IMPLEMENTASI *INTRUSION DETECTION SYSTEM (IDS)* DI JARINGAN UNIVERSITAS BINA DARMA

Maria Ulfa
Dosen Universitas Bina Darma
Jalan Jenderal Ahmad Yani No.12 Palembang
Pos-el: mariakurniawan2009@gmail.com

Abstract: *Computer security systems, in recent years has become a major focus in the world of computer networks, this is due to the high threat of suspicious and attacks from the internet. Bina Darma University is one of the agencies which activities using the internet network services, ranging from the processing of existing data, including the KRS online system, mail server and web portal in each unit and others. Bina Darma University network manager for this building system is a network security by implementing a firewall and proxy server on each server in the network unit. To further optimize the network security system at the University of Bina Darma, the author will implement a network Intrusion Detection System at the Bina Darma University as network security solutions for both the intranet and internet network of Bina Darma University, where the author will build an IDS (Intrusion Detection System) using a snort.*

Keywords: *Networking Security, Firewall, Proxy Server, IDS (Intrusion Detection System), and Snort*

Abstrak: *Sistem keamanan komputer, dalam beberapa tahun ini telah menjadi fokus utama dalam dunia Jaringan Komputer, hal ini disebabkan tingginya ancaman yang mencurigakan dan serangan dari Internet. Universitas Bina Darma merupakan salah satu instansi yang aktivitasnya menggunakan layanan jaringan internet, mulai dari mengolah data yang ada, diantaranya adalah sistem KRS online, mail server dan web portal di tiap unit dan lain-lain. Pengelolah jaringan Universitas Bina Darma selama ini membangun sistem keamanan jaringan dengan menerapkan sistem firewall dan proxy sever pada tiap unit server di jaringannya. Untuk lebih mengoptimalkan sistem keamanan jaringan di universitas Bina Darma maka Pada penelitian ini penulis akan mengimplementasikan Intrusion Detection System pada jaringan Universitas Bina Darma sebagai solusi untuk keamanan jaringan baik pada jaringan Intranet maupun jaringan Internet Universitas Bina Darma. Dimana penulis akan membangun sebuah IDS (Intrusion Detection System) dengan menggunakan snort.*

Kata kunci*: Keamanan Jaringan, Firewall, Proxy Server, IDS (Intrusion Detection System), and Snort*

1. PENDAHULUAN

Keamanan jaringan komputer sebagai bagian dari sebuah sistem informasi adalah sangat penting untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya. Sistem harus dilindungi dari segala macam serangan dan usaha-usaha penyusupan oleh pihak yang tidak berhak. Menurut Stiawan (2009) Sistem keamanan komputer, dalam beberapa tahun ini telah menjadi fokus utama dalam dunia jaringan komputer, hal ini disebabkan tingginya ancaman yang mencurigakan (*Suspicious Threat*) dan serangan dari Internet. Keamanan Komputer (*Security*) merupakan salah satu kunci yang dapat mempengaruhi tingkat *Realibility* (keandalan) termasuk *Performance* (kinerja) dan *Availability* (tersedianya) suatu *Internetwork*.

Kerusakan yang terjadi pada suatu jaringan akan mengakibatkan pertukaran data yang terjadi pada jaringan tersebut akan melambat atau bahkan akan merusak suatu sistem jaringan. Insiden keamanan jaringan adalah suatu aktivitas terhadap suatu jaringan komputer yang memberikan dampak terhadap

keamanan sistem yang secara langsung atau tidak bertentangan dengan *security policy* sistem tersebut (Wiharjito, 2006).

Universitas Bina Darma merupakan salah satu instansi yang aktivitasnya didukung oleh layanan jaringan internet, mulai dari mengolah data yang ada, diantaranya adalah sistem KRS *online*, *mail server* dan *web portal* di tiap unit dan lain-lain. Pengelolah jaringan Universitas Bina Darma selama ini membangun sistem keamanan jaringan dengan menerapkan sistem *firewall* dan *proxy* sever pada tiap unit server di jaringannya.

Pada dasarnya, menurut Arief (2010) *firewall* adalah titik pertama dalam garis pertahanan sebuah sistem jaringan komputer. Seharusnya *firewall* diatur agar melakukan penolakan (*deny*) terhadap semua *traffic* yang masuk kedalam sistem dan kemudian membuka lubang-lubang yang perlu saja. Jadi tidak semua lubang dibuka ketika sistem melakukan hubungan ke jaringan luar. Idealnya *firewall* diatur dengan konfigurasi seperti diatas. Beberapa *port* yang harus dibuka untuk melakukan hubungan keluar adalah *port* 80 untuk mengakses internet atau *port* 21 untuk FTP file server. Tiap-tiap *port* ini mungkin penting untuk tetap dibuka tetapi lubang-lubang ini juga merupakan potensi kelemahan atas terjadinya serangan yang akan masuk kedalam jaringan. *Firewall* tidak dapat melakukan pemblokiran terhadap jenis serangan ini karena administrator sistem telah melakukan konfigurasi terhadap *firewall* untuk membuka kedua *port* tersebut. Untuk tetap dapat memantau *traffic* yang terjadi di kedua *port* yang terbuka tersebut dibutuhkan sebuah sistem yang dapat melakukan deteksi terhadap *traffic* yang membahayakan dan berpotensi menjadi sebuah serangan.

Oleh karena itu, penerapan *IDS* (*Intrusion Detection System*) diusulkan sebagai salah satu solusi yang dapat digunakan untuk membantu pengaturan jaringan dalam memantau kondisi jaringan dan menganalisa paket-paket berbahaya yang terdapat dalam jaringan tersebut, hal ini bertujuan untuk mencegah adanya penyusup yang memasuki sistem tanpa otorisasi (misal: *cracker*) atau seorang user yang sah tetapi menyalahgunakan *privilege* sumber daya sistem.

Penelitian ini akan mengimplementasikan *Intrusion Detection System* pada jaringan Universitas Bina Darma sebagai solusi untuk keamanan jaringan baik pada jaringan intranet maupun jaringan internet Universitas Bina Darma. Di mana akan membangun sebuah *IDS* (*Intrusion Detection System*) dengan menggunakan *snort*, karena *snort* merupakan *IDS open source* dan dinilai cukup bagus kinerjanya.

2. METODOLOGI PENELITIAN

Metode penelitian yang digunakan adalah penelitian tindakan (*action research*) menurut Davison, Martinsons dan Kock (2004, dalam Chandrax 2008). Penelitian tindakan yaitu mendeskripsikan, menginterpretasi dan menjelaskan suatu situasi atau keadaan pada jaringan *VLAN* server di Universitas Bina Darma dan melakukan analisis terhadap penerapan *Intrusion Detection System*.

Pada penerapan *Intrusion Detection System* yaitu dengan menggunakan beberapa

komponen *Intrusion Detection System* yang terdiri dari *snort engine*, *rule database*, dan *alert* dengan menggunakan *software* atau modul tambahan seperti *webmin* dan program *BASE* (*Basic Analysis and Security Engine*) atau *ACID* (*Analisys Console for Intrusion Databases*) serta sistem operasi Linux Ubuntu 10.04 server.

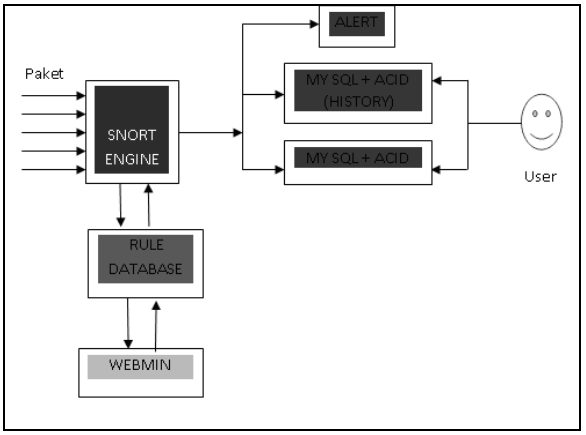
Adapun tahapan penelitian yang merupakan siklus dari *action research* ini yaitu : 1) Melakukan diagnosa dengan melakukan identifikasi masalah pokok yang ada pada objek penelitian. Dimana pada penelitian ini penulis melakukan diagnosa terhadap jaringan *VLAN* server Universitas Bina Darma yaitu dengan mengenal dan mempelajari jenis-jenis serangan yang sering terjadi dalam jaringan; 2) Membuat rencana tindakan yaitu memahami pokok masalah yang ditemukan dan menyusun rencana tindakan yang tepat. Pada tahapan ini penulis melakukan rencana tindakan yang akan dilakukan pada jaringan dengan membuat perancangan dan penerapan *Intrusion Detection System* pada jaringan *VLAN* server Universitas Bina Darma; 3) Melakukan tindakan disertai dengan implementasi rencana yang telah dibuat dan mengamati kinerja *Intrusion Detection System* pada jaringan *VLAN* server Universitas Bina Darma yang telah dibangun; 4) Melakukan evaluasi hasil temuan setelah proses implementasi, pada tahapan evaluasi penelitian yang dilakukan adalah hasil implementasi *Intrusion Detection System* terhadap jaringan *VLAN* server Universitas Bina Darma. Evaluasi ini dilakukan untuk mengetahui kelebihan dan kekurangan *Intrusion Detection System* yang sudah diterapkan pada jaringan *VLAN* server Universitas Bina Darma dalam meningkatkan keamanan jaringan; 5) Pembelajaran yaitu mengulas tahapan yang telah dilakukan dan mempelajari prinsip kerja *Intrusion Detection System* serta untuk memperbaiki kelemahan dari penerapan *Intrusion Detection System* pada jaringan *VLAN* server Universitas Bina Darma.

2.1 Lokasi Penelitian

Penelitian ini dilakukan pada Jaringan Universitas Binadarma khususnya pada Unit Pelayanan Terpadu atau MIS Universitas Bina Darma Palembang yang terletak di Jl. Jenderal Ahmad Yani No.12 Palembang.

2.2 Kerangka Penelitian

Dalam penelitian ini dapat dilihat alur perancangan sistem *Intrusion Detection System* (*IDS*) pada jaringan *VLAN* server Universitas Bina Darma, penelitian yang dilakukan dengan menggunakan beberapa komponen *IDS* seperti *Rule Snort*, *Snort Engine* dan *Alert* yang akan diterapkan pada jaringan *VLAN* server Universitas Bina Darma.



Gambar 1. Kerangka Penelitian

2.3 Jenis-jenis *IDS*

IDS adalah sebuah aplikasi perangkat lunak atau perangkat keras yang bekerja secara otomatis untuk memonitor kejadian pada jarigan komputer dan menganalisis masalah keamanan jaringan. Sasaran *Intrusion Detection System* (*IDS*) adalah memonitoring aset jaringan untuk mendeteksi perilaku yang tidak lazim, kegiatan yang tidak sesuai, serangan atau menghentikan serangan (penyusupan) dan bahkan menyediakan informasi untuk menelusuri penyerang. Pada umumnya ada dua bentuk dasar *IDS* yang digunakan yaitu (Thomas, 2005) :

1) *Network based Intrusion Detection System* (*NIDS*): Menempati secara langsung pada jaringan dan melihat semua aliran yang melewati jaringan. *NIDS* merupakan strategi yang efektif untuk melihat *traffic* masuk / keluar maupun *traffic* di antara *host* atau di antara segmen jaringan lokal. *NIDS* biasanya dikembangkan di depan dan di belakang *firewall* dan *VPN gateway* untuk mengukur keefektifan peranti-peranti keamanan tersebut dan berinteraksi dengan mereka untuk memperkuat keamanan jaringan.
2) *Host-Based Intrusion Detection System (HIDS). HIDS* hanya melakukan pemantauan pada perangkat komputer tertentu dalam jaringan. *HIDS* biasanya akan memantau kejadian seperti kesalahan login berkali-kali dan melakukan pengecekan pada file.

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

1) *Knowledge-based* atau *misuse detection*. *Knowledge-based IDS* dapat mengenali adanya penyusupan dengan cara menyadap paket data kemudian membandingkannya dengan *database rule IDS* (berisi *signature-signature* paket serangan). Jika paket data mempunyai pola yang sama dengan (setidaknya) salah satu pola di *database rule IDS*, maka paket tersebut dianggap sebagai serangan, dan demikian juga sebaliknya, jika paket data tersebut sama sekali tidak mempunyai pola yang sama dengan pola di *database rule IDS*, maka paket data tersebut dianggap bukan serangan.
2) *Behavior based (anomaly*). *IDS* jenis ini dapat mendeteksi adanya penyusupan dengan mengamati adanya kejanggalan-kejanggalan pada sistem, atau adanya penyimpangan-penyimpangan dari kondisi normal, sebagai contoh ada penggunaan memori yang melonjak secara terus menerus atau ada koneksi paralel dari 1 buah *IP* dalam jumlah banyak dan dalam waktu yang bersamaan. Kondisi-kondisi diatas dianggap kejanggalan yang kemudian oleh *IDS* jenis *anomaly based* dianggap sebagai serangan.

2.4 Tujuan Penggunaan *IDS*

IDS merupakan *software* atau *hardware* yang melakukan otomatisasi proses monitoring kejadian yang muncul di sistem komputer atau jaringan, menganalisanya untuk menemukan permasalahan keamanan (Bace dan Mell, 2005). *IDS* adalah pemberi sinyal pertama jika seorang penyusup mencoba membobol sistem keamanan komputer kita. Secara umum penyusupan bisa berarti serangan atau ancaman terhadap keamanan dan integritas data, serta tindakan atau

percobaan untuk melewati sebuah sistem keamanan yang dilakukan oleh seseorang dari internet maupun dari dalam sistem.

IDS tidak dibuat untuk menggantikan fungsi *firewall* karena kegunaannya berbeda. Sebuah sistem *firewall* tidak bisa mengetahui apakah sebuah serangan sedang terjadi atau tidak. *IDS* mengetahuinya. Dengan meningkatnya jumlah serangan pada jaringan, *IDS* merupakan sesuatu yang diperlukan pada infrastruktur keamanan di kebanyakan organisasi.

Secara singkat, fungsi *IDS* adalah pemberi peringatan kepada administrator atas serangan yang terjadi pada sistem kita. Alasan mempergunakan *IDS* (Bace and Mell, 2005), (Balasubramaniyan dkk, 2008): 1) Untuk mencegah resiko timbulnya masalah; 2) Untuk mendeteksi serangan dan pelanggaran keamanan lainnya yang tidak dicegah oleh perangkat keamanan lainnya. Biasanya penyusupan berlangsung dalam tahapan yang bisa diprediksi. Tahapan pertama adalah probing, atau eksploitasi pencarian titik masuk. Pada sistem tanpa *IDS* , penyusup memiliki kebebasan melakukannya dengan resiko kepergok lebih kecil. *IDS* yang mendapati probing, bisa melakukan blok akses dan memberitahukan tenaga keamanan yang selanjutnya mengambil tindakan lebih lanjut; 3) Untuk mendeteksi usaha yang berkaitan dengan serangan misal *probing* dan aktivitas *dorknob rattling*; 4) Untuk mendokumentasikan ancaman yang ada ke dalam suatu organisasi. *IDS* akan mampu menggolongkan ancaman baik dari dalam maupun dari luar organisasi. Sehingga membantu pembuatan keputusan untuk alokasi sumber daya keamanan jaringan; 5) Untuk bertindak sebagai pengendali kualitas pada administrasi dan perancangan keamanan, khususnya pada organisasi yang besar dan kompleks. Saat ini *IDS* dijalankan dalam waktu tertentu, pola dari pemakaian sistem dan masalah yang ditemui bisa nampak. Sehingga akan membantu pengelolaan keamanan dan memperbaiki kekurangan sebelum menyebabkan insiden; 6) Untuk memberikan informasi yang berguna mengenai penyusupan yang terjadi, peningkatan diagnosa, *recovery*, dan perbaikan dari faktor penyebab. Meski jika *IDS* tidak melakukan *block* serangan, tetapi masih bisa mengumpulkan informasi yang relevan mengenai serangan, sehingga membantu penanganan insiden dan *recovery*. Hal itu akan membantu konfigurasi atau kebijakan organisasi.

2.5 Respon *IDS*

Respon yang diberikan oleh suatu *IDS* biasanya dikelompokkan dalam tiga kategori: pemberitahuan (*notification*), *storage*, dan *active response*. Contoh respon yang ada (*Internet Security Systems , www.iss.net.net*) :

Tabel 1. Respon *IDS*

Respon	***NIDS***	***HIDS***
Notification	*Alarm* ke *console*	*Alarm* ke *console*
	E-mail	*E-mail*
	SNMP trap	*SNMP trap*
	Melihat *session* yang aktif	
Storage	Laporan *log*	Laporan *log*
	Data *log* mentah	
Aktif	Memutuskan koneksi (*TCP reset*)	Menghentikan *login User*
	Konfigurasi ulang *firewall*	Melakukan *disable Account user*

2.6 Karakteristik IDS

Berikut adalah beberapa kriteria yang diinginkan untuk suatu *IDS* yang ideal yaitu (Balasubramaniyan dkk, 2008), Bambang (2011): 1) Meminimalkan *overhead* sistem untuk tidak mengganggu operasi normal; 2) Mudah dikonfigurasi untuk disesuaikan dengan kebijakan keamanan sistem; 3) Mudah diinstalasi (*deploy*); 4) Mudah beradaptasi dengan perubahan sistem dan perilaku user, misal aplikasi atau *resource* baru; 5) Mampu memonitor sejumlah *host* dengan tetap memberikan hasil yang cepat dan tepat; 6) Dampak negatif yang minimal; 7) Memungkinkan konfigurasi dinamis, khususnya bila pemantauan dilakukan pada sejumlah besar *host*; 8) Berjalan secara kontinu dengan supervisi minimal dari manusia; 9) Mampu mendeteksi serangan: tidak salah menandai aktivitas yang legitimate (*false positive*), tidak gagal mendeteksi serangan sesungguhnya (*false negative*), segera melakukan pelaporan penyusupan yang terjadi, cukup general untuk berbagai tipe serangan; 10) Mampu *fault tolerant* dalam arti: bisa melakukan *recover* dari sistem yang *crash* baik secara insidental atau karena aktivitas tertentu, setelah itu bisa melanjutkan *state* sebelumnya tanpa mempengaruhi operasinya; 11) Mampu menolak usaha pengubahan: adanya kesulitan yang tinggi bila penyerang mencoba memodifikasinya, mampu memonitor dirinya sendiri dan mendeteksi bila dirinya telah dirubah oleh penyerang.

Kebanyakan *IDS* memiliki permasalahan sebagai berikut, Eugene (2008), (Balasubramaniyan dkk, 2008): 1) Tingkat sentralisasi. Kebanyakan deteksi dilakukan secara terpusat; 2) Konsumsi sumberdaya. Karena sentralisasi tersebut maka terjadi kebutuhan sumberdaya pemrosesan yang besar; 3) Batasan skalabilitas; 4) Masalah keamanan, misalnya *single point failure*; 5) Kesulitan untuk melakukan konfigurasi ulang atau penambahan kemampuan

2.7 Pemilihan *IDS*

IDS paling baik diimplementasikan dengan mengkombinasikan penggunaan solusi berbasis *host* dan *network*. Tahapan evaluasi umumnya terdiri dari tiga fase yaitu, (www.infolinux.web.id): Fase 1: Penentuan kebutuhan *IDS*: untuk mencakup aset penting dan kelengkapan dengan kebijakan keamanan. Tingkatan keamanan ini bisa mencakup perlindungan perimeter, aplikasi, *e-business*, server kunci, kebijakan dan perlindungan hukum. Hal ini harus diurutkan sesuai prioritas. Fase 2: Evaluasi solusi *IDS*: 1) Memilih produk yang tepat untuk memenuhi kebutuhan; 2) Pemahaman bagaimana tiap *IDS* mendeteksi penyusupan dalam jaringan; 3) Pemahaman bagaimana tiap produk menempatkan prioritas dan menjelaskan penyusupan pada jaringan, termasuk false positif yang dihasilkan; 4) Pemahaman kemampuan pelaporan dari tiap produk, kelengkapan, fleksibilitas dan penerapan teknisnya. Fase 3: Deployment *IDS*: penempatan solusi dalam organisasi secara efektif

Fleksibilitas *IDS* sendiri bisa didasarkan pada (Bace and Mell, 2005): 1) Kustomisasi: adaptasi *IDS* pada kebijakan tertentu dari organisasi; 2) *Deployment*: penempatan pada

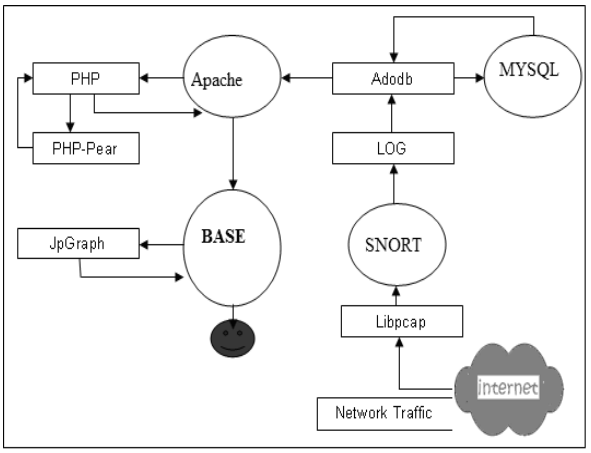
jaringan yang heterogen; 3) Skalabilitas manajemen.

2.8 *SNORT*

Snort tidak lain sebuah aplikasi atau tool sekuriti yang berfungsi untuk mendeteksi intrusi-intrusi jaringan (penyusupan, penyerangan, pemindaian dan beragam bentuk ancaman lainnya), sekaligus juga melakukan pencegahan. Istilah populernya, *snort* merupakan salah satu *tool Network Intrusion Prevention System* (*IPS*) dan *Network Intrusion Detection System* (*NIDS*). (Rafiudin, 2010).

2.9 Perancangan Sistem *Intrusion Detection System (IDS*)

Perancangan sistem yang akan digunakan untuk membangun *Intrusion Detection System*, terlebih dahulu yang dilakukan adalah mengumpulkan komponen yang akan digunakan sebagai *IDS*. *Intrusion Detection System* (*IDS*) yang akan dibangun adalah *IDS* yang bisa menyimpan *alert* dalam *database* dan setup otomatis jika komputer di hidupkan. Untuk mendapatkan *IDS* yang seperti itu dalam penelitian ini menggunakan beberapa komponen tambahan yang memudahkan user dalam menggunakan *IDS*. Komponen-komponen yang digunakan adalah: *snort*, *libpcap 0.8-dev, adodb, JpGraph, php5 dan mysql-server, php-pear, apache2*, *BASE* (program *ACID*)



Gambar 2. Arsitektur *IDS*

Pengembang lebih lanjut dari sistem *Intrusion Detection System*, memerlukan berbagai *tool* tambahan, sehingga *IDS* lebih *user friendly* sehingga *alert* lebih terorganisir dan mudah untuk dimengerti seperti digambarkan pada gambar 2. Libpcap mengirim *packet capture* ke *snort* untuk dianalisis oleh *system engine*, output plugin *snort* akan mengirim *alert* ke *database* yang mana *variable* dari *database* telah didefinisikan pada *snort config*. File *log* dan *alert* akan disimpan di dalam *database* pengimplementasian menggunakan *BASE*, tapi terlebih dahulu harus ada penghubung antara *database* dengan *web server* yaitu *adodb*. Untuk melihat *alert* pada *BASE* console dibutuhkan php sebagai penghubung ke *Basic Analysis Security Engine* (*BASE*).

2.10 Perancangan Penempatan *Intrusion Detection System (IDS)*

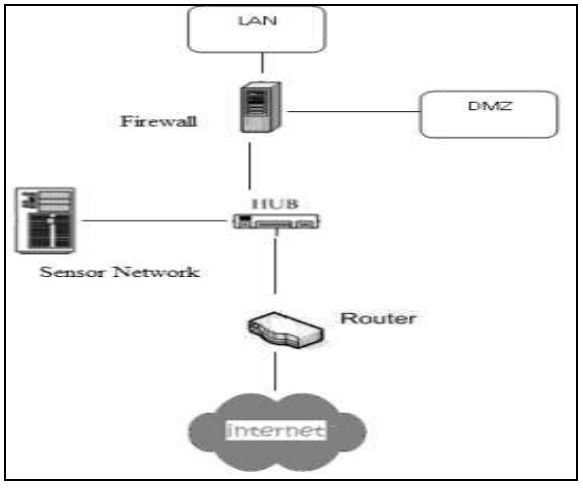
Intrusion Detection System (*IDS*) pada suatu jaringan akan dapat bekerja dengan baik, tergantung pada peletakannya. Secara prinsip pemahaman penempatan komponen *Intrusion*

Detection System (*IDS*) akan menghasilkan *IDS* yang benar-benar mudah untuk dikontrol sehingga pengamanan jaringan dari serangan menjadi lebih efisien (Ariyus: 2007).

2.10.1 Penempatan Sensor *Network* di Jaringan UPT-SIM

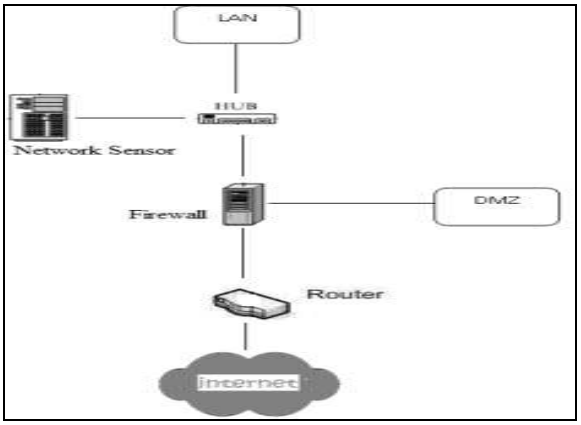
Sensor merupakan suatu komponen yang sangat penting dari suatu *Intrusion Detection System* (*IDS*). Oleh karena itu penempatannya benar-benar harus diperhatikan. *Sensor network* untuk *Intrusion Detection System* (*IDS*) biasanya terinstall pada lokasi berikut, (Ariyus, 2007):

1) Antara Router dan *Firewall*. Untuk melindungi jaringan dari serangan eksternal, fungsi *sensor network* sangat penting. Yang pertama dilakukan adalah menginstalasi *sensor network* diantara router dengan *firewall*. Sensor ini akan memberikan akses untuk mengontrol semua lalu lintas jaringan, termasuk lalu lintas pada *Demilitarized Zone*. (Ariyus: 2007)



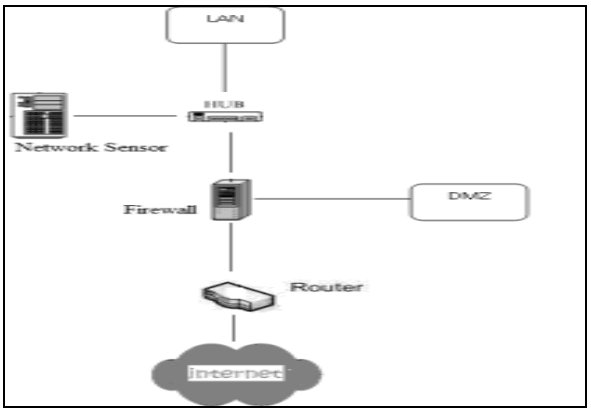
Gambar 3. Penempatan *Sensor Network* antara *Firewall* dan *Router*

2) Pada "*Demilitarized Zone*" (*DMZ*). Penempatan sensor pada lokasi ini untuk melindungi *Demilitarized Zone (DMZ)* yang meliputi *Web, FTP* dan *SMTP server*, external DNS server dan *host* yang diakses oleh *external user*. Sensor *IDS network* tidak akan menganalisis lalu-lintas jaringan jika tidak melewati *zona* yang dikontrol oleh suatu *IDS*, karena *IDS* juga mempunyai keterbatasan. (Ariyus: 2007)



Gambar 4. Penempatan *Sensor Network* pada *Demilitarized Zone* (*DMZ*)

3) Di belakang *firewall*. Sensor network bisa diletakkan di belakang *firewall*, bersebelahan dengan *LAN*. Keuntungan dari penempatan ini adalah bahwa semua lalu-lintas jaringan biasanya melintasi *firewall*. *Administrator* harus mengkonfigurasikan *sensor network* dan *firewall* dengan benar sehingga bisa melindungi jaringan secara maksimal. Dengan penempatan seperti ini administrator bisa mengontrol semua lalu-lintas *inbound* dan *outbound* pada *Demilitarized Zone*, karena semua lalu lintas jaringan akan berputar pada segment sebagai *gateway* jaringan. (Ariyus: 2007)



Gambar 5. Penempatan *Sensor Network* di Belakang *Firewall*

2.10.2 Analisis Kinerja *Intrusion Detection System*

Pada penelitian ini penulis melakukan analisis dari kinerja yang dilakukan *Intrusion Detection System* (*IDS*) yaitu untuk meningkatkan keamanan pada sistem seperti, merekam aktivitas yang tidak sah untuk digunakan untuk keperluan forensik atau *criminal prosecution* (tuntunan pidana) dari serangan penyusup. Banyak kemungkinan analisis data untuk analisis *engine* dan dalam rangka memahami proses yang terjadi, ketika data dikumpulkan dari sensor *Intrusion Detection System* (*IDS*) maka data diklasifikasikan dalam beberapa bentuk dimana tergantung pada skema analisis yang digunakan. Seperti jika *rule-based detection* atau *misuse detection* yang digunakan maka klasifikasi akan melibatkan aturan dan *pattern* (pola) dan jika *anomaly detection* yang digunakan maka akan selalu menggunakan algoritma yang berbeda untuk *baseline* dari waktu ke waktu untuk menganalisis apapun yang berasal dari luar jaringan yang tidak dikenal.

Dalam mengenali sebuah serangan yang dilakukan oleh *cracker* atau *hacker* dilakukan menggunakan data yang telah diperoleh. Dimana pada penelitian ini penulis melakukan pendekatan dengan menggunakan *misuse detection*, detektor melakukan analisis terhadap aktivitas sistem, mencari *event* atau *set event* yang cocok dengan pola perilaku yang dikenali sebagai serangan. Pola perilaku serangan tersebut disebut sebagai *signature*s, sehingga *misuse detection* banyak dikenal sebagai *signature*s based detection. Ada empat tahap proses analisis yang ada pada *misuse detector*: 1) *Preprocessing*, langkah pertama mengumpulkan data tentang pola dari serangan dan meletakkannya pada skema klasifikasi atau *pattern desciptor*. Dari skema klasifikasi, suatu model akan dibangun dan kemudian dimasukkan ke dalam bentuk format yang umum seperti: *Signature* Name: nama panggilan dari suatu tandatangan, *Signature* ID: ID yang unik, *Signature Description*: Deskripsi tentang tandatangan, Kemungkinan deskripsi yang palsu, Informasi yang berhubungan dengan *Vulnerability* (kerentanan): *field* yang berisi semua informasi tentang *Vulnerability*, *User Notes*: *field* ini mengijinkan *professional security* untuk menambahkan suatu catatan khusus yang berhubungan dengan jaringan.; 2) *Analysis*, data dan formatnya akan dibandingkan dengan *pattern* yang ada untuk keperluan analisis *engine pattern matching*. Analisis *engine* mencocokkan dengan pola serangan yang sudah dikenalnya; 3) *Response*, jika ada yang *match* (cocok) dengan pola serangan, analisis *engine* akan mengirimkan alarm ke server; 4) *Refinement* (perbaikan), perbaikan dari analisis *pattern-matching* yang diturunkan untuk memperbarui *signature*, karena *Intrusion Detection System* (*IDS*) hanya

mengijinkan tandatangan yang terakhir yang di-*update*.

2.11 Variabel dan Data Penelitian

Dalam penelitian ini *variable* dan data yang digunakan untuk kemudian diolah menjadi sebuah acuan adalah, (Ariyus:2007): 1) Paket *sniffer*: untuk melihat paket yang lewat di jaringan; 2) Paket *logger* : untuk mencatat semua paket yang lewat di jaringan untuk dianalisis dikemudian hari; 3) *NIDS*, deteksi penyusup pada network: untuk mendeteksi serangan yang dilakukan melalui jaringan komputer.

3. HASIL DAN PEMBAHASAN

3.1 Install Server I*ntrusion Detection System* (*IDS*)

Pada penelitian ini tahapan pertama yang harus dilakukan adalah menginstall semua komponen *Intrusion Detection System* (*IDS*) pada sebuah *PC* yang akan difungsikan sebagai server *Intrusion Detection System* (*IDS*).

3.2 Konfigurasi Server *Intrusion Detection System* (*IDS*)

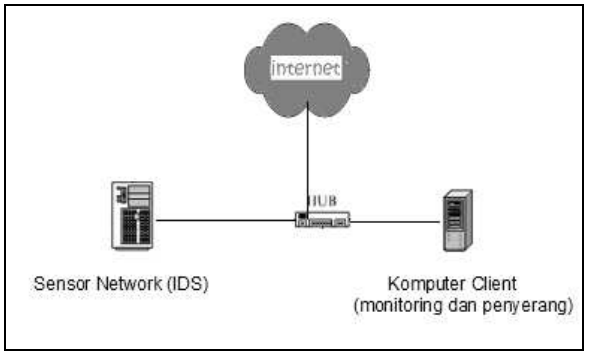
Setelah melakukan semua proses instalasi komponen *Intrusion Detection System* (*IDS*), maka tahap selanjutnya adalah tahap pengkonfigurasian, dimana pada penelitian ini sangat diperlukan untuk melakukan konfigurasi terhadap beberapa file yang sangat berpengaruh terhadap proses kerja dari sistem Server *Intrusion Detection System* (*IDS*) yang akan dibangun.

3.3 Implementasi Server *Intrusion Detection System* (*IDS*)

Pada penelitian ini Implementasi dari *Intrusion Detection System* (*IDS*) adalah mendeteksi kemungkinan *bad traffic* yang melintas suatu jaringan komputer. Fungsi dasar dari *Intrusion Detection System* (*IDS*) itu sendiri mengumpulkan kode-kode dari suatu paket yang polanya dikenali dari *rule* dan *signature* yang disimpan di dalam suatu *folder* dalam bentuk *file log* kemudian ditransfer ke *database* dengan menggunakan fasilitas *adodb*. *File log* yang tersimpan bisa dipelajari untuk melakukan antisipasi dikemudian hari, supaya yang telah terjadi tidak terulang kembali di kemudian hari. Agar *Intrusion Detection System* (*IDS*) lebih *friendly* dan *user interfaces* maka dibutuhkan komponen-komonen lain yang mendukungnya seperti : *PHP, PHP-pear, apache, Mysql, BASE, JPGraph, adodb*.

Pada tahapan implementasi pada penelitian ini yang harus dilakukan adalah melakukan pengujian terhadap *Intrusion Detection System* (*IDS*) yang telah dibangun dengan menggunakan Sistem Operasi Linux 10.04 server dan program *snort* sebelum di implementasikan langsung ke jaringan UPT-SIM Universitas Bina Darma. Adapun proses pengujian *Intrusion Detection System* (*IDS*) ini dilakukan dengan cara diantaranya adalah:

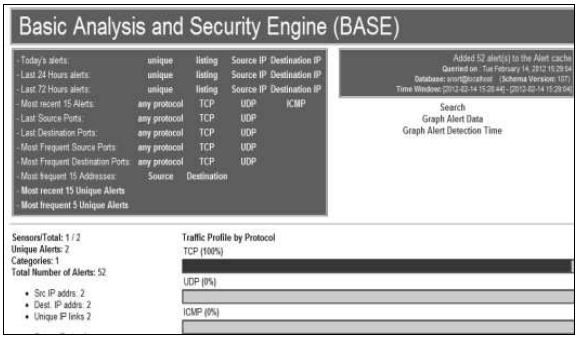
Melakukan perancangan jaringan yang terdiri dari komputer yang berfungsi sebagai server *Intrusion Detection System* (*IDS*), komputer client yang berfungsi sebagai monitoring dan penyerang



Gambar 6. Perancangan Jaringan Pengujian *IDS*

Setelah melakukan perancangan jaringan tahapan berikutnya adalah melakukan penyerangan terhadap *sensor network* (*IDS*) yang telah dibangun dengan melancarkan beberapa serangan seperti mengirimkan paket *ICMP* dalam ukuran besar sehingga dikategorikan oleh *Intrusion Detection System* (*IDS*) sebagai *DOS attack* (*Denial of Service*).

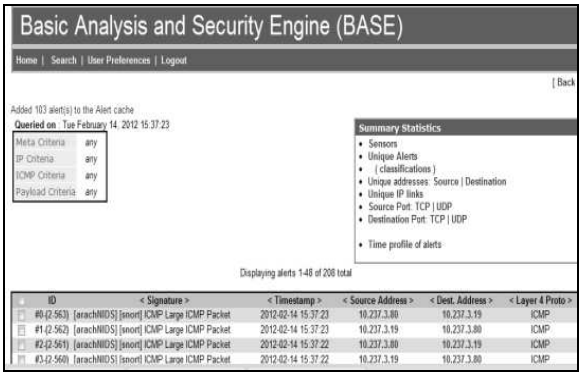
Lalu kemudian *DOS attack* ini akan segera terdeteksi oleh *snort engine* yang kemudian *snort engine* akan mengirimkan *alert* ke *alert log* dan kemudian ke *MySQL BASE*, untuk melihat hasil serangan yang terdeteksi maka *user admin* membuka aplikasi *BASE* melalui *web browser* seperti mozila dengan mengetikkan alamat http://10.237.3.80/base maka akan terlihat berapa persen jenis serangan yang masuk ke aplikasi *BASE* tersebut, dapat dilihat pada gambar 7 di bawah ini:



Gambar 7. Bentuk Serangan *DOS* pada Aplikasi *BASE*

Pada tahap selanjutnya untuk menganalisis jenis serangan yang terjadi *user admin* dapat melihat pada aplikasi *BASE* tersebut dengan mengklik pada bagian persen seperti pada protokol *TCP* terdapat 100% serangan yang terdeteksi maka akan tampil informasi dari serangan tersebut diantaranya: 1) ID adalah nomor identifikasi yang unik untuk *alert* yang terdeteksi oleh *snort*; 2) *Signature*: menunjukkan *link* dari *signature* yang merujuk dari jenis serangan yang terdapat pada *reference*; 3) *Timestamp*: waktu dan jam terjadinya suatu serangan; 4) *Source Address*: merupakan alamat *IP* dari sumber serangan; 5) *Destination Address*: merupakan alamat *IP* dari tujuan serangan; 6) *Layer 4 Protocol*: merupakan keterangan dari jenis protokol yang diserang.

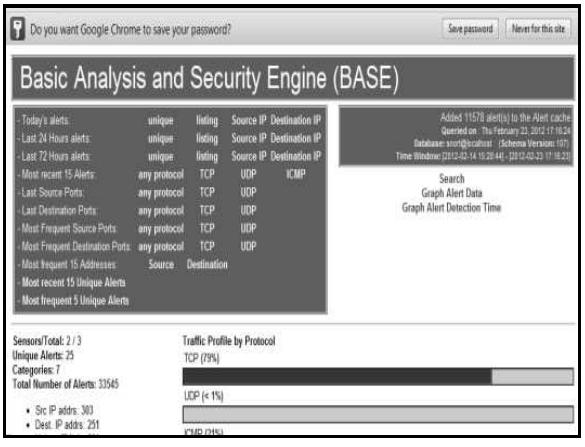
Untuk lebih jelas dapat dilihat seperti pada gambar 8 berikut :



Gambar 8. Analisis Bentuk Serangan

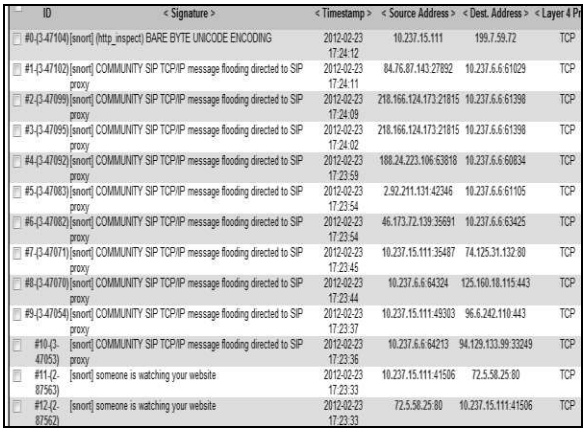
3.4 Pengujian Server *Intrusion Detection System* (*IDS*) di Jaringan UPT-SIM Universitas Bina Darma

Setelah melakukan pengujian terhadap server *Intrusion Detection System* (*IDS*) dengan melakukan beberapa serangan, sekarang saatnya melakukan pengujian langsung ke jaringan *VLAN* Universitas Bina Darma dengan meletakkan server *Intrusion Detection System* (*IDS*) pada jaringan server di UPT-SIM, dimana *sensor network* akan ditempatkan pada *Demilitarized Zone* (*DMZ*) penempatan sensor pada lokasi ini untuk melindungi *Demilitarized Zone* (*DMZ*) yang meliputi *Web, FTP, SMTP* server dan sebagainya. Langkah pertama yang harus dilakukan adalah meletakkan *PC network sensor* di jaringan *VLAN* UPT-SIM pada *Demilitarized Zone (DMZ)*, Kemudian melalui *PC client* penulis melakukan monitoring terhadap serangan yang terjadi dengan membuka alamat http://10.237.2.69/base seperti pada gambar 9 di bawah ini :



Gambar 9. Bentuk Serangan Pada Jaringan UPT-SIM

Selanjutnya adalah mengamati bentuk-bentuk serangan yang sudah terekam pada *database* aplikasi *BASE* seperti serangan melalui protokol *TCP, UDP ICMP* dan *Raw IP*. Beberapa bentuk serangan yang terjadi dapat dilihat pada gambar 10 dan 11 di bawah ini :



ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Pr
#0-(3-47104)	[snort] (http_inspect) BARE BYTE UNICODE ENCODING	2012-02-23 17:24:12	10.237.15.111	199.7.59.72	TCP
#1-(3-47102)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:24:11	84.76.87.143:27892	10.237.6.6:61029	TCP
#2-(3-47099)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:24:09	218.166.124.173:21815	10.237.6.6:61398	TCP
#3-(3-47095)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:24:02	218.166.124.173:21815	10.237.6.6:61398	TCP
#4-(3-47092)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:59	188.24.223.106:63818	10.237.6.6:60834	TCP
#5-(3-47083)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:54	2.92.211.131:42346	10.237.6.6:61105	TCP
#6-(3-47082)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:54	46.173.72.139:35691	10.237.6.6:63425	TCP
#7-(3-47071)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:45	10.237.15.111:35487	74.125.31.132:80	TCP
#8-(3-47070)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:44	10.237.6.6:64324	125.160.18.115:443	TCP
#9-(3-47054)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:37	10.237.15.111:49303	96.6.242.110:443	TCP
#10-(3-47053)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:23:36	10.237.6.6:64213	94.129.133.99:33249	TCP
#11-(2-87563)	[snort] someone is watching your website	2012-02-23 17:23:33	10.237.15.111:41506	72.5.58.25:80	TCP
#12-(2-87562)	[snort] someone is watching your website	2012-02-23 17:23:33	72.5.58.25:80	10.237.15.111:41506	TCP

Gambar 10. Serangan Melalui Protokol *TCP*

#11-(2-22307)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:10:11	10.237.4.3:32770	128.8.10.90:53	UDP
#12-(2-22306)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:09:07	10.237.15.111:46730	10.237.4.3:53	UDP
#13-(2-22304)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:08:46	10.237.4.3:53	10.237.15.111:49271	UDP
#14-(2-22303)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:08:26	10.237.15.111:20430	10.237.4.3:53	UDP
#15-(2-22301)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:07:40	10.237.4.3:32770	77.73.32.118:53	UDP
#16-(3-36)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:07:07	10.237.15.111:32791	10.237.4.3:53	UDP
#17-(2-22300)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:07:07	10.237.15.111:32791	10.237.4.3:53	UDP
#18-(3-34)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:06:37	10.237.4.3:53	10.237.6.6:1046	UDP
#19-(2-22298)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:06:37	10.237.4.3:53	10.237.6.6:1046	UDP
#20-(3-33)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:06:07	10.237.15.111:49401	10.237.4.3:53	UDP
#21-(2-22297)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:06:07	10.237.15.111:49401	10.237.4.3:53	UDP
#22-(3-19)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:05:54	10.237.4.3:32770	192.52.178.30:53	UDP
#23-(2-22283)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:05:54	10.237.4.3:32770	192.52.178.30:53	UDP
#24-(3-18)	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy	2012-02-23 17:04:13	10.237.15.111:56331	10.237.4.3:53	UDP
#25-(2-	[snort] COMMUNITY SIP TCP/IP message flooding directed to SIP	2012-02-23	10.237.15.111:56331	10.237.4.3:53	UDP

Gambar 11. Serangan Melalui Protocol UDP

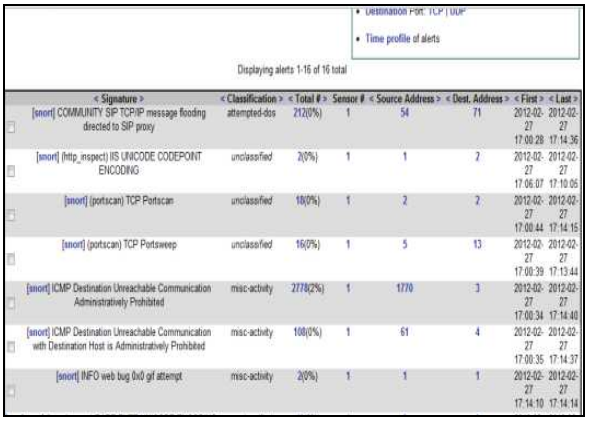
3.5 Analisis *Alert* melalui *BASE Console* pada Server *Intrusion Detection System* (*IDS*) di Jaringan UPT-SIM Universitas Bina Darma

Pada bagian informasi *alert* bisa didapatkan informasi tentang unique *alert* dan total number of number *alert*. Jika angka yang terdapat pada unique *alert* diklik maka akan tampil semua *alert* yang sudah diklasifikasikan. Hal ini dapat dilihat pada gambar 12.

Kesimpulan dari analisis *alert* melalui *BASE* console pada penelitian ini adalah serangan yang telah dikenali oleh *signature* dan *rule* pada server *Intrusion Detection System* (*IDS*) pada jaringan UPT-SIM Universitas Bina Darma diantaranya :

Tabel 2. Nama Tabel Bentuk Serangan

No	Bentuk Serangan
1.	*Portscan TCP Portsweep*
2.	*http_inspect BARE BYTE UNICODE ENCODING*
3.	*http_inspect OVERSIZE REQUEST-URI DIRECTORY*
4.	*Portscan ICMP Sweep*
5.	*ICMP Destination Unreacheable Communication with Destination Network is Administratively rohibited.*
6.	*(portscan) TCP Portscan*
7.	*(portscan) TCP Filtered Portscan*
8.	*Community SIP TCP/IP message flooding directed to SIP Proxy*
9.	*Someone is watching your website*
10.	*Community WEB-MISC Proxy Server Access*



Gambar 11. Nama Gambar *Unique Alert* dan *Number Alert*

Dari bentuk-bentuk serangan yang terjadi pada jaringan UPT-SIM diatas maka dapat disimpulkan beberapa persen serangan melalui protokol *TCP* (82%), *UDP* (1%), *ICMP* (16%) dan *Raw IP* (1%).

Untuk menghindari dari bentuk serangan diatas pada penelitian ini penulis memberikan solusi dengan cara, seperti pada bentuk serangan *flooding* maka di setiap server jaringan Universitas Bina Darma agar pada setiap server *firewall* melakukan proses pencegahan paket *flood syn Attack* dan paket *ping flood attack*. Kemudian untuk bentuk serangan *port scaning* yang terjadi agar melakukan pemblokiran terhadap *port-port* yang terbuka yang sudah dimasuki oleh penyusup melalui server *firewall*, selain itu juga dapat menggunakan perangkat lunak seperti *portsentry* dimana *portsentry* memiliki fitur diantaranya (Aulya, 2011): 1) Berjalan di atas soket *TCP & UDP* untuk mendeteksi scan *port* ke sistem; 2) Mendeteksi *stealth scan*, seperti *SYN/half-open*, *FIN, NULL, X-MAS*; 3) *PortSentry* akan bereaksi secara *real-time* (langsung) dengan cara memblokir *IP address* si penyerang. Hal ini dilakukan dengan menggunakan *ipchains/ipfwadm* dan memasukan ke file /etc/host.deny secara otomatis oleh *TCP Wrapper*; 4) *PortSentry* mempunyai mekanisme untuk mengingat mesin / *host* mana yang pernah *connect* ke sistem. Dengan cara itu, hanya mesin / *host* yang terlalu sering melakukan sambungan (karena melakukan *scanning*) yang akan di blokir; 5) *PortSentry* akan melaporkan semua pelanggaran melalui syslog dan mengindikasikan nama sistem, waktu serangan, IP mesin penyerang, *TCP / UDP port* tempat serangan dilakukan. Jika hal ini di integrasikan dengan *Logcheck* maka administrator sistem akan memperoleh laporan melalui *e-mail*.

4. SIMPULAN

Berdasarkan hasil penelitian dan pembahasan yang telah diuraikan pada bab –bab sebelumnya, sehingga dalam penelitian yang berjudul Implementasi *Intrusion Detection System* (*IDS*) di Jaringan Universitas Bina Darma Palembang maka didapatkanlah beberapa kesimpulan yang terdiri dari: 1) Serangan dapat terdeteksi atau tidak tergantung pola serangan tersebut ada didalam *rule IDS* (*Intrusion Detection System*) atau tidak. Oleh karena itu pengelola *Intrusion Detection System* (*IDS*) harus secara rutin meng-*update rule* terbaru;(2). Untuk mempermudah pengelolaan *rule* perlu *user interface* (*front end*) yang lebih baik seperti aplikasi *webmin* yang ditambahkan *plugin snort rule*; 3) Untuk mempermudah analisa terhadap catatan-catatan *Intrusion Detection System* (*IDS*) atau *security event* perlu ditambahkan program tambahan seperti *BASE* (*Basic Analysis and Security Engine*) atau *ACID* (*Analysis Console for Intrusion Database*s).

DAFTAR RUJUKAN

Arief, Rudyanto, Muhammad. 2010. *Penggunaan Sistem IDS (Intrusion Detection System) untuk Pengamanan Jaringan Komputer*, (Online), (http://rudy.amikom.ac.id, diakses tanggal 9 Oktober 2011).

Ariyus, Dony. 2007. *Intrusion Detection System*. Andi Offset. Yogyakarta.

Aulya, M. O. 2011. *Intrusion Detection System (Portsentry),* (Online), (http://www.psionic.com, diakses tanggal 20 September 2011).

Bace, Rebecca and Petter Mell. 2005. *Intrusion Detection System*. NIST Special Publication on *IDS*.

Bambang. 2011. *Kajian Aplikasi Mobile Agent untuk Deteksi Penyusupan pada Jaringan Komputer*. Yogyakarta.

Balasubramaniyan, Jai Sundar, Jose Omar, David Isacoff, and Diedo Samboni. 2008. *An Architecture for Intrusion Detection Using Autonomous Agent,, Center for Education and Research in Information Assurance and Security*. Departemen of Computer Sciences Purdue University. [Diakses 25 Oktober 2011].

Davison, R. M., Martinsons, M. G., Kock N. 2005. Journal: Information Systems, Journal: Principles of Canonical Action Research.

Eugene, Spafford. 2008. *A Framework and Prototype for Distributed Intrusion Detection System*. Departement od Computer Sciences Purdue University.

InfoLinux. 2011. *Sistem Pendeteksian Intrusi*, (Online), (http://www.infolinux.web.id, diakses 10 November 2011).

Internet *Security* Systems. 2011. *Network VS Host-based Intrusion Detection: A Guide to Intrusion Detection Technology*, (Online), (http://www.iss.net.net, diakses 5 November 2011).

Rafiudin, Rahmat. 2010. *Mengganyang Hacker dengan Snort*. Andi Offset. Yogyakarta.

Stiawan, Deris. 2009. *Intrusion Prevention System (IPS) dan Tantangan dalam Pengembangannya*. (Diakses 2 November 2011).

Thomas, Tom. 2005. *Networking Security First-Step*. Andi Offset. Yogyakarta.

Wiharjito, Tony. 2006. *Keamanan Jaringan Internet*. PT. Gramedia. Jakarta.