Perancangan Sistem Pencegahan *Flooding* Data Pada Jaringan Komputer

Alva S. M. Tumigolung(1) Arie S. M. Lumenta(2) Arthur M. Rumagit(3)
(1)Mahasiswa, (2)Pembimbing 1, (3)Pembimbing 2
Email: Alfa4jc@gmail.com
Jurusan Teknik Elektro-FT UNSRAT, Manado-95115

Abstrak - Suatu serangan ke dalam Server jaringan komputer dapat terjadi kapan saja. Baik pada saat administrator sedang kerja atupun tidak. Dengan demikian dibutuhkan sistem pertahanan didalam server itu sendiri yang bisa menganalisa langsung apakah setiap paket yang masuk tersebut adalah data yang diharapkan ataupun data yang tidak diharapkan. Kalau paket tersebut merupakan data yang tidak diharapkan, diusahakan agar komputer bisa mengambil tindakan yaitu dengan medeteksi IP dan memblokir IP target.

Pemodelan suatu sistem yang digunakan untuk mengatasi *flooding* data pada suatu jaringan. Sistem didesain dengan jalan membuat suatu sistem penditeksi yang aktif yang bisa mendifinisikan setiap data yang masuk kedalam *server*, apakah data yang datang itu merupakan sebuah data *flood* atau data yang diperlukan oleh *user*. Pemodelan dibuat dengan menggunakan bahasa pemrograman *Snort*, dan IDS (*Istrusion Detection System*) dan dalam lingkungan jaringan komputer berbasis *ip address*.

Kata Kunci : BASE (Basic Analisis and Security System), Flooding Data, IDS (ISTRUSION DETECTION SYSTEM), Snort.

Abstract - An attack into Server of Computer network earn happened any time. Whether at the time of administrator is working or not. Thereby be required by a defender system in server of itself which the analysis can be direct whether every the incoming packet is expected data and or the data which is not expected. If the packet represent data which is not expected to be, afforded in order to the computer can bring an action against that is by detection and blockit the source of IP address of the origin packet.

Modelling of an used system to overcome flooding data at one particular network. Design of the system by way of making an active detection which can check every incoming data into server, whether that incoming data represent a data of flood or data that is needed by user.Modelling made by using Ianguage of Snort, and IDS (ISTRUSION DETECTION SYSTEM in environment of computer network base on ip address.

Keyword : BASE (Basic Analisis and Security System), Flooding Data, IDS (ISTRUSION DETECTION SYSTEM), Snort.

I. PENDAHULUAN

Sudah banyaknya perusahaan yang menggunakan internet sebagai sarana untuk membantu dalam melaksanakan aktifitas rutin perusahaan dan aktifitas rutin lainya.Dalam hal ini tidak hanya perusahaan yang bergerak di bidang telekomunikasi saja yang menggunakan internet,tetapi juga perusahaan lain yang tidak bergerak di bidang tersebut. Kecenderung pengguna internet ini di sebabkan oleh dengan adanya internet akan didapatkan dengan kemudahan dalam hal komunikasi dan *transfer* data. Kenyataan ini kita bisa lihat pada bidang perbankan *system* komunikasi data sangat berguna membantu perusahaan tersebut untuk melayani parah nasabahnya, juga dalam bidang marketing suatu barang hasil industri suatu perusahaan. Kemudahan dan kepraktisan merupakan kunci dari mengapa dipilihnya internet ini.

Tetapi di samping keuntungan yang banyak,internet juga banyak menyimpan banyak kekurangan yang sangat mengkhawatirkan bagi para penggunanya. Salah satu yang sangat menjadi kendala adalah dalam bidang keamanan. Banyak kasus yang menbuktikan bahwa perusahaan yang tersambung dengan internet sering kali mendatatkan gangguan baik dalam data yang dimiliki maupun peralatannya. Kerugian yang di derita akan hal ini biasa dibilang tidak kecil. Kasus pencurian atau manipulasi data perusahaan saja dapat mencapai kerugian sampai miliaran Rupiah bahkan sampai jutaan dollar US. Belum lagi kerusakan peralatan yang di gunakan oleh perusahan tersebut, yang bisa dibilang tidak murah.

Dalam faktor keamanan ini biasanya perusahaan mendapatkan administrator untuk bekerja. Tetapi fungsi administrator tentunya akan terbatas waktunya,pada saat jam kerja. Meskipun di jam kerja pun kadang kala karena terlalu banyaknya aliran data yang diterima oleh *server* adalah data yang di harapakan atau data yang tidak diharapkan. Sedangkan suatu serangan ke *system* keaamanan yang bisa terjadi kapan saja. Baik pada saat administrator sedang bekerja ataupun tengga malam dimana tidak ada yang menjaga *server* tersebut. Dengan demikian di butuhkan system pertahanan didalam *server* itu sendiri yang bisa menganalisa langusng apakah setiap paket yang masuk tersebut adalah data yang diharapkan atupun data yang tidak diharapakan. Kalau paket tersebut merupakan data yang tidak diharapkan,diusahakan agar komputer bisa mengambil tindakan untuk mengantisipasi agar serangan yang terjadi tidak menimbulkan kerugian yang besar. Akan lebih baik kalau *server* bisa mengantisipasi langsung,sehingga kerugian bisa mendekati nol atau tidak ada sama sekali.

II. LANDASAN TEORI

A. *Model Referensi OSI (Open System Interconnection)*

Untuk menyelanggarakan komunikasi berbagai macam *vendor computer* diperlukan sebuah aturan baku yang standard an disetujui sebgai fihak. Seperti halnya dua orang yang berlainan bangsa, maka untuk berkomunikasi

memerlukan penerjemah/interpreter atau satu bahasa yang dimengerti kedua bela fihak. Dalam dunia computer dan telekomunikasi interpreter identik dengan protocol. Untuk itu maka badan dunia yang menangani masalah standarisasi ISO (*Open System Interconnection*). Membuat aturan baku yang dikenal dengan nama model referensi OSI (*Open System Interconnection*). Dengan demikian diharapakan semua vendor perangkat telekomunikasi haruslah berpedoman dengan model referensi ini dalam mengembangkan protokolnya. Model referensi OSI (*Open System Interconnection*) terdiri dari 7 lapisan, mulai dari lapisan fisik sampai aplikasi. Model referensi tidak hanya berguna untuk produk-produk LAN (*Local Area Network*) saja tetapi dalam membangun jaringan internet sekalipun sangat diperlukan. Hubungan antara model referensi OSI (*Open System Interconnection*) dengan protokol internet pada tabel I dan tabel II.

B. Internet Protocol (IP)

IP (*internet protocol*) merupakan protokol yang paling penting yang harus berada pada *layer Internet TCP/IP*. Semua *protokol TCP/IP* yang berasal dari layer mengirmkan data melalui *protokol IP* ini. Seluruh data dilewatkan, dioalah oleh protokol *IP* dan dikirimkan sebagai datagram *IP* untuk sampai ke sisi penerima. Dalam melakukanpengiriman data, *protokol IP* ini bersifat *unrealible connectionless*, dan *datagram delivery service*.

Unrealible berarti *protokol IP* yang tidak menjamin datagram yang dikirim pasti sampai ke tujuan. *Protokol IP* hanya melakukan cara terbaik untuk menyampaikan datagram yang dikirim ke tujuan. Jika pada perjalanan datagram tersebut terjadi hal-hal tidak dinginkan (putusnya jalur, kemacetan, atau sisi penerima yang dituju sedang mati), *protokol IP* hanya akan memberikan pemberitahuan pada sisi kirim kalau telah terjadi pemasalahan pengiriman data ketujuan melalui *protokol* ICMP. Sedangkan *Connectionless* berarti tidak melakukan pertukaran kontrol informasi (*handshake*) untuk membentuk koneksi sebelum mengirimkan data.

Datagram delivery service berarti setiap datagram yang dikirim tidak tergantung pada datagram lainya. Akibatnya jalur yang ditempuh oleh masing-masing datagram *IP* ketujuan bisa berbeda satu sama lainya. Dengan demikian kedatangan datagram pun bisa jadi tidak berurutan. *Metode* ini dipakai untuk menjamin sampainya datagram ke tujuannya walaupun salah satu jalur menuju ke tujuan mengalami masalah.

Ada juga lapisan Internet yang bertangung jawab untuk mengirimkan data melalui jaringan. Protokol lapisan *Internet* yang utama adalah *Internet Protokol (IP)*. IP merupakan *protokol Internet* yang mempunyai fungsi sebagai berikut: Pengalamatan, Frakmentasi datagram pada antar jaringan, dan Pengiriman datagram antar jaringan.

Diantara fungsi tersebut yang paling berkepentingn dengan administrator jaringan adalah fungsi pengalamatan. *IP* mempunyai pola pengalamatan yang unik yang membutuhkan waktu untuk membiasakanya.

Format Alamat IP

Pada *terminology TCP / IP*, suatu jaringan terdiri dari sekelompok host yang dapat berkomunikasi secara langsung tanpa *router*. Semua host *TCP / IP* yang menempati jaringan yang sama harus diberi netid yang sama. *Host* yang mempunyai *netid* harus berkomunikasi melalui router.

Suatu *TCP / IP Internetwork* adalah sebuah jaringan dari beberapa jaringan, dan dapat menggabungkan banyak jaringan yang dihubungkan oleh *router*. Setiap jaringan pada *internetwork* harus diberi netid yang unik.

TABEL I MODEL OSI (*OPEN SYSTEM INTERCONNECTION*)

Lapisan	Fungsi Lapisan
Application (Aplikasi)	Lapisan yang menangani program aplikasi yang digunakan oleh user dalam mengirim/menerima data, misalnya program e-mail, Messenger, Browser, dsb
Presentation (Presentasi)	Lapisan ini melakukan presentasi data, perubahan format agar terjadi kesesuaian antara pengirim dan penerima
Session (Sessi)	Lapisan ini yang membuka koneksi antara dua komponen yang berkomunikasi, menjaga koneksi selama komunikasi berlangsung dan memutuskan-nya ketika selesai
Transport (Transport)	Lapisan ini yang menjamin pengiriman data dari satu komponen ke komponen lainnya yang berkomunikasi
Network (Jaringan)	Lapisan yang mengatur rute dari paket data melalui jaringan, sehingga paket ini bisa sampai ke tujuan
Data Link (Sambung Data)	Lapisan yang menjamin paket-paket data terbebas dari kesalahan ketika disampaikan ke penerima
Physical (Fisik)	Lapisan yang menangani medium fisik / koneksi listrik yang menghubungkan dua komponen yang berkomunikasi.

TABEL II LAPISAN TCP/IP

Lapisan	Fungsi Lapisan
Physical (Fisik)	Lapisan yang menangani antarmuka antara medium transmisi dengan peralatan. Karakteristik fisik, seperti medium, bentuk signal, kecepatan signal, ditentukan pada lapisan ini.
Network Access (Jaringan)	Lapisan ini menangani rute data dan akses antara dua komputer yang saling berkomunikasi dalam jaringan yang sama. Lapisan ini juga memeriksa alamat penerima data, menetapkan prioritas pengiriman.
Internet	Lapisan ini menangani rute data dan akses antara dua komputer yang berkomunikasi dalam jaringan yang berbeda. Lapisan ini menggunakan protokol Internet untuk memilih rute data dalam jaringan yang beragam.
Transport	Lapisan yang menjamin reliabilitas pengiriman paket-paket data, serta mengatur urutan paket tersebut. Protokol TCP digunakan pada lapisan ini.
Application (Aplikasi)	Lapisan ini menangani berbagai aplikasi yang akan menggunakan jaringan.

C. Macam-macam Protokol IP

Internet Control Message Protocol (ICMP)

ICMP (*Internet Control Message Protocol*) merupakan IP yang tidak didesain sebagai protokol yang handal *ICMP* hanya bertugas untuk mengirimkan pesan-pesan khusus atau pesan-pesan kesalahan yang memerlukan perhatian, dan tidak memerlukan keamanan yang tinggi karena pengirimannya dalam bentuk datagram *IP*. Pesan ICMP ini akan dikirm jika terjadi masalah *layer IP* dan *layer* di atasnya *(TCP atau UDP)*

Pesan *ICMP* tersebut ditentukan dari kombinasi tipe dan kodenya. Pesan kesalahan yang mungkin dikirim dengan *ICMP* diantaranya adalah:

Destination Unreachable. Pesan yang menyediakan informasi ketika host jaringan *port* atau protokol tertentu tidak dapat dijangkau.

Time Exceeded. Pesan ini memberitahu bahwa pengiriman datagram tidak dapat dikirim karena *Time To Live* sudah habis.

Parameter *Problem*. Pesan ini dikirim jika terdapat kesalahan parameter pada header datagram *IP*.

Source Quench. Pesan *ICMP* ini dikirim jika *router* atau tujuan mengalami kemacetan/kongesti proses dan sebagi *respon* balik atas pesan ini pada sistem pengirim paket data yang harus memperlambat pengiriman paket datanya.

Redirect. Pesan yang dikirim jika pada *router* merasa pengirim melewatkan data pada router yang salah, sehingga harusnya dikirim melalui lain.

Echo dan *Echo Reply*. Merupakan pesan yang menyediakan mekasisme pengujian keaktifan alamat pengiriman dan alamat tujuan.

Time Stamp dan *Time Stamp Reply*. Menyedikan mekasisme untuk mengetahui informasi waktu yang diperlukan sistem tujuan untuk memproses suatu paket data.

Addres Mask Requst dan *Addres Mask Reply*. Untuk mengetahui pengalamatan yang harus digunakn oleh *host/komputer* dalam suatu alamat jaringan.

IP tidak didesain dengan keandalan pengiriman data yang mutlak. Tujuan dari *ICMP* ini dalah untuk memberikan pesan balik terhadap permasalahan yang terjadi dalam jaringan komunikasi *IP*, bukan untuk membuat *protokol IP* menjadi andal (*reliable*) *[RFC 792]*. Pesan *ICMP* sendiri dikirim dalam bebrapa situasi, misal: jika sebuah datagram tidak dapat mencapai tujuannya, jika *gateway/router* tidak mampu meneruskan datagram dikarenakan penuhnya kapasitas buffer yang ada, dan jika tidak dapat menemukan alamat tujuan.

Transmission Control Protocol (TCP)

TCP (*Transmission Control Protocol*) merupakan protokol yang berada pada layer *transfer* dari layer *TCP/IP*. *TCP* juga merupakan protokol yang handal, protokol ini berusaha keras secara saksama untuk mengirimkan data ketujuan,memeriksa kesalahan,mengirim data ulang bila diperlukan dan mengirim *eror* kelapisan atas hanya bila *TCP* tidak berhasil mengadakan komunikasi. *TCP* didesain untuk memenuhi kebutuhan (*DOD*) akan pengiriman data akurat pada masa dimana jaringan *wide-area* masih belum begitu handal, dan tetap sesuai digunakan untuk aplikasi yang membutuhkan pengiriman data yang handal. Tetapi perlu dicatat bahwa keandalan *TCP* dapat tercapai dengan mengorbankan *bandwith* yang besar. *TCP (RFC 792)* memeberikan komunikasi yang handal antar proses yang berjalan pada host yang saling terhubung. Komunikasi *host-to-host* ini fungsinya *independen* terhadap sruktur jaringan yang dipakai. *TCP* tidak mengurusi proses *routing* data melalui *internetwork*, infrastruktur jaringan adalah tanggung jawab *IP*. Pada lapisan *host-to-host*, *TCP* pada host yang satu berkomunikasi langsung dengan *TCP* pada host yang lain, tidak peduli apakah kedua *host* ini berada pada satu jaringan atau jaringan mereka terpisah satu dengan yang lainnya. Pada kenyataannya *TCP* tidak terdapat pada *router* kecuali fungsi *router* tersebut dilakukan pada host yang menjalankan proses lapisan atas (*misal: windows NT* dapat melakukan *routing* pada komputer yang dihunakan *workstation).* Pada kenyataannya, *TCP* diabaikan oleh jaringan. Banyak jenis teknologi jaringan local maupun *wide-area*, termasuk *circuit switching* dan paket *switching*, *TCP* mengenali *host* menggunakan *IP address* dan tidak memperdulikan alamat fisik.

Karakteristik dan fungsi *TCP* adalah sebagai berikut : Penanganan aliran data dengan proses dan aplikasi lapisan atas, Tersedianya komunikasi yang handal, Penanganan hubunhgan yang baik, dan Tersedianya jaringan dan keamanan.

Format Header TCP

TCP memiliki format header untuk tiap segmen yang dikirimkan ke *IP*. *Header TCP* mengikuti *header IP datagram.*

Data yang diterima pada sisi penerima akan disusun berdasarkan nomor urut yang diberikan oleh sisi pengirim. Unruk mengatasi kerusakan data yang diterima, *TCP* menggunakan sebuah untuk memastikan bahwa data tersebut tidak rusak.

Model komunikasi dua arah antara komputer sisi kirim dan sisi terima sebelum terjadi proses pengiriman data disebut *handshake.* Tipe *handshake* yang digunakan *TCP* adalah karena menggunakan tiga segmen. Tujuan ini adalah untuk pembentukan koneksi, sinkronisasi segmen, dan pemberitahuan besar data yang bisa diterima pada suatu saat antara sisi kirim dan sisi terima.

Protokol TCP/IP

Transmission Control Protocol/Internet Protocol (TCP/IP) adalah standar komunikasi data yang digunakan oleh internet dalam proses tukar-menukar data dari komputer ke komputer lain di dalam jaringan internet. Protokol ini tidaklah dapat berdiri sendiri, karena memang protokol ini berupa kumpulan protokol (*protocol suite*). Protokol ini juga merupakan protokol yang paling banyak digunakan saat ini. Data tersebut diimplementasikan kedalam bentuk perangkat lunak (*software*) disistem operasi. Istilah yang diberikan perangkat lunak ini adalah *TCP/IP stack*.

Protokol *TCP/IP* dikembangkan pada akhir dekade 1970-an hingga awal 1980-an sebagai sebuah protokol standar untuk menghubungkan komputer-komputer dan jaringan untuk membentuk sebuah jaringan yang luas WAN (*Wide Area Network*). *TCP/IP* merupakan sebuah standar jaringan

terbuka yang bersifat independen terhadap mekasisme transport jaringan fisik yang digunakan, sehingga dapat digunakan dimana saja. Protokol ini menggunakan skema pengalamatan yang sederhana yang disebut sebagai alamat *IP* (*IP Address*) yang mengizinkan hingga beberapa ratus juta computer untuk dapat saling berhubungan satu sama lainnya di Internet. Protokol ini juga bersifat *routable* yang berarti protokol ini cocok untuk menghubungkan sistem-sistem berbeda (seperti *Microssoft* dan keluarga *UNIX)* untuk membentuk jaringan yang heterogen.

Protokol *TCP/IP* selalu berevolusi seiring dengan waktu, mengingat semakin banyaknya kebutuhan terhadap jaringan komputer Internet. Pengembangan ini dilakukan oleh beberapa badan, seperti halnya *Internet Socisty* (ISOC). *Internet Architecture Board (IAB),*dan *Internet Engineering Task Force (IETF).* Macam-macam protocol yang berjalan diatas TCP/IP, skema pengalamatan, dan konsep *TCP/IP* didefinisikan dalam dokumen yang disebut *Request For Comment (RFC)* yang dikeluarkan oleh *IETF.*

D. *Flooding Data*

Data

Data merupakan kumpulan huruf atau angka yang belum diolah sehingga tidak memiliki arti, atau bisa juga disebut sebagai catatan atas kumpulan fakta. Data merupakan bentuk jamak dari *datum*, berasal dari bahasa latin yang berarti "sesuatu yang diberikan". Dalam penggunaan sehari-hari data berarti suatu peryataan yang diterima secara apa adanya. Pernyataan ini adalah hasil pengukuran atau pengamatan suatu variable yang bentuknya dapat berupa angka, kata-kata citra.

Dalam fakta yang dikumpulkan untuk menjadi data. Data kemudian diolah sehingga dapat diaturkan secara jelas dan tepat sehingga dapat dimengerti oleh orng lain yang tidak langsung mengalaminya sendiri, hal ini dinamakan *deskripsi*. Pemillihan banyak data sesuai dengan persamaan atau perbedaan yang dikandungnya dinamakan klasifikasi.

Flooding

Pengiraman data yang berlebihan baik dari besar paket maupun jumlah paket kedalam suatu jaringan dan umumnya merupakan data yang tidak berguna disebut dengan *Flood Data,* adakalanya data-data yang berbedah dalam *traffic* merupakan data yang tidak perlu. Data-data tersebut memang sengaja dikirim oleh seseorang meneruskan jaringan data yang ada. Pengiriman data tersebut dapat mengakibatkan lambatnya jalur *traffic* yang ada dalam jaringan dan juga bisa mengakibatkan kerugian lain yang cukup berarti, misalnya kerusakan *program* karena adanya *intruder* yang masuk kedalam jaringan.

Traffic data yang ada dalam suatu jaringan akan mengalami turun naik selama pemakaiannya. Pada jam-jam sibuk *traffic* suatu data akan sangat padat, sehingga *traffic* data tersebut akan terganggu. Baik data yang akan dikirim maupun data yang akan datang akan mengalami antrian data yang mengakibatkan kelambatan dalam pengiriman dan penerimaan data.

E. *Firewall*

Firewall adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas yang dianggap aman untuk melaluinya dan mencegah lalu lintas jaringan yang tidak aman. Umumnya sebuah *firewall* diplementasikan dalam sebuah mesin terdedikasi yang berjalan pada pintu gerbang *(gateway)* antara jaringan *local* dan jaringan lainnya.

Firewall umumnya juga digunakan untuk mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan pribadi dari pihak luar. Saat ini istilah *firewall* menjadi istilah generika yang merujuk pada sistem yang mengatur komunikasi antar dua jaringan yang berbeda.

Fungsi Firewall

Mengontrol dan mengawasi paket data yang mengalir dijaringan *firewall* harus dapat mengatur, memfilter, dan mengontrol lalu lintas data yang diizinkan untuk mengakses jaringan *private* yang dilindungi *firewall. Firewall* harus dapat melakukan pemeriksaan terhadap paket data yang akan melewati jaringan privat. Beberapa kriteria yang dilakukan *firewall* apakah memperbolehkan paket data lewat atau tidak antara lain :

Manfaat dari *firewall* yaitu Mengatur *lalulintas/trafik* data antara jaringan, Dapat mengontrol *port* atau paket data yang diperbolehkan atau ditolak dan Memonitoring atau mencatat lalulintas jaringan.

Cara kerja firewall

Firewall pada dasarnya merupakan penghalang antara komputer anda atau jaringan dan internet. F*irewall* bisa hanya dibandingkan dengan seorang penjaga keamanan yang terdiri dari pintu masuk rumah anda dan menyaring pengunjung yang datang ke tempat anda. Dia mungkin mengizinkan pengujung masuk sementara menyangkal orang lain yang dia tersangka penyusup. Demikian pila *firewall* adalah sebuah *program* perangkat lunak atau perangkat keras yang menyaring informasi paket yang datang melalui internet ke komputer pribadi atau jaringan computer lainnya.

Firewall dapat memutuskan atau memblokir lalu lintas jaringan antara perangkat berdasarkan aturan yang dikonfirmasi atau ditentukan oleh *administrator firewall.*

Firewall memberikan keamanan disejumlah ancaman *online seperti login remote, backdoors, Trojan,* pembajakan sesi, serangan *DOS* dan *DDOS, virus* , dan banyak lagi. Efektifitas keamanan tergantung pada cara anda mengkonfirmasi *firewall* dan bagaimana anda mengatur aturan *filter.* Namun ancaman utama dari *DOS* dan serangan *DDOS* kadang-kadang dapat mengelolah untuk melewati *firewall* dan melakukan kerusakan *server.* Meskipun *firewall* bukanlah jawaban yang lengkap terhadap ancaman *online,* dapat paling efektif mengenai serangan dan memberikan keamanan komputer sampai batas maksimal.

F. IDS (Intrusion Detection System)

IDS (Intrusion Detection System) dapat didefinisikan sebagai tool,metode,sumber daya yang memberikan bantuan untuk melakukan identifikasi, memberikan laporan terhadap aktifitas jaringan komputer.

III. ANALISA DAN PERANCANGAN SISTEM

Analisis sistem didefinisikan sebagai pengurayan dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya, dengan maksud untuk mengidentifikasi dan mengevaluasi permaslahan, kesempatan, hambatan yang terjadi dan kebetuhan yang diharapkan sehingga dapat diusahakan.

Pada bab ini dibahas tentang bagaimana proses yang digunakan untuk merancang dan mensimulasikan sistem pengamanan *server* yang akan dibuat. Pembahasan yang akan dilakukan meliputi spesifikasi dan mekasisme kerja program yang diinginkan. Setelah itu berlanjut ke pembahasan mengenai alat bantu yang akan digunakan. Setelah masing-masing alat bantu itu dijabarkan, pembahasan menyentuh hal-hal lebih spesifik dari sistem, yaitu bagaimana data diambil dan bagaimana pengolahan data-data tersebut. Kemudian akan dijabarkan secara lebih jelas mengenai desain sistem secara keseluruhan. Proses implementasi dijelaskan melalui *algoritma* dari *program-program* yang akan dibuat.

Langkah-langkah yang dilalui dalam analisis sistem perancangan pencegahan *flooding* data yaitu dengan melakukan.

Untuk mengidentifikasi terjadinya *flooding* digunakan suatu konsep 4W+1H yaitu : *What, Why, Where, When* dan *How.* Dimana konsep ini digunakan untuk mengumpulkan suatu informasi agar dapat memperoleh data yang utuh kenapa terjadi *flooding* data pada suatu jaringan. Berikut merupakan penjelasan dari konsep 4W+1H:

What : Apa yang dimaksud dengan *Flooding Data* Pengiriman data yang berlebihan baik dari besar paket maupun jumlah paket ke dalam suatu jaringan dan umumnya merupakan data yang tidak berguna.

Why: Kenapa terjadi *Flooding Data*. Karena keterbatasan ketahan sistem dalam menangani *traffic* data yang besar

Where : Dimana terjadinya *flooding data* Pada sebuah sistem jaringan yang tidak dapat terkontrol ketika paket data masuk secara kontinu dan besar.

When: Kapan terjadinya *flooding data* Ketika sistem tidak bisa menditeksi paket data *flooding* dan bisa melakukan pengeblokan paket data yang dianggap *flooding* pada sebuah sistem jaringan.

How: Bagaimana pencegahan agar dapat tidak terjadinya *flooding data.* Yaitu merancang dan membangun suatu sistem yang mampu menditeksi terjadinya *flooding* data.

Langkah berikutnya agar dapat didefinisikan permasalahan yang ada sehingga masalah tersebut bisa dipecahkan, masalah yang terjadi kenapa terjadinya *flooding* pada jaringan adalah karena :

Human Error. Sesuatu yang telah dilakukan, yang tidak diharpakan oleh pelaku, tidak diinginkan oleh sesuatu aturan yang ditetapkan atau oleh pengamat luar atau yang membuat sistem tidak berjalan atau melampui batasnya. (*Sunder and Moray 1991).*

Kelemahan Hardware. Hardware tidak dapat bekerja berdasarkan perintah yang telah ditentukan atau disebut *untruction set* atau perintah yang dimengerti oleh *hardware* tersebut untuk melakukan berbagai kegiatan yang ditentukan oleh si pemberi perintah, karena apabila sistem *hardware* tidak berfungsi akan mengakibatkan tidak berfungsinya suatu sistem yang ssudah didesain sebelumnya.

Kelemahan Software. Salah satu kelemahan *software* aplikasi yang digunakan tidak berlaku di *flatfom linux.*

Kelemahan Sistem Jaringan . Ada banyak definisi sistem jaringan begituh lemah terhadap ancaman dari luar,diantaranya karena jaringan tidak di lengkapi dengan akses kontrol yang kuat dan pertahan seperti menggunakan *Firewall* dan sebagainya.

Berikutnya yakni Memahami kinerja sistem yang ada. Mempelajari secara terperinci bagaimana sistem yang ada beroperasi, menganalisa permasalahan, kelemahan dan kebutuhan pemakai sistem untuk dapat memberikan rekomendasi dan pemecahannya.

Baru kemudian menganalisa sistem. Dalam tahapan ini yang dilakukan adalah mentrasformasikan dua masukan utama diatas kedalam spesifikasi sistem yang diataur. Ada tiga kelompok utama penyebab terjadinya *flooding data pada jaringan* yang menyebabkan sistem terjadi hang bahkan bisa menjadi rusak. Diantaranya adalah: *File* yang dikirim secara *broadcast* mengandung *Virus*, Sistem keamanan jaringan lemah, dan Fasilitas jaringan *unavailable.*

A. Spesifikasi Sistem

Sebelum melakukan proses pembuatan sistem, terlebih dahulu ditentukan spesifikasi sistem. Spesifikasi sistem akan menjadi titik tolak sekaligus menjadi acuan untuk pembuatan sistem dan juga menentukan kapabilitas dan kemampuan apa saja yang harus bisa dipenuhi sistem yang dimaksud.

Sistem yang dibangun memiliki spesifikasi sebagi berikut: Sistem beroperasi pada platform *Linux Ubuntu 12.04*, Sistem yang digunakan harus bisa mengambil data-data dari jaringan, Semua data yang dikumpulkan disimpan dalam *database*, Resource yang digunakan harus seminimal mungkin, serta Sistem harus bersifat *multiuser* dan *multitasking*. Dikembangkan dengan alat Bantu yang mudah dan/atau gratis.

B. Sistem Operasi Dan Alat Bantu Yang Digunakan

Setelah spesifikasi dari sistem dan mekanisme kerja sistem telah dapat dijabarkan maka proses selanjutnya adalah memilih sistem operasi dan *tool* yang akan digunakan untuk membangun sistem tersebut. Alat Bantu tersebut merupakan paket aplikasi dan bahasa pemograman yang memiliki kemampuan sesuai dengan kebutuhan untuk membangun sistem ini.

Beberapa hal yang menjadi dasar pemikiran untuk memilih tool adalah: Alat bantu harus murah atau sedapatnya gratis, agar sesuai dengan tujuan semula yaitu membuat sistem yang semurah mungkin, Alat Bantu yang digunakan sedapat mungkin adalah alat Bantu yang biasa digunakan dalam platform *Linux Ubuntu* dan akan lebih baik lagi bila merupakan *tool default* pada *platform* itu, dan Alat Bantu harus memiliki dokumentasi yang lengkap dan juga harus bersifat *open source.*

Berikut ini akan membahas mengenai sistem operasi dan alat Bantu yang akan digunakan. Pembahasan meliputi deskripsi umum, kegunaan dan keunggulan sistem tersebut.

Linux Ubuntu 12.04

Linux awalanya adalah sistem operasi yang handal dipakai untuk jaringan. Akan tetapi, kini kegunanaanya semakin meluas untuk keperluan *desktop* baik diperkantoran maupun dirumah.

Switch

Switch merupakan perangkat jaringan yang bekerja pada *OSI Layer 2*, Data *Link Layer*. dia bekerja sebagai penyambung / *concentrato*r dalam Jaringan. *Switch* mengenal *MAC Adress* sehingga bisa memilah paket data mana yang akan diteruskan ke mana.

Snort

Snort merupakan suatu perangakat lunak untuk menditeksi penyusup dan mampu menganalisa paket yang melintas jaringan secara *real time traffic* dan *logging* ke dalam *database* serta mampu mendeteksi berbagai serangan yang berasal dari dalam jaringan maupun diluar jaringan. *www.Snort.org.*

Komputer server IDS menggunakan *OS Linux Ubuntu 12.04*
Prosesor Intel Prosesor core i5
Memori 2 GB
Hardisk 500 GB
DVD-ROM 52x
Enternet Card
Monitor Calor 14”

Spesifiksai Perangkat Keras Jaringan
Switch
Kabel UTP 4
Konektor RJ-45

C. Analisa Perangkat Lunak

Perangkat lunak ini sudah memenuhi staandar menjalankan *IDS Snort.* Seperti menggunakan *Ubuntu Server 12.04* untuk *server IDS atau windows 7* untuk *client*. Untuk mengelola *database* digunakan *mysql sever, webmin* dan *bas.*

Analisa Pemakai

Adapun profil pengguna *IDS* ini adalah :
Administrator. Sistem deteksi penyusup ini digunakan oleh *administrato*r komputer sebagai sarana utama mendiktesi adanya penyusup.
Operator . Operator adalah staf yang diberikan wewenang / hak oleh *administrator* untuk mengantikan tugan dan tangung jawab *administrator* jika diperlukan.

D. Pembuatan IDS (Instrusion Detection System)

IDS merupakan solusi yang sangat tepat untuk mengatasi permasalahan pada keamanan jaringan yang ada didalam sebuah jaringan *Server.* Karena *IDS* memiliki beberapa kemampuan dalam mengatasi permasalahan mengenai keamanan jaringan.Dengan memanfaatkan sebuah Laptop, hal tersebut akan dimaanfaatkan untuk pembuatan *Server* yang terinstal *IDS.*

E. Pengembangan Jaringan

Dengan adanya tujuan untuk mengatasi permasalahan pada keamanan jaringa pada sebuah jaringan *Server,* pengembangan keamanan pun dibuat sebagai solusi keamanan jaringa untuk menjalankan sesuai kebutuhan yang ada pada jaringan tersebut *IDS* sendiri menggunakan aplikasi *Snort.*

Kebutuhan Perangkat Lunak

IDS (Intrusion Detection System) adalah sebuah aplikasi perangkat lunak yang memonitor aktivitas jaringan atau sistem untuk kegiatan berbahaya atau pelanggaran kebijakan dan menghasilkan laporan-laporan Management Station. *IDS* dibagi menjadi 2 jenis ;

Network Intrusion Detection System (NIDS). Semua lalu lintas yang mengalir ke semua jaringan akan dianalis untuk mencari apakah ada percobaan serangan atau penyusupan ke dalam sistem jariingan. *NIDS* umumnya terletak didalam segmen jaringan penting dimana server berada atau terdapat pada pintu masuk jaringan.

Host-based Intrusion Detection System (HIDS). Aktivitas sebuah *host* jaringan individual akan dipantau apakah terjadi sebuah percobaan serangan atau penyusupan ke dalamnya atau tidak. *HIDS* seringnya diletakkan pada *server-server* kritis di jaringan, seperti halnya *firewall, web server,* atau server yang terkoneksi ke Internet.

Penerapan *IDS* sendiri menggunakan aplikasi Snort yang berfungsi sebagai software atau aplikasi atau juga sebuah *tools Security*. *IDS* sudah banyak mendukung berbagai macam *driver hardware,* beberapa hal yang perlu diperhatikan dalam instalasi antara lain dapat dilihat pada tabel III dan tabel IV berikut.

TABEL III SPESIFIKASI PERANGKAT KERAS

NO	HARDWARE	UNIT	SPESIFIKASI
1	Processor	1 Unit	Core i5
2	RAM	1 Unit	2 GB
3	Hard disc	1 Unit	500 GB
4	LAN Card	1 Unit	

TABEL IV SPESIFIKASI PERANGKAT LUNAK

NO	SOFTWARE	SPESIFIKASI
1	Sistem Operasi	Ubuntu Server 12.04LTS
2	Paket Snort	Snort 2.9.2
3	Paket Rules	Snortrules-pr-2.4
4	Paket BASE	BASE 1.4.5

F. Tinjau Fungsional IDS

Snort

Snort adalah aplikasi *open source* yang berbasis jaringan sistem deteksi intrusi (*NIDS)* yang memiliki kemampuan untuk melakukan analisis lalu lintas secara *real-time* dan paket logging pada jaringan Internet Protokol (IP). Snort melakukan analisis protokol, mencari konten, dan pencocokan konten. Program ini juga dapat digunakan untuk mendeteksi *probe* atau serangan, termasuk, namun tidak terbatas pada upaya sidik jari sistem operasi, antarmuka *gateway* umum, *buffer overflows*, probe server pesan blok, dan *stealth port scan*. *Snort* dapat dikonfigurasi dalam tiga mode utama: *sniffer, logger* paket, dan deteksi intrusi jaringan.

Dalam mode *sniffer*, program akan membaca paket jaringan dan menampilkannya pada konsol. Dalam mode paket *logger*, program ini akan log paket ke disk. Dalam mode deteksi intrusi, program ini akan memonitor lalu lintas jaringan dan menganalisany terhadap ruleset yang didefinisikan oleh pengguna. Maka program akan melakukan tindakan tertentu berdasarkan apa yang telah diidentifikasi. Kelebihan Snort dibanding IDS lainnya adalah:

Snort dapat dijalankan di *background* sebagai sebuah daemon, Cepat dalam meganalisa paket. Apakah paket merupakan paket 'aneh' atau normal, *Multiplatform*, dapat digunakan pada jaringan yang memilki *platform* berbeda dan sistem operasi mana saja, *Open source*, dan Merupakan *IDS* yang paling banyak dipakai di dunia.

Snort kadang meloloskan intrusi jika suatu intrusi terlalu besar atau melebihi beban *scanning* biasa.

Tujuan Snort

Untuk menditeksi instruksi-instruksi jaringan, menditeksi penyusupan yang memasuki jaringan, penyerangan terhadap jaringan,pemindaian dan berbagai bentuk ancaman lain di jaringan.jika terdapat pola serangan baru dan termasuk dalam kategori serangan berbahaya maka tambakan rule baru pada IDS Snort,sehingga sistem akan benar-benar menjadi sistem yang aman.

Cara kerja Snort

Snort menggunakan deteksi *signature* pada lalu lintas jaringan mencocokkan lalu lintas jaringan dengan daftar *signature* serangan yang disebut Snort *rules*. Jika aksi atau paket yang melintasi jaringan itu sesuai dengan *rules*, maka Snort *engine* akan menganggapnya sebagai intrusi dan dicatat pada log kemudian disimpan di database.

Snort Rules

Snort Rules merupakan database yang berisi pola-pola serangan berupa *signature* jenis-jenis serangan. *Snort Rules IDS* ini, harus diupdate secara rutin agar ketika ada suatu teknik serangan yang baru, serangan tersebut dapat terdeteksi. *Rule Snort* dapat didownload di *www.snort.org.*

Cara kerja Snort rules

Rules dibaca ke dalam struktur atau rantai data *internal* kemudian dicocokkan dengan paket yang ada. Jika paket sesuai dengan *rules* yang ada, tindakan akan diambil, jika tidak paket akan dibuang. Tindakan yang diambil dapat berupa *logging* paket atau mengaktifkan *alert.*

G. Basic Analysis and Security Engine (BASE)

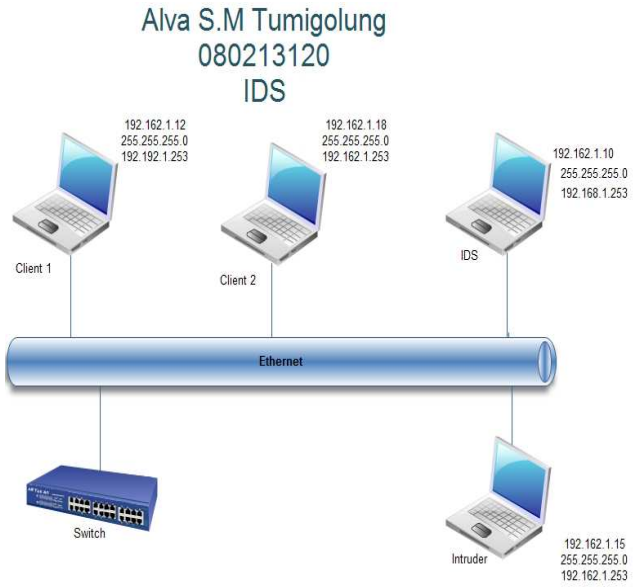
Ini merupakan program berbasis web yang memungkinkan implementasi antar platfon. Merupakan perintis antarmuka untuk *snort* dan paling banyak digunakan oleh pengguna *IDS*. *BASE* merupakan rekomendasi dari *snort* itu sendiri. *Multi language*, antar muka beberapa bahasa selain bahasa ingris. Layanan *real time*. *Open source*, dapat diimplementasikan pasa *IDS* manapun.

H. Cara Kerja IDS (Intrusin Detectoin System)

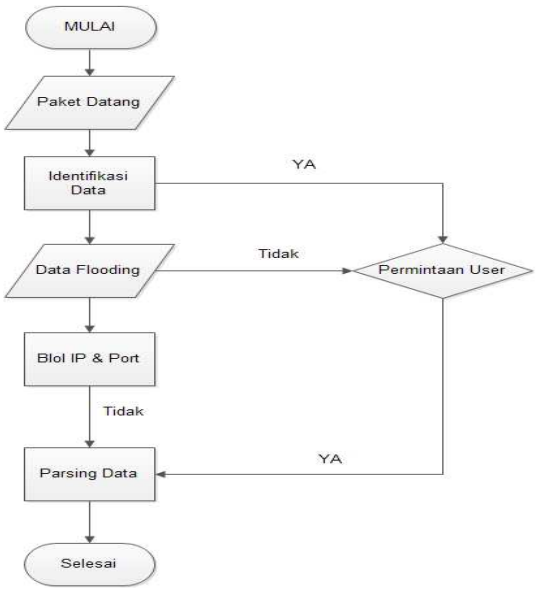
Pada gambar 1 terdiri dari 4 laptop, 2 diantaranya sebagai *Client, 1 Intruder* dan *1 IDS* dan *switch*. Dimana masing-masing berfungsi sebagai *client* atau pemakai dan intruder sebagai si penyerang didalam satu jaringan local, dan *IDS* sebagai penditeksi jika apabila terjadinya suatu serangan. Dan *switch* berfungsi sebagai penghubung jaringan ke semua penguna.

I. Diagram Alir Sistem

Pada gambar 2 berikut ini adalah merupakan diagram alir dari sistem flooding data.



Gambar 1. Topologi *IDS (Intrusin Detectoin System)*



Gambar 2. Diagram Alir Sistem

Input dari program adalah data jaringan yang masuk kemudian akan di proses apakah data yang ada tersebut melakukan *flooding* atau tidak. Jika data yang datang adalah *flooding* maka komputer akan mencari apakah data merupakan permintaan *user* atau tidak. Jika terbukti tidak maka secara otomatis akan memblok *ip* dan *port* darimana data itu berasal dan kalau ya berarti data akan ditujukan kepada tujuanya.

J. Desain Pengambilan Data

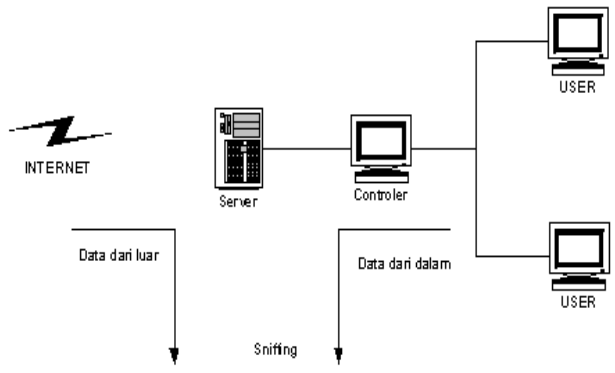
Pengambilan data yang kita gunakan adalah pengambilan data pada *Basic Analysis And Security Engine (BASE). BASE* menbaca log yang ditulis pada *snort* pada *database* secara otomatis.*BASE* akan menampilkan sebagai alaram dan *list log* dengan antar muka yang mudah dipahami dan dapat dilihat oleh *administrator*.

Pada gambar 3, data yang akan masuk ataupun yang akan keluar dibelokan terlebih dahulu untuk diambil datanya sebelum dilanjutkan ke tujuan sebenarnya. Dalam pembelokan ini tidak berarti bahwa data paket ditahan dahulu untuk diteliti melainkan data hanya dating maupun keluar di *capture headernya*.

Pengolahan Data Transmission Control Protocol (TCP)

Flood yang disebabkan oleh *TCP* mungkin lebih jelas karena jenis *flood* yang digunakan adalah *SYN TCP flood*. Yaitu pengiriman paket *TCP SYN* untuk request koneksi hubungan host-to-host menggunakan protokol *TCP SYN* sebagai perantaranya. Meskipun demikian flood yang diakibatkan oleh *TCP SYN* adalah fatal bagi koneksi. Mengingat paket *TCP SYN* adalah paket yang besar membutuhkan bandwithnya. Dengan demikian untuk mendefinisikan suatu paket *TCP* apakah suatu paket yang dianggap *flood* adalah dengan melihat berapa kali munculnya paket *TCP SYN* dalam jumlah waktu 10 detik. Apabila ternyata melebihi batasan maka bisa di kategorikan dalam *flood TCP SYN*.

Paket *TCP SYN* biasanya sudah di atur bekerja di port tertentu, Apabila kemudian diketahui bahwa ada paket *TCP SYN* yang menggunakan *port-port* yang tidak di definisikan, atau tidak melalui port yang diperbolehkan maka bisa dianggap sebagai intruder. Paket intruder ini pun juga harus di cegah keberadaaanya karena biasanya paket-paket tersebut membawa file-file berbahaya bagi *server*, misalnya virus.



Gambar 3. Proses Pengambilan Data

Pengolahan Data Internet Control Message Protocol (ICMP)

Pengolahan data-data paket yang menggunakan ICMP, untuk mengantisipasi *flood* data yang disebabkan oleh paket yang menggunakan protokol *ICMP*. Dari bab sebelumnya diketahui bahwa kebanyakan *flood* yang disebabkan oleh paket yang menggunakan protokol *ICMP* adalah *PING Flood*. Pada kondisi normalnya penggunaan protokol *ICMP* untuk suatu kegiatan PING adalah paket dalam ukuran yang kecil, Tentunya Tetapi dalam kondisi lain hal ini sangatlah mengganggu, misalnya pada jam-jam sibuk. Akan mengakibatkan lambatnya alur keluar masuknya data. Dengan asumsi tersebut tentunya pengiriman paket *ICMP* dengan skala besar akan sangat lah mengganggu. Untuk itu diberi batasan bahwa paket *ICMP* yang masuk adalah kurang dari 100 *byte*. Sedangkan paket *ICMP* besar tidak diperbolehkan masuk, atau bisa dianggap sebagai *flood ICMP*. Begitu pula perlu juga dibatasi bahwa paket *ICMP* dari luar hanya diperbolehkan 5 buah paket dalam setiap detiknya.

Pengolahan Data User Datagram Protocol (UDP)

Seperti halnya *protokol ICMP* pengiriman paket melalui *UDP* juga merupakan jenis pengiriman paket berupa datagram. Yang pada skala normalnya juga merupakan paket data yang berukuran kecil. Penggunaan protokol ini pun juga termasuk jarang digunakan untuk hubungan antar *host*, mengingat sifatnya yang tidak baik keamanannya. Sehingga apabila ada hubungan *UDP* dengan kontinuitas yang tinggi atau besar paket *UDP* nya besar bisa dianggap sebagai suatu *flooding.*

K. Proses Penginstalan

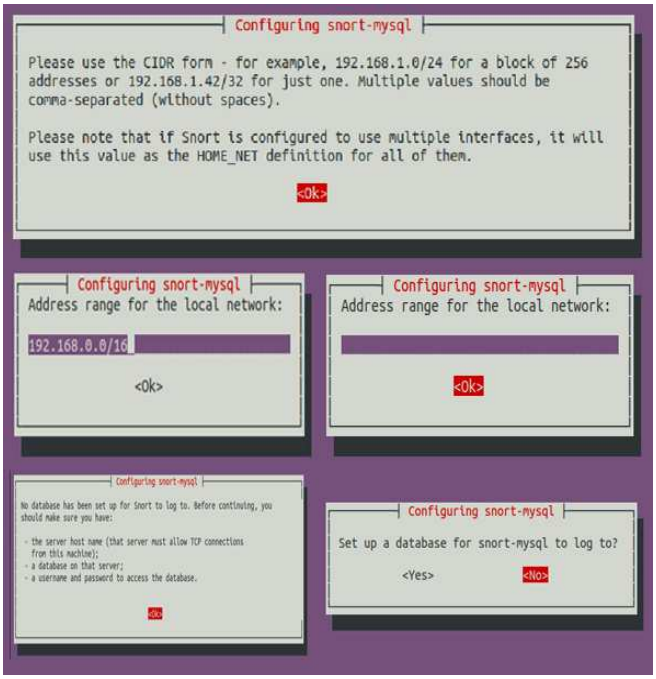
Proses Penginstalan Snort-Mysql

Buka terminal untuk penginstalan snort-mysql dan kemudian ketik perintah yang berikut. *~$ sudo apt-get install snort-mysql*. seperti terlihat pada gambar 4. untuk melanjutkanya tekan Y.

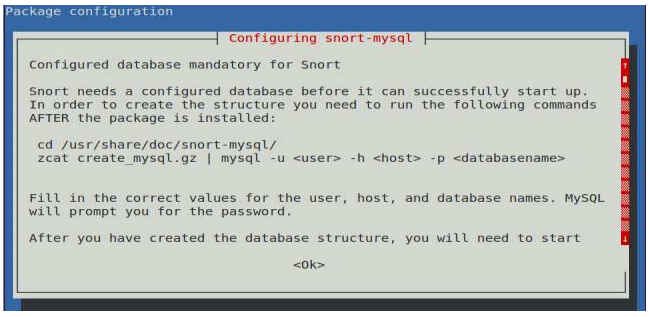
#apt-get install snort_mysql

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libprelude2 oinkmaster snort-common snort-common-libraries
  snort-rules-default
Suggested packages:
  snort-doc
The following NEW packages will be installed:
  libprelude2 oinkmaster snort-common snort-common-libraries snort-mysql
  snort-rules-default
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
1 not fully installed or removed.
Need to get 2,513kB of archives.
After this operation, 10.4MB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Gambar 4. Proses Instalasi *Snort-Mysq*



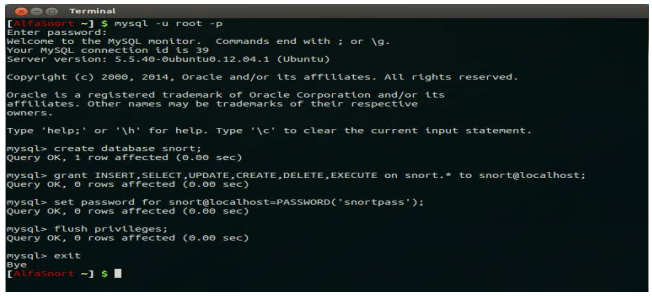
Gambar 5. *konfing Snort-Mysql*



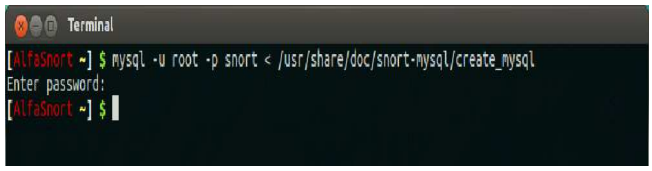
Gambar 6. Petunjuk *Database Snort*

Gambar 7. *Snort Complete*

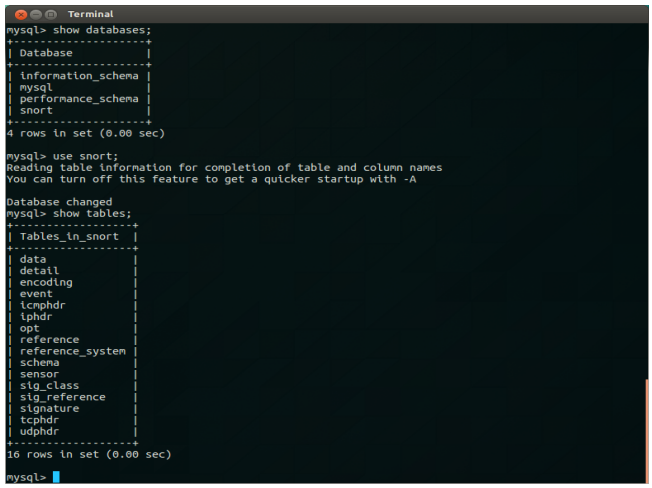
Gambar 8. *IDS* telah *Running*



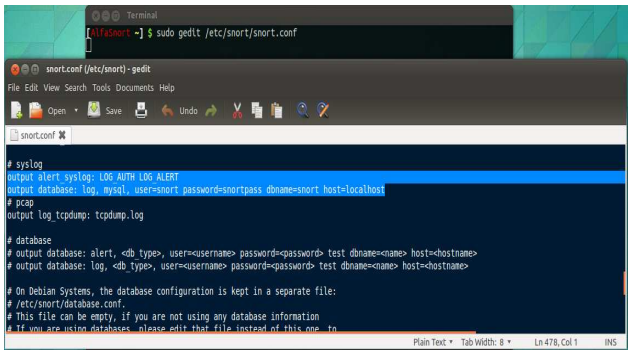
Gambar 9. Pembuatan *Database Snort*



Gambar 10. Perintah Untuk Membuat Tabel

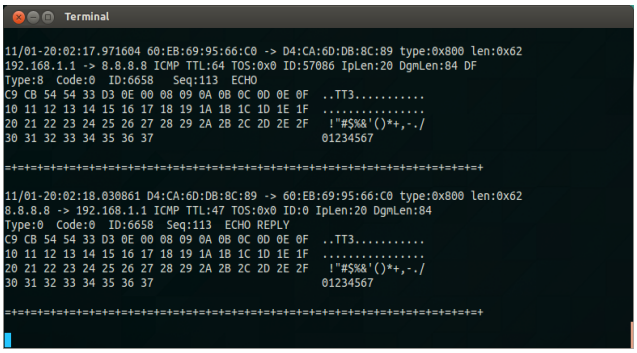


Gambar 11. *Database* dan Tabel

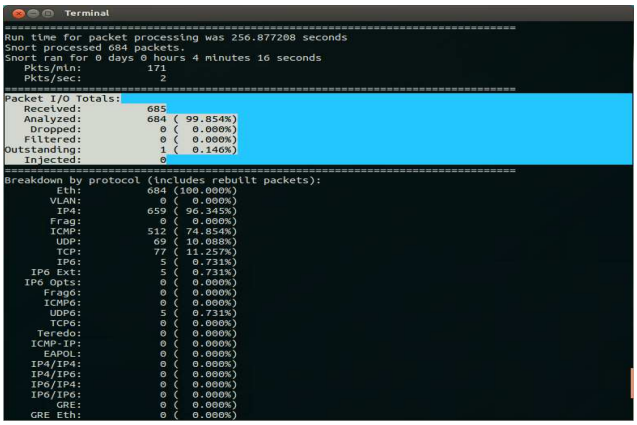


Gambar 12. *Derektori Snort*

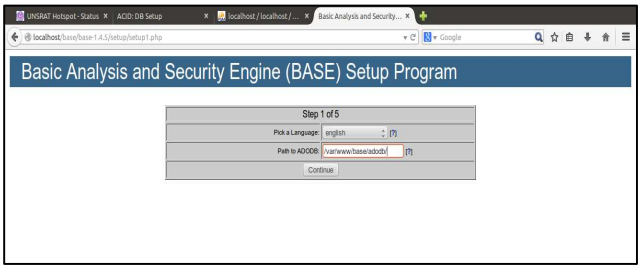
Gambar 13. *Snort Running*



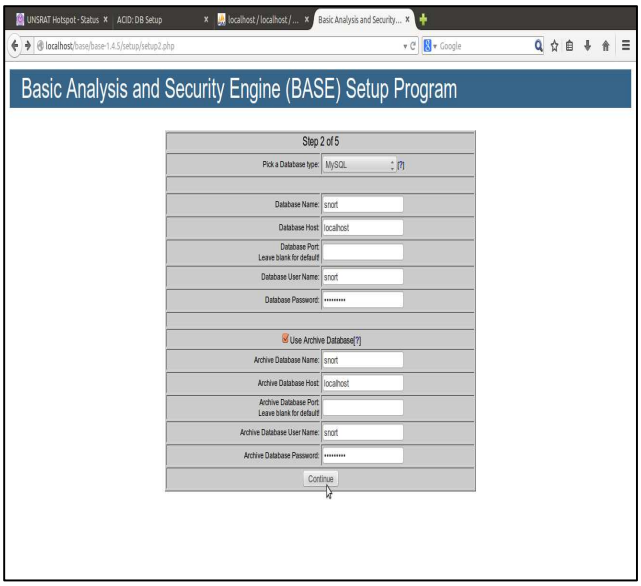
Gambar 14. Tampilan *Signature*



Gambar 15 *Statistik Output*

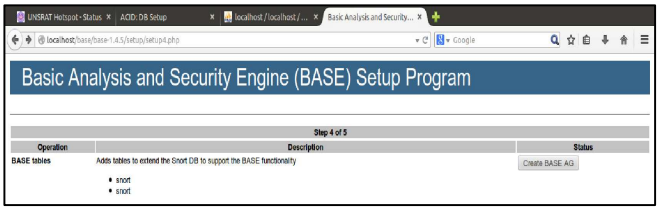


Gambar 16. *Base Setup Page*

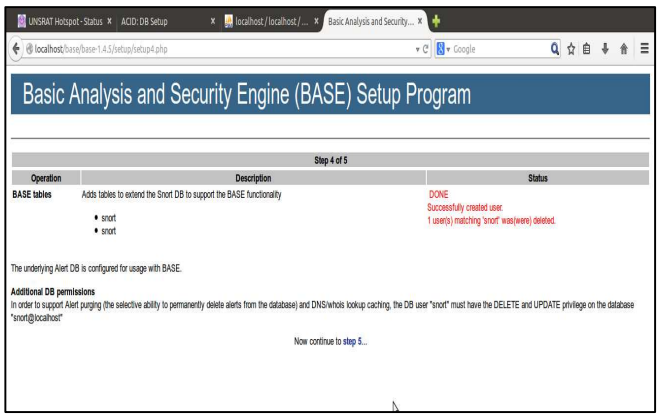


Gambar 17. *Database Port*

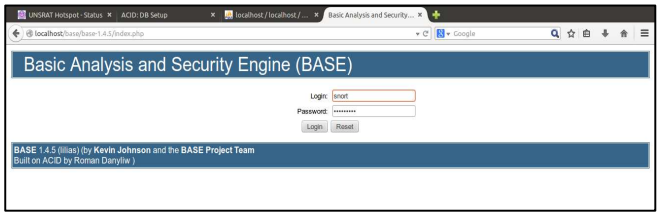
Gambar 18. *Database Username*



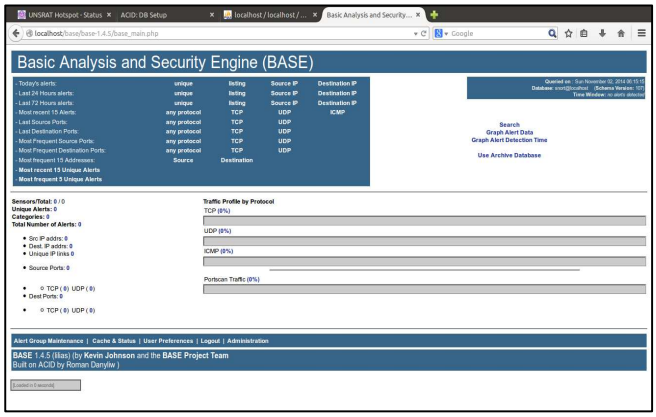
Gambar 19. *Create Table* Untuk *BASE*



Gambar 20. *Create table user Admin* sukses



Gambar 21. *Admin Login*



Gambar 22. *Home Page Base*

Setelah itu untuk melanjutkanya kita harus mengkonfigurasi *snort-mysql*. Lalu tekan OK. Lihat pada gambar 5. Kemudian kita harus mengatur *range IP* untuk *network local*.stelah itu OK untuk melanjutkan. Setelah itu pada alamat *range network local* kita kosongkan, kemudian OK untuk melanjutkanya. Setelah itu lanjutkan kembali config *snort-mysql*, lalu pilih OK untuk mengkonfirmasi selanjutnya. Selanjutnya kita pilih NO pada data penyimpan aktifitas jaringan karena pada paket konfig database yang sebelunya kita telah kosongkan.

Selanjutnya (gambar 6) merupakan petunjuk instalsi database pada *snort*. Setelah proses konfirmasi selesai maka penginstalan *snort* telah selesai dan kita cek apakah proses *instalsi* telah berhasil. Masukan perintah *# sudo snort –V*. (gambar 7). Setelah selesai diinstal sekarang kita cek apakah telah running atau masih ada yang *error*. Terlihat pada gambar 8 bahwa Itelah running dan siap dijalankan pada . Untuk proses pengecekan kita buka terminal dan masuk admin user. *# sudo /etc/init.d/snort restar*.

Setelah paket *snor*t telah terinstal, maka kemudian proses selanjutnya adalah pembuatan *database snort* (gambar 9). dan Perintahnya adalah

 # sudo mysql –u root –p
Enter password: "password_db"

Kemudian selanjutnya mengeksekusi tabel pada file *create_mysql* kedalam *database snort*, (gambar 10). Selanjutnya untuk melihat apakah tabel suskes di buat (gambar 11). kemudian kita masukan perintah berikut:

 # mysql –u root –p "password_db"
 # mysq> show database;
 # mysql> use snort;
#mysql> show tables;.

Selanjutnya mengatur file konfiguarsi *snort.conf* yang ada didalam direktori ***/etc/snort*** (gambar 12).

Snort mempunyai aneka macam *format* untuk *output file log*, tapi tidak semua *file log* bisa ditampilkan sekaligus. Hal ini tergantung konfigurasi dan komponen yang digunakan. Pada umumnya *snort* mempunyai dua *output format* :

ASCII (American Standard Code for Information Interchange) logging : pada dasarnya *snort* menyimpan data *traffic* jaringan komputer dalam bentuk file *ASCII*. Ini merupakan *default* dari *setting snort*.

Logging ke database : *Log ke database* membutuhkan komponen tambahan yang akan dibahas pada bab implementasi dan hasil.

Ada tiga kombinasi yang dimiliki snort mempunyai output yang berbeda satu sama lain. Pada kasus snifer mode, snort jilankan pada command prompt/terminal dengan menggunakan perintah sebagai berikut :

#snort –v untuk melihat header TCP/IP paket yang lewat.
#snort –d untuk melihat isi paket.
#snort –e untuk melihat header link layar paket seperti eternet header.

Setelah semua sudah terinstal, selanjutnya kita lakukan pengecekan pada *snort* apakah *snort* sudah bisa *running.* Dengan perintah *# sudo snort –dev* (gambar 13). Aksi *snort* saat merekam paket lalu lintas jaringan *(Signature)* pada gambar 14 dijalankan dengan paket *capture mode*, bisa dibaca sebagai berikut. :

11/01-20:02:17.971604 : Tangal dan waktu paket di capture.
8.8.8.8 : Source Address.
192.162.1.1 : Destination Address.
ICMP : Paket yang dikirim menggunakan Message Protocol.
TTL : 47 : Nilai time to live atau TTL pada IP header adalah 47.
TOS : 0x0 : jenis servis dan nilai TOS
ID : 0 : ID paket
IPLEN : 20 : Panjang IP header.
DGMLEN : 84 : IP payload dihitung dalam jumlah byte
Seq : 113 : ICMP sequence number.
..TT3.. : Satu ICMP flags yang aktif.

Setelah *snort* dinon-aktifkan maka akan telihat statistik dari *capture* paket (gambar 15). Pada data tersebut paket *TCP,UDP, ICMP* yang di *capture* oleh *IDS* dengan jumlah paket *TCP* : 77, *UDP* : 69 dan *ICMP* : 152.

Peroses penginstalan Base (Basic Analisis and Security Engine)

Ini merupakan proses terakhir dalam penginstalan dalam menjalankan *snort IDS*. Untuk penginstalan Base kita ambil versi yang terakhir Base 1.4.5.

Paket-paket yang diperlukan adalah sebagai berikut :

PHP. Bahasa *scrip* yang digunakan adalah *PHP 5.04 versi scrip*. Konfigurasi *PHP* untuk keperluan *IDS* atau (*Intrusion Detection System).*

Jpgraph. Jpgraph merupakan suatu objek *orienta*si untuk membuat *library graph* untuk alaram *IDS.*

Apache 2.0. Apache merupakan *web server* yang digunakan sebagai perantara *web interface* supaya bisa menggunakan *Basic Analysis Security Engine (BASE*).

Mysql. Fungsi dari Mysql digunakan untuk mengkoneksikan *script PHP* dengan database *Mysq* ke database *Basic Analysis Security Engine (BASE).*

Adodb. Adodb merupakan database *abstration library* yang merupakan tambahan liblary untuk *PHP* yang berfungsi untuk menghubungkan database dengan (*Intrusion Detction System.) IDS*

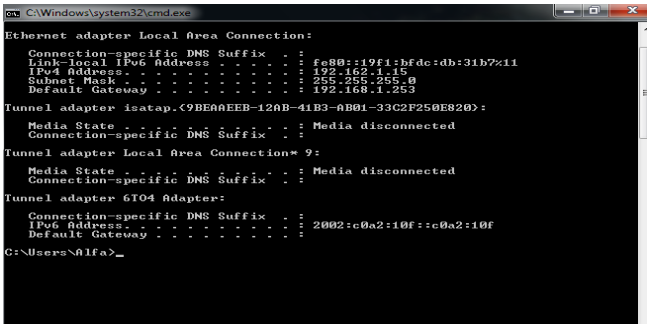
Oinkmaster. Oinkmaster seperti antivirus yang memerlukan apdate, *rule snort* juga perlu diupdate. *Update* dilakukan untuk memperoleh *rule* terbaru sehingga nantinya dapat memperoleh sebuah *rule* yang baru dan mampu mengetahui jenis-jenis serangan baru.

Proses Pada Testing BASE (Basic Analisis and Security Engine)

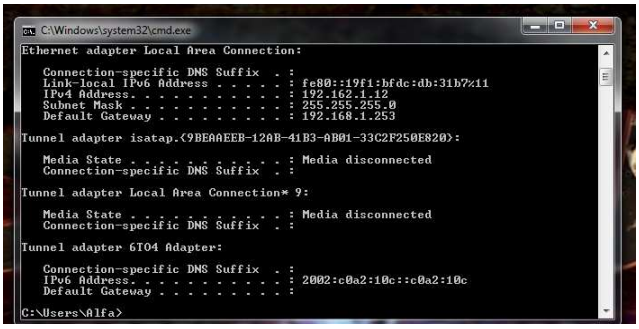
Testing terhadap *Base* merupakan testing terakhir untuk mengoperasikan *Intrusion Detection System (BASE).* pada jaringan komputer. Langka-langka yang dilakukan adalah sebagai berikut :

Testing yang dilakukan mengunakan browser http://localhost/base/base-1.4.5/setup/php.

Step pertama pengaturan *path* ke *adodb* (gambar 16). *Step* ke dua (gambar 17) pengaturan *database* pada *Base*, kemudian klik *continue* utuk ke *step* berikut. *Step* ke tiga (gambar 18) merupakan pengaturan *admin* pada *database Base*, kemudian

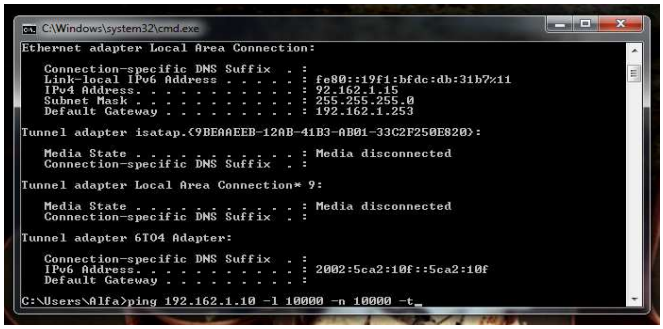


Gambar 23. Alamat IP *Client* 1



Gambar 24. Alamat IP *Client* 2

Gambar 25. Contoh perintah Serangan



Gambar 26. *Ping Of Death*

klik *continue* untuk proses selanjutnya. *Select Create BASE AG* maka akan tampil pesan '*database is completely configurad'* (gambar 19). Pada step ke lima (gambar 20) yakni pengaturan *create* pada *user admin* telah suskes. Halaman *admin login* untuk masuk ke *home Base* (gambar 21).

Setelah *admin login* berhasil masuk maka akan terlihat tampilan home page pada *Base* (gambar 22). Maka *Snort* dan *Base* siap untuk dijalankan.

IV. IMPLEMENTASI DAN PENGUJIAN

A. Konfigurasi sistem

Komputer Server

Komputer ini berfungsi sebagai host yang kita miliki yang akan di gunakan sebagai korban dari *flooding* data. Komputer ini dirancang agar bisa melakukan pemblokiran *IP* dilakukan sebagai berikut

ETH 0 :

IP : 192.168.1.10

Netmask : 225.225.225.0

Gateway : 192.168.1.253

Komputer Client

Komputer *Client* adalah seperangkat komputer yang memungkinkan pengguna untuk mengakses servis atau layanan dari komputer server. Istilah Komputer *Client* bisa di sebut dengan *Workstation* atau *Node*. komputer *client* juga bisa berfungsi sebagai penyerang pada sebuah *server*.

Pengesetan IP dilakukan sebagai berikut (gambar 23 dan gambar 24)

Client 1:

IP : 192.168.1.15

Netmask : 255.255.255.0

Gateway : 192.168.1.253

Client 2:

IP : 192.168.1.12

Netmask : 255.255.255.0

Gateway : 192.168.1.253

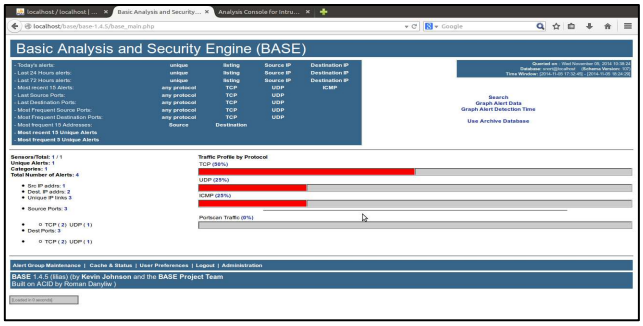
B. Proses Pengujian

Dalam pengujian ini dilakukan 2 pengujian yaitu menggunakan simulasi serangan *Syn Attack, dan Ping of Death*, *Syn Attack* terjadi bila suatu *host* hanya mengirimkan paket *SYN TCP* saja secara kontinyu tanpa mengirimkan paket *ACK* sebagai konfirmasinya. Hal ini akan menyebabkan *host* tujuan akan terus menunggu paket tersebut dengan menyimpan keadalam *baclog.* Meskipun ukuran paket kecil, tetapi apapbila pengiriman *SYN t*erus menerus akan memperbesar *backlog.* Hal ini terjadi apabila *backlog* sudah besar akan mengakibatkan host tujuan akan otomatis menolak semua paket *SYN* yang datang, sehingga host tersebut tidak akan otomatis menolak semua paket *SYN* yang datang, sehingga host tersebut tidak bisa koneksi oleh *host-host* yang lain. Sedangkan *Ping Of Death* merupakan pengiriman paket echo request ICMP ke dalam suatu jaringan secara berlebihan. Pengiriman paket ini dapat mengakibatkan sistem *crash,hang* ataupun *reboot.*

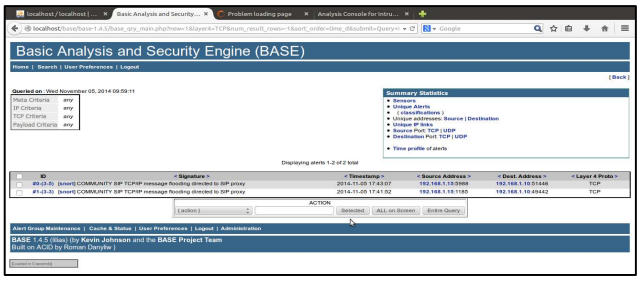
Cara kerja sistem yang dibangun ini adalah ketika *IDS* menditeksi adanya suatu serangan maka alert *BASE* akan merekam *IP* penyerang dan memerintahkan iptables melakuan diteksi terhadap *IP* yang telah tercatat. Sedangkan *BASE* dapat digunakan sebagai *user interface* dengan web-based. Pada *home page informasi BASE* menunjukan bahwa *BASE* menerima alarm dan menampilakan *source port, TCP,UDP,ICMP.*

C. Pengujian SYN Flooding Attack

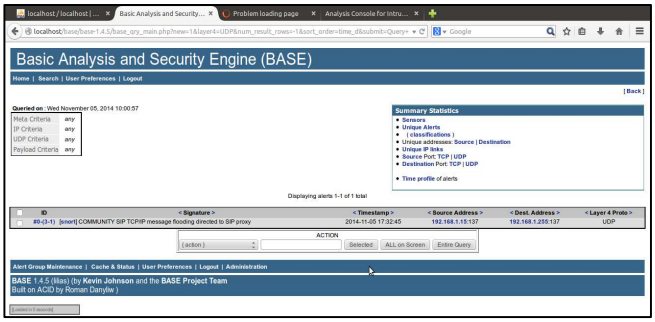
Pengujian dilakukan dengan menggunakan perintah cmd dengan *IP client 192.168.1.15* dengan target ke *IP korban 192.168.1.10.* contoh serangan paket ke target. *Ping*



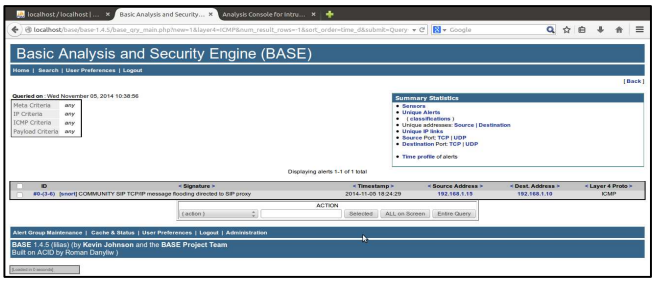
Gambar 27. Alarm pada Home Page



Gambar 28. Alarm protokol *TCP*



Gambar 29. *Alarm* protokol *UDP*



Gambar 30. *Alarm* protokol *ICMP*

192.168.1.10 –l 10000 –n 10000 –t. Serangan ini melibatkan satu komputer / koneksi internet untuk (membanjiri) sebuah *server* dengan paket *ICMP/TCP/UDP*,tujuan dari serangan ini adalah untuk membuat *bandwith server* menjadi *overload*, sehingga *server* tidak bisa lagi menangani trafik yang masuk dan *server* akhirnya *down* (gambar 25).

Pada proses pengujian ini, client yang bertindak sebagai penyerang mengirimkan paket *SYN* ke dalam *port-port* yang sedang berada dalam *server target*. Pada kondisi normal, paket *SYN* yang dikirmkan berisi alamat sumber yang menunyukan data-data aktual. Tetapi pada serangan *SYN* attack, paket-paket tersebut memiliki alamat sumber yang tidak menunjukan data aktual. Ketika server menerima paket *SYN* tersebut, *server* merespon dengan sebuah paket *SYN/ACK* sesuai dengan *SYN* paket yang dia terima dan kemudian akan menunggu paket *ACK* sebagai balsan untuk melengkapi proses tersebut. Karena alamat sumber dalam paket *SYN* yang dikirmkan oleh penyerang tidaklah *valid*, maka paket *ACK* tidak akan pernah terkirim ke target, dan port yang menjadi target serangan akan menunggu hingga waktu pembuatan koneksi telah habis atau *time-out*. Selanjutnya untuk pengujian telah disiapkan lokal rule pada snort yang digunakan untuk menditeksi adanya suatu penyeranagan *SYN*.

Attack. *Rule snort* akan membandingkan setiap paket data dari jaringan luar yang mengalir masuk ke *server* dengan *protokol TCP*. Maka ketika ada paket data yang sesuai dengan rule snort tersebut akan mengangapnya sebagai sebuah serangan dan memberikan peringatan memalui *web BASE*. Pesan peringatan yang ditampilkan halaman *web BASE* adalah ”*Syn Flooding Simalotion*”.

D. Pengujian Ping Of Death

Pengujian simulasi *Ping of Death* dilakukan dengan melakukan ping yang ada menyertakan paket data sebesar *10000 byte* terhadap komputer *server* dari komputer client atau si penyerang. Sebuah ping berukuran *64 byte*, perintah yang diberikan pada terminal adalah *ping 192.168.1.10 –l 10000 –n 10000 -t* (gambar 26).

Selanjutnya untuk pengujian telah disisipkan lokal *rule* pada *Snort* yang digunakan untuk menditeksi adanya suatu serangan *ping of death Atttack*. Dan bisa dilihat dalam bentuk *GUI* pada *database BASE* (gambar 27).

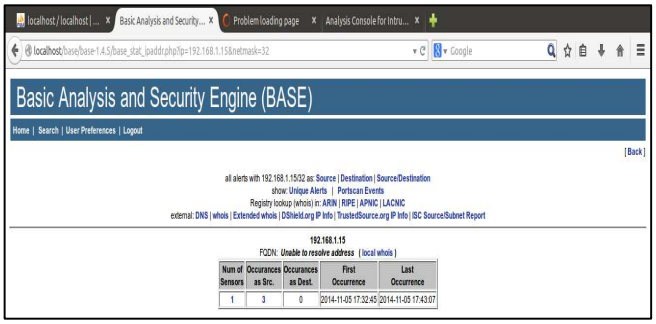
Rule snort sebagaimana sebagaimana, akan membandingkan setiap paket data dari jaringan luar yang mengalir masuk ke *server* dengan protokol *ICMP*. Parameter perbandingan yang digunakan mengacu pada ukuran file, yang ditandai adanya opsi ***dsize*** pada baris rule. Apabila ukuran *ICMP* berukuran lebih besar dari *1500 byte*, maka *snort IDS* akan menganggap paket *ICMP* tersebut sebuah serangan dan memberikan peringatan melalui *web BASE*. Pesan peringatan yang ditampilkan pada halaman *web BASE* adalah ”*Ping Of Death Simulation*”

Untuk melihat adanya alaram yang terbaru bisa dipilih most recent alaram. *BASE* akan menampilkan yang terbaru, dibagi menjadi beberapa kategori menampilkan semua protokol, menampilkan hanya protokol *TCP*,*UCP*,*ICP*. Hasil dari pencarian alarm menurut *port* (gambar 28, gambar 29 dan gambar 30). pada gambar 28 bahwa adanya alaram terbaru melalu protokol *TCP*, pada tanggal 5 november 2014, jam 17.43. adalah paket *flooding*.

Kemudian juga terjadi alarm terbaru melalui protokol *UDP*, pada tanggal yang sama juga tanggal 5 november 2014, jam 17:32 (gambar 29).Selanjutnya terjadi alarm terbaru melalui protokol *ICMP*, pada tangal yang sama juga tanggal 5 november 2014, jam 18:24 (gambar 30).

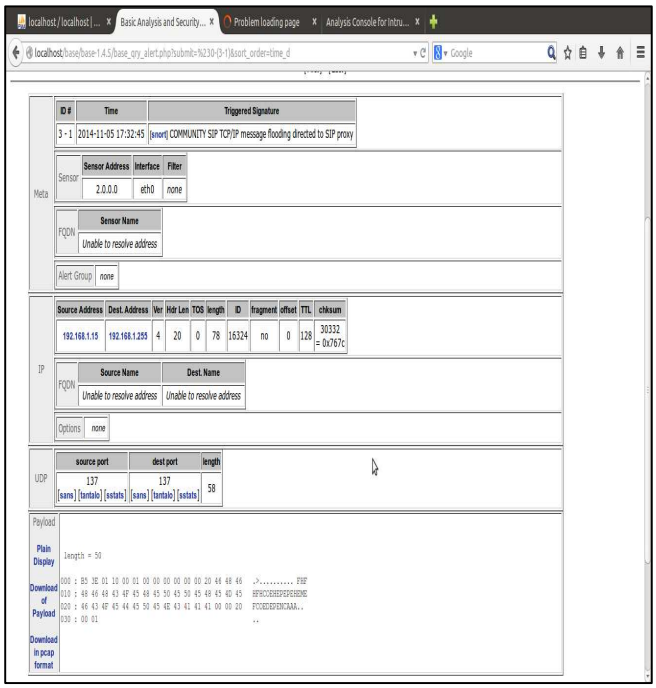
Dari hasil pencarian alarm dapat dibaca sebagai berikut : *ID # (3-5)* adalah nomor identifikasi yang unik untuk alarm yang dideteksi oleh snort. Semua informasi tentang alarm disimpan pada *ID di database*. Pada dasarnya untuk melihat informasi lebih mendetail untuk alaram terdapat pada *ID*, (gambar 33).

Timestamp: waktu dan jam terjadi suatu serangan di sini pada tanggal 5 november 2014. *Sourcre address* : *192.162.1.15* tapi *DNS* dari *IP* tersebut tidak bisa dideteksi

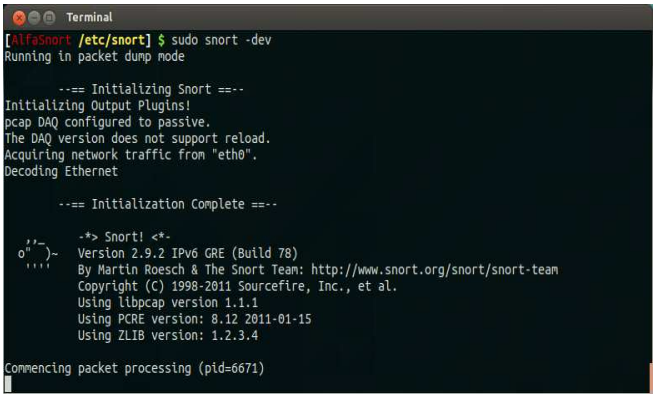


Gambar 31. Informasi *IP Address* Penyerang

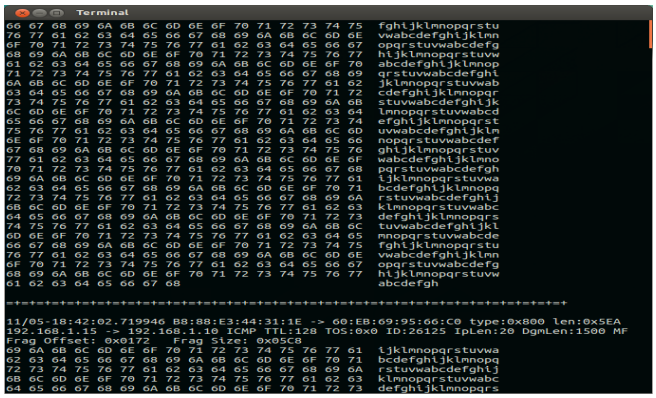
Gambar 32. Informasi *IP Address* Target



Gambar 33. Informasi Untuk *Alarm*



Gambar 34. *Running Sno*



Gambar 35. Hasil *Sniffing* Paket Pada Jaringan

sebagaimana tampilan *FQDN*: *(Unable To Resolve Address*). (gambar 31).

Destination address: Tujuan dari *IP 192.162.1.15* melakukan *Simple Network Management Protocol (SNMP) broadcast* trap terhadap *broadcast IP* korban 192.162.1.10 (gambar 32).

E. Output Paket Intrusion Detection Mode

Contoh hasil *snifing* paket dijaringan mengunakan perinta *# snort –dev* pemilihan dari interface yang digunakan (gambar 34).

Pada gambar 35 *snort* dijalankan dengan paket *capture mode*, contoh diatas bisa dibaca sebagai berikut. :

11/05-18:42:02.719946 : Tanggal dan waktu paket di capture.
192.162.1.15 : Source Address.
192.162.1.10 : Destination Address.
ICMP : Paket yang dikirim menggunakan Message Protocol.
TTL : 128 : Nilai time to live atau TTL pada IP header adalah 47.
TOS : 0x0 : jenis servis dan nilai TOS
ID : 26125 : ID paket
IPLEN : 20 : Panjang IP header.
DGMLEN : 1500 mf : IP payload dihitung dalam jumlah byte
..ijklmno.. : Satu ICMP flags yang aktif.

Setelah *snort* dinon-aktifkan maka akan telihat statistik dari *capture* paket. Pada data tersebut paket *TCP,UDP,ICMP* yang di capture oleh *IDS* denganjumlah paket *TCP : 564, UDP : 472 dan ICMP : 7. ARP : 556.*

F. Pendeteksi Flooding

Pendeteksi ini dapat dilakukan dengan cara menjalankan *Snort IDS*, sehingga serangan dapat terdeteksi, maka disaat bitu pula serangan yang masuk dapat dicegah. Kemudian dari *BASE* bisa mengabil data dan memberitahukan alamat *IP* si penyerang.

G. Kemampuan Sistem Dalam Mengambil Data

Sistem mempunyai kemampuan melihat semua paket yang datang dalam bentuk apapun. Meskipun demikian sistem hanya mengambil paket-paket dari tiga protokol utama yang biasa digunakan untuk mentransfer data. Protokol itu adalah *TCP/SYN ATTACK, UPD* dan *ICMP/PING OF DEATH ATTACK*. Hal ini disebabkan karena *flood* yang biasa terjadi

dalam jaringan dilakukan melalui dua protokol tersebut. Sedangkan protokol lain hampir tidak pernah mengalami data *flooding*.

H. Kemampuan Sistem Dalam Mengelolah Data

Pengolahan data dari setiap paket datang yang masuk ditujukan untuk mengoptimasikan kerja komputer agar tidak terjadi komputer mengalami *crash* atau *hang*. Karena sistem mengolah data-data yang tidak perlu. Pengolahan data tersebut meliputi:

Pemisahan paket data

Pada pengolahan ini sistem akan memisahkan data-data mana yang perlu ditampilkan pada tabel dan data-data mana yang tidak perlu ditampilkan pada tabel. Untuk paket *ICMP* data-data yang tidak masuk kategori *flood* dihapus langsung. Untuk data yang menggunakan *TCP* sebagai protokol, data yang di tampilkan hanya data yang merupakan paket *SYN* saja, data *acknowledge* tidak ditampilkan.

Pengidentifikasian data flooding

System mampu mengidentifikasikan apakah data yang datang itu *flooding* atautidak. Secara visual data yang dikategorikan *flooding* akan masuk lansung dalam tabel *blacklist* untuk di *blok IP* pengirimnya.

Pengiriman data TCP, UDP, UCMP secara otomatis untuk blokir IP

Apabila setelah teridentifikasi *flooding* maka control akan mengirimkan satu paket data melalui protokol *UDP*,

TCP, *ICMP*. Pengiriman paket ini digunakan untuk memerintahkan *server* melakukan pengeblokan *IP sever* yang melakukan *flooding* terhadap kita.

I. Hasil Pengujian

Snort berhasil mendeteksi intruksi *flooding* dan mecatatnya dalam ke dalam alarm dan log ke dalam database. *BASE* merima alarm tersebut dan menyajikanya ke dalam bentuk *GUI*. Kemudian *firewall* berhasil melakukan *dropping* paket dan *memblocking IP* penyerang.

V. KESIMPULAN

Berdasrkan hasil pengujian yang dilakukan *Snort* dapat diimplementasikan sebagai (*Instruction Detection System) IDS.* Pada sebuah system operasi *Ubuntu Server* untuk menditeksi serangan *DOS Attack* yaitu *flooding data* dengan *SYN Flood Attack dan Ping Of Death* sebagai sempel pengujian.

ada masih mengalami ganguan, yaitu berupa penuhnya jaringan yang ada. Proses pengiriman data dan penerimaan data akan mengalami kelambatan.

Snort dapat memberikan peringatan adanya sebuah serangan keamanan sehingga dapat meningkatkan keamanan jaringan. Dapat atau tidaknya sebuah serangan terdeteksi oleh *Snort* tergantung dari ada atau tidaknya *rules* dengan jenis *Signature* pada sebuah pola serangan. Kemudian *BASE* akan membaca database dan menampilkan alaram dengan *GUI WEB-BASE*.

Sistem dapat mendeteksi *flooding* data. Data yang keluar masuk akan dideteksi,sehingga semua data bisa dilihat apakah data itu merupakan *flooding* atau bukan, sehingga data bisa mengklasifikasikan bahwa data tersebut benar-benar melakukan *flooding* atau tidak.

DAFTAR PUSTAKA

[1] D. Ariyus, Istrusion Detection System, Yogjakarta. 2007.
[2] S. Prakoso, Jaringan Komputer Linux, Yogjakarta. 2005.
[3] R. Rafiudin, Mengganyang Hacker Dengan Snort, Yogjakarta. 2010.
[4] W. Andi, Konsep Dan Implementasi Jaringan Komputer Dengan Linux ubuntu, Yogjakarta. 2014.