutuhan.

Jalur yang digunakan dalam transaksi data oleh masing-masing sensor IDS dengan pusat manajemen sistem dapat berupa private network ataupun menggunakan infrastruktur jaringan yang telah ada. Apabila menggunakan infrastruktur jaringan yang ada, sangat direkomendasikan untuk mengaplikasikan konsep teori keamanan seperti enkripsi data maupun teknologi VPN sebagai sistem keamanan tambahan.

Sebagai kelengkapan sebuah sistem perangkat lunak. Snort IDS mempunyai 3 komponen utama vang saling berhubungan satu sama lain. Komponen utama dari Snort tersebut adalah snort engine, rule snort dan alert. Snort engine merupakan pemroses yang akan melakukan pembandingan paket data dengan rule Snort untuk dapat menghasilkan output berupa alert. Ketiga komponen tersebut bekerja dalam satu siklus yang saling berpengaruh, seperti ditunjukkan pada Gambar 4 tentang komponen *Snort* tersebut adalah sebagai berikut:



Gambar 4 Komponen *Snort*

Penjelasan lebih lanjut tentang komponen Snort tersebut adalah sebagai berikut.

a. Snort engine.

Snort engine adalah program yang berjalan sebagai daemon process yang bekerja untuk membaea paket dan membandingkannya dengan rule Snort. Pada sistem UNIX, untuk melihat apakah snort engine dalam keadaan

aktif atau tidak dapat dilakukan dengan perintah sebagai berikut:

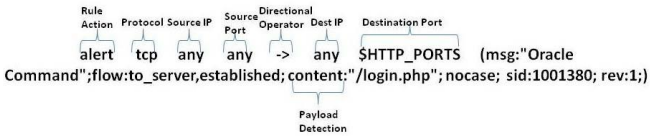
ps aux | grep snort

Apabila Snort engine aktif, maka akan ditunjukkan ID proses pada mesin tersebut.

b. Rule Snort.

Rule Snort merupakan database yang berisi pola-pola serangan yang berupa signature jenis serangan, rule snort ini harus secara rutin di-update. sehingga ketika terjadi pola serangan baru, Snort dapat mendeteksi pola tersebut sebagai sebuah serangan. Penulisan rule Snort mempunyai aturan yaitu rule harus ditulis dalam satu baris (*single line*). Rule Snort dibagi menjadi 2 bagian, *rule header* dan *rule options. Rule header* berisi *rule action. protocol*, *source* dan *destination port*. Sementara itu, rule oplion berisi pesan peringatan dan informasi dimana seharusnya paket tersebut diletakkan.

Contoh dari *rule Snort* tersebut seperti ditunjukkan pada Gamhar 5. dengan penjelasan seeara lebih rinci.



Gambar 5 Contoh dari *rule Snort*

1) Rule Header.

Bagian-bagian dari rule header menunjukkan pengertian sebagai berikut:

a. Rule Actions.

Rule Actions akan memberitahukan kepada Snort apa yang harus dilakukan ketika menemukan paket yang sesuai dengan signuture yang ada pada rule Snort. Aksi yang dapat dilakukan adalah *alert*. *log* dan *pass*.

b. Protocols.

Isi dari rule header selanjutnya adalah jenis protokol yang digunakan, misalnya tep. udp dan icmp.

c. IP Address.

Penulisan nomor IP pada rute Snort ditulis lengkap beserta netmask yang digunakan atau dapat juga menggunakan kata 'any' untuk banyak alamat IP.

d. Port Number.

Port number dapat dituliskan dengan angka yang menunjukkan jenis protokol ataupun dengan kata 'any' untuk semua port yang tersedia ataupun dengan range port seperti ‘1:1023'.

e. Directions Operator.

Tanda '->' merupakan orientasi atau direction. sebelah kiri adalah source host dan sebelah kanan adalah destination host. Orientasi juga dapat bersilat bidirectional dengan tanda '< >'.

2) Rule Options.

Rute Options dapat dipisahkan dengan tanda ‘:’ pada saat penulisan. Jenis options yang digunakan pada Snort terdapat 15 kata kunci, yaitu:

a. *msg*: menuliskan pesan dari alert dan paket log
b. logto: memasukkan log ke standart output file.
c. minfrag.
d. ttl: nilai time to life pada IP beader.
e. dsize: ukuran nilai data vang ditangkap.
f. content: pola dari data yang dianalisa.
g. offset.

h. *depth.*

i. *flags: nilai TCP flags.*

j. *seq: nilai TCP sequence.*

k. *ack: nilai TCP acknowledgcment*

l. *itype: nilai ICMP type.*

m. *icode: nilai ICMP codc.*

n. *session.*

3) *Alert.*

Alert merupakan catatan atau peringatan serangan pada pendeteksian Snort. Apabila Snort engine menyatakan sebuah paket data sebagai serangan, maka Snort engine akan mengirimkan alert berupa log file. Untuk kebutuhan analisa, alert dapat disimpan di dalam sebuah database.

Damn Vulnerable Web Application

Damn Vulnerable Web Application (DVWA) adalah sebuah aplikasi web berbasis *PHP* dan *MySQL* yang sangat rentan terhadap sebuah serangan keamanan. Tujuan utama dari dibuatnya aplikasi ini adalah sebagai sarana bagi praktisi keamanan untuk menguji kemampuan yang bersifat legal. DVWA juga ditujukan untuk membantu pengembang web dalam

memahami lebih dalam tentang proses pengamanan web serta dapat digunakan oleh kalangan akademik untuk mempelajari sistem keamanan berbasis aplikasi wcb.

Kode sumber aplikasi ini dapat diunduh dari situs resmi *DVWA* yaitu www.dvwa.co.uk secara gratis. Versi terakhir aplikasi ini yang dirilis adalah versi 1.0.7. Aplikasi ini didistribusikan dengan lisensi GPU *General Public I.icense*. sehingga dapat didistribusikan secara bebas, tanpa jaminan dari pihak pengembang.

Penggunaan aplikasi ini sangat disarankan hanya digunakan pada web server lokal, karena akan sangat berbahaya jika digunakan pada web server publik. Pemanfaatan aplikasi web *DVWA* ini dalam penelitian adalah sebagai sarana pengujian, terutama pada pengujian *SQL Injection*.

Gambar 6 Contoh tampilan DVWA

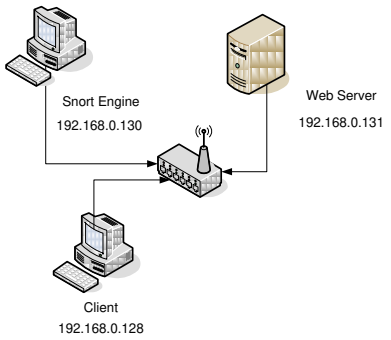
ANALISIS DAN PEMODELAN

Pemodelan

Pemodelan yang digunakan pada penelitian ini adalah model jaringan *Client/Server*. Pada model jaringan *Client/Server* diperlukan satu atau lebih komputer khusus yang disebut *server* untuk mengatur lalu lintas data informasi dalan jaringan computer. Komputer selain *server* disebut sebagai *client*. *Server* biasanya bersifat pasif, hanya menunggu berbagai permitaan dari client untuk kemudian melayani permintaan tersebut. Client biasanya bersifat aktf dan mengirim permintaan ke server serta menerima layanan dari server. (Arief Ramadhan, 2006).

Jaringan yang digunakan dalan penelitian ini adalah jaringan *intranet*. Yang

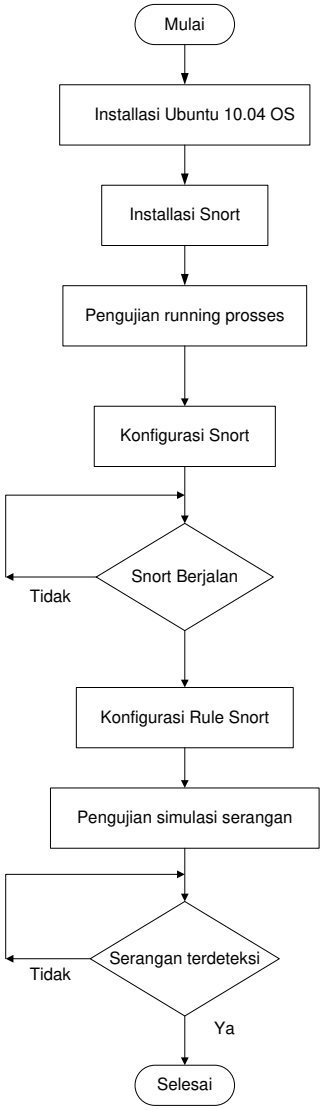
diimplementasikan dengan konsep *Snort Network – Based Intrusion Detection System*. Dengan skema seperti ditujukan pada gambar 7.



Gambar 7 Skema Jaringan Penelitian.

Karena keterbatasan peralatan, maka penelitian hanya melakukan pengamatan terhadap *web server*, monitoring dilakukan pada komputer yang digunakan sebagai *Snort sensor*. Komputer server dan komputer client terhubung melalui *switch/hub* dalam subnet yang sama.

Langkah perancangan, konfigurasi server dan penujian *Snort IDS* merrupakan proses yang lebih kompleks dibandingkan langkah lainnya. Sehingga langkah – langkah tersebut dapat digambarkan dalam diagram alir pada gambar 8.



Gambar 8. Diagram Alir Perancangan dan Pengujian.

Perancangan

Berdasarkan diagram alir pada gambar 3.2. dapat diuraikan langkah – langkah yang dilakukan selama perancangan, konfigurasi dan pengujian. Penjabaran dari langkah – langkah tersebut adalah :

1. Installasi Ubuntu 10.04 OS.

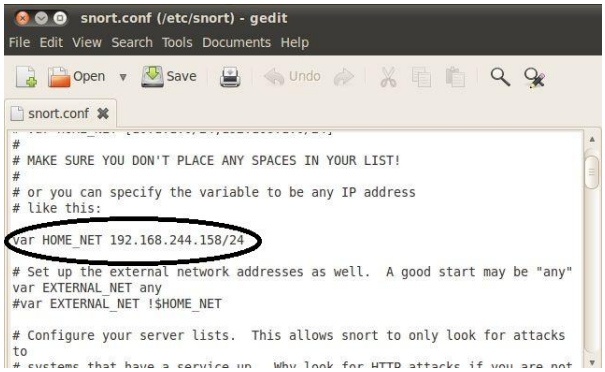
Ubuntu pada umumnya dapat diinstall dari CD atau dapat diinstall melalui USB. Sistem operasi Ubuntu dapat dijalankan dari Live CD, pengguna juga dapat juga

menginstall Ubuntu pada komputer secara permanen.

2. Installasi dan Konfigurasi *Snort*.

Langkah installasi Snort dapat dilakukan dapat dengan package apt, synaptic atau mengunduh program secara langsung pada situs resmi *Snort* (www.snort.org). pada penelitian ini menggunakan *Snort* versi 2.8.6.2 dan rule *Snort* dengan versi yang sama.

File konfigurasi *Snort* terdapat pada file *snort.conf* yang terletak pada direktori ***/etc/snort/snort.conf***. Pengaturan utama yang diperlukan adalah konfigurasi pada jaringan. Pengaturan jaringan pada *Snort* seperti ditunjukkan pada gambar 9.



Gambar 9. Pengaturan network *Snort*.

3. Konfigurasi Rule *Snort.*

Rule *Snort* didapatkan dengan mengunduh dari web resmi *Snort* dengan mendaftar akun sebelum mengunduh file rule tersebut. File – file hasil ekstraksi dari rule *Snort* yang telah diunduh kemudian dapat dipasang dengan cara dipindahkan pada direktori ***/etc/snort/rules***.

4. Pengujian Snort IDS.

Jenis serangan yang sering terjadi sebagai bagian untuk melakukan intrusi terhadap suatu system dapat berupa *nmap port scan,* eksploitasi, *SQL Injection* dan pengaksesan database. Oleh karena itu, langkah pengujian system *Snort* IDS dalam penelitian ini dilakukan melalui 5 macam pengujian.

PENGUJIAN DAN HASIL

a. Data Uji coba.

- **Uji coba 1: ping.**

Ping adalah sebuah program yang digunakan untuk memeriksa indukivitas jaringan berbasis teknologi (TCP/IP). Dengan menggunakan program ini, dapat diuji apakah sebuah komputer terhubung dengan komputer lainnya. Hal ini dilakukan dengan mengirimkan sebuah paket kepada alamat ip yang hendak diujicoba konektivitasnya dan menunggu respon darinya.

Client akan melakukan ping pada komputer *server* dengan memasukkan perintah **ping 192.168.0.130**.

- **Uji coba 2: nmap port scan.**

Pengujian 2 dilakukan dengan melakukan nmap port scan dengan menggunakan aplikasi nmap. Informasi yang diinginkan adalah port – port yang terbuka pada *server*, sistem operasi, dan versi sistem operasi dari *server.*

Proses *port scan* dilakukan dengan teknik nmaping, dimana server hanya akan memberikan *reply* terhadap paket data yang dikirimkan *client.*

Pihak penyerang dalam simulasi ini adalah komputer client dengan *IP* Address 192.168.0.128 dengan target *web server* dengan *IP address* 192.168.0.131.

Perintah yang dimasukkan pada nmap dalam pengujian ini adalah **nmap 192.168.0.131.**

- **Uji coba 3: eksploitasi.**

Eksploitasi adalah bagian dari perangkat lunak, sebagian data, atau urutan perintah yang mengambil keuntungan dari kesalahan, bug atau kerentanan untuk menyebabkan tidak diinginkan atau perilaku tak terduga terjadi pada perangkat lunak komputer, perangkat keras, atau sesuatu yang elektronik (biasanya komputerisasi). Hal ini sering mencakup beberapa hal seperti mendapatkan kontrol dari sistem komputer.

Program yang digunakan pada pengujian ini adalah **msfconsole** yang dijalankan pada komputer *client*.

- **Uji coba 4: SQL Injection.**

Teknik *SQL Injection* saat ini sering dilakukan dalam serangan keamanan untuk mendapatkan akses *root,* terutama pada web berbasis *php-MySQL.* Teknik ini dilakukan dengan memberikan kode – kode khusus dalam bahasa *MySQL* terhadap masukan yang dimana oleh sebuak halaman web, agar server memberikan informasi yang seharusnya.

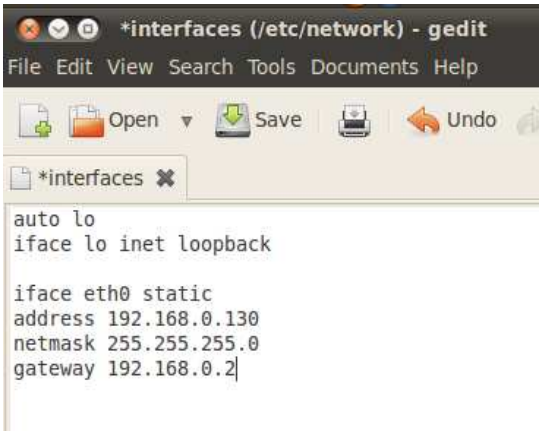
Pengujian simulasi ini dilakukan terhadap halaman web DVWA yang telah terinstall pada server. Pada komputer client digunakan aplikasi *sqlmap* untuk melakukan teknik *SQL-Injection*.

- **Uji coba 5: Pengaksesan database.**

Pengujian ini dilakukan sebagai simulasi serangan keamanan terutama layanan *MySQL* pada port 3306 untuk menguji kemampuan *Snort* IDS dalam mendeteksi serangan dengan pola yang sama. Apabila Snort mendeteksi paket data pada nomor port tersebut sesuai dengan rule *Snort* yang ada.

b. Konfigurasi peralatan (Hardware dan Software).

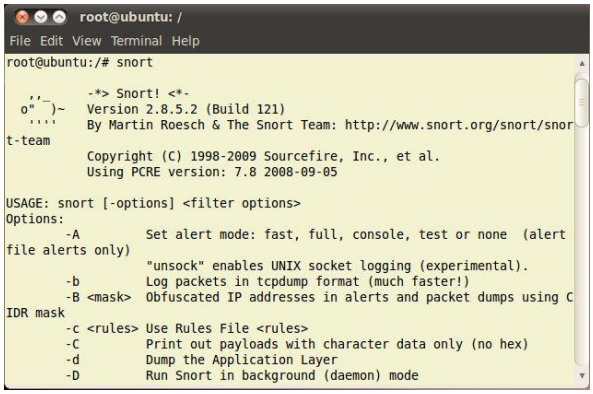
Konfigurasi utama pada IDS yang digunakan terutama pada konfigurasi IP Address pada kartu jaringan sesuai dengan skema jaringan yang telah ada sebelumnya. Untuk mengkonfigurasi kartu jaringan, yang harus dimodifikasi adalah sebuah file, yaitu ***/etc/network/interfaces***. Adapun konfigurasi kartu jaringan pada IDS adalah ditunjukkan pada gambar 10 berikut.



Gambar 10. Konfigurasi kartu jaringan pada IDS.

Testing / Implementasi Rancangan Interface.

Untuk menjalankan *Snort* pada terminal atau CLI pada Ubuntu, Kita tinggal memanggil aplikasi *Snort*, sama seperti menjalan aplikasi lainnya di *Linux*. *Interface Snort* di tunjukkan pada gambar 11.



Gambar 11. Contoh tampilan Interface *Snort* IDS pada CLI.

Analisis Hasil.

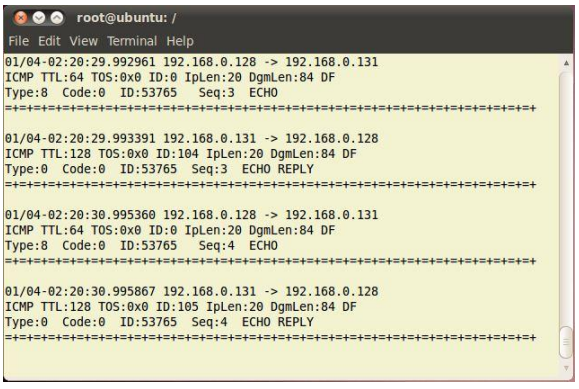
a. Pengujian ping.

Ping bekerja dengan mengirimkan sebuah paket data yang disebut dengan *Internet Control Message Protocol* (ICMP) *Echo Request*. Paket ICMP ini biasanya digunakan untuk mengirimkan informasi tentang kondisi jaringan antara dua host (komputer).

Mekanisme kerjanya yaitu ketika melakukan ping terhadap situs target (objek) maka akan tampil pada layar hasil respon berupa informasi nomor *IP* dari mana ping memperoleh *Echo* Reply, waktu (dalam milisekon) yang diperlukan program ping mendapatkan balasan dan yang terakhir adalah *Time To Live* (TTL). Setelah proses ping berhenti maka akan ditampilkan *summary* dari dari keseluruhan paket data yang telah dikimkan. Pada *summary* tersebut menampilkan beberapa informasi yaitu perhitungan paket data yang telah dikirim, telah diterima oleh target dan perhitungan data yang hilang ditengah jalan. Selain itu juga menampilkan waktu respon minimum, maksimum, dan rata-rata.

Proses ping hanya akan mendapat respon setiap 1 detik. Informasi waktu yang diberikan oleh ping adalah waktu perjalanan pulang pergi ke remote host yang diperlukan oleh satu paket. Satuan yang dipakai adalah mili detik, semakin kecil angka yang dihasilkan, berarti semakin baik cepat koneksinya. Setiap paket data yang dikirimkan melalui jaringan memiliki informasi yang disebut TTL, biasanya TTL ini diisi dengan angka yang relatif tinggi, (paket ping pada MS-DOS memiliki TTL 64). Dengan informasi ini dapat diketahui kira-kira berapa *router* yang dilewati oleh paket tersebut, dalam hal ini 64 dikurangi dengan N, dimana N adalah TTL pada *Echo Reply*. Selain itu dapat diketahui juga bila terjadi data yang hilang pada urutan router yang dilalui.

Hasil pengujian ping dengan perintah **ping 192.168.0.131** dari komputer client dengan IP Address 192.168.0.128 menghasilkan hasil seperti ditunjukkan pada gambar 12.

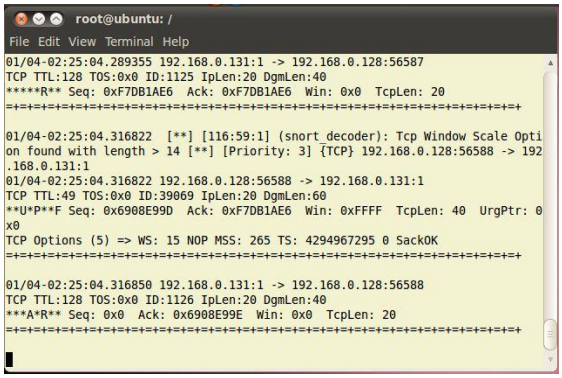


Gambar 12. Hasil pengujian ping pada CLI IDS.

b. Pengujian nmap port scan.

Hasil pengujian port scan dengan perintah **nmap –sX –o –v 192.168.0.131** dari

client dengan IP Address 192.168.0.131 menghasilkan hasil seperti di tunjukkan pada gambar 13.



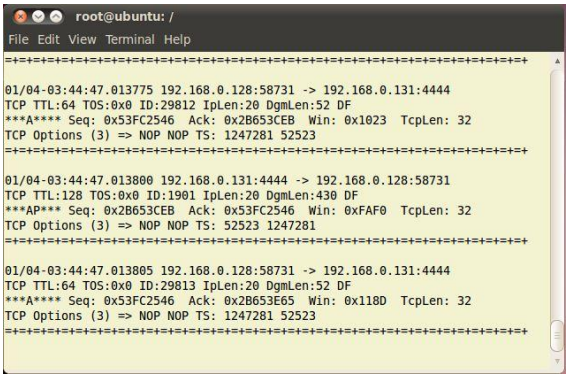
Gambar 13. Hasil pengujian port scan pada CLI IDS.

Pengertian dari perintah scan pada pengujian ini adalah memeritahkan nmap untuk melakukan *port scan* dengan opsi **–sX** untuk mode XMAS scan. **–O** untuk deteksi sistem operasi target, **-v** untuk menampilkan versi dari system operasi target dengan *IP Address* target adalah **192.168.0.131.**

Hasil peringatan Snort pada gambar 9. menunjukkan bahwa decoder Snort mendeteksi serangan nmap XMAS *scan* terhadap *server* oleh *client* dengan *IP Address* 192.168.0.131. Simulasi serangan ini dapat terdeteksi *Snort* karena paket data yang dikirimkan *client* ke *server* memenuhi kriteria *rule*.

c. Pengujian eksploitasi.

Hasil deteksi *Snort* terhadap pengujian eksploitasi yang dilakukan menghasilkan peringatan pada Snort yang ditunjukkan pada gambar 14.

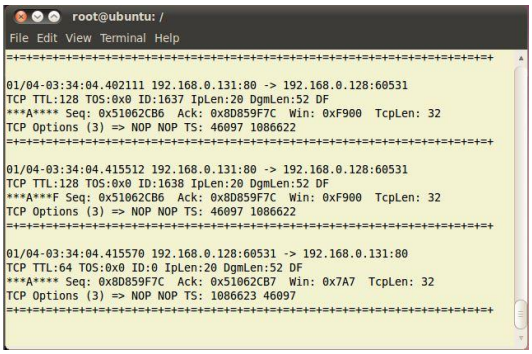


Gambar 14. Hasil pengujian eksploitasi pada CLI IDS.

Didalam sistem operasi *Windows XP*, yang digunakan sebagai *web server* terdapat bug yang dinamakan Dcom bug. Bug ini terdapat pada port 135, 137, 139 dan 445. Itulah mengapa ketika kita mengeksploitasi *Windows XP* kita menggunakan *port* tersebut.

Pengujian SQL Injection.

Hasil deteksi Snort terhadap pengujian *SQL Injection* yang dilakukan menghasilkan peringatan dalam CLI yang ditunjukkan pada gambar 15.



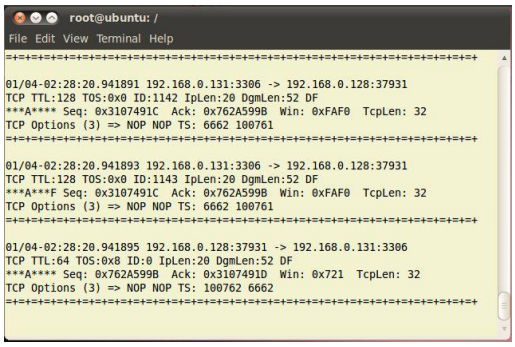
Gambar 15. Hasil pengujian *SQL Injection* pada CLI IDS.

Peringatan *Snort* IDS sebagaimana ditunjukkan pada gambar 12 menunjukkan bahwa telah terjadi serangan SQL Injection melalui protocol TCP pada layanan http *port*.

Peringatan *Snort* ini mengacu pada rule sebagaimana ditunjukkan pada gambar 12 yang membandingkan paket data ke *server* pada *port – port* layanan http. Pada pengujian ini, client dengan IP address 192.168.0.128 mengakses *web server* melalui port 80 dan memberi input terhadap halaman web DVWA pada form input *SQL Injection*.

d. Pengujian Pengaksesan Database.

Hasil pengujian pengaksesan *database server* dengan melakukan *login MySQL. Root* dan perintah untuk menampilkan database dari cliemt dengan hasil tampilan informasi pada gambar 16.



Gambar 16. Hasil pengujian pengaksesan database pada CLI IDS.

Hasil peringatan *Snort* pada gambar 16 menunjukkan bahwa terjadi serangan pengasksesan database melalui *protocol TCP* pada layanan MySQL. Penyerang melakukan *login* sebagai *MySQL root* dan melakukan pengaksesan terhadap *database* dengan menampilkan seluruh database yang ada. Serangan ini terdeteksi oleh mesin *Snort*.

Kesimpulan

Kesimpulan yang dapat diambil berdasarkan hasil penelitian ini adalah sebagai berikut :

1. Berdasarkan hasil pengujian yang dilakukan, *Snort* dapat diimplementasikan sebagai *Intrusion Detection System* pada sistem operasi Ubuntu 10.04 Linux untuk mendeteksi serangan berupa *ping*, *nmap port scan*, eksploitasi, *SQL Injection*, pengaksesan *database*.
2. *Snort* dapat memberikan peringatan adanya sebuah serangan keamanan, sehingga dapat meningkatkan keamanan jaringan. Dapat atau tidaknya sebuah serangan terdeteksi oleh Snort IDS tergantung dari ada tidaknya rule dengan jenis *signature* pada sebuah pola serangan.
3. Kemudahan dalam langkah *installasi Snort* IDS pada sistem operasi Ubuntu Linux didukung adanya sistem *aptitude* pada system operasi linux.

Saran.

Sistem keamanan dengan IDS Snort dapat memberikan manfaat lebih apabila Snort diintegraskan dengan *firewall*. Snort cukup efektif untuk mendeteksi adanya sebuah serangan terhadap system, namun Snort bukanlah sebuah *Intrusion Prevention Sistem* (IPS) yang dapat mencegah atau memblokir usaha – usaha penyusupan kedalam sistem.

DAFTAR PUSTAKA

Beale, Jay. 2003. ***"Snort 2.0 Intrusion Detection"***, Masachusset : Syngress Publishing, Inc.

Fauziah, Lilis. 2009. ***"Pendeteksian Serangan Pada Jaringan Komputer Berbasis***

IDS Snort Dengan Algoritma Clustering K-Means", Surabaya : Institut Teknologi Sepuluh November, Surabaya.

Rafiudin, Rahmat. 2010. ***"Mengganyang Hacker dengan SNORT"***. Surabaya : ANDI OFFSET

Snort Teams. Desember 7, 2011. ***"Snort User Manual 2.9.2".*** Columbia: Sourcefire, Inc.

The Ubuntu Manual Team. Juli 30, 2012. ***"Getting Started with Ubuntu 12.04"*** California: Creative Commons.

Wagoner, Richard. 2007. ***"Performance Testing An Inline Network Intrusion Detection System Snort"***. Master Thesis, Morehead State University.