IMPLEMENTASI DAN UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTRUKTUR BERBASIS IDPS (INTRUSION DETECTION PREVENTION SYSTEM)

SKRIPSI

YUDHA KRISTANTO

0606078563

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM TEKNIK KOMPUTER

DEPOK

JUNI 2010

IMPLEMENTASI DAN UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTRUKTUR BERBASIS IDPS (INTRUSION DETECTION PREVENTION SYSTEM)

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

SKRIPSI

YUDHA KRISTANTO

0606078563

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

PROGRAM TEKNIK KOMPUTER

DEPOK

JUNI 2010

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,

dan semua sumber baik yang dikutip maupun dirujuk

telah saya nyatakan dengan benar.

Nama	**: Yudha Kristanto**
NPM	**: 06060078563**
Tanda Tangan	**: .....**
Tanggal	**: Juni 2010**

LEMBAR PENGESAHAN

Tugas akhir ini diajukan oleh :

Nama : Yudha Kristanto

NPM : 0706078563

Program Studi : Teknik Komputer

Judul Skripsi : **IMPLEMENTASI DAN ANALISA UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTRUKTUR BERBASIS IDPS (INTRUSION DETECTION AND PREVENTION SYSTEM)**

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : **Muhammad Salman, ST, MIT** **(_____)**

Penguji : **Ir Endang Sriningsih, MT , SI** **(_____)**

Penguji : **Prima Dewi Purnamasari, ST, MT, M.Sc** **(_____)**

Ditetapkan di : Depok

Tanggal : 1 Juli 2010

KATA PENGANTAR

Segala puji bagi Allah Subhanahu wa Ta'ala yang telah memudahkan untuk terselesaikannya tugas akhir ini. Shalawat serta salam semoga senantiasa dilimpahkan kepada Baginda Nabi Besar Muhammad Shalallahu 'alaihi wasalam, para salafussalih, dan insya Allah kepada kita semua.

Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan program pendidikan SI Universitas Indonesia pada program pendidikan Teknik Komputer. Judul dari seminar ini adalah :

IMPLEMENTASI DAN ANALISA UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTRUKTUR BERBASIS IDPS (INTRUSION DETECTION PREVENTION SYSTEM)

Skripsi ini disusun berdasarkan penelitian,percobaan, pengamatan dan analisa yang dilakukan oleh penulis .

Dengan terselesaikannya skripsi ini, penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Bapak Muhammad Salman. ST,MIT selaku pembimbing tugas akhir yang banyak memberikan masukan serta semangat dalam menyelesaikan seminar ini.
2. Ayah dan Bunda tercinta yang tak habis-habisnya memberikan support dan dorongan baik moril maupun materil.

3. Mercator Office And Multimedia Lab selaku tempat riset penulis dalam melakukan uji coba sistim yang dibangun oleh penulis.
4. Seorang yang selalu ada dihati yang tanpa disadari tak henti-hentinya memberikan semangat untuk terselesaikannya skripsi ini.
5. Monika Kusumawati Teman seperjuangan,senasib,sepenanggungan dalam grup riset IDS (*Intrusion Detection System*) di Mercator Office.
6. Yomma Hendra Putra ,Winda Actarina Teman yang besedia meminjamkan laptopnya untuk dijadikan target serangan.Semangat yah Insya Allah semester besok selesai.
7. Teman-teman angkatan Teknik Komputer 2006 Universitas Indonesia yang banyak membantu dan memberi dorongan untuk terselesaikannya skripsi ini.
8. Semua pihak yang tidak dapat disebutkan satu persatu

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna untuk itu dengan segala kerendahan hati, penulis menerima segala kritik dan saran yang membangun untuk melengkapi skripsi ini.

Akhir kata penulis mengharapkan semoga skripsi ini dapat bermanfaat dan berguna untuk kemajuan ilmu pengetahuan.

Jakarta , Juni 2010

Yudha Kristanto

Ku persembahkan Skripsi ini untuk Ayah dan Ibundaku yang kucintai

Ndoeng Waloeyo dan Margaretha yang banyak memberikan pengertian, kesabaran dan semangat untuk tidak pernah menyerah dalam menyelesaikan Skripsi ini.

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama	:Yudha Kristanto
NPM	: 0606078563
Program Studi	: Teknik Komputer
Departemen	: Teknik Elektro
Fakultas	: Teknik
Jenis karya	: Tugas akhir

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif *(Non-exclusive Royalty-Free Right)* atas karya ilmiah saya yang berjudul :

IMPLEMENTASI DAN ANALISA UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTRUKTUR BERBASIS IDPS (INTRUSION DETECTION PREVENTION SYSTEM)

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / formatkan, mengelola dalam bentuk pangkalan data *(database)*, merawat, dan mepublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya

Dibuat di : Depok

Pada tanggal : Juni 2010

Yang menyatakan

ABSTRAK

Nama **:Yudha Kristanto**

Program Studi **: Teknik Komputer**

Judul **:**IMPLEMENTASI DAN ANALISA UNJUK KERJA KEAMANAN JARINGAN PADA INFRASTUKTUR BERBASIS IDPS (INTRUSION DETECTION AND PREVENTION SYSTEM)

ABSTRAK

Dalam melakukan pengembangan jaringan saat ini keamanan jaringan adalah suatu bagian yang amat penting yang harus diperhatikan, Keamanan jaringan komputer sebagai bagian dari sebuah sistem sangat penting untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya.Dimana sebuah sistem harus dilindungi dari segala macam serangan dan bentuk usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak. Untuk itu diperlukannya sebuah pengembangan sistim penanaganan terhadap bahaya serangan yang dilakukan oleh orang yang tidak berhak didalam jaringan.

Perancangan IDPS (Intrusion Detection Prevention System)yang digunakan pada penulisan ini adalah perancangan yang berbasiskan pada software SNORT yang awalnya berupa IDS (Intrusion Detection System) kemudian dikembangkan menjadi software IDPS (Intrusion Detection Prevention System) yang memiliki kemampuan dalam melakukan prevention terhadap jaringan. Yang dapat menahan menahan pengujian yang dilakukan oleh yaitu : Ip Scanning , Port Scanning ,OS Finger Printing, Vulnerability Scanning dan Flooding

Kata Kunci : SNORT, IDS ,IDPS

ABSTRACT

Name : Yudha Kristanto

Study Program : Computer Engineering

Title : IMPLEMENTATION AND SECURITY PERFORMANCE ANALYSIS on IDPS (INTRUSION DETECTION PREVENTION SYSTEM) BASED NETWORK SECURITY INFRASTRUCTURE

ABSTRACT

In developing this network when network security is a very important part that must be considered, computer network security as part of a system is very important to maintain the validity and integrity of data and ensure availability of services for user. When a system must be protected from all kinds of attacks and forms of intrusion attempts or scanning by unauthorized parties. Therefore the need for a development system to the danger handling attacks carried out by unauthorized people in the network.

The design of IDPs (Intrusion Detection Prevention System) used in this paper is based on the software design which was initially in the form of Snort IDS (Intrusion Detection System) and then developed into a software IDPs (Intrusion Detection Prevention System) which has the ability to do prevention on the network. That can withstand tests conducted by that is: Ip Scanning, Port Scanning, OS Finger Printing, Scanning Vulnerability and Flooding

Key words : SNORT, IDS ,IDPS

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Dalam melakukan pengembangan jaringan saat ini keamanan jaringan adalah suatu bagian yang amat penting yang harus diperhatikan, Keamanan jaringan komputer sebagai bagian dari sebuah sistem sangat penting untuk menjaga validitas dan integritas data serta menjamin ketersediaan layanan bagi penggunanya. Sebuah sistem harus dilindungi dari segala macam serangan dan bentuk usaha-usaha penyusupan atau pemindaian oleh pihak yang tidak berhak.

Dengan semakin banyaknya cara untuk melakukan pengaksesan terhadap data dan semakin berkembangnya teknologi yang di gunakan tentunya akan menyebabkan meningkatnya ancaman keamanan terhadap suatu jaringan. Hal ini tentunya sangat berbahaya terutama pada sektor-sektor yang memiliki tingkat keamanan data yang sensitif seperti perbankan. Untuk itulah diperlukan sebuah perhatian khusus dalam bidang keamanan jaringan yang bertujuan untuk mencegah terjadinya pencurian data-data perusahaan.

Untuk itu diperlukan sebuah perancangan sistem yang dapat mengamankan jaringan tersebut dari ancaman pencurian data-data perusahaan. Namun. Kebanyakan dari sistem yang diterapkan biasanya hanya berupa IDS (*Intrusion Detection System*) atau IPS (*Intrusion Prevention System*) yang mengakibatkan tidak maksimalnya tingkat keamanan jaringan pada suatu perusahaan yang hanya mampu mendeteksi berbagai jenis serangan tetapi tidak mampu mengambil tindakan lebih lanjut. Hal ini sangat berbahaya sehingga diperlukan sebuah sistem yang mampu menjadi IDS sekaligus menjadi IPS sehingga seorang administrator jaringan tidak harus memantau setiap kejadian yang berkaitan dengan adanya sebuah serangan.

Untuk itulah dikembangkan suatu sistem *StrataGuard* yang sudah ada menjadi sebuah sistem yang lebih efektif sehingga dapat digunakan tidak hanya sebagai sebuah IDS namun juga sekaligus sebagai IPS.Sistem ini sangatlah menarik untuk dikembangkan karena sistem yang berbasiskan linux ini memiliki fitur yang memudahkan seseorang administrator jaringan untuk melakukan konfigurasi aturan-aturan yang akan diterapkan pada jaringan, tidak seperti perangkat IDS atau IPS yang lainnya yang dalam melakukan konfigurasinya harus menggunakan terminal.

Perancangan sistem ini akan memberikan suatu yang baru dalam melakukan penanganan terhadap keamanan jaringan yaitu membangun suatu IDPS yang mampu menangani kondisi jaringan.

1.2. TUJUAN PENELITIAN

Melalui Skripsi ini diharapkan dapat membangun suatu sistem yang aman dengan mengunakan IDPS *StrataGuard* yang berbasiskan *web monitoring* dengan melakukan pengukuran terhadap performa *respon time* dan Unjuk Kerja.

1.3. PEMBATASAN MASALAH

Masalah yang terdapat pada skripsi ini dibatasi hanya pada :

1. Perfoma *respon time* dan unjuk kerja IDPS dalam menangani sebuah serangan.
2. Mengimplementasikan suatu sistem keamanan jaringan yang handal dan memiliki GUI (*General User Interface*) yang baik

1.4. METODE PENELITIAN

Metode penelitian yang digunakan pada skripsi ini adalah :

1. **Studi literatur dan pustaka,**

   Dengan melakukan berbagai diskusi pembahasan baik dengan dosen pembimbing maupun dengan orang yang berkompeten pada kasus ini serta dari pustaka yang mendukung.

2. **Pendefinisian masalah dan kebutuhan sistem.**
3. **Analisa dan perancangan sistem,**

   meliputi tahapan terstruktur sebagai berikut :

   a. Perancangan sistem dengan menggunakan program *StrataGuard*.
   b. Perancangan *interface* untuk menampilkan hasil dari sistem.
   c. Implementasi dan Uji Coba

4. **Implementasi perancangan Sistem,**

   Sistem yang akan diimplementasikan adalah sistem yang menggunakan program *Strata Guard,* yaitu sistem yang dapat mendeteksi adanya serangan yang masuk ke dalam jaringan berdasarkan IP *address* asal dan IP *address* tujuan dan melakukan pengamanan jaringan.

5. **Uji Coba dan Evaluasi Sistem,**

   Melakukan uji coba dan mengevaluasi sistem yang telah diimplementasikan.

6. **Mengambil kesimpulan,**

   Apakah sistem *monitoring* yang ada memiliki performansi (kinerja) yang baik dan tingkat keamanan yang baik.

1.5. SISTEMATIKA PENULISAN

Sistematika penulisan skripsi ini dibagi menjadi beberapa bab yang meliputi:

BAB I Pendahuluan

Bab ini berisi tentang latar belakang, tujuan penulisan, batasan masalah, metologi penelitian yang dipakai dalam penelitian, dan sistematika penulisan yang memuat susunan penulisan Skripsi ini.

BAB II Landasan Teori

Bab ini membahas definisi-definisi dan konsep-konsep dasar yang digunakan dalam penelitian ini, meliputi teori *IPS (Intrusion Prevention System), IDS (Intrusion Detection System)* dan, *Snort*

BAB III Perancangan Sistem

Bab ini berisi perancangan dari sistem yang diharapkan dapat bertindak sebagai IPS dan IDS.

BAB IV Implementasi Sistem

Bab ini berisi implementasi, uji coba dan analisa sistem keamanan jaringan yang dibuat

BAB IV Kesimpulan dan Saran

Bab ini berisi tentang kesimpulan yang di dapat dari hasil penelitian yang dilakukan serta saran untuk pengembangan lebih lanjut.

BAB II

LANDASAN TEORI

2.1. Intrusion Detection System (IDS)

Intrusion Detection System (IDS) merupakan suatu sistem yang mampu melakukan pendeteksian terhadap suatu gangguan, yang berupa penyerangan. Adapun tipe pendeteksian gangguan ini terbagi menjadi dua bagian, yaitu pertama *Host Based Attack Detection* (HBAD) atau pendeteksian gangguan yang dilakukan terhadap suatu komputer tempat disimpannya aplikasi ini. Kedua adalah *Network Based Attack Detection* (NBAD), biasa dikatakan sebagai pendeteksian yang dilakukan terhadap suatu kondisi dengan lingkup jaringan, misalnya saja gangguan yang dilakukan oleh aplikasi paket *sniffer*.Sistem yang berbasis IDS hanya mampu mendeteksi saja namun kelanjutan aksi yang akan dilakukan terhadap suatu gangguan tergantung pada aplikasi lainnya, misalnya saja *Firewall*, dengan menggunakan perangkat ini aksi akan lebih lengkap dan berkelanjutan, sehingga dapat terhindar dari setiap gangguan. Umumnya aplikasi IDS mampu memberikan suatu indikasi apabila ada gangguan untuk kemudian disampaikan ke administrator agar dapat dilakukan aksi terhadap gangguan tersebut**.**

Pada sistem yang berbasis IDS, hanya dapat mendeteksi suatu serangan saja namun untuk menindak lanjutinya tergantung pada aplikasi lainnya. Pada umumnya aplikasi tersebut adalah *firewall*, dimana dengan menggunakan perangkat ini keputusan yang diambil dapat lebih sempurna yang akan menyebabkan dapat terhindar dari berbagai macam gangguan.

Berdasarkan cara melakukan analisa dari suatu gangguan maka, IDS dapat dibagi atas dua kategori yaitu :

1. *Misuse Detection Model*, merupakan sistem deteksi yang melihat suatu aktivitas yang secara jelas dianggap sebagai *Pattern Signature* suatu gangguan dan biasanya deteksi ini cenderung untuk kondisi internal sistem.
2. *Anomaly Detection Model*, merupakan sistem deteksi yang cenderung melihat suatu gangguan karena suatu aktifitas yang dianggap sebagai kondisi yang tidak normal dan biasanya untuk kondisi yang disebabkan oleh eksternal sistem.

metode *Misuse Detection*, dimana pola-pola gangguan selalu di*update* oleh pihak *vendor* pembuat aplikasi tersebut. Bila dibandingkan dengan *Anomaly Detection Model*, metode ini mampu melakukan analisa tanpa melihat secara spesifik terhadap pola suatu gangguan dan terkadang melakukan suatu kesalahan akibat sensitifitas yang ditimbulkan

2.2. Intrusion Prevention System (IPS)

Intrusion Prevention System adalah sebuah perangkat monitoring jaringan komputer atau aktivitas sistem terhadap *malicious* atau kebiasaan yang tidak biasa. Sistem ini adalah sistem yang *realtime* untuk memblok atau mencegah suatu aktivitas yang membahayakan jaringan. IPS beroperasi dalam jaringan. *Intrusion Prevention System* (IPS) memiliki kemampuan lebih lengkap dibandingkan dengan IDS yang hanya mampu mendeteksi adanya penyusupan dalam jaringan, kemudian mengaktifkan peringatan kepada pengguna untuk segera mengambil langkah-langkah pencegahan berbeda dengan IPS yang dapat langsung mengatasi penyusupan tersebut.

Pada awalnya, pasar saat itu memandang skeptis terhadap keberhasilan teknologi IPS yang menggunakan filter dalam menangkal serangan dan penyusupan. Saat itu aktif IDS, merupakan teknologi asal mula dari IPS yang mampu secara aktif mendeteksi serangan dan mengubah aturan *firewall* dan *router* untuk mengantisipasi serang, dinilai mengganggu, sehingga tidak diterima oleh administrator jaringan. Namun, pada perkembangannya frekuensi serangan terhadap jaringan meningkat sementara rentang waktu antara ditemukannya celah keamanan dan tersedia *patch* untuk menutup celah itu semakin sempit. Yang tentu saja mengakibatkan para administrator jaringan tidak memiliki cukup banyak waktu untuk mengantisipasi serangan dengan memasang *patch*.

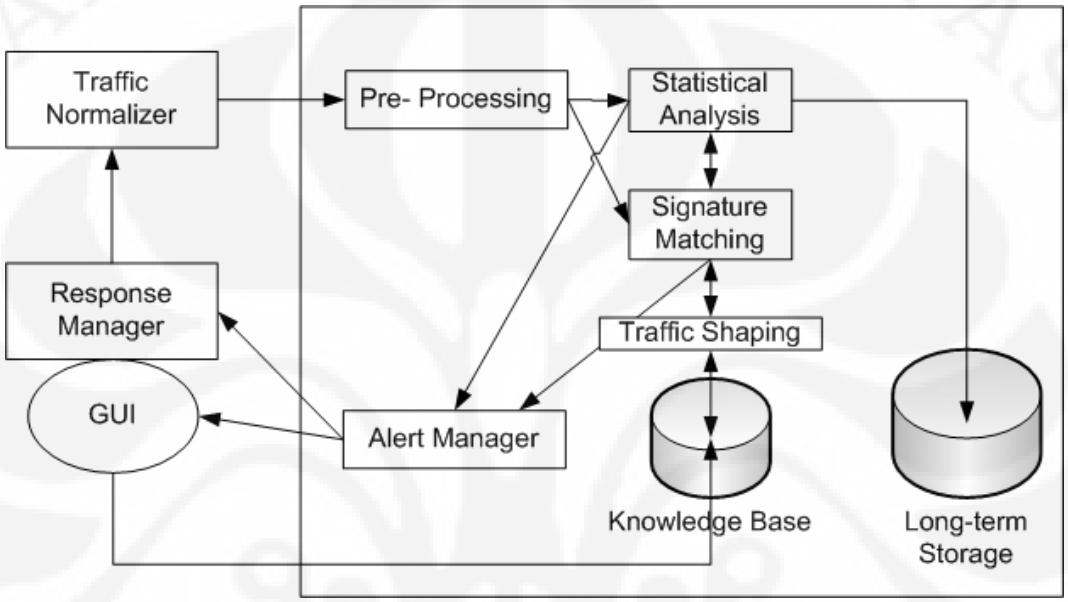
Sistem *setup* pada IPS sama dengan sistem *setup* pada IDS. IPS bisa sebagai *client*-based IPS (HIPS) yang bekerja untuk melindungi aplikasi dan juga sebagai *network based* IPS (NIPS). IPS tentunya lebih unggul daripada IDS hal ini desebabkan Karena IPS mampu mencegah serangan yang datang dengan bantuan administrator secara minimal atau bahkan tidak sama sekali. Tidak seperti IDS, secara *logic* IPS akan menghalangi suatu serangan sebelum terjadi eksekusi pada memori, metode lain dari IPS membandingkan *file checksum* yang tidak semestinya *dengan file checksum* yang semestinya mendapatkan izin untuk dieksekusi.

Pada dasarnya IPS memiliki empat komponen utama:

- *Normalisasi traffic*
- *Services scanner*
- *Detection engine*
- *Traffic shaper*

Normalisasi *traffic* akan menginterpretasikan lalu-lintas jaringan dan melakukan analisis terhadap paket yang disusun kembali, seperti halnya fungsi blok sederhana. Lalu-lintas paket bisa dideteksi dengan *detection engine* dan *service scanner*. *Service scanner* membangun suatu tabel acuan untuk mengelompokkan

informasi dan membantu pembentukan lalu-lintas serta mengatur lalu-lintas informasi. *Detection engine* melakukan *pattern matching* terhadap tabel acuan dan *respons* yang sesuai.[1]



Gambar 2.1. Sistem IPS Standar [1]

Teknologi IDS dan IPS masing-masing mempunyai kemampuan dalam melindungi suatu sistem. Teknologi IPS merupakan teknologi yang diperbarui dari IDS. Perbedaan dari kedua program itu adalah seperti pada gambar dibawah ini

Tabel 2.1. Perbandingan IDS dan IPS

IDS	IPS
Install pada segmen jaringan (NIDS) dan pada *client* (HIDS)	Install pada segmen jaringan (NIPS) dan pada *client* (HIPS)
Berada pada jaringan sebagai sistem yang pasif	Berada pada jaringan sebagai sistem yang aktif
Tidak bisa menguraikan lalu-lintas enkripsi	Lebih baik untuk melindungi aplikasi
Managemen control terpusat	Managemen control terpusat
Baik untuk mendeteksi serangan	Ideal untuk memblokir serangan
Alerting (reaktif)	Blocking (proaktif)

2.3. JENIS SERANGAN

Pada dasarnya jenis-jenis serangan yang mengganggu dalam jaringan komputer amat beragam namun dapat dikelompokkan dalam beberapa jenis yaitu :

- **Back Orifice (BO)** adalah sebuah alat bantu *remote* administrasi komputer dari jarak jauh yang dapat digunakan untuk mengontrol sistem operasi Microsoft Windows, yang dikembangkan oleh kelompok peretas profesional *Cult of the Dead Cow*. *Back Orifice* dirilis pertama kali untuk *platform* Windows NT pada tahun 1997. Namanya merupakan pelesetan dari Microsoft BackOffice *Server*. Pada tahun 1999, grup yang sama merilis versi baru, yang disebut sebagai *Back Orifice* 2000 atau sering disebut BO2K. Meskipun pada dasarnya alat bantu ini merupakan salah satu bentuk dari Trojan horse, yang dapat digunakan untuk mendapatkan akses dan kontrol penuh terhadap mesin target, program mini menawarkan banyak fitur, khususnya untuk mengendalikan sistem operasi Windows NT. Tampilan yang digunakannya sangatlah mudah dan sederhana, sehingga para peretas pemula pun dapat menggunakannya.

- **Denial of Service (DOS)** adalah jenis serangan terhadap sebuah komputer atau *server* di dalam jaringan internet dengan cara menghabiskan *resource* yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut. Dalam sebuah serangan *Denial of Service (DoS),* penyerang akan mencoba untuk mencegah akses seorang pengguna terhadap sistem atau jaringan dengan menggunakan beberapa cara, yakni sebagai berikut:
  - Membanjiri lalu lintas jaringan dengan banyak data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk ke dalam sistem jaringan. Teknik ini disebut sebagai *traffic flooding*.
  - Membanjiri jaringan dengan banyak *request* terhadap sebuah layanan jaringan yang disediakan oleh sebuah *client* sehingga *request* yang datang dari pengguna terdaftar tidak dapat dilayani oleh layanan tersebut. Teknik ini disebut sebagai *request flooding*.
  - Mengganggu komunikasi antara sebuah *client* dan kliennya yang terdaftar dengan menggunakan banyak cara, termasuk dengan mengubah informasi konfigurasi sistem atau bahkan perusakkan fisik terhadap komponen dan *server*.
- ***Port scan*ning:** merupakan suatu proses untuk mencari dan membuka port pada suatu jaringan computer. Dari hasil scanning akan didapat letak kelemahan sistem tersebut. Pada dasarnya sistem *port scanning* mudah untuk dideteksi, tetapi penyerang akan menggunakan berbagai metode untuk menyembunyikan serangan.
- **Teardrop:** Merupakan suatu teknik yang dikembangkan dengan mengekploitasi proses *assembly-reassembly* paket data. Dalam jaringan internet seringkali data harus dipotong kecil-kecil untuk menjamin reablitas dan proses multiple akses jaringan. Potongan paket data ini kadang harus

dipotong ulang menjadi lebih kecil lagi pada saat disalurkan melalui saluran *Wide Area Network (WAN)* agar pada saat melalui saluran WAN yang tidak reliable. Pada proses pemotongan data paket yang normal, setiap potongan diberi informasi *offset* data yang kira-kira berbunyi "potongan paket ini merupakan potongan 600byte dari total 800 byte paket yang dikirim. Program *teardrop* akan memanipulasi *offset* potongan data sehingga akhirnya terjadi overlapping antara paket yang diterima di bagian penerima, setelah potongan paket ini di *reassembly* seringkali *overlapping* ini menimbulkan sistem yang *crass, hang,* dan *reboot* di penerima.

- **IP-Spoofing:** adalah suatu serangan teknis yang rumit yang terdiri dari beberapa komponen. Ini adalah eksploitasi keamanan yang bekerja dengan menipu komputer, seolah-olah yang menggunakan komputer tersebut adalah orang lain. Hal ini terjadi karena *design flaw* (salah rancang). Lubang keamanan yang dapat dikategorikan ke dalam kesalah desain adalah desain urutan nomor *sequence numbering* dari paket TCP/IP. Kesalahan ini dapat dieksploitasi sehingga timbul masalah.
- **Smurft Attack:** Serangan jenis ini biasanya dilakukan dengan menggunakan IP *spoofing*, yaitu mengubah nomor IP dari datangnya *request*. Dengan menggunakan IP *spoofing*, respons dari ping tadi dialamatkan ke komputer yang IP-nya di*spoof*. Akibatnya, komputer tersebut akan menerima banyak paket. Hal ini dapat mengakibatkan pemborosan *bandwith* jaringan yang terhubung dengan komputer tersebut.
- **UDP *Flood*:** Pada dasarnya mengaitkan dua sistem tanpa disadari. Dengan cara *spoofing, User Datagram Protocol* (UDP) *flood attack* akan menempel pada servis UDP chargen di salah satu mesin yang digunakan untuk keperluan "percobaan" akan mengirimkan sekelompok karakter ke mesin lain, yang diprogram untuk meng-echo setiap kiriman karakter yang diterima melalui *service chargen*. Karena paket UDP tersebut di *spoofing* di antara ke dua mesin tersebut maka yang terjadi adalah banjir tanpa henti kiriman karakter yang tidak berguna diantara kedua mesin. Untuk mengulangi *UDP flood*, anda

dapat men*disable* semua *service* UDP di semua mesin dijaringan, atau yang lebih mudah adalah dengan memfilter pada *firewall* semua *service* UDP yang masuk.

- **ICMP *flood*:** Seorang penyerang melakukan eksploitasi sistem dengan tujuan untuk membuat suatu target *client* menjadi *crash*, yang disebabkan oleh pengiriman sejumlah paket yang besar ke arah target *client*. *Exploting* sistem ini dilakukan dengan mengirimkan suatu perintah ping dengan tujuan *broadcast* atau *multicast* di mana si pengirim dibuat seolah-olah adalah target *client*. Semua pesan balasan dikembalikan ke target *client*. Hal inilah yang membuat target *client* menjadi crash dan menurunkan kinerja jaringan. Bahkan hal ini dapat mengakibatkan *denial of service*

2.3.1. Klasifikasi Serangan

Berikut adalah tabel klasifikasi serangan yang nantinya menjadi *priority* di dalam *snort. Priority 1 = high, priority 2 = medium dan priority 3 = low.*[2] Klasifikasi serangan dapat diubah sesuai keinginan administrator. Untuk mengubah klasifikasi serangan dengan mengubah isi pada *file classification.conf.*

Tabel 2.2.Klasifikasi Serangan Berdasarkan Tingkat Prioritas[2]

Classtype	Description	Priority
attempted-admin	Attempted Administrator Privilege Gain	high
attempted-user	Attempted User Privilege Gain	high
kickass-porn	SCORE! Get the lotion!	high
policy-violation	Potential Corporate Privacy Violation	high
shellcode-detect	Executable code was detected	high
successful-admin	Successful Administrator Privilege Gain	high
successful-user	Successful User Privilege Gain	high
trojan-activity	A Network Trojan was detected	high
unsuccessful-user	Unsuccessful User Privilege Gain	high
web-application-attack	Web Application Attack	high
attempted-dos	Attempted Denial of Service	medium
attempted-recon	Attempted Information Leak	medium
bad-unknown	Potentially Bad Traffic	medium
default-login-attempt	Attempt to login by a default username and password	medium
denial-of-service	Detection of a Denial of Service Attack	medium
misc-attack	Misc Attack	medium
non-standard-protocol	Detection of a non-standard protocol or event	medium
rpc-portmap-decode	Decode of an RPC Query	medium
successful-dos	Denial of Service	medium
successful-recon-largescale	Large Scale Information Leak	medium
successful-recon-limited	Information Leak	medium
suspicious-filename-detect	A suspicious filename was detected	medium
suspicious-login	An attempted login using a suspicious username was detected	medium
system-call-detect	A system call was detected	medium
unusual-client-port-connection	A client was using an unusual port	medium
web-application-activity	Access to a potentially vulnerable web application	medium
icmp-event	Generic ICMP event	low
misc-activity	Misc activity	low
network-scan	Detection of a Network Scan	low
not-suspicious	Not Suspicious Traffic	low
protocol-command-decode	Generic Protocol Command Decode	low
string-detect	A suspicious string was detected	low
unknown	Unknown Traffic	low
tcp-connection	A TCP connection was detected	very low

2.4. Snort

Snort IDS merupakan IDS *open source* yang secara umum menjadi standar IDS di dunia keamanan jaringan. Snort dapat diimplementasikan dalam jaringan yang *multiplatform*, salah satu kelebihannya adalah mampu mengirimkan *alert* dari mesin Unix atupun Linux ke platform Microsoft Windows dengan melalui SMB. Pada dasarnya Snort dapat berkerja dalam 3 mode:

- *Sniffer mode (penyadap)*: untuk melihat paket yang lewat di jaringan.
- *Packet logger*: untuk mencatat semua paket yang lewat di jaringan untuk dianalisa.
- *Network Intrusion Detection* (NIDS) mode: pada mode ini snort akan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer. Untuk menggunakan mode IDS ini diperlukan *setup* dari berbagai *rules* atau aturan yang akan membedakan sebuah paket normal dengan paket yang membawa serangan.

2.4.1. Komponen – Komponen Snort

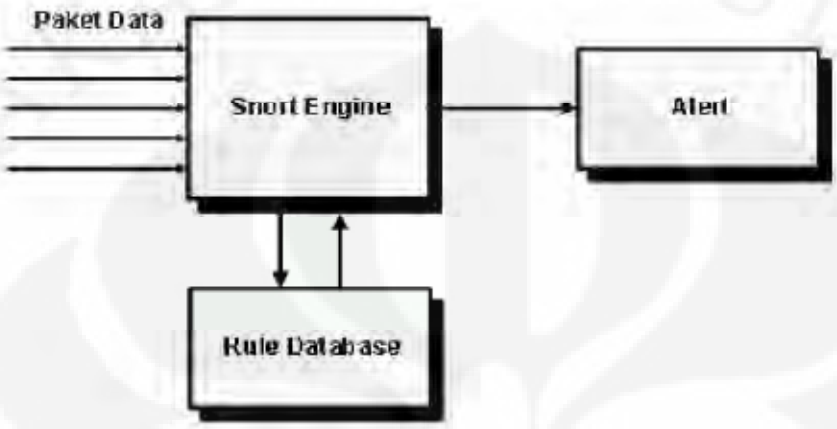
Snort mempunyai lima komponen dasar yang bekerja saling berhubungan satu dengan yang lain seperti berikut:[3]

1. **Decoder**: sesuai dengan paket yang di-*capture* dalam bentuk struktur data dan melakukan identifikasi protokol, *decode* IP dan kemudian TCP atau UDP tergantung informasi yang dibutuhkan, seperti *port number, IP address*. Snort akan memberikan *alert* jika menemukan paket yang cacat.
2. **Preprocessors**: Merupakan suatu saringan yang mengidentifikasi berbagai hal yang harus diperiksa seperti *Detection Engine.* Pada dasarnya *preprocessors* berfungsi mengambil paket yang mempunyai potensi berbahaya yang kemudian dikirim ke *detection engine* untuk dikenali polanya.

   Example: HTTPInspect

HTTPInspect menggantikan http_decode sebagai *preprocessor* yang bertanggung jawab untuk mendecodekan lalu-lintas http dan mendeteksi lapisan aplikasi serangan eksploit http design atau implementasi. Hal ini akan terlihat dalam buffer data paket yang berusaha mencari celah-celah di dalam lalu lintas http dan berusaha melakukan normalisasi data.

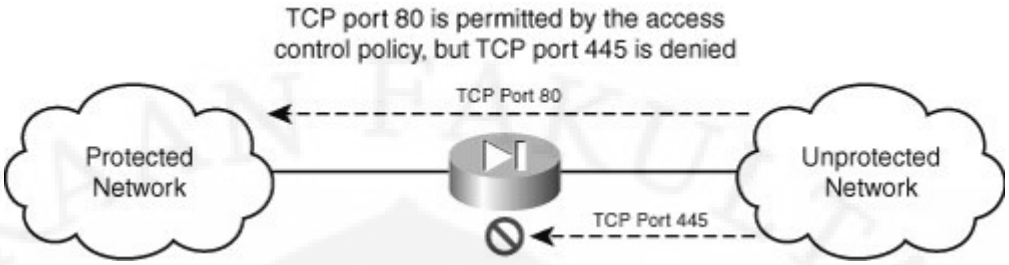
3. **Rules Files:** Merupakan suatu file teks yang berisi daftar aturan sintaks-nya sudah diketahui. Sintaks ini meliputi protokol, *address, output plug-ins* dan hal-hal yang berhubungan dengan berbagai hal. *Rules* file akan selalu diperbaharui setiap ada kejadian di dunia maya. *Rule* snort lebih dari 100 ribu tipe. Setiap hari bisa diupdate melalui situs resmi snort www.snort.org atau dari forum yang disediakan oleh komunitas snort.
4. **Detection Engine:** Menggunakan *detection plug-ins*, jika ditemukan paket yang cocok maka snort akan menginisialisasi paket tersebut sebagai suatu serangan.
5. **Output Plug-ins**: Merupakan suatu modul yang mengatur format dari keluaran untuk *alert* dan file logs yang biasa diakses dengan berbagai cara seperti *console, extern file, database* dan sebagainya.
6. **Alert:** merupakan catatan serangan pada deteksi penyusupan. Jika *snort engine* menilai paket data yang lewat sebagai serangan, maka *snort engine* akan mengirimkan *alert* berupa *log file*. Untuk kebutuhan analisa, alert dapat disimpan di dalam *database*, sebagai contoh BASE *(Basic Analys Security Engine*) sebagai modul tambahan pada Snort.



Gambar 2.2. Hubungan komponen Snort [3]

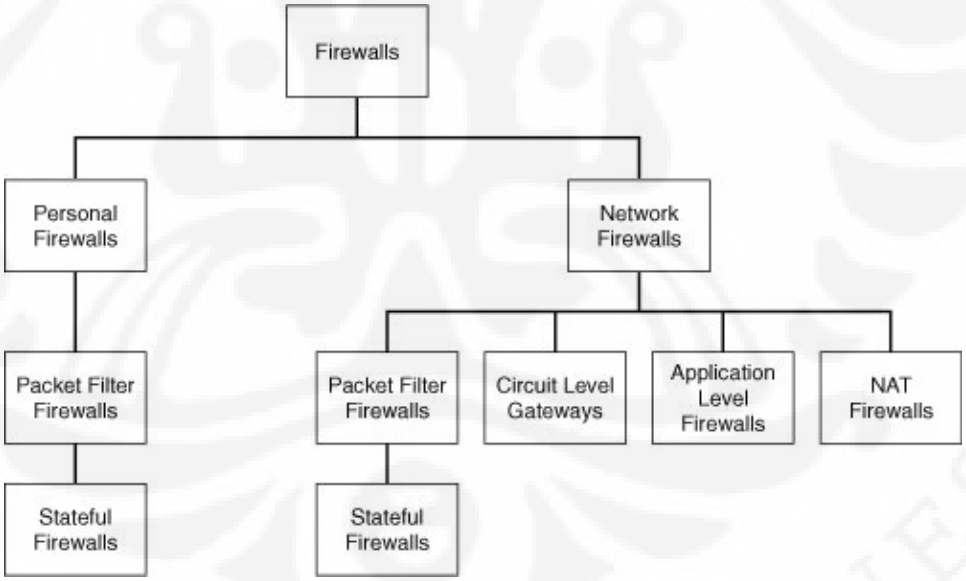
2.5. Firewall

Firewall, terlepas dari bagaimana kompleks dalam desain dan implementasi, memiliki tanggung jawab sederhana untuk bertindak sebagai pelaksana penegakan kebijakan pada keamanan. *Firewall* melakukannya dengan memeriksa data yang diterima dan melacak koneksi yang dibuat untuk menentukan data apakah koneksi tersebut diijinkan atau ditolak. [4] *Firewall* dapat juga bertindak sebagai perantara dan permintaan *proxy host* yang dilindungi, sementara pada saat yang sama menyediakan sarana otentikasi akses untuk lebih memastikan bahwa hanya perangkat akses diberikan. Akhirnya, *firewall* dapat melakukan pelaporan sehingga seorang administrator dapat waspada terhadap kejadian-kejadian yang berkaitan dengan semua proses ini agar *administrator* mengetahui apa yang terjadi dengan *firewall*. [5]



Gambar 2.3. *Firewall* melindungi jaringan dari koneksi yang tidak mempunyai izin. [4]

Ada beberapa motif yang menyebabkan seseorang untuk melakukan ancaman terhadap sistem jaringan. Dengan memeriksa ancaman dan tanggapan yang sesuai, dapat dikembangkan kebijakan keamanan yang meminimalkan risiko yang dapat timbul oleh ancaman tersebut melalui pelaksanaan dan konfigurasi *firewall* yang tepat. Meskipun *firewall* tidak dapat mencegah semua serangan *Firewall* setidaknya lebih dapat membantu membuat data aman daripada tanpa *firewall* sama sekali.



Gambar 2.4. Diagram jenis-jenis *firewall* [5]

2.5.1. Personal *Firewall*

Personal *firewall* dirancang untuk melindungi sebuah *host* dari akses yang tidak legal. Saat ini personal *firewall* modern mengintegrasikan kemampuan tambahan seperti pemantauan perangkat lunak antivirus dan dalam beberapa kasus mampu meng-analisa perilaku serta *intrusion detection* untuk melindungi jaringan.

Personal *firewall* membuat arti besar di jaringan *internet* dan pengguna rumahan karena mereka memberikan perlindungan *end-user* serta mampu mengendalikan kebijakan dalam melindungi sistem komputer. Mungkin kekhawatiran terbesar bagi perusahaan pengguna yang berkaitan dengan personal *firewall* adalah kemampuan untuk menyediakan mekanisme kontrol yang terpusat pada *firewall* itu sendiri. Kebutuhan untuk mensentralisasi kontrol sangat penting untuk menggunakan personal *firewall* di lingkungan perusahaan untuk meminimalkan beban administrasi. Oleh karena itu, sangatlah penting bahwa ketika jumlah *firewall* meningkat, kemampuan untuk mengelola *firewall* tersebut harus tidak menjadi terlalu membebani jaringan. Dengan sentralisasi kontrol dan pemantauan banyak *vendor* berharap mampu mengurangi upaya konfigurasi *firewall* pada *end-user*.

2.5.1 *Network Firewall*

Network firewall dirancang untuk melindungi seluruh jaringan dari serangan. *Network firewall* terdiri dalam dua bentuk utama: *special tools* atau perangkat lunak *firewall* suite yang diinstal di atas sistem operasi *host*. Contoh *tools* jaringan yang berbasis *firewall* adalah *Cisco PIX, Cisco ASA, NetScreen Juniper firewall, Nokia firewall, dan Symantec Enterprise Firewall]*. *Network firewall* yang lebih populer merupakan *firewall* berbasis *software* termasuk *Check Point Firewall*-1 NG atau NGX *Firewall*, Microsoft ISA *Server*, IPTables berbasis Linux, dan BSD pf *filter* paket. *Firewall* berbasiskan pada jaringan memiliki lebih banyak fitur baru seperti *in-line* intrusi deteksi dan *virtual private network* (VPN), juga mempunyai kontrol yang

baik untuk *LAN-to-LAN* VPN serta *akses-remote-user* VPN. *Firewall* dapat digunakan mengidentifikasi *traffic protocol,* sehingga dapat membuat keputusan mengenai cara terbaik untuk menangani arus *traffic* jaringan.

2.6. Strata Guard

Strata Guard adalah salah satu jenis dari distro linux turunan dari *Red Had* yang dapat difungsikan sebagai IDS maupun IPS dimana didalamnya telah diintegrasikan Snort sebagai standar dari IDS ataupun IPS selain itu juga telah diintegrasikan *Rules* serta memiliki pembuatan database tersendiri. Penggunaan Strata Guard Sebagai IPS ataupun sebagai IDS semuanya tergantung bagaimana seorang network designer mendesain dari bentuk jaringan yang ingin diterapkan. Distro StrataGuard memiliki kelebihan dibanding yang aplikasi ataupun distro linux lainnya karena hampir semua pengaturan dari *rules* dilakukan dengan basis web sehingga tidak akan merepotkan bagi pengguna biasa yang belum terbiasa dengan aplikasi linux. Namun tetap saja untuk menghasilkan hasil yang maksimal diperlukan juga penyetingan *rules* pada tingkatan *command line* dimana pada strata guard seorang admin dapat mengaturnya pada IP Table yang terdapat pada /etc/sysconfig/iptables hal ini digunakan untuk mengatur pengkategorian DOS.

Strata Guard juga memungkinkan melakukan pelaporan melalui email terhadap kondisi dari jaringan sehingga seorang admin akan mendapatkan *update* secara *realtime* mengenai kondisi dari jaringan.

Strata Guard memberikan keleluasaan terhadap seorang admin dalam melakukan konfigurasi sistim IDPS dalam jaringan seperti dalam mengkonfigurasi suatu bentuk serangan yang belum terdefinisikan dalam *rules* ataupun *firewall*.

BAB III

PERANCANGAN SISTIM IDPS

(Intrusion Detection Prevention System)

Pada dasarnya banyak cara yang dapat dilakukan untuk melakukan perancangan sistim penanganan *intrusion* pada jaringan komputer. Dalam perancangan sistim penanganan *intrusion* , akan dikelompokkan menjadi dua yaitu :

1. Sistim Pervasif
2. Sistim Reaktif

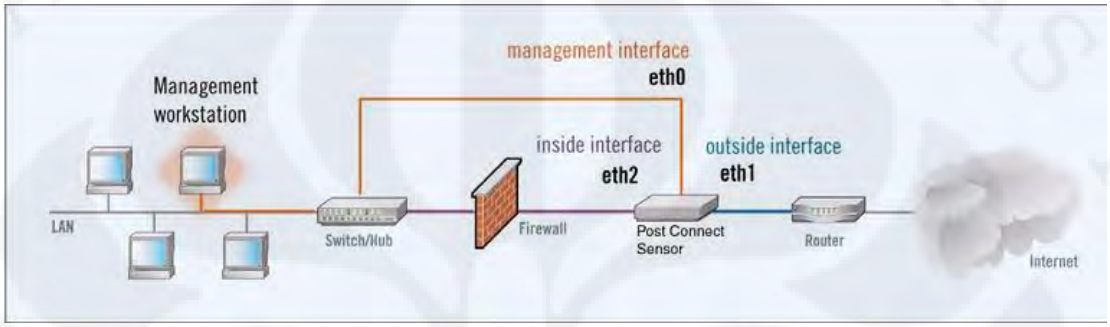
Sistim pervasif ini akan berkerja untuk mencegah jika terjadi serangan sehingga hasil yang diharapkan dari sistim pervasif adalah sistim semakin memiliki kemampuan untuk menahan serangan sedangkan pada sistim reaktif sistim ini bertujuan untuk menangkal serangan yang terjadi dalam jaringan sehingga sistim bisa terselamatkan dari serangan namun sistim reaktif terkadang tidak bersifat fleksibel dalam melakukan deteksi sehingga bisa terjadi kesalahan deteksi pada implementasinya. Karena alasan inilah dibangun suatu perancangan IDPS yang dapat mencegah dan mengendalikan serangan yang terjadi pada jaringan.

3.1 SISTIM STRATA GUARD

Sistim staraguard adalah sistim operasi yang merupakan pengembangan linux Red Had yang bersifat freeware untuk versi lite. Linux Stara Guard digunakan untuk membangun sistim IDS ataupun sistim IPS. Pembangunan sistim baik IPS ataupun IDS tergantung dari desain jaringan yang dibuat oleh desiner jaringan. Stara Guard didalamnya telah dilengkapi oleh sistim pendeteksian serangan yang umum digunakan yaitu Snort yang merupakan aplikasi freeware dalam mendeteksi sebuah serangan yang dilakukan oleh hacker atau orang yang tidak memiliki wewenang dalam mengakses jaringan tersebut.

3.2 PERANCANGAN SISTIM IDPS (*Intrusion Detection Prevention System*)

Berikut ini gambar 3.1 adalah bentuk perancangan sistim yang akan digunakan sebagai IDPS (*Intrusion Detection Prevention System*).



Gambar 3.1 Perancangan Sistim IDPS

Pada gambar 3.1 akan terdapat 3 buah PC yang akan digunakan sebagai *user/ host* dan terdapat 1 buah PC manajement jaringan yang akan disambungkan menggunakan swich yang terhubung dengan firewall serta gateway strataguard. Kemudian strata guard akan dihubungkan juga dengan firewall serta router yang tersambungkan dengan internet.

3.3 Kebutuhan Pendukung Infrastruktur

Kebutuhan akan infrastruktur terbagi menjadi dua macam, yaitu software dan hardware dimana keduanya saling mendukung satu sama lain.

3.3.1Kebutuhan Hardware

Kebutuhan akan penggunaan hardware dalam melakukan perancangan sistim IPS Stara Guard ini antara lain meliputi : NIC *(Network Interface Card),* Switch, Router, Firewall,bebrapa PC sebagai Gateway serta sebagai Client.

NIC *(Network Interface Card)*

Sebuah kartu yang berfungsi sebagai jembatan dari komputer ke sebuah jaringan komputer. Jenis NIC yang beredar, terbagi menjadi dua jenis, yakni NIC yang bersifat fisik, dan NIC yang bersifat logis. Contoh NIC yang bersifat fisik adalah NIC Ethernet, Token Ring, dan lainnya; sementara NIC yang bersifat logis adalah loopback adapter dan Dial-up Adapter. Disebut juga sebagai Network Adapter. Setiap jenis NIC diberi nomor alamat yang disebut sebagai MAC address, yang dapat bersifat statis atau dapat diubah oleh pengguna.Penggunaan NIC pada sistim ini adalah sebagai sarana penghubung komputer dengan jaringan yang ingin dibangun.

Router

Router adalah sebuah alat jaringan komputer yang mengirimkan paket data melalui sebuah jaringan atau internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing. Proses routing terjadi pada layer 3 pada OSI layer. Router berfungsi sebagai penghubung antar dua atau lebih jaringan untuk meneruskan data dari satu jaringan ke jaringan lainnya. Router berbeda dengan switch. Switch merupakan penghubung beberapa alat untuk membentuk suatu Local Area Network (LAN). Dalam perancangan ini router yang digunakan adalah router Linksys series WRVS 4400

Switch

Switch digunakan sebagai konsentrator yang menghubungkan antar client. Switch yang digunakan pada skripsi ini yaitu Switch 10/100 Fast Ethernet 3 Com, switch tersebut terdiri dari 16 port dan mendukung full duplex.

Firewall

Firewall digunakan sebagai pelindungan jaringan dari serangan yang menyebabkan down nya sistim. Pada sistim ini digunakan Firewall yang ada pada

Router WRVS 4400 yang mana pada sistim ini router itu hanya difungsikan sebagai firewall.

Gateway

Gateway digunakan sebagai pintu keluar dan masuknya paket. Pada percobaan ini digunakan sebuah Komputer yang akan diinstal Stara Guard didalamnya.Tabel 3.1 berikut ini adalah spesifikasi dari komputer yang digunakan dalam penginstalasian IDPS :

Tabel 3.1 Spesifikasi Minimum Reqruitmen Penggunaan IDPS [7]

Processor	Single Core (2.8 GHz)	Quad Core, 2.33 GHz minimum Eight Core for Multi-Gig Speeds
RAM	2 GB	4 GB
Disk space	36 GB	160 GB to 250 GB
Server-class network interfaces: Strata Guard standard mode – 2 ea. Strata Guard gateway mode – 3 ea.	10/100/1000 server-quality (Intel)	10/100/1000 server-quality (Intel)
CD R/W ROM drive: • CD W drive to create an install CD • CD R drive to use for first-time installation • DVD	Yes	Yes
An Internet connection that allows outbound SSL communications	Yes	Yes
High-availability (HA) bypass card [optional])	Not available	Optional

3.3.2 Kebutuhan software

Kebutuhan akan penggunaan software pada pembangunan dan uji coba sistim ini antara lain meliputi : wire shark, Zen Map , WinArp Attacker ,Stara Guard Operating System .

Wireshark

Wireshark merupakan software yang digunakan untuk melakukan analisa jaringan komputer, wireshark dapat menganalisa beberapa parameter QoS seperti bandwidth, delay, throughput, dan packet loss dan lain lain serta dapat mengcapture protokol yang sedang berjalan dalam jaringan tersebut, versi wireshark yang digunakan untuk pengujian adalah *wireshark-setup-1.0.10* dan dapat didownload secara gratis pada website www.wireshark.org

Zen Map

Zen Map adalah sebuah software yang digunakan untuk memetakan port yang terbuka dalam satu jaringan. Zen Map merupakan software untuk sistim operasi windows dari Nmap yang memiliki tampilan GUI yang lebih baik. Zen Map selain digunakan untuk memetakan port yang terbuka juga di gunakan untuk melakukan OS Fingerprinting yang mana sangat berguna dalam mengetahui sistim operasi yang apa digunakan oleh target. Software ini dapat didownload secara gratis dari www.nmap.org

WinARP Attacker

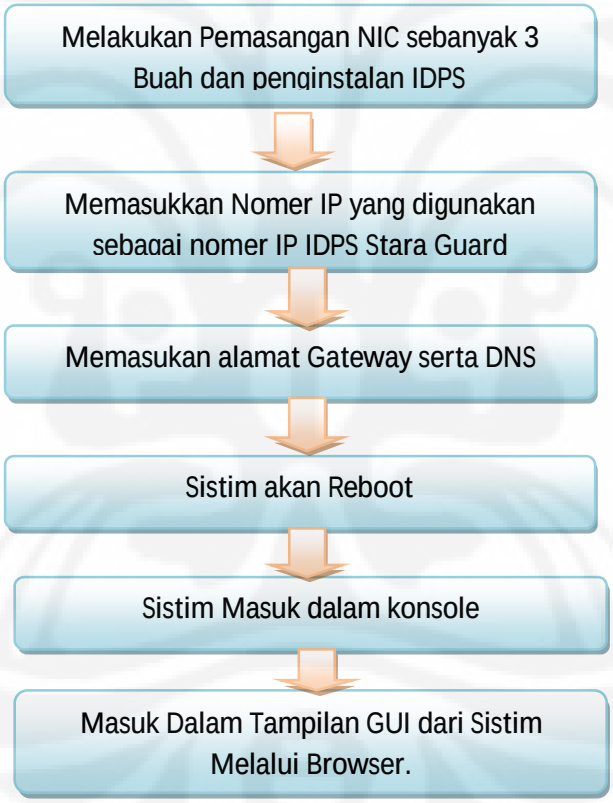
WinARP Attacker adalah sebuah software yang digunakan untuk melakukan Ip scanning dan melakukan flooding paket ARP ,UDP dan TCP dalam suatu jaringan sehingga sebuah jaringan akan mengalami RTO (Request Time Out) yang disebabkan terlalu banyaknya sebuah permintaan akan sebuah service .

3.4 Instalasi Infrastruktur

Pada bagian ini akan dibahas mengenai proses instalasi hardware dan software sistem IPS Strata Guard.

3.4.1 Instalasi Stara Guard sebagai Gateway

Untuk mengaktifkan sebuah sistim IPS diperlukannya sebuah Gateway yang digunakan untuk sebagai pintu keluar dan masuknya paket-paket data. Adapun cara instalsi sistim dari sistim IDPS digambarkan dalam bentuk diagram alir seperti dibawah ini :



Gambar 3.2 Flow Chart Strata Guard

3.4.2 Instalasi Wireshark

Sebelum melakukan instalasi wireshark kita perlu mendownload program wireshark dari alamat http://www.wireshark.org, untuk instalasi kita tinggal mengeklik double program *wireshark-setup-1.0.10*.exe dan ikuti petunjuk selanjutnya, pada program wireshark juga diperlukan program WinpCap untuk mengcapture protocol yang sudah terintegrasi pada wireshark

3.4.3 Instalasi Client

Pada dasarnya urutan instalasi pada sisi client sama dengan sisi server, perbedaannya hanya pada setting IP Address, untuk client diberikan IP Address 192.168.0.x dengan netmask 255.255.255.0 sesuai topologi yang telah direncanakan serta menyertakan gateway 192.168.0.11.

3.4.4 Instalasi Router

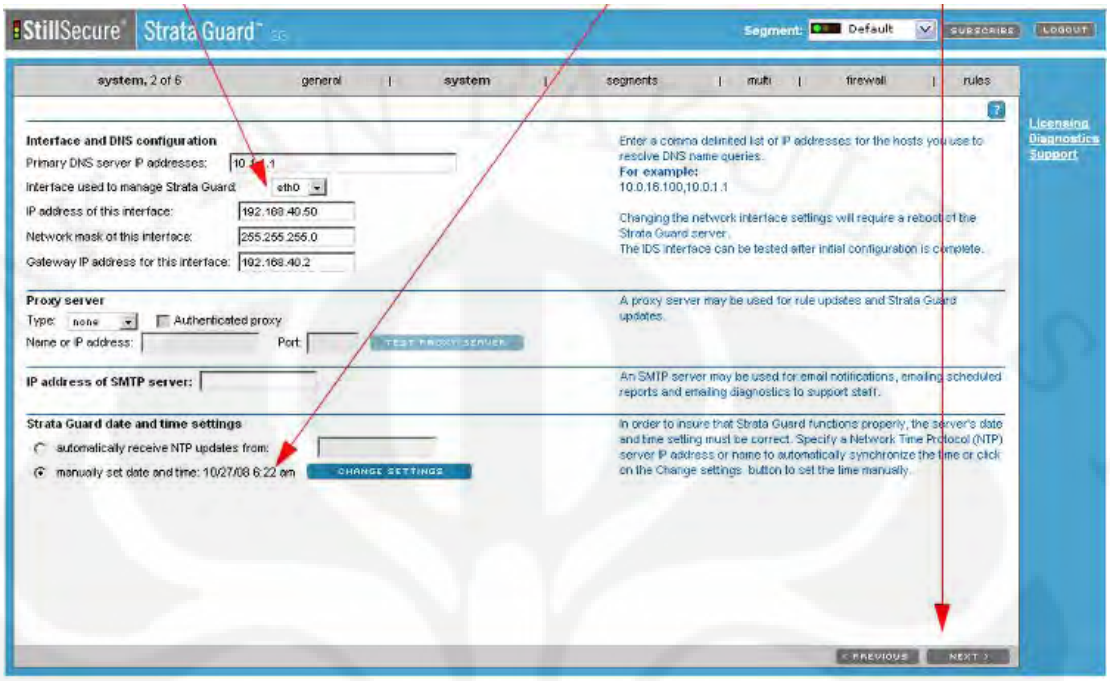
Router digunakan untuk meneruskan paket menuju network yang berbeda dalam hal ini adalah dari Gateway IPS 192.168.0.11 menuju koneksi internet dimana dalam jaringan IPS StaraGuard ini Router menggunakan IP 192.168.0.1

3.4.5 Instalasi Firewall

Firewall digunaka untuk memfilter paket-paket yang mencurigakan yang tidak dapat dideteksi secara baik oleh sistim sehingga terjadi sinkronisasi yang baik dengan sistim pada jaringan IPS ini firewall dalam sistim ini meggunakan IP 192.168.0.6

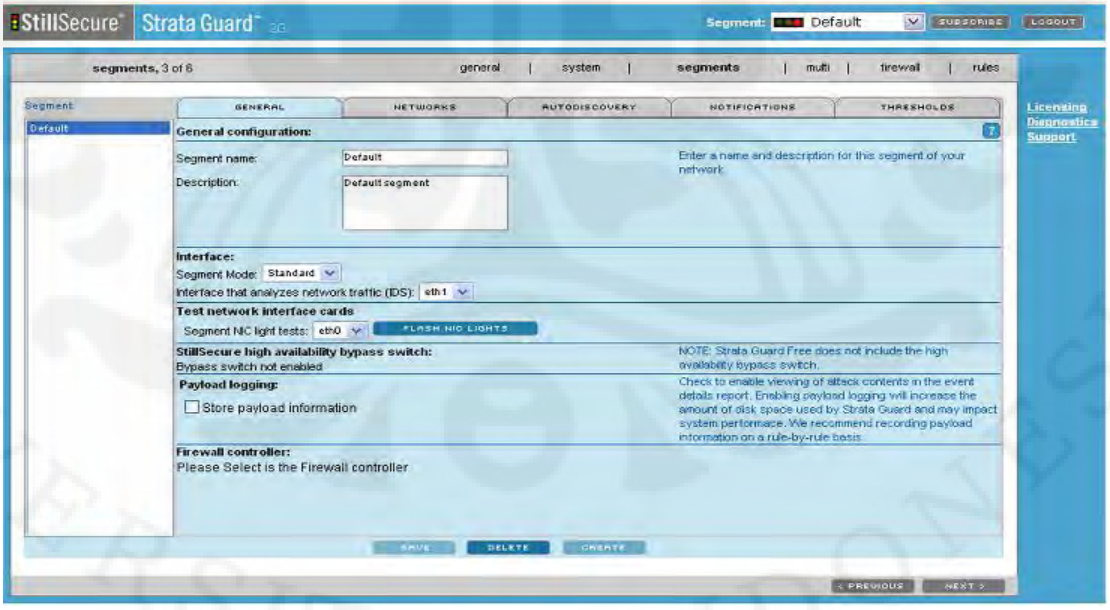
3.5 Konfigurasi Strata Guard

Strata Guard adalah inti dari sistim ini yang mana diperlukan konfigurasi yang maksimal untuk memberikan hasil yang optimum pada saat pengujian jaringan IPS Strata Guard dibawah ini adalah tampilan konfigurasi IP dan DNS pada StaraGuard[8]



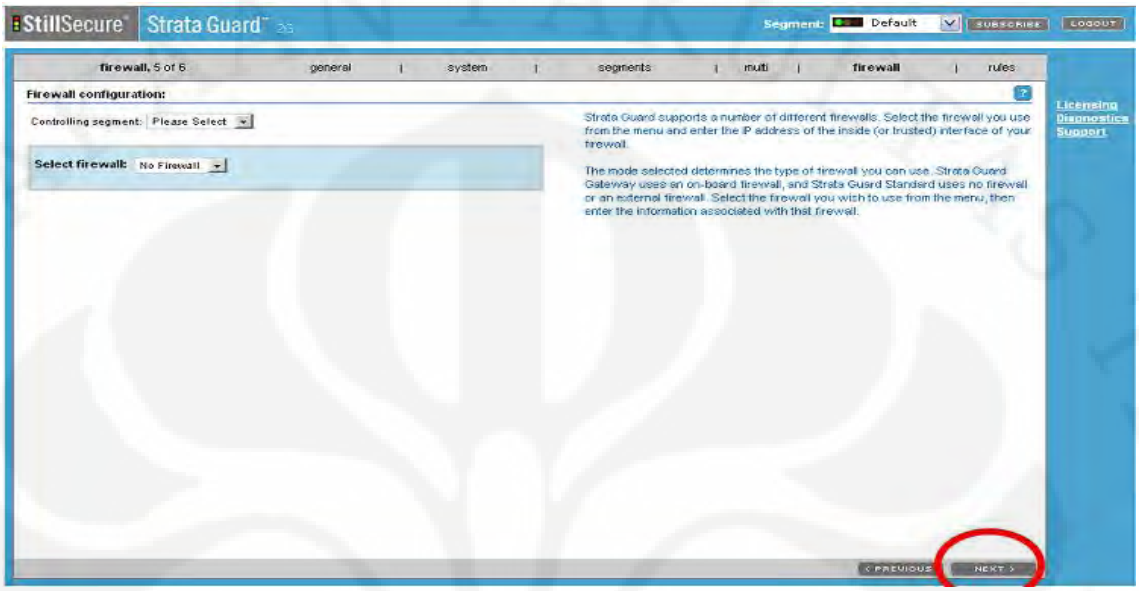
Gambar 3.3 konfigurasi IP dan DNS pada StaraGuard [8]

Gambar dibawah berikut ini adalah tampilan saat melakukan penyetingan segmen dari stara guard [9]



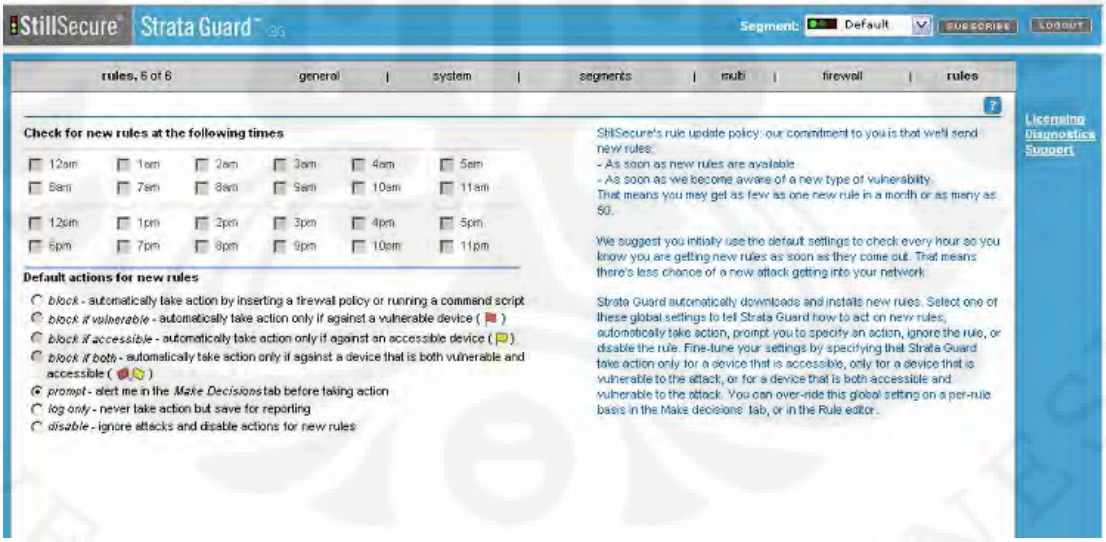
Gambar 3.4 Konfigurasi Segmen pada StaraGuard [9]

Tampilan gambar dibawah ini menunjukkan tampilan konfigurasi penggunaan firewall [10]



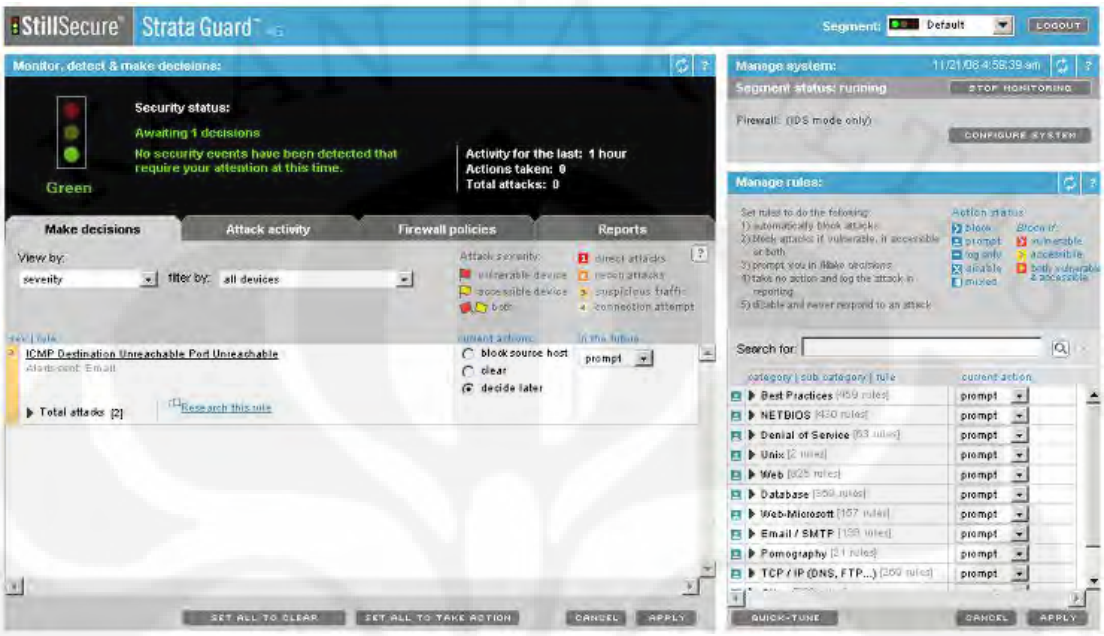
Gambar 3.5 konfigurasi penggunaan firewall [10]

Gambar dibawah ini adalah gambar yang menujukkan sistim update rules yang dilakukan oleh StaraGuard [11]



Gambar 3.6 Sistim update Rules [11]

Gambar dibawah ini adalah gambar setelah semua sistim terkonfigurasi semuanya [12]



Gambar 3.7 Tampilan Awal Strata Guard [12]

Selain konfigurasi diatas perlu juga dinyatakan pensetingan berapa besar paket yang akan dikategorikan sebagai DOS dalam jaringan sehingga dilakukan perubahan parameter pada strara guard. Berikut ini adalah cara dalam melakukan perubahan parameter dalam stara guard.

Masuk melalui terminal dengan memasukkan username dan password dari root.

Lalu masukkan perintah

cd /etc

vi sysconfig

Pilih IP tables

Lalu lakukan perubahan pada limit paket nya

-A syn-flood -m limit --limit 300/sec --limit-burst 600 -j RETURN

Untuk konfigurasi paket ACK

-A ack-flood -m limit --limit 100/sec --limit-burst 200 -j RETURN

-A ack-flood -m limit --limit 1/sec --limit-burst 2 -j LOG --log-tcp-options --log-ip-options --log-prefix "BGIPTables SID: 9001 "

-A ack-flood -j DROP

Untuk konfigurasi paket SYN

-A syn-flood -m limit --limit 100/sec --limit-burst 200 -j RETURN

-A syn-flood -m limit --limit 1/sec --limit-burst 2 -j LOG --log-tcp-options --log-ip-options --log-prefix "BGIPTables SID: 9000 "

-A syn-flood -j DROP

Untuk konfigurasi General Flood

-A general-flood -m limit --limit 200/sec --limit-burst 400 -j RETURN

-A general-flood -m limit --limit 1/sec --limit-burst 2 -j LOG --log-tcp options --log-ip-options --log-prefix "BGIPTables SID: 9002 "

-A general-flood -j DROP

Untuk konfigurasi DNS reply Flood

-A dns-reply-flood -m limit --limit 100/sec --limit-burst 200 -j RETURN

-A dns-reply-flood -m limit --limit 1/sec --limit-burst 2 -j LOG --log-tcp-options --log-ip-options --log-prefix "BGIPTables SID: 9003 "

-A dns-reply-flood -j DROP

3.6 Konfigurasi Router

Router adalah sebuah device yang berfungsi untuk meneruskan paket-paket dari sebuah network ke network yang lainnya (baik LAN ke LAN atau LAN ke WAN) sehingga host-host yang ada pada sebuah network bisa berkomunikasi dengan host-host yang ada pada network yang lain. Router menghubungkan network-network tersebut pada network layer dari model OSI, sehingga secara teknis Router adalah

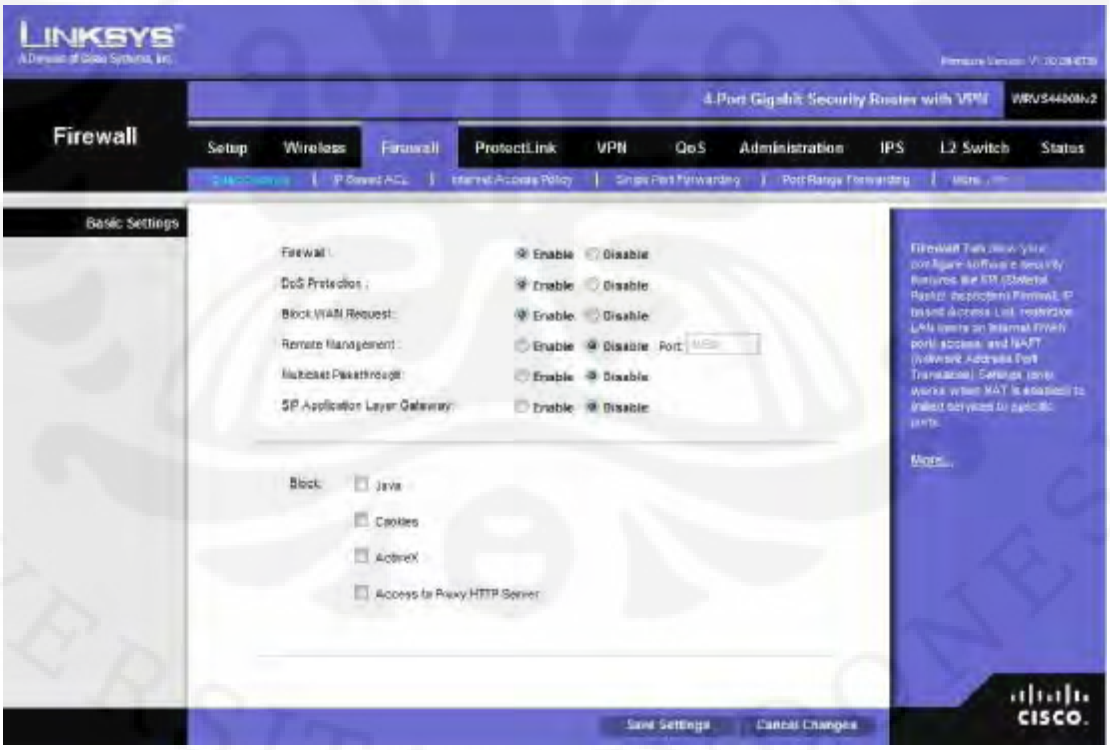
Layer 3 Gateway.Pada sistim IPS ini router digunakan untuk melajutkan paket-paket tersebut ke internet. Dalam melakukan konfigurasi terhadap router Linksys WRVS 4400 dilakukan dengan cara melalui tampilan GUI dari web browser.Berikut ini adalah tampilan router setelah dikonfigurasi.

Gambar 3.8 Konfigurasi Router Linksys

3.7 Konfigurasi firewall

Firewall adalah device yang digunakan sebagai penahan dan pem filter sebuah serangan. Firewall bias dikatakan bersifat pasif dimana firewall hanya akan memblock hal – hal yang telah diatur sebelumnya untuk itu pad sistim jaringan IPS firewall diletakkan dibelakang dari IPS Strata Guard. Berikut ini adalah tampilan dari konfigurasi firewall.

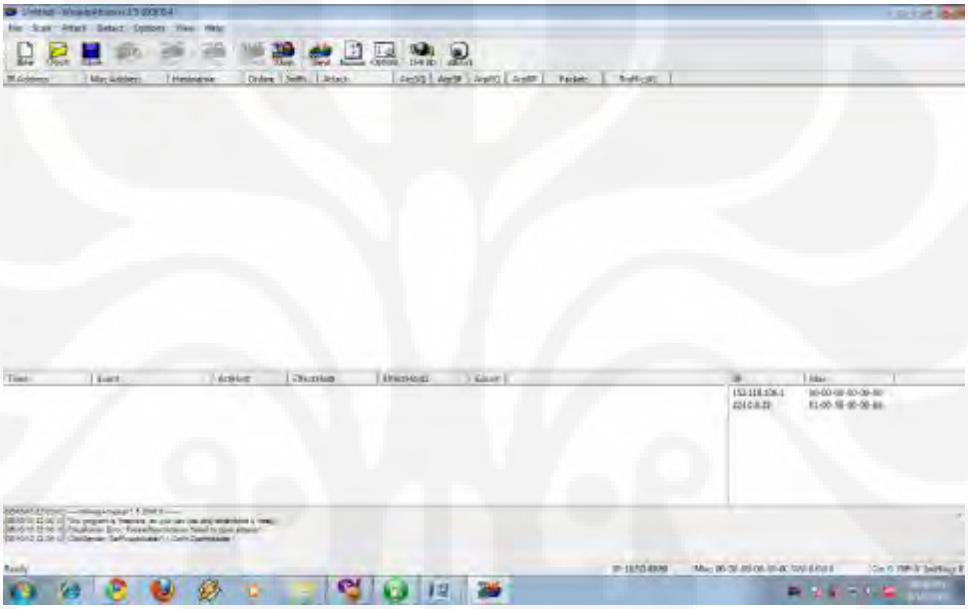
Gambar 3.9 Konfigurasi IP dari Firewall



Gambar 3.10 GambarKonfigurasi Firewall Linksys

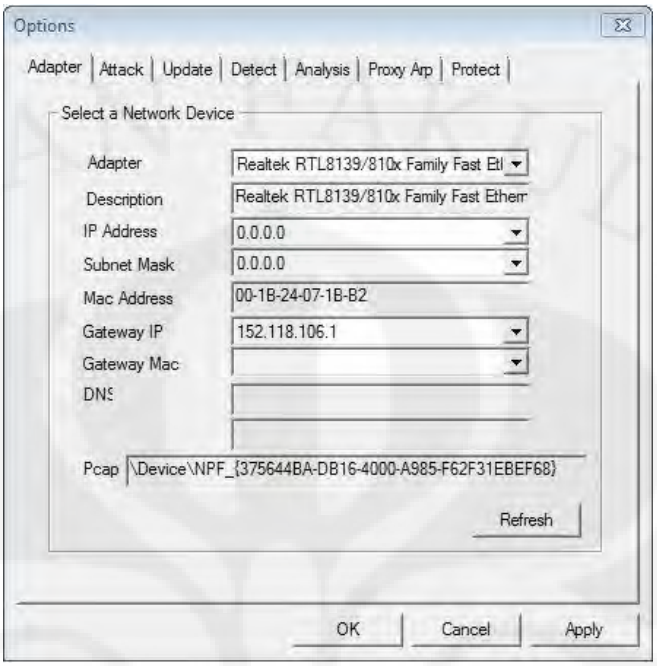
3.8 Konfigurasi WinARP Attacker

WinArp adalah tools yang digunakan untuk melakukan IP scaning dan flooding pada jaringan IPS StrataGuard tools ini sangat penting untuk menguji sampai pada tinggkatan mana kemampuan dari IPS StaraGuard dalam mendeteksi sebuah ancaman dalam jaringan.Berikut ini adalah tampilan dari program WinARP Attacker.



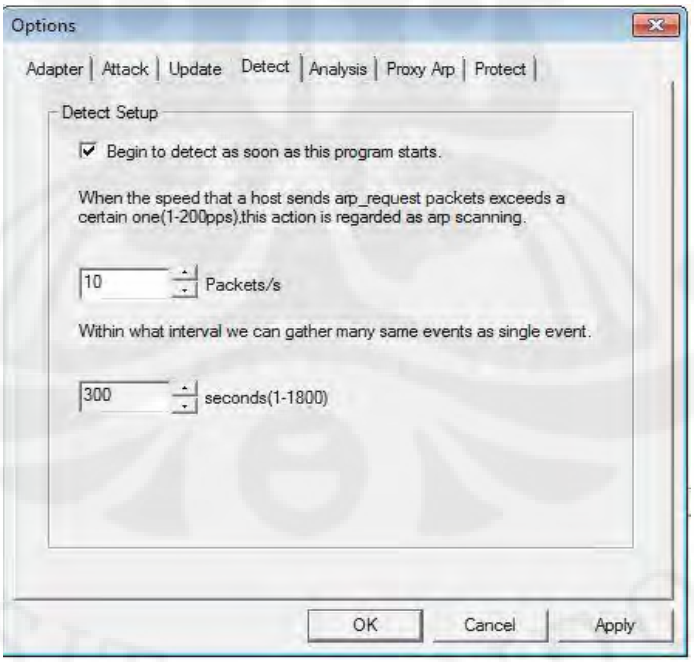
Gambar 3.11 Tampilan Awal dari program WinARP Attacker

Selanjutnya untuk membuat program ini berjalan diperlukannya penyetingan terhadap device mana yang akan digunakan untuk melakukan flooding serta IP Scan



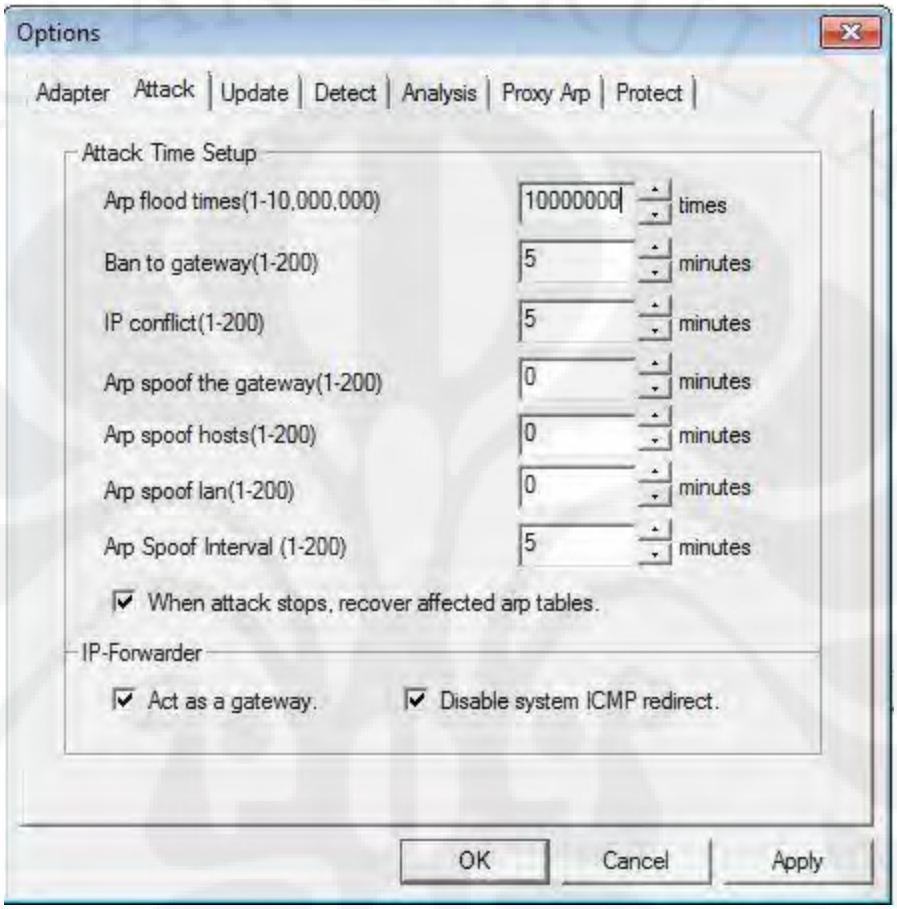
Gambar 3.12 Tampilan saat memlih device

Kemudian yang diperlu dilakukan adalah menentukan besarnya paket yang akan dikirim dalam 1 detik.



Gambar 3.13 Penentuan besarnya paket yang akan dikirimkan

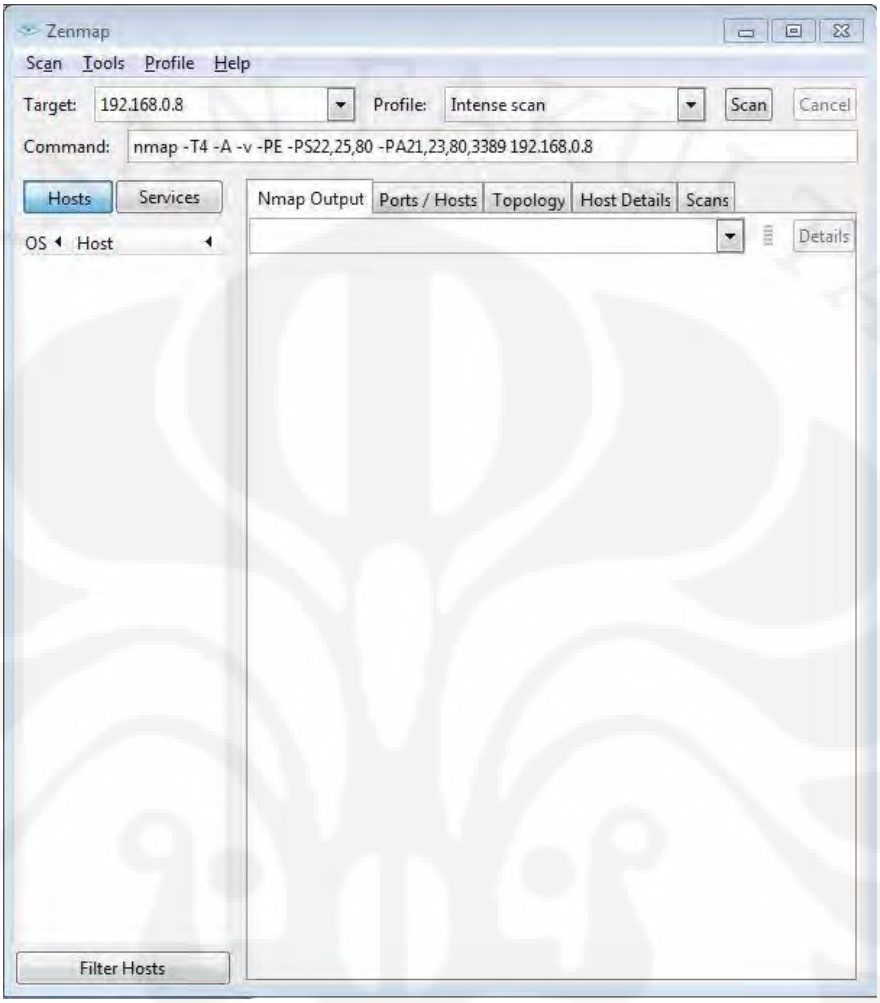
Selanjutnya setelah mendefinisikan berapa besarnya paket yang dikirimkan selanjutnya adalah mendefinisikan lamanya waktu flooding dilakukan.



Gambar 3.14 Mendefinisikan lama waktu flooding

3.9 Konfigurasi Zen Map

Zen Map adalah tools yang berfungsi dalam menguji coba jaringan IPS Stara Guard ini sebagai software *scanning port* dan *Operating System Finger Printing* yang beguna untuk menentukan banyaknya port yang terbuka dan jenis Sistim operasi apa yang digunakan. Berikut ini adalah tampilan dari Zen Map.



Gambar 3.15 Tampilan dari Zen Map

BAB IV

PENGUJIAN SISTEM DAN ANALISA

4.1 UMUM

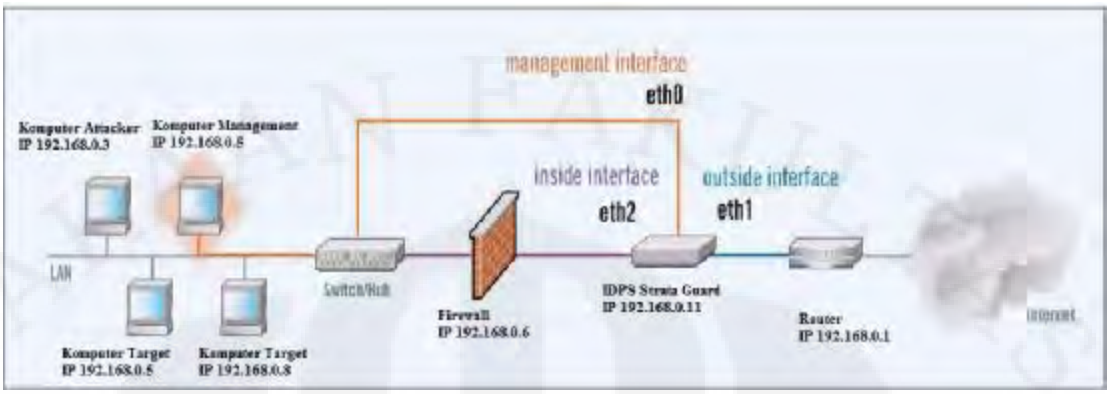
Pada bagian ini akan dilakukan pengujian sistem yang sudah dibuat berdasarkan perancangan pada bab sebelumnya. Pengujian sistem dilakukan dengan melakukan beberapa variasi serangan. Untuk mengetahui apakah IPS *Strata Guard* dapat berfungsi dengan baik.

4.2 METODE PENGUJIAN

Pada skripsi untuk menguji IDPS *Strata Guard* apakah sistem yang dikembangkan berfungsi dengan baik dan memiliki tingkat *reliability* atau kehandalan maka akan dilakukan dua metode pengujian yaitu:

1. *Functionality Test*
2. *Response Time*

Pada pengujian ini akan digunakan 1 buah laptop yang akan menjadi target serangan serta digunakan sebuah *gateway* yang telah diinstalasikan Strata Guard. Untuk menghitung *response time* dan *action time* akan digunakan Software Wireshark yang diletakkan pada laptop yang digunakan sebagai *management station* dalam jaringan yang akan digunakan untuk memonitor kegiatan serangan yang berhasil ditangkap oleh gateway IPS .Untuk desain jaringan yang digunakan dapat dilihat pada gambar 4.1



Gambar 4.1 Desain jaringan IDPS

4.2.1 Functionality Test

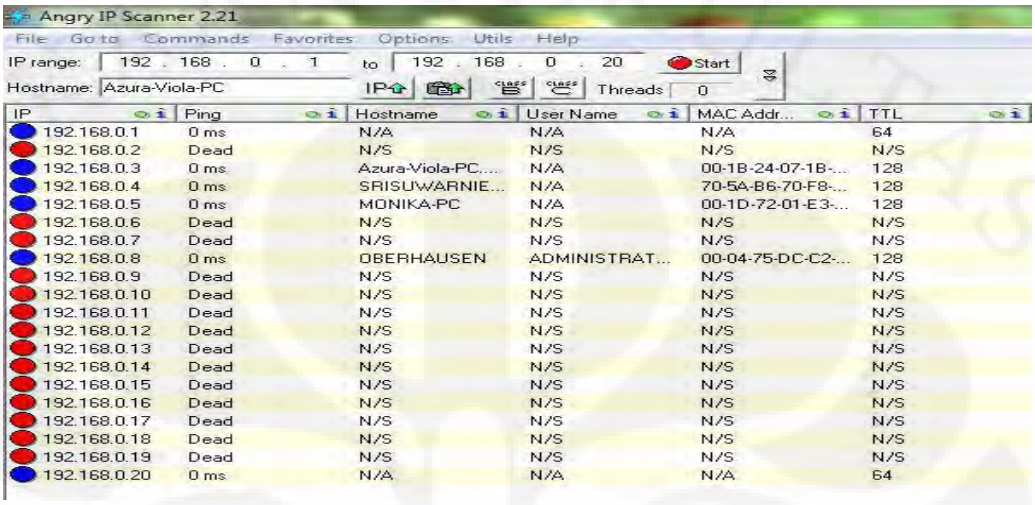
Functionality test bertujuan untuk menguji *gateway strata guard* apakah dapat berfungsi dengan baik sesuai dengan skenario yang diinginkan yaitu dapat melakukan pendeteksian serta prevention dari serangan dimulai dari :

- *Network Surveying* dimana didalamnya dilakukan percobaan-percobaan :
  - *IP Scanning*
  - *Port Scanning*
  - *Os Finger Printing*
  - *Vulnerability Scaning*
- Selanjutnya dilakukan *Enumeration Test*
  - *Flooding*

IP Scanning

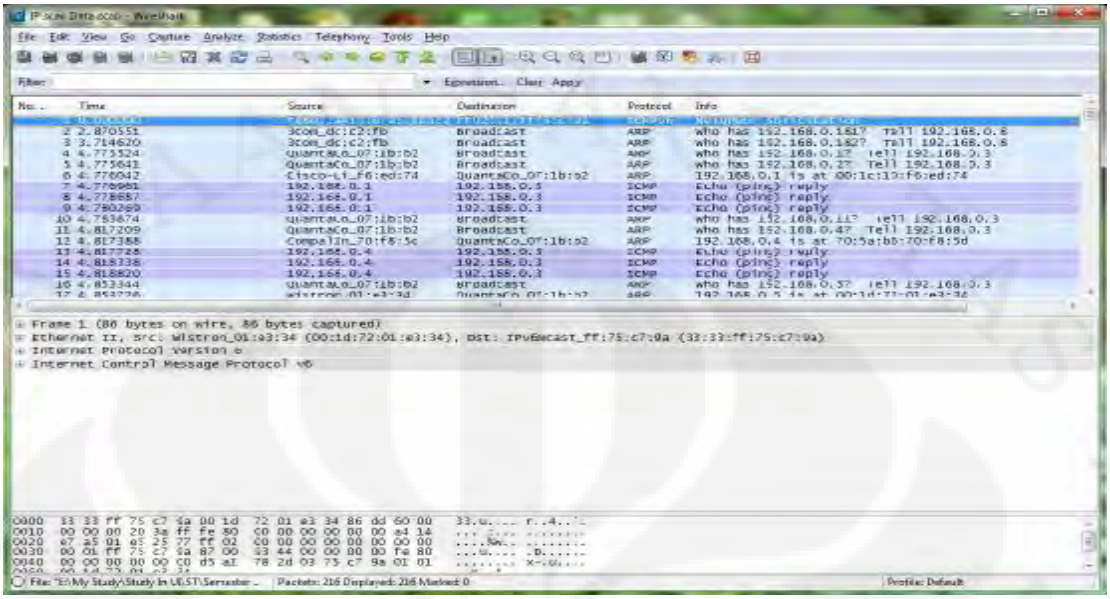
IP *Scanning* adalah metode yang digunakan untuk mengetahui keberadaan sebuah user apakah dalam keadaan aktif atau *off*. Untuk itu diperlukaan sebuah software yang dapat memastikan keberadaan dari user-user yang ada dalam jaringan. Dalam hal ini kita dapat menggunakan berbagai macam *software* namun pada uji coba sistem ini digunakan software IP Angry yang digunakan untuk mengecek

kondisi IP apakah IP tersebut hidup atau dalam kondisi mati . Berikut ini adalah tampilan dari software IP angry.



Gambar 4.2 *Capture Angry IP Scanner*

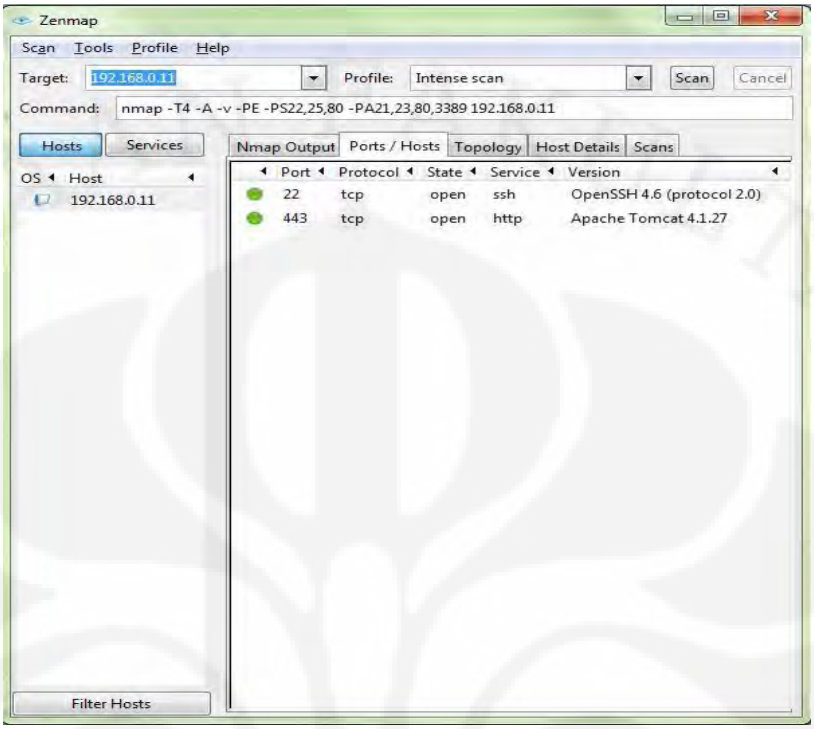
Dari hasil *capture* paket yang dilakukan oleh wireshark didapat bahwa untuk mengetahui bahwa sistem itu Angry IP scanner menggunakan paket ARP *who has* yang menanyakan kepemilikan IP addres tersebut yang jika IP itu dalam kondisi hidup IP tersebut akan melakukan reply paket ICMP terhadap IP dari computer yang menjalankan software Angry IP Scanner. Berikut ini hasil capture yang diambil oleh wire shark.



Gambar 4.3 Capture WireShark Saat Angry IP dijalankan

Port Scanning

Port scanning merupakan suatu proses untuk mencari dan membuka *port* pada suatu jaringan computer. Dari hasil scanning akan didapat letak kelemahan sistem tersebut. Pada dasarnya sistem *port scan*ning mudah untuk dideteksi, tetapi penyerang akan menggunakan berbagai metode untuk menyembunyikan serangan. Dalam uji coba ini *port scaning* dilakukan terhadap dua IP *address* yaitu IP 192.168.0.11 dengan jenis port scaning yang dilakukan adalah *intense scan , intense scan plus UDP*,dan *intense scan all TCP port*.Berikut ini adalah tampilan dari *port scaning* saat melakukan *port scanning intense scan* pada IP 192.168.0.11



Gambar 4.4 Tampilan ZenMap saat melakukan Port Scan

Dari tampilan diatas dapat terlihat port yang terbuka adalah port 22 dan port 443, Hasil proses *scanning* ini sering digunakan penyerang untuk melakukan serangan disalah satu port tersebut.berikut ini adalah hasil laporan dari *scan port* .

NSE: Loaded 49 scripts for scanning.

Initiating ARP Ping Scan at 14:33

Scanning 192.168.0.11 [1 port]

Completed ARP Ping Scan at 14:33, 0.59s elapsed (1 total hosts)

Initiating Parallel DNS resolution of 1 host. at 14:33

Completed Parallel DNS resolution of 1 host. at 14:33, 16.59s elapsed

Initiating SYN Stealth Scan at 14:33

Scanning 192.168.0.11 [1000 ports]

Discovered open port 443/tcp on 192.168.0.11

Discovered open port 22/tcp on 192.168.0.11

Completed SYN Stealth Scan at 14:33, 9.02s elapsed (1000 total ports)

Initiating Service scan at 14:33

Scanning 2 services on 192.168.0.11

Completed Service scan at 14:34, 12.19s elapsed (2 services on 1 host)

PORT STATE SERVICE VERSION

22/tcp open ssh OpenSSH 4.6 (protocol 2.0)

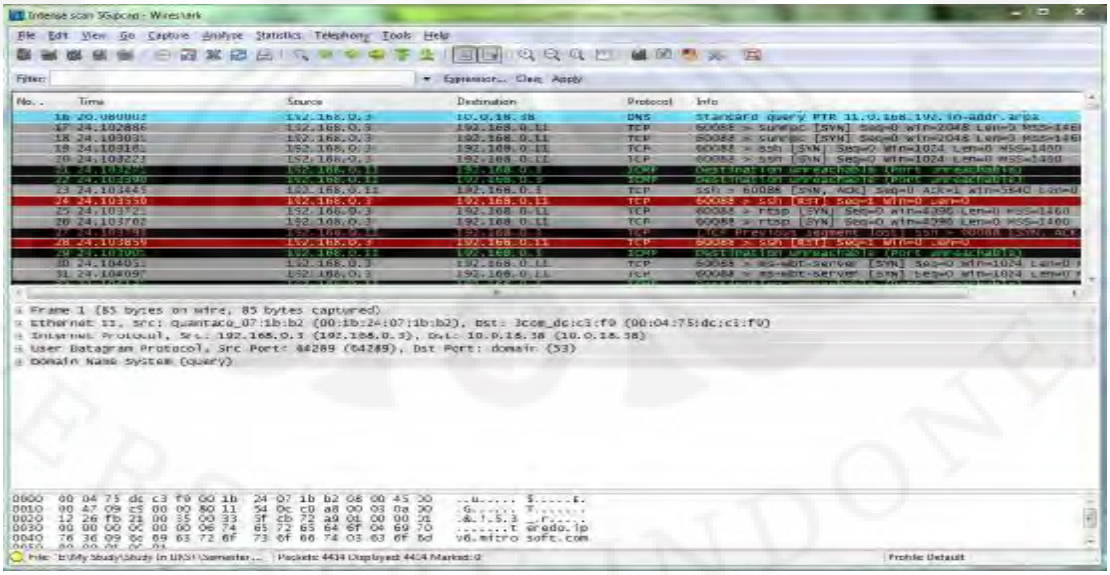
| ssh-hostkey: 1024 03:de:06:a1:c3:16:8c:42:cb:de:c3:1d:1f:fd:f6:f8 (DSA)

|_2048 4c:61:70:bf:c4:93:0a:a5:61:bb:37:e7:d0:74:d8:81 (RSA)

443/tcp open ssl/http Apache Tomcat 4.1.27

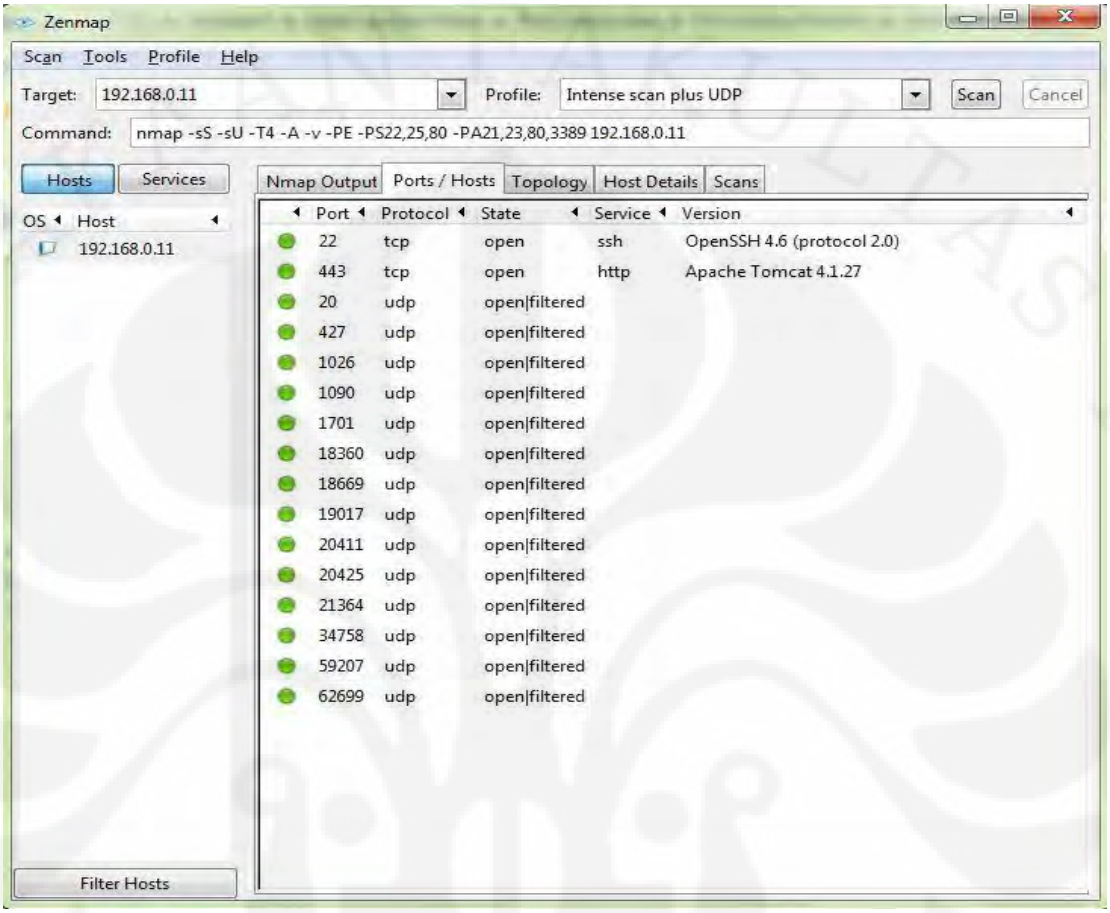
| http-methods: GET HEAD POST PUT DELETE TRACE OPTIONS

Dari hasil *Scanning Por*t yang dilakukan terlihat hasil *capture* dari wireshark yang tampak seperti dibawah ini.



Gambar 4.5 Hasil *Capture* paket dengan wireshark saat *Intense Scan Port*

Untuk serangan *scan port intense* scan plus UDP didapatkan hasil sebagai berikut :



Gambar 4.6. Intense Scan Plus UDP

Hasil yang didapatkan pada intense scan Plus UDP didapatkan tambahan port yang terbuka yaitu port 22, 443, 20, 427, 1026, 1090, 1701, 18360, 18669, 19017, 20411, 20425, 21364, 34758, 59207 ,62699. Terlihat jelas adanya penambahan port setelah dilakukan intense scan Plus UDP .Berikut ini adalah hasil dari laporan port scanning nya.

NSE: Loaded 49 scripts for scanning.

Initiating ARP Ping Scan at 14:45

Scanning 192.168.0.11 [1 port]

Completed ARP Ping Scan at 14:45, 0.48s elapsed (1 total hosts)

Initiating Parallel DNS resolution of 1 host. at 14:45

Completed Parallel DNS resolution of 1 host. at 14:46, 16.72s elapsed

Initiating SYN Stealth Scan at 14:46

Scanning 192.168.0.11 [1000 ports]

Discovered open port 22/tcp on 192.168.0.11

Discovered open port 443/tcp on 192.168.0.11

Completed SYN Stealth Scan at 14:46, 8.08s elapsed (1000 total ports)

Initiating UDP Scan at 14:46

Scanning 192.168.0.11 [1000 ports]

Increasing send delay for 192.168.0.11 from 0 to 50 due to max_successful_tryno increase to 5

Increasing send delay for 192.168.0.11 from 50 to 100 due to max_successful_tryno increase to 6

Warning: 192.168.0.11 giving up on port because retransmission cap hit (6).

Increasing send delay for 192.168.0.11 from 100 to 200 due to 11 out of 21 dropped probes since last increase.

UDP Scan Timing: About 5.40% done; ETC: 14:55 (0:09:03 remaining)

Increasing send delay for 192.168.0.11 from 200 to 400 due to 11 out of 11 dropped probes since last increase.

Increasing send delay for 192.168.0.11 from 400 to 800 due to 11 out of 15 dropped probes since last increase.

UDP Scan Timing: About 8.93% done; ETC: 14:57 (0:10:22 remaining)

UDP Scan Timing: About 11.71% done; ETC: 14:59 (0:11:26 remaining)

UDP Scan Timing: About 16.37% done; ETC: 15:00 (0:12:05 remaining)

UDP Scan Timing: About 28.71% done; ETC: 15:02 (0:11:20 remaining)

UDP Scan Timing: About 35.54% done; ETC: 15:02 (0:10:27 remaining)

UDP Scan Timing: About 41.87% done; ETC: 15:02 (0:09:36 remaining)

UDP Scan Timing: About 47.80% done; ETC: 15:03 (0:08:45 remaining)

UDP Scan Timing: About 53.17% done; ETC: 15:03 (0:07:54 remaining)

UDP Scan Timing: About 58.54% done; ETC: 15:03 (0:07:01 remaining)

UDP Scan Timing: About 64.10% done; ETC: 15:03 (0:06:07 remaining)

UDP Scan Timing: About 69.46% done; ETC: 15:03 (0:05:13 remaining)

UDP Scan Timing: About 74.64% done; ETC: 15:03 (0:04:21 remaining)

UDP Scan Timing: About 79.71% done; ETC: 15:03 (0:03:29 remaining)

UDP Scan Timing: About 85.01% done; ETC: 15:03 (0:02:35 remaining)

UDP Scan Timing: About 90.20% done; ETC: 15:03 (0:01:42 remaining)

UDP Scan Timing: About 95.49% done; ETC: 15:03 (0:00:47 remaining)

Completed UDP Scan at 15:04, 1097.96s elapsed (1000 total ports)

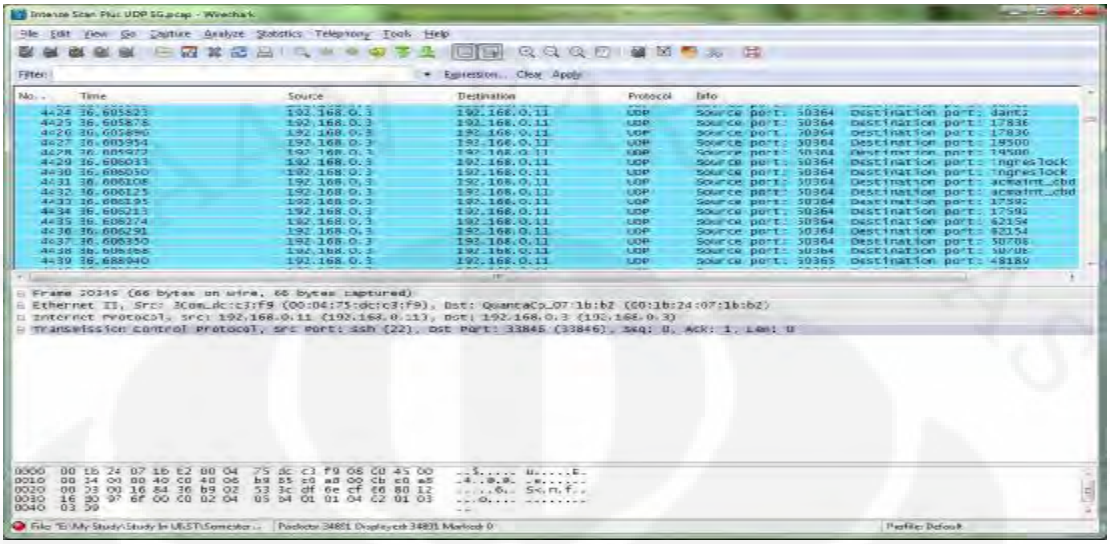
Initiating Service scan at 15:04

Scanning 16 services on 192.168.0.11

Service scan Timing: About 18.75% done; ETC: 15:08 (0:03:15 remaining)

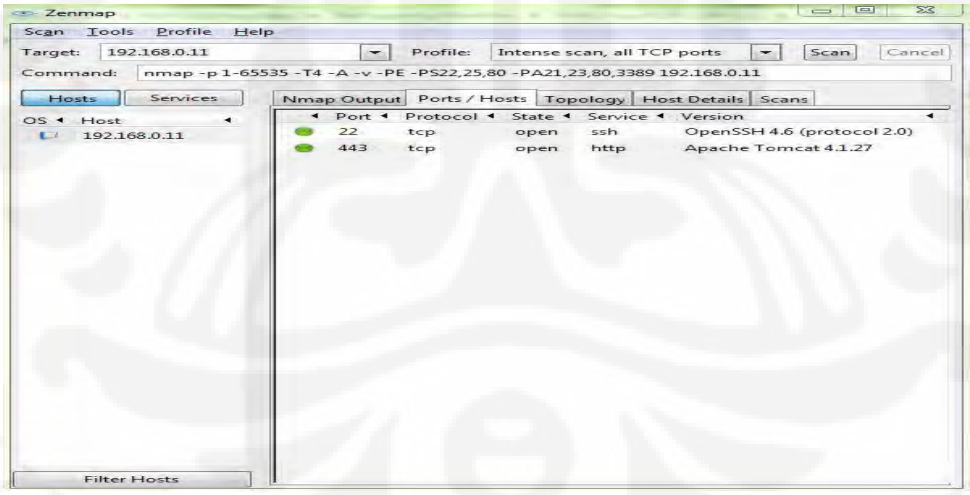
Completed Service scan at 15:05, 62.69s elapsed (16 services on 1 host)

Dari hasil scanning Port yang dilakukan terlihat hasil capture dari wireshark yang tampak seperti dibawah ini.



Gambar 4.7. Hasil *capture* dari wireshark

Sedangkan untuk intense scan all TCP port hasil yang didapat tidak jauh berbeda dengan Intense scan. Gambar 4.8 adalah tampilan dari intense scan all TCP port.



Gambar 4.8. *Intense scan all TCP port*

Berikut ini adalah hasil dari laporan dari Zenmap.

NSE: Loaded 49 scripts for scanning.

Initiating ARP Ping Scan at 15:11

Scanning 192.168.0.11 [1 port]

Completed ARP Ping Scan at 15:11, 0.69s elapsed (1 total hosts)

Initiating Parallel DNS resolution of 1 host. at 15:11

Completed Parallel DNS resolution of 1 host. at 15:11, 16.55s elapsed

Initiating SYN Stealth Scan at 15:11

Scanning 192.168.0.11 [65535 ports]

Discovered open port 22/tcp on 192.168.0.11

Discovered open port 443/tcp on 192.168.0.11

SYN Stealth Scan Timing: About 9.48% done; ETC: 15:17 (0:04:56 remaining)

SYN Stealth Scan Timing: About 21.76% done; ETC: 15:16 (0:03:39 remaining)

SYN Stealth Scan Timing: About 33.70% done; ETC: 15:16 (0:02:59 remaining)

SYN Stealth Scan Timing: About 47.02% done; ETC: 15:16 (0:02:16 remaining)

SYN Stealth Scan Timing: About 59.92% done; ETC: 15:15 (0:01:41 remaining)

SYN Stealth Scan Timing: About 73.02% done; ETC: 15:15 (0:01:07 remaining)

SYN Stealth Scan Timing: About 87.00% done; ETC: 15:15 (0:00:32 remaining)

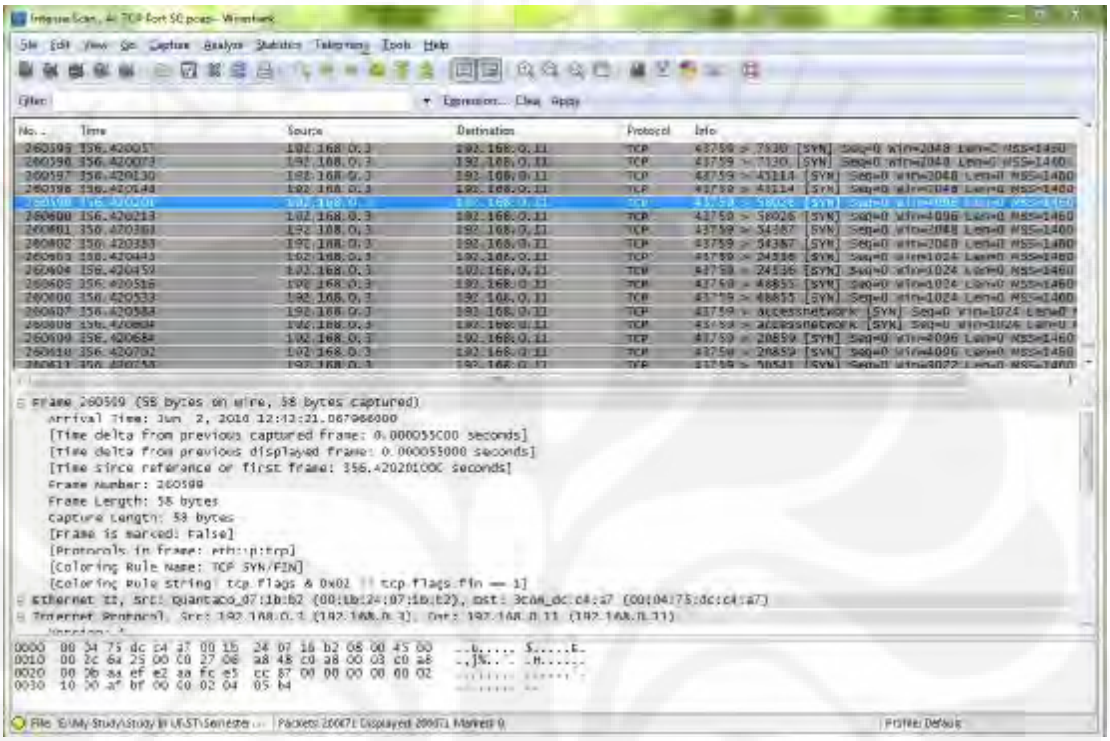
Completed SYN Stealth Scan at 15:15, 239.82s elapsed (65535 total ports)

Initiating Service scan at 15:15

Scanning 2 services on 192.168.0.11

Completed Service scan at 15:15, 12.24s elapsed (2 services on 1 host)

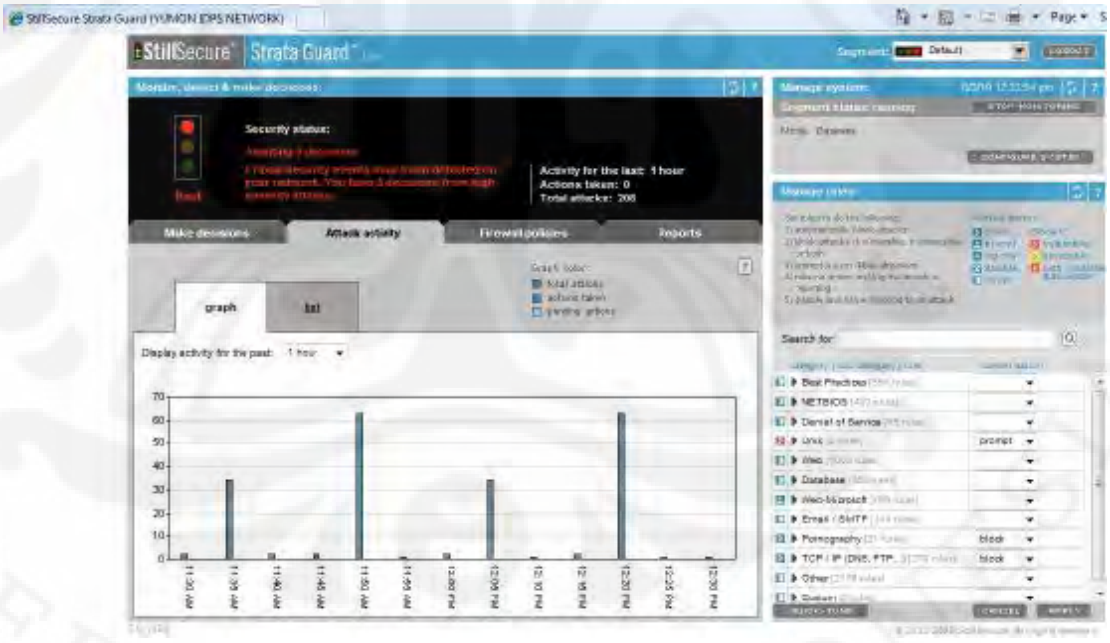
Selanjutnya berikut adalah chapture wireshark saat dilakukan scan.



Gambar 4.9. *Chapture wireshark* saat dilakukan *scan*

Dari serangan tersebut dilakukan deteksi oleh IPS *Strata Guard* yang menunjukkan adanya suatu *scanning port* yang masuk dalam kategori *recorn attack* dimana termasuk dalam kategori membahayakan dalam jaringan berikut ini adalah hasil deteksi yang dilakukan oleh IPS *Strata Guard*. Hal ini terdeteksi sebagai *prompt* yang mana sistem akan memberikan pengingatan terhadap suatu kejadian yang dikatakan membahayakan jaringan. Namun jika suatu kejadian tersebut telah dinyatakan sebelum nya maka kejadian itu tidak akan dimunculkan dalam sebuah peringatan karena hasil keputusan terhadap peristiwa tersebut telah dinyatakan sehingga keputusan pun dapat diambil.

Gambar 4.10. Tampilan Strata Guard saat mendeteksi port scanning

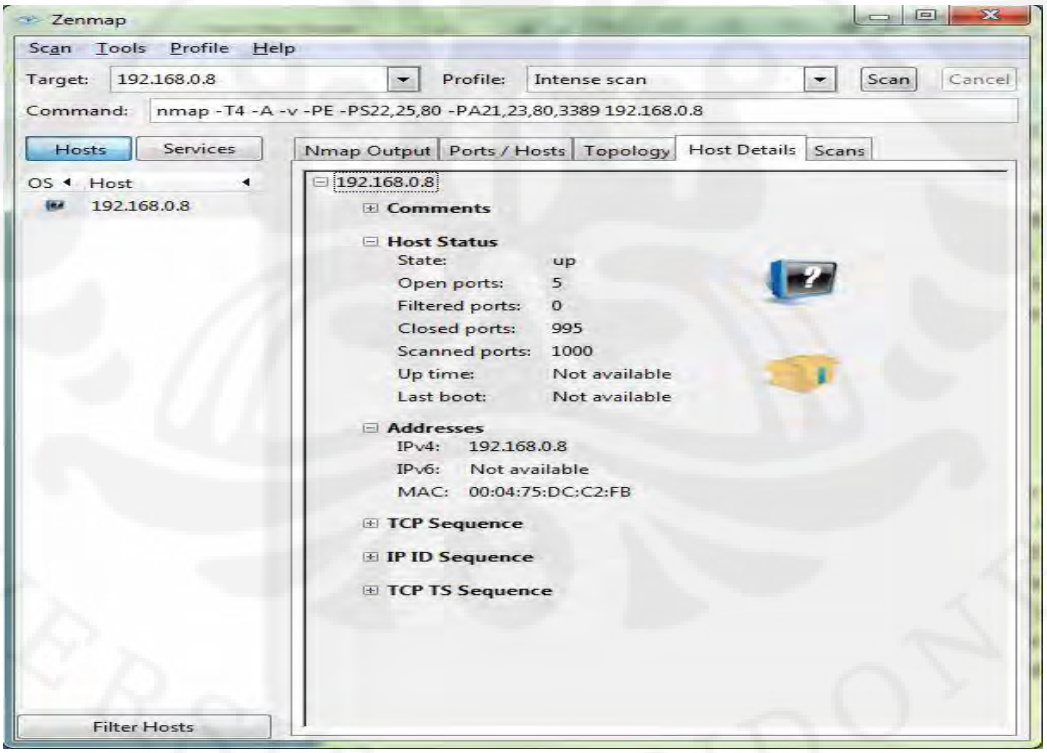


Gambar 4.11.Tampilan Grafik banyaknya serangan yang terdeteksi oleh Sistim IDPS Strata Guard

Dari peringatan tersebut seorang administrator dapat menghasilkan kebijakan untuk mem*block* jika terjadi *scan port* kembali pada jaringan.

OS Finger Printing

OS Finger Printing merupakan suatu proses yang digunakan untuk mengetahui sistem operasi apa yang digunakan pada komputer target. Pada dasarnya terdapat dua cara pendeteksian yaitu *Active Finger Prin*ting dan *Passive Finger Printing* .Dalam uji coba ini yang kita gunakan adalah *Active Finger Printing* yang telah digabungkan dengan *Port Scanning* .Namun pada *OS Finger Printing* keputusan untuk mem*block* telah dinyatakan sebelumnya pada *Gateway StaraGuard* sehingga hasil pendeteksiannya tidak dapat terlihat pada diagram secara langsung tapi berdampak pada hasil OS *Finger Printing* yang dilakukan pada IP 192.168.0.8. Berikut ini tampilan dari OS *Finger Printing*.



Gambar 4.12. Tampilan Tidak terdeteksinya Sistim Operasi Windows 2000 Sp 4

Initiating OS detection (try #1) against 192.168.0.8

Retrying OS detection (try #2) against 192.168.0.8

Retrying OS detection (try #3) against 192.168.0.8

Retrying OS detection (try #4) against 192.168.0.8

Retrying OS detection (try #5) against 192.168.0.8

Dari tampilan tersebut terlihat jelas bahwa Operating System yang terdapat dalam jaringan strata guard tidak dapat dideteksi oleh ZenMap hal ini disebabkan karena keputusan untuk tidak memperbolehkan terjadinya pendeteksian OS Finger Printing.

Vulnerability Scanning

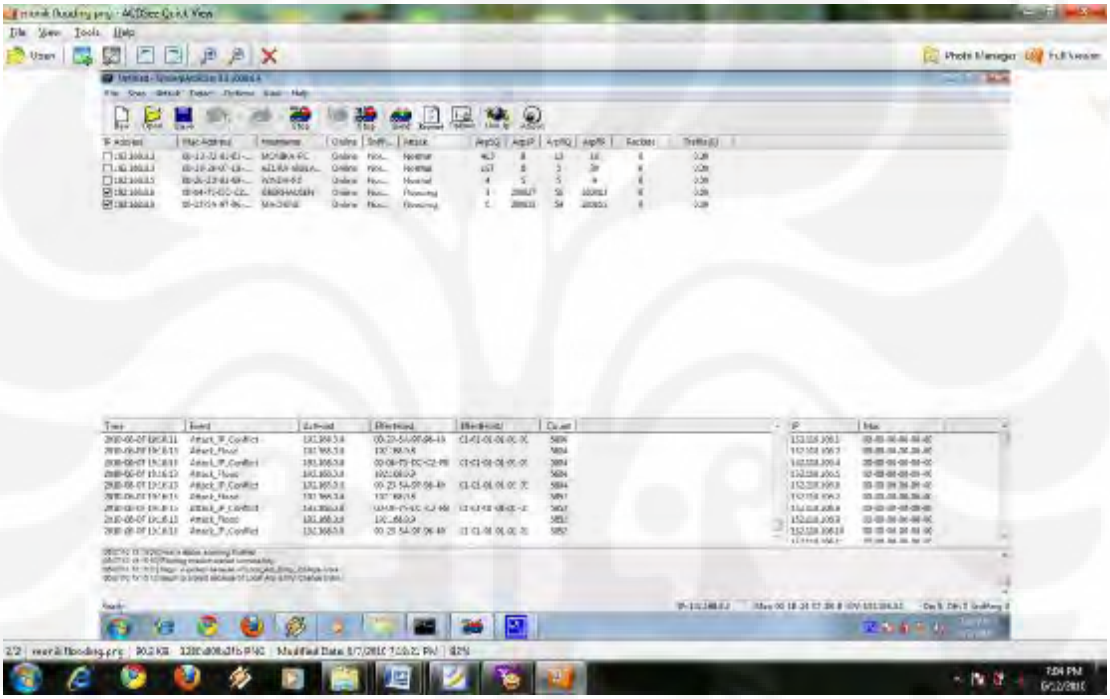
Vulnerability Scanning merupakan suatu proses yang digunakan untuk mengetahui celah keamanan yang terdapat dalam suatu user hal ini biasa dilakukan setelah melakukan *Port scanning* dan setelah melakukan OS *Finger Print.* Pada percobaan ini uji coba *Vulnerability Scanning* dilakukan bersamaan dengan *port scanning* serta OS *finger Print* sehingga akan menghemat waktu dan *resource* yang digunakan dalam menyelidiki suatu jaringan. Berikut ini adalah tampilan dari hasil *Vulnerability Scanning* yang dilakukan terhadap User dengan IP 192.168.0.8 .

Gambar 4.13. Tampilan Pendeteksian Vulnerability

Dari tampilan berikut dapat terlihat bahwa index dari TCP Sequance adalah 134 dengan tingkat Difficulty Good luck yang memiliki arti bahwa prediksi pengurutan paket TCP pada IP 192.168.0.8 adalah 134 yang mana rentangan penomeran yang ada adalah dari 1- 4,294,967,295 sehingga hasil dari TCP sequence tersebut memiliki arti bahwa prediksi pengurutan TCP sequence dari IP 192.168.0.8 adalah 134.

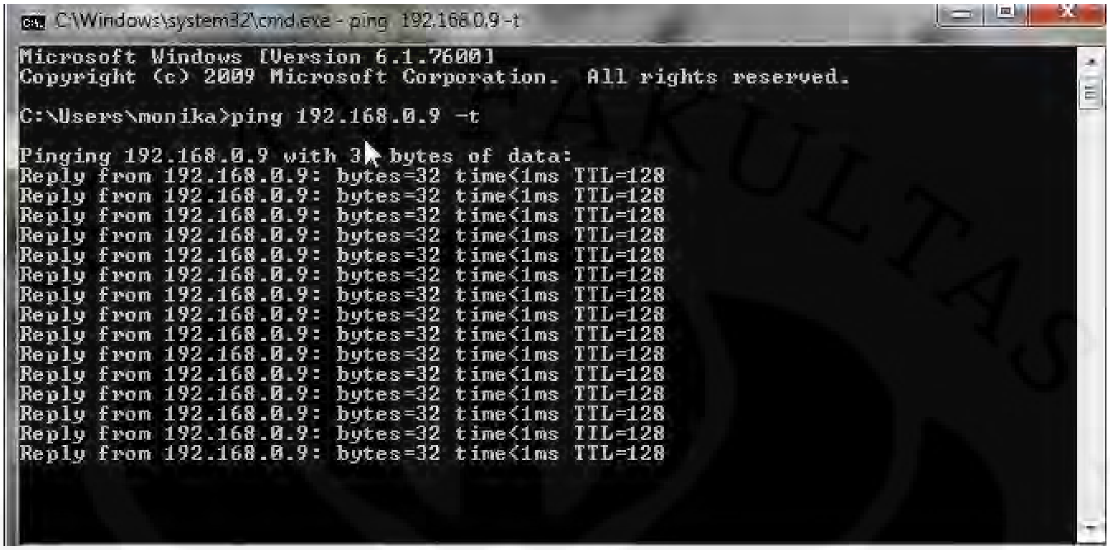
Flooding

Flooding adalah jenis serangan terhadap sebuah komputer atau *server* di dalam jaringan dengan tujuan mengirimkan *request* paket sehingga menghabiskan sumber *(resource)* yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut

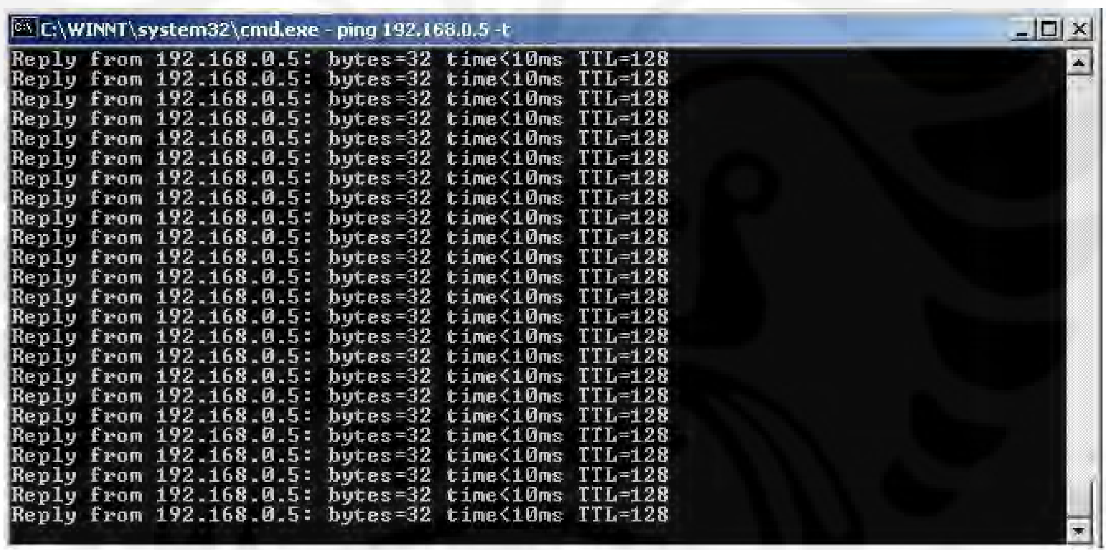


Gambar 4.14.Win ARP Attack Saat melakukan Flooding

Dalam uji coba sistem ini dilakukan 2 kali uji coba sistem yaitu pada IP address 192.168.0.5 dan 192.168.0.8. Dari uji coba tersebut didapat bahwa selama sistem berada dalam jaringan *Strata Guard* maka sistem tidak akan mengalami RTO (*Request Time Out*) hal ini terlihat jelas dalam gambar dibawah ini.

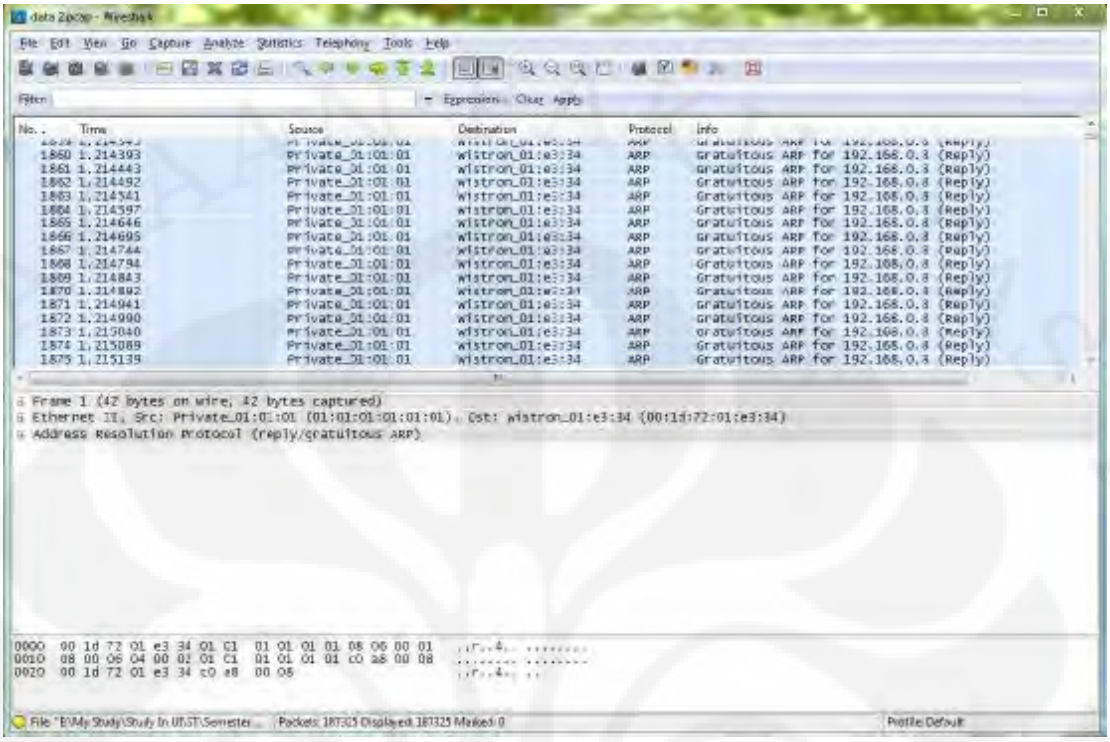


Gambar 4.15 Hasil PING Ip dari IP 192.168.0.5



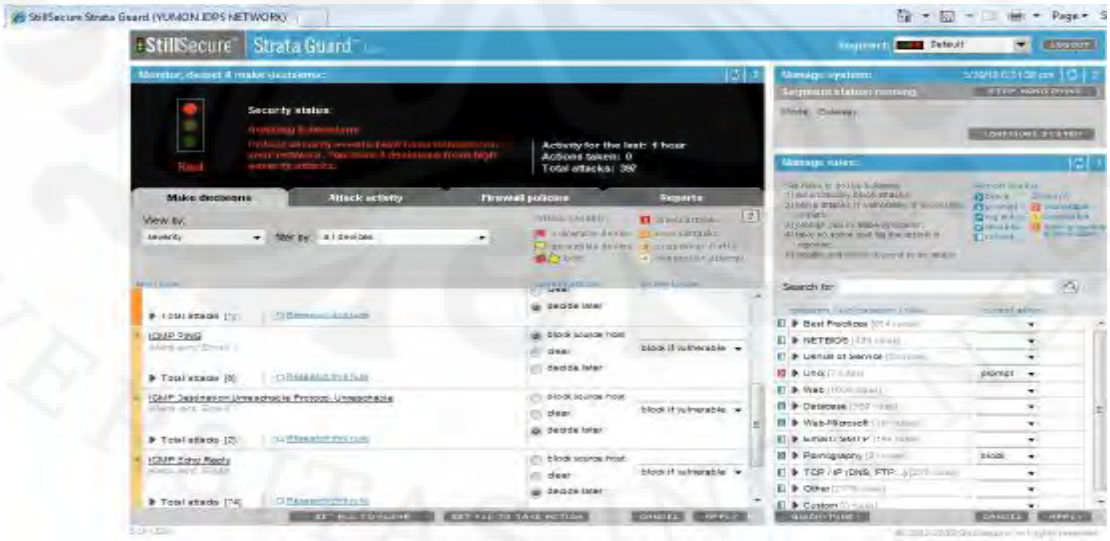
Gambar 4.16. Hasil PING Ip dari IP 192.168.0.8

Dari uji coba *flooding* yang dilakukan dilakukan *capture* paket oleh wireshark yang mana pada *capture* paket tersebut didapati bahwa paket yang dihasilkan semuanya dinyatakan sebagai paket *gratuitous* ARP yang mana paket yang dikirimkan adalah paket pengecekan nomor IP yang akan menyebabkan IP konflik yang tentunya akan menyebabkan RTO (*Request Time Out*) karena IP tujuan tidak mengetahui alamat IP yang benar dari pengiriman paket PING ICMP.



Gambar 4.17. Capture Tampilan Wire shark Saat jaringan dilakuka Flooding

Untuk itu sistem IPS akan memblok alamat IP yang mengirimkan paket secara berlebihan. Berikut ini adalah tampilan dari *Gateway Strata Guard* saat mendeteksi terjadinya *anomaly* paket.



Gambar 4.18.Tampilan StaraGuard Mendeteksi terjadinya Flooding

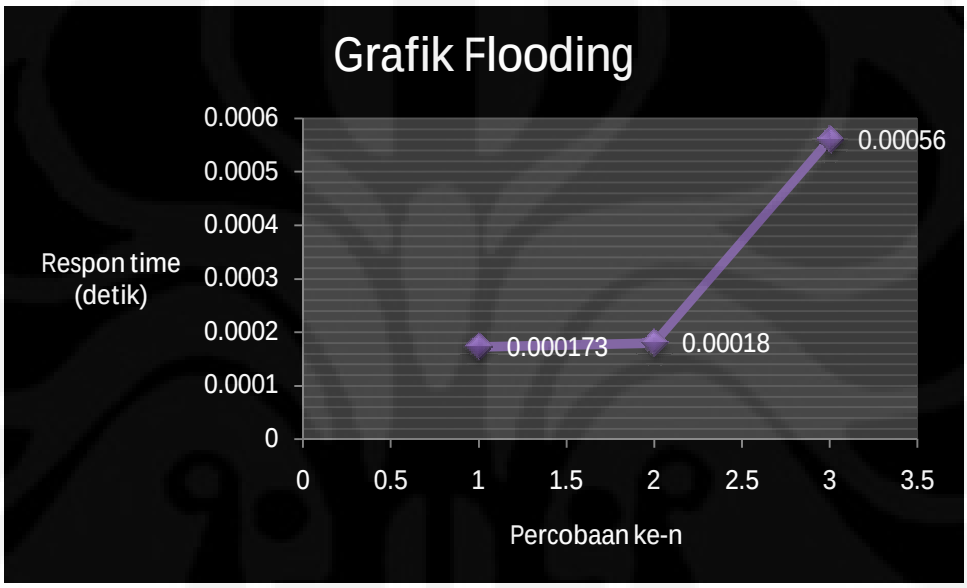
Gambar 4.19. Tampilan Strata Guard untuk menganalisa serangan.

4.2.2 Respon Time

Tingkat kehandalan dari IDS *server* dapat dilihat dari beberapa parameter. Salah satu parameter yang penting adalah *response time. Response time* adalah waktu yang dibutuhkan untuk *server* merespon sebuah serangan. Pada percobaan ini pengukuran *response time* dilakukan pada saat serangan dimulai sampai pada saat *server* pertama kali memberikan respon. Di bawah ini adalah *response time* yang dihasilkan dari 2 buah percobaan yang dilakukan yaitu *Port Scaning* dan *Flooding.*Untuk metode *Flooding* penyerangan dilakukan sebanyak 3 kali dimana menggunakan sistem operasi yang sama yaitu windows 2000 SP 4. Dari data-data yang ada didapat tabel sebagai berikut:

Tabel 4.1 Percobaan Respon Time Flooding

Percobaan ke-n	Respon Time
1	0.000173 detik
2	0.00018 detik
3	0.00056 detik
Rata-Rata	0.0019 detik



Gambar 4.20 Grafik Respon Time Terhadap Flooding

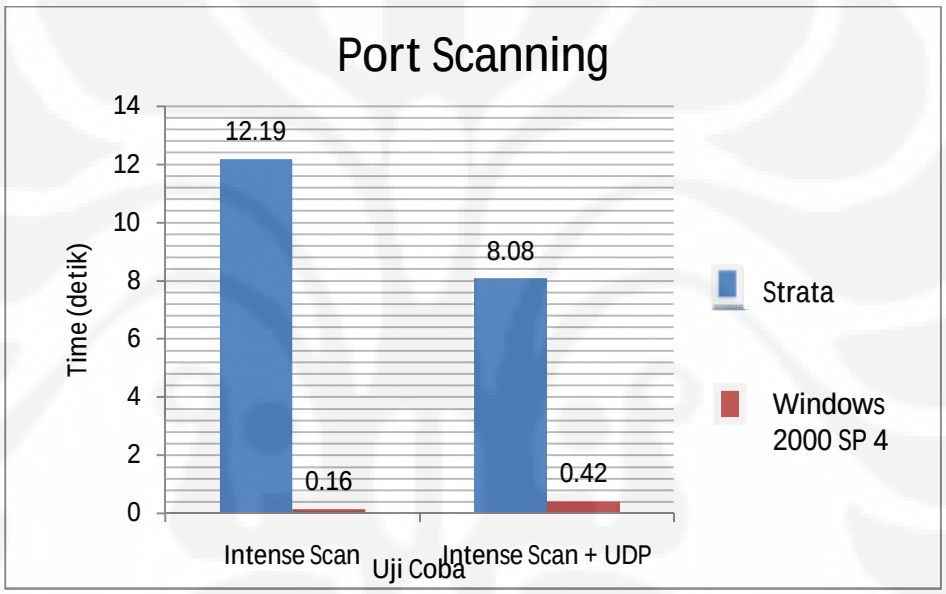
Dari percobaan yang dilakukan dilihat bahwa rata-rata respont time yang dilakukan oleh strata guard adalah 0.00019 detik dari hasil tersebut terlihat jelas bahwa dari data tersebut bahwa respon time yang terjadi untuk percobaan flooding semakin lama-semakin meningkat hal ini disebabkan karena prevention system yang dijalankan oleh IDPS sehingga flooding yang dilakukan dianggap sebagai paket yang tidak penting untuk diteruskan terlihat dengan jelas semakin pada percobaan ketiga yang mana respon time aktivitas flooding baru di respont pada saat detik ke 0,00056 detik.

Untuk respont time saat terjadi nya port scanning dapat terlihat dari grafik yang ada dimana pada garfik ini port scan yang dilakukan adalah intense Port scanning dan

Intense Port Scanning Plus UDP yang mana dilakukan terhadap sistim Strata Guard dan Windows 2000 SP 4 Berikut ini adalah Table dan grafiknya.

Table 4.2 Intense Scan Port dan Intense Scan + UDP

Operating System	Intense Scan Port	Intense Scan + UDP
Strata Guard	12.19 detik	8.08 detik
Win 2000	0.16 detik	0.42 detik



Gambar 4.21 Diagram Intense Scan Port dan Intense Scan + UDP

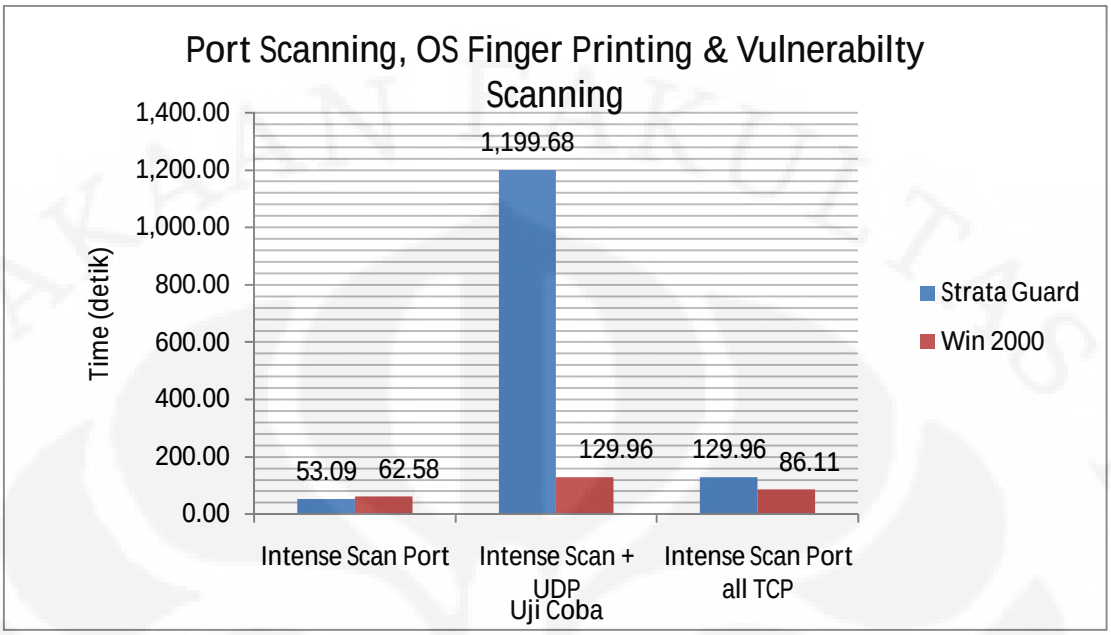
Dari hasil table diatas terlihat bahwa perbandingan respon time antara strata guard dan windows 2000 SP 4 terjadi perbedaan yang signifikan untuk intens Scan waktu yang dibutuhkan software flooding untuk melakukan pengecekkan port pada srata guard yaitu 12,19 s sedangkan pada windows 2000 Sp 4 waktu yang dibutuhkan adalah 0.16 s hal ini disebabkan karena pada saat pengecekan port pada strata guard, Strata Guard melakukan pertahanan dan membaca apakah pengecekan port tersebut masuk kedalam kategori membahayakan yang patut untuk di block.Sedangkan pada saat melakukan intense scan pada windows 2000 Sp 4 dapat berjalan lebih cepat dikarenakan sistim strata guard telah memblock sistim port scan sehingga port

scanning yang dilakukan menjadi gagal dalam mendapatkan informasi secara benar akan kondisi port yang terbuka. Hal yang sama pun berlaku pada Intense scan Plus UDP dimana stratagurd akan melakukan mekanisme pertahanan sehingga waktu dalam melakukan scanning port pun menjadi lebih lama dan hasil yang didapatkan tidaklah valid.

Untuk pengujian port scanning , OS Finger Printing dan vulnerability scanning yang dilakukan dengan pengujian intense scan , intense scan plus UDP , dan Intense scan port all TCP didapatkan bahwa keseluruhan hasil yang didapat menyatakan waktu pengujian terhadap strata gurard lebih lama dibandingkan pada windows 2000 Sp 4 hal ini bias terlihat pada table dan grafik dibawah ini.

Table 4.3 Hasil Waktu Pengujian port pengujian intense scan , intense scan plus UDP , dan Intense scan port all TCP

Operating System	Intense Scan Port	Intense Scan + UDP	Intense Scan Port all TCP
Strata Guard	53.09 detik	1,199.68 detik	282.11 detik
Win 2000	62.58 detik	129.96 detik	86.11 detik



Gambar 4.22 Grafik Hasil Waktu Pengujian port intense scan , intense scan plus UDP , dan Intense scan port all TCP

Kondisi ini terjadi disebabkan karena sistim mekanisme IDPS yang terdapat pada sistim sehingga dapat menahan serangan yang didalamnya terdapat pengujian port scanning , OS Finger Printing dan vulnerability scanning sehingga hasil yang didapatkan tidaklah valid sedangkan untuk windows 2000 Sp 4 mengapa mendapatkan waktu pengujian yang pendek hal ini disebabkan sistim operasi Windows 2000 Sp 4 terdapat dalam jaringan IDPS yang tentunya ketika terjadi serangan port scanning , OS Finger Printing dan vulnerability scanning sudah di block terlebih dahulu oleh sistim yang tentu saja akan menghasilkan hasil port scanning , OS Finger Printing dan vulnerability scanning yang tidak dapat dipercaya.

BAB V

KESIMPULAN

1. Pada *functionality test* IDPS *Gateway* dapat merespon adanya serangan berupa :
   - *IP Scanning*
   - *Port Scanning*
   - *Os Finger Printing*
   - *Vulnerability Scaning*
   - *Flooding*

2. Respon Time Pada saat *Flooding* akan semakin meningkat dengan rata-rata 0.00019 detik. Hal ini disebabkan IDPS gateway melakukan tindakan *prevention system* pada alamat IP tersebut sehingga hasil *flooding* paket ARP yang dikirim tidak dilanjutkan ke alamat IP sebenarnya.

3. Pemindaian port yang dilakukan pada IDPS gateway akan memakan waktu yang sangat lama yaitu 12.19 detik untuk intense scan port dan 8.08 detik untuk intense scan port + UDP dibandingkan pemindaian port pada windows 2000 Sp 4 yaitu 0.16 detik untuk intense scan port dan 0.42 detik untuk intense scan port + UDP hal ini disebabkan karena adanya mekanisme pengecekkan dan penyelubungan port yang terbuka oleh IDPS serta adanya *prevention system* untuk melakukan pengeblockkan alamat IP jika dinilai alamat IP tersebut membahayakan jaringan.

4. Hasil yang dihasilkan oleh *OS Finger printing* software untuk menguji kemampuan jaringan IDPS Strata Guard dalam menangani pemindaian jenis sistim operasi pada jaringan bisa dikatakan tidak valid karena IDPS Strata Guard telah melakukan penyelubungan jenis sistim operasi yang digunakan sehingga data jenis sistim operasi yang dihasilkan dari software OS Finger Printing tidaklah valid.

DAFTAR ACUAN

[1] "Internet World Stats" , diakses tanggal 29 Mei 2010 dari : http://www.internetworldstats.com/stats.htm

[2] Mirkovic, Jelena, dkk, "Internet Denial of Service". Prentice Hall, 2004.

[3] "CSI/FBI Computer Crime and Security Survey" 2005, didapat dari : Distributed Denial of Service.ppt, anegroni@cisco.com

[4] Tannenbaum, Andre S, "Computer Network", 4th Edition, Prentice Hall, 2003.

[5] "The Physical Layer", diakses tanggal 29 Mei 2010 dari : http://www.mindspring.com/~cari/networks/physlayer.html

[6] Sito," STO CEH 100% ",Jasakom, Juni 2009

[7] Strata Guard installation.pdf www.sgfree.stillguard.com tanggal akses 2 Mei 2010

[8] "DNS and IP configuration" www.stillguard.com tanggal akses 30 Mei 2010

[9] "Segment Configuration" www.stillguard.com tanggal akses 2 Juni 2010

[10] "Firewall Configuration" www.stillguard.com tanggal akses 10 juni 2010

[11] " Rules Up Data" www.stillguard.com tanggal akses 10 Juni 2010

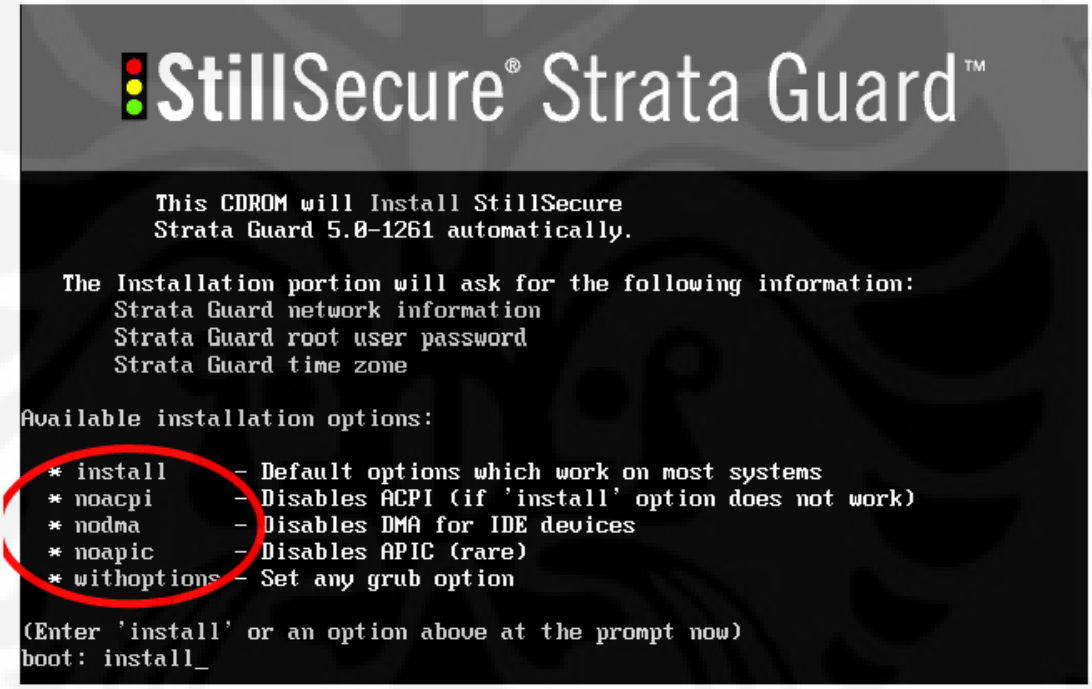
[12] "Begins of Strata Guard " www.stillguard.com tanggal akses 10 juni 2010

LAMPIRAN

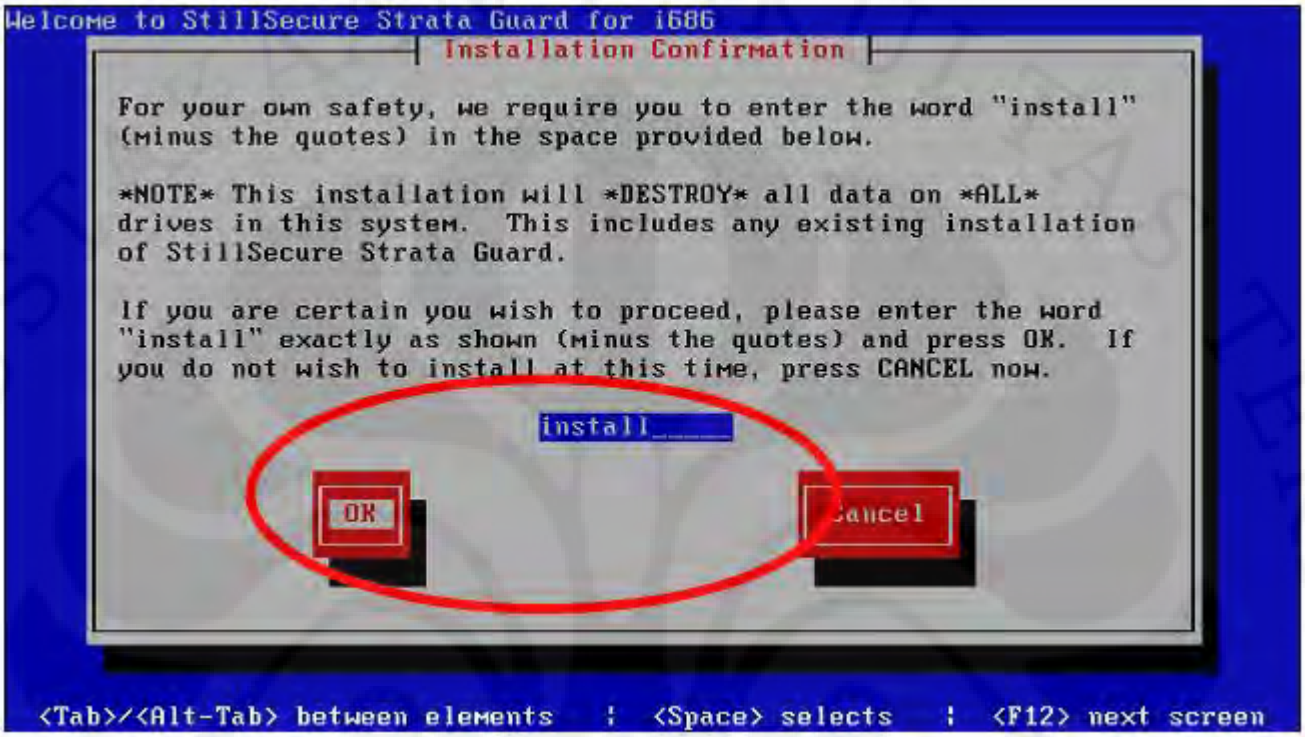
Lampiran I Instalasi Strata Guard

1. Memasangkan 3 buah NIC (*Network Interface Card*) yang akan digunakan untuk memonitoring jaringan, *Firewall* , dan *Router*.

2. Melakukan instalasi *Strata Guard* didalam *hardisk.*

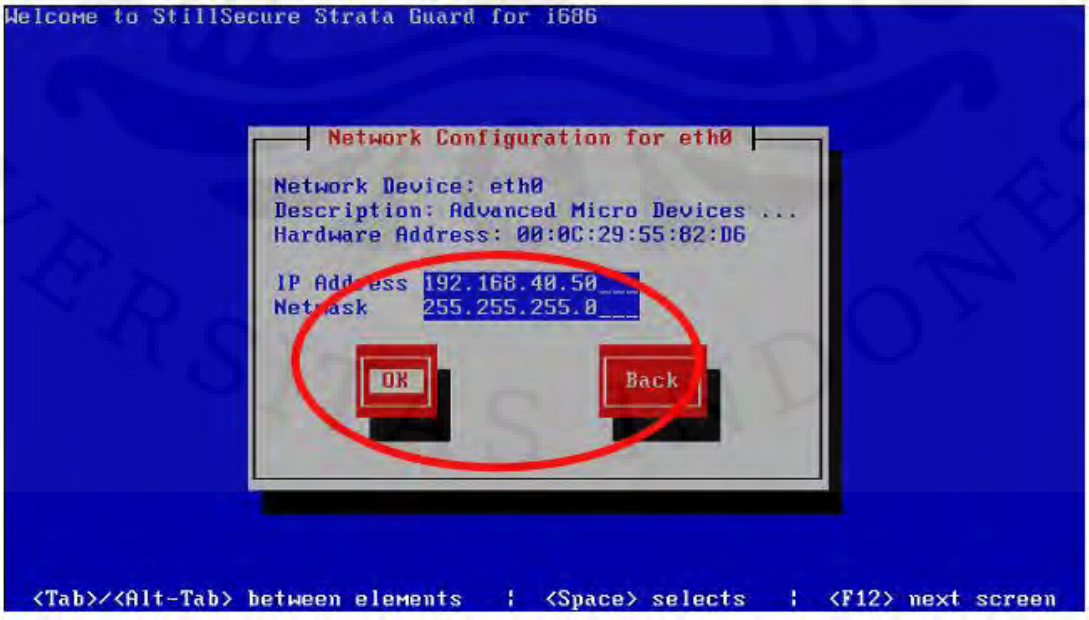


Gambar 1 Tampilan Awal Saat Instalasi *Strata Guard*



Gambar 2 Tampilan Gambar Saat akan melakukan Instalasi *Strata Guard*

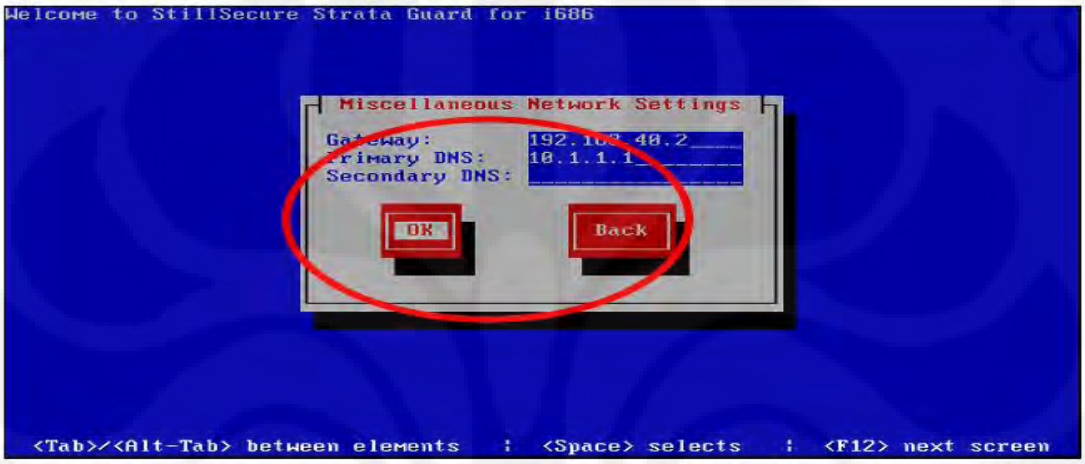
3. Memberikan IP yang akan digunakan sebagai *gateway* dalam hal ini nomor yang digunakan adalah 192.168.0.11



nesia

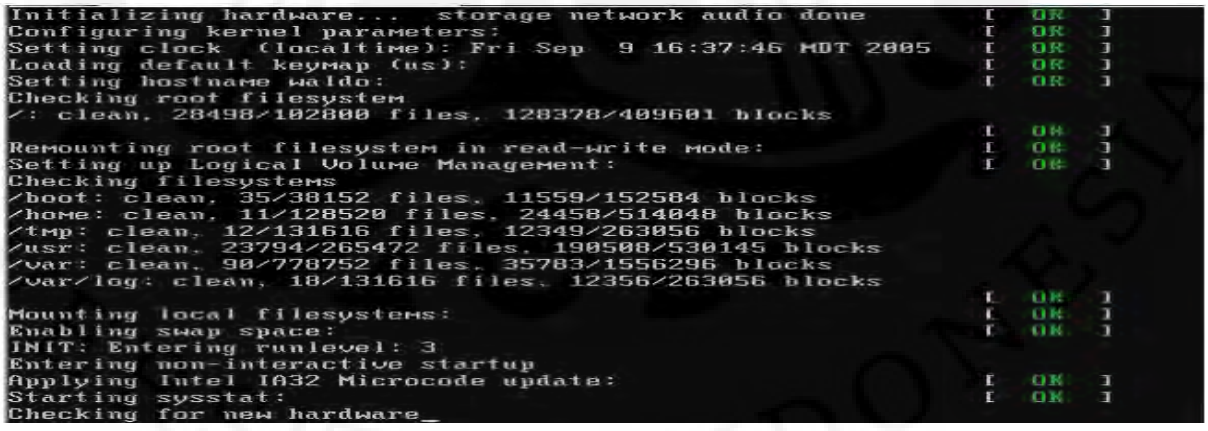
Gambar 3 Tampilan *Stara Guard* saat meminta IP *address*

4. Memberikan nomor *gateway* yang akan di gunakan oleh *Strata Guard* untuk melakukan koneksi ke luar dalam hal ini nomor IP yang digunakan adalah 192.168.0.2.



Gambar 4 Tampilan StaraGuard saat meminta IP *Gateway*

5. Saat setelah instalasi selesai komputer akan *reboo*t dan akan menampilkan tampilan seperti dibawah ini



Gambar 5 Tampilan saat akan masuk *booting* dan setelah masuk kedalam sistem

6. Setelah sistem ter*install* maka kita akan masuk ke dalam jaringan dengan menggunakan *browser* dan masukkan nomor IP yang kita telah pilih yaitu https://192.168.0.11

Gambar 6 Tampilan *Strata Guard* saat login dari *browser* dalam jaringan

Lampiran 2 Hasil Data Finger Printing

Berikut ini adalah lampiran saat melakukan Os finger Printing pada IP 192.168.0.8

Initiating OS detection (try #1) against 192.168.0.8

Retrying OS detection (try #2) against 192.168.0.8

Retrying OS detection (try #3) against 192.168.0.8

Retrying OS detection (try #4) against 192.168.0.8

Retrying OS detection (try #5) against 192.168.0.8

NSE: Script scanning 192.168.0.8.

NSE: Starting runlevel 1 scan

Initiating NSE at 11:29

Completed NSE at 11:29, 0.06s elapsed

NSE: Starting runlevel 2 scan

Initiating NSE at 11:29

Completed NSE at 11:29, 22.08s elapsed

NSE: Script Scanning completed.

Host 192.168.0.8 is up (0.0013s latency).

Interesting ports on 192.168.0.8:

Not shown: 995 closed ports

PORT STATE SERVICE VERSION

80/tcp open tcpwrapped

135/tcp open msrpc

Microsoft Windows RPC

139/tcp open netbios-ssn

445/tcp open microsoft-ds Microsoft Windows XP microsoft-ds

1025/tcp open mstask
 Microsoft mstask (task server - c:\winnt\system32\Mstask.exe)

MAC Address: 00:04:75:DC:C2:FB (3 Com)

No exact OS matches for host (If you know what OS is running on it, see http://nmap.org/submit/).

TCP/IP fingerprint:

OS:SCAN(V=5.00%D=6/2%OT=80%CT=1%CU=41451%PV=Y%DS=1%G=Y%M=000475%
TM=4C05DE2E

OS:%P=i686-pc-windows-
windows)SEQ(SP=89%GCD=1%ISR=9A%TI=I%CI=I%II=I%SS=S%TS

OS:=0)SEQ(SP=87%GCD=1%ISR=9B%TI=I%CI=I%II=I%SS=S%TS=0)SEQ(SP=82%GCD=
1%ISR=9

OS:B%TI=I%CI=I%II=I%SS=S%TS=0)SEQ(SP=85%GCD=1%ISR=9B%CI=I%II=I%TS=0)S
EQ(SP=

OS:86%GCD=1%ISR=9B%CI=RD%II=I%TS=0)OPS(O1=M5B4NW0NNT00NNS%O2=M5B
4NW0NNT00NN

OS:S%O3=M5B4NW0NNT00%O4=M5B4NW0NNT00NNS%O5=M5B4NW0NNT00NNS%O
6=M5B4NNT00NNS)

OS:WIN(W1=FB7C%W2=FB7C%W3=FB7C%W4=FB7C%W5=FB7C%W6=FB7C)ECN(R=
Y%DF=Y%T=81%W=

OS:FB7C%O=M5B4NW0NNS%CC=N%Q=)T1(R=Y%DF=Y%T=81%S=O%A=S+%F=AS%
RD=0%Q=)T2(R=Y%

OS:DF=N%T=81%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T3(R=Y%DF=Y%T=81%W=
FB7C%S=O%A=S+%F

OS:=AS%O=M5B4NW0NNT00NNS%RD=0%Q=)T4(R=Y%DF=N%T=81%W=0%S=A%A=
O%F=R%O=%RD=0%Q

OS:=)T5(R=Y%DF=N%T=81%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=N
%T=81%W=0%S=A

OS:%A=O%F=R%O=%RD=0%Q=)T7(R=Y%DF=N%T=81%W=0%S=Z%A=S+%F=AR%O=
%RD=0%Q=)T7(R=Y

OS:%DF=N%T=81%W=0%S=Z%A=O%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=81%IPL=38%UN=0%RIPL

OS:=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=S%T=81%CD=Z