IMPLEMENTASI *INTRUSION DETECTION SYSTEM* SEBAGAI KEAMANAN *WEB SERVER* UNIVERSITAS DEHASEN BENGKULU

Khairil[1], Toibah Umi Kalsum [2]

[1] Sistem Informasi, Ilmu Komputer, Universitas Dehasen Bengkulu, Jln. Meranti raya No. 32 Sawah Lebar Bengkulu
[2] Teknik Komputer, Ilmu Komputer, Universitas Dehasen Bengkulu, Jln. Meranti raya No. 32 Sawah Lebar Bengkulu

[1]khaereal@yahoo.com
[2]ci2k_umie@yahoo.co.id

Abstrak: Firewall mampu menghadapi ancaman terhadap sistem keamanan *web server*. Namun *firewall* masih dapat dilewati oleh *hacker* dengan menyerang *web server* melalui celah keamanan yang terbuka pada *firewall.* Disaat firewall di eksploitasi oleh *hacker* maka *Intrusion Detection System* (IDS) berperan sebagai pemberi peringatan adanya ancaman. Penelitian dilaksanakan di UPT Puskom Universitas Dehasen Bengkulu diruangan *server* pada bulan Juni sampai dengan September 2014. Pelaksanaan penelitian ini dilakukan beberapa tahap. Tahap pertama adalah penginstalan *software snort*. Tahap kedua adalah konfigurasi *snort* s, konfigurasi *rule snort* dan konfigurasi *output log file*. Tahap ketiga adalah melakukan pengetesan fungsi IDS. Tahap keempat adalah pengujian dan analisa IDS. Hasil dari penelitian ini adalah aplikasi *snort* berfungsi sebagai *network intrusion detection system* dalam mendeteksi penyusup yang melakukan *scanning port. Snort* menampilkan peringatan ancaman secara *real time* dalam bentuk tanggal, waktu, *ip address* pengirim dan jenis ancamannya. *Snort* dalam merespon adanya ancaman membutuhkan waktu 2 menit. *Snort* juga menyimpan alert dalam bentuk *log* kedalam file alert.ids. *File log* ini sebagai analisa bagi administrator jaringan untuk meningkatkan keamanan terhadap *web server* Universitas Dehasen Bengkulu.
Kata Kunci: *Intrusion Detection System*, Keamanan Sistem Informasi, *Web Server*

Abstract: Firewall is able to deal with the threats against the security system of the web server. While firewall is operated by hackers, the Intrusion Detection System (IDS) plays a part as a warning announcer of the threat. This research is conducted in order to find out or log-file the activity of data packets periodically on the network connected to the web server in Dehasen University of Bengkulu, especially the data packets that have threaten the security of web server such as intrusions or attacks. The implementation of the research carried out several stages. The first stage is the installation software snort. The second stage is the snort configuration. The third stage is to test. The fourth stage is the trial and the analysis of IDS. The results of this research are the application has a function as a snort network intrusion detection system in detecting the intruders. Snort displays the threat alerts periodically. Snort takes 2 minutes to respond the threat. In addition, snort also saves the alerts in the form of log files into alert.ids.

Keywords : Intrusion Detection System, Security of Information System, Web server

I. PENDAHULUAN

1.1. Latar Belakang

Keamanan sistem informasi sangat dibutuhkan pada saat perkembangan teknologi komunikasi jaringan yang cukup pesat saat ini. Kemudahan dalam mendapatkan Informasi selalu tersedia saat dibutuhkan melalui jaringan internet. Informasi yang diperoleh tersebut dengan harapan tidak dimodifikasi oleh orang yang tidak berhak (*attacker*) dalam perjalanan informasi tersebut.

Salah satu cara mengamankan informasi pada jaringan komputer adalah dengan memasang teknologi *firewall*. *Firewall* melakukan kebijakan

keamanan dengan memberikan aturan-aturan (*Rule Filter*) pada jaringan untuk akses keluar masuknya paket data pada jaringan. Namun keamanan yang dilakukan firewall tidak dapat sepenuhnya dijamin. Karena biasanya *firewall* dirancang hanya untuk memblokir trafik mencurigakan tanpa membedakan trafik mana yang berbahaya dan trafik mana yang tidak berbahaya. Sehingga paket yang dicurigai akan langsung ditindaki oleh *firewall*.

Misalnya *firewall* melakukan kebijakan pada port 23 yang digunakan protokol Telnet untuk memblokir port tersebut. *Attacker* tidak dapat melaksanakan panetrasi pada port yang diblokir *firewall* tersebut. Sedangkan pada port 80 yang digunakan untuk *protocol* http kebijakan *firewall* adalah status *allow* (dibuka). Jika port tersebut di blokir oleh administrator akan mengakibatkannya klien tadak dapat melakukan *browsing* internet. *Attacker* dapat memanfaatkan port 80 untuk melakukan *ekploitasi* http. Sehingga *attacker* dikatakan sudah berhasil mem by-pass ke web server sehingga *firewall* disini dikatakan sudah tidak digunakan lagi.

Intrusion detection system (IDS) merupakan sebuah perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS dapat melakukan inspeksi terhadap lalu lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti percobaan intrusi penyusupan.

Tugas IDS adalah memberikan peringatan kepada administrator jaringan bahwa mungkin ada serangan atau gangguan terhadap jaringan. IDS bekerja dengan cara menggunakan pendeteksian dengan pencocokan lalu lintas data pada jaringan. Hasil monitoring dari IDS dapat ditindaklanjuti oleh administrator jaringan dalam hal mengamankan informasi yang ada pada server.

Ada beberapa sistem informasi berbasis web server pada Universitas Dehasen, seperti website, sistem informasi akademik, perpustakaan, mail serever. Dan beberapa portal yaitu portal alumni. Portal forum, portal *e-jurnal*, *blog* dosen dan lain-lain. Kesemua sistem informasi dan portal tersebut sudah menerapkan sistem keamanan dengan mengandalkan kemampuan *firewall* dalam menghadapi ancaman dari *attacker*.

Aplikasi yang digunakan untuk melakukan pengawasan terhadap paket dalam jaringan (*signature based*), memonitoring keadaan trafik pada jaringan (*anomaly based*) dan pendeteksi dan pemberi peringatan (*passive IDS*). Pada hal hasil monitoring IDS ini sangat penting sekali bagi administrator jaringan dalam mengambil keputusan untuk meningkatkan langkah keamanan jaringan pada universitas dehasen Bengkulu selanjutnya.

Dari latar belakang diatas, penulis melihat bahwa keamanan jaringan pada web server Universitas Dehasen sudah dibangun dengan menggunakan teknologi *firewall*. Namun *attacker* masih dapat membypass *firewall* tanpa diketahui oleh administrator jaringan. Penulis disini mengaplikasikan IDS dengan *software snort* untuk memonitor dari serangan *attacker* yang menyusup ke web server Universitas Dehasen Bengkulu.

1.2. Tujuan Penelitian

Penelitian ini dilakukan dengan tujuan untuk mengetahui secara *real-time* ataupun *log-file* aktifitas paket-paket data pada jaringan yang terkoneksi dengan *web server* Universitas Dehasen Bengkulu. Terutama paket data yang mengancam terhadap keamanan web server seperti penyusupan atau penyerangan. Hasil dari dibangunnya system IDS merupakan laporan yang dapat digunakan bagi

administrator jaringan universitas untuk menganalisa keamanan terhadap jaringan komputer universitas.

1.3. Target Luaran

Membangun sistem yang mampu memantau aktifitas paket data yang dapat menjadi ancaman dari penyusupan terhadap web server Universitas Dehasen Bengkulu. Ancaman tersebut ditampilkan dalam bentuk alert dari snort secara real time dan hasil outputnya juga akan tersimpan dalam bentuk log file yang dihasilkan oleh snort.

II. KAJIAN LITERATUR

2.1. Jaringan Komputer

Jaringan komputer adalah kumpulan dua atau lebih komputer yang saling berhubungan satu sama lain untuk melakukan komunikasi data dengan menggunakan *protocol* komunikasi melalui media komunikasi (kabel atau nirkabel), sehingga komputer – komputer tersebut dapat saling berbagi informasi, data, program-program, dan penggunaan perangkat keras secara bersama. Dengan adanya jaringan komputer komunikasi data dapat berupa teks, gambar, video dan suara [1].

1. *Local Area Network (LAN)*

   LAN adalah sebuah jaringan komputer dengan jangkauan area yang terbatas dan hubungan fisik antar komputer saling berdekatan. Misalkan Jaringan komputer di sebuah kantor, jaringan komputer di sebuah ruangan kerja (laboratorium).

   Manfaat menggunakan LAN: Pertukaran file, data antar komputer dapat dilakukan dengan mudah (*file sharing*). Penggunaan printer dapat dilakukan oleh semua pengguna *(printer sharing)*. Pertukaran data antar komputer dapat dikendalikan sehingga keamanan data dapat terjaga. Proses *backup* data menjadi lebih mudah dan cepat. Komunikasi antar user dapat dilakukan dengan *email* atau *chat*. Komputerasi jaringan mudah dan cukup murah.

2. Metropolitan Area Network (MAN)

   MAN biasanya meliputi area yang lebih besar dari LAN, area yang digunakan adalah dalam sebuah Negara. Jaringan komputer menghubungkan beberapa buah jaringan - jaringan LAN kedalam lingkungan area yang lebih besar, sebagai contoh: jaringan pada BANK (sistem online Perbankan). Setiap bank memiliki kantor pusat dan kantor cabang. Disetiap kantor baik kantor pusat maupun kantor cabang memiliki LAN, Penggabungan LAN-LAN disetiap kantor ini akan membentuk sebuah MAN.

   Manfaat menggunakan MAN: Server kantor pusat dapat berfungsi sebagai pusat data dari kantor cabang. Transaksi yang Real Time (data di server pusat diupdate saat itu juga, contoh ATM Bank untuk wilayah Nasional). Komunikasi antar kantor bias menggunakan *e-mail*, *chatting* dan *video conference* (ViCon).

3. Wide Area Network (WAN)

   WAN adalah jaringan komputer dengan jangkauan area geografi yang paling luas, antar Negara, antar benua bahkan keluar angkasa (sebagai contoh jaringan internet yang menggunakan sistem koneksi satelit).

   Manfaat menggunakan WAN: Penggunaan kartu kredit di seluruh dunia, Pengambilan uang dengan jaringan internasional (ATM Internasional). Komunikasi antar kantor biasa menggunakan *e-mail*, *chatting* dan *video conference (ViCon)*. Pooling data dan update data antar kantor dapat dilakukan setiap hari pada waktu yang ditentukan. Data dapat dikirim melalui *e-mail*.

2.2. Web Server

Web Server adalah aplikasi (*software*) yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* [2]. Komputer yang sudah terkomputer dengan *web server* komputer tersebut dapat di berfungsi sebagai *server* yang dapat melayani permintaan HTTP dari klien.

Web server secara umum terbagi menjadi dua tipe yaitu *web server* secara *local* atau *offline* dan *web server online* atau *web server* yang terhubung dengan internet. Jenis-jenis web server antara lain :

1. *Apache Web Server*

   Apache web server adalah *server web* yang dapat dijalankan di banyak sistem operasi (UNIX, BSD, Linux, Microsoft Windows dan Novell Netware serta *platform* lainnya) yang berguna untuk melayani dan mengfungsikan situs *web*. Protokol yang digunakan untuk melayani fasilitas web atau www ini menggunakan HTTP. *Apache* memiliki fitur canggih seperti pesan kesalahan yang dapat dikonfigurasi, autentikasi berbasis data dan lain-lain. Antarmuka pengguna atau tampilan juga didukung berbasis grafis (GUI) yang memungkinan penanganan server menjadi mudah.

2. *Apache Tom Cat*

   Apache Tom Cat merupakan *servlet* atau *JSP container* yang dibuat oleh *Apache Software Foundation*. *Container* yang bisa dibilang *server* untuk menjalankan bahasa pemrograman web JSP *(Java Server Pages)*.

3. *Internet Information Service (IIS)*

   IIS atau *Internet Information Service* adalah sebuah HTTP *web server* yang digunakan dalam sistem operasi *server windows*.

2.3. Keamanan Sistem Informasi

Keamanan komputer *(computer security*) melingkupi empat aspek, yaitu *privacy, integrity, authentication,* dan *availability*. Selain keempat hal di atas, masih ada dua aspek lain yang juga sering dibahas dalam kaitannya dengan *electronic commerce*, yaitu *access control* dan *non-repudiation* [3].

a. *Privacy/Confidentiality*

   Inti utama aspek *privacy* atau *confidentiality* adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. *Privacy* lebih kearah data-data yang sifatnya privat sedangkan *confidentiality* biasanya berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu (misalnya sebagai bagian dari pendaftaran sebuah servis) dan hanya diperbolehkan untuk keperluan tertentu tersebut. Contoh hal yang berhubungan dengan *privacy* adalah *e-mail* seorang pemakai (*user*) tidak boleh dibaca oleh administrator.

b. *Integrity*

   Aspek ini menekankan bahwa informasi tidak boleh diubah tanpa seijin pemilik informasi. Adanya virus, *trojan horse*, atau pemakai lain yang mengubah informasi tanpa ijin merupakan contoh masalah yang harus dihadapi. Sebuah *e-mail* dapat saja “ditangkap” (*intercept*) di tengah jalan, diubah isinya (*altered*, *tampered, modified*), kemudian diteruskan ke alamat yang dituju. Dengan kata lain, integritas dari informasi sudah tidak terjaga. Penggunaan *enkripsi* dan *digital signature*, misalnya, dapat mengatasi masalah ini.

c. *Authentication*

   Aspek ini berhubungan dengan metoda untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan

informasi adalah betul-betul orang yang dimaksud, atau *server* yang kita hubungi adalah betul-betul *server* yang asli.

d. *Availability*

Aspek *availability* atau ketersediaan berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi. Contoh hambatan adalah serangan yang sering disebut dengan "*denial of service attack*" (DoS attack), dimana *server* dikirimi permintaan (biasanya palsu) yang bertubitubi atau permintaan yang diluar perkiraan sehingga tidak dapat melayani permintaan lain atau bahkan sampai *down*, *hang*, *crash*.

2.4. Intrusion Detection System

Ada 2 bentuk *intrusion detection sistem* yaitu IDS berbasis jaringan dan IDS berbasis *host* [4]. *Network-Based IDS* (NIDS) menempati secara langsung pada jaringan dan melihat semua aliran yang melewati jaringan. *Host-Based IDS* (HIDS) merupakan aplikasi perangkat lunak khusus yang diinstal pada komputer untuk melihat semua aliran komunikasi masuk dan keluar ke dan dari server tersebut dan untuk memonitor sistem file jika ada perubahan. HIDS sangat efektif untuk aplikasi *internet-accesible*, seperti web atau *e-mail* server karena mereka dapat melihat aplikasi pada source-nya untuk melindungi mereka.

HIDS memonitor server dengan menyediakan informasi terkait dengan hal berikut :

a. Usaha-usaha penyusupan atau perilaku yang mencurigakan oleh pengguna resmi
b. Memindai *host* untuk memastikan meraka dapat memenuhi praktek keamanan yang telah ditentukan seperti memiliki semua *patch* terbaru dan tidak memiliki layanan yang tak semestinya.
c. Manajemen dan sentralisasi kebijakan pemeriksaan, menyuplai *forensic* data, analisis statistik dan dukungan yang jelas, dan dalam *instance-instance* tertentu, beberapa ukuran kontrol akses.

Intrusion Detection System (IDS) dapat didefinisikan sebagai *tool*, metode, sumber daya yang memberikan bantuan untuk melakukan identifikasi, memberikan laporan terhadap aktivitas jaringan komputer [5]. IDS tidak cocok diberi pengertian tersebut karena IDS tidak mendeteksi penyusup tetapi hanya mendeteksi aktivitas pada lalu-lintas jaringan yang tidak layak terjadi. *Intrusion detection system* secara khusus berfungsi sebagai proteksi secara keseluruhan dari sistem yang telah dikomputer IDS. IDS tidak berdiri sendiri dalam melindungi suatu sistem.

Ada beberapa kelebihan dari *Network-based Intrusion Detection System* yaitu :

a. Biaya yang lebih rendah. *Network based* IDS memungkinkan pengawasan yang strategis pada titik akses yang kritis untuk menampilkan *network traffic* untuk beragam sistem yang akan diamati sehingga sistem ini tidak memerlukan perangkat lunak yang digunakan dan diatur pada banyak *host*. Karena kebutuhan titik deteksi yang lebih sedikit, maka biayanya lebih rendah.
b. Deteksi serangan yang tidak terdeteksi oleh *Host-Based* IDS. Sistem ini memeriksa semua paket *header* untuk mencari aktivitas yang mencurigakan. Beberapa serangan yang tidak dapat dideteksi oleh *Host-Based* IDS adalah *IPbased Dos* dan *Fragmented Packet (Tear Drop)*. Serangan tersebut dapat diidentifikasi dengan membaca *header* dari paket yang ada.

c. Kesulitan bagi penyerang untuk menghapus jejak. *Network-based* IDS menggunakan data *live network traffic* untuk mendeteksi secara *real time*.
d. Deteksi respon secara *real-time*. Sistem ini mampu mendeteksi serangan yang sedang terjadi sehingga notifikasi dan respon juga dapat dilakukan dengan cepat.
e. Deteksi serangan yang gagal dan kecenderungan serangan. *Network-based* IDS yang ditempatkan diluar *firewall* dapat mendeteksi serangan yang sebenarnya telah dicegah oleh *firewall*. Data serangan tersebut dapat menjadi bahan evaluasi bagi pengembangan kebijakan selanjutnya.
f. Tidak tergantung pada system operasi, *Network-based* IDS tidak tergantung sistem operasi yang digunakan pada *host* yang dilindungi, karena evaluasi yang dilakukan tidak harus berada pada *host* tersebut.

Sebuah IDS akan mendeteksi semua serangan yang dapat melalui jaringan komputer (internet maupun intranet) ke jaringan komputer yang kita miliki. Sebuah NIDS biasanya digunakan bersamaan dengan *firewall*, hal ini untuk menjaga supaya *snort* tidak terancam oleh serangan.

2.5. SNORT

Snort adalah sebuah aplikasi atau *tool* sekuriti yang berfungsi untuk mendeteksi intrusi-intrusi jaringan (penyusupan, penyerangan, pemindaian, dan beragam bentuk ancaman lainnya), sekaligus juga melakukan pencegahan [6]. *Snort* andal dalam membentuk loging paket-paket dan analisis trafik-trafik secara *real-time* dalam jaringan-jaringan berbasis TCP/IP.

Snort bekerja melakukan analisis *protocol*, pencocokan/pencarian konten, dan biasanya digunakan untuk secara aktif menangkal atau secara pasif mendeteksi suatu ancaman serangan dan probe tertentu, seperti :

a. *Buffer overflow*
b. *Stealth port scan*
c. Serangan aplikasi berbasis web
d. SMB *probe*
e. Usaha-usaha *fingerprint OS* dll.

Snort dibuat untuk *platform UNIX-like*, dan dengan pengguna *windows* mengkomputer *snort* sedikit rumit. Karena pada awalnya *snort* dibuat untuk *platform UNIX-like*, opsi-opsi baris perintah (*command-line*) dan file-file konfigurasi UNIX adalah berformat teks. Sehingga pengguna *windows* yang terbiasa dengan konfigurasi berbasis klik *mouse* akan terasa sangat berat saat berada dilingkungan *command-line*.

Sistem *snort* akan membutuhkan dua komponen, yaitu *library packet capture Winpcap*, dan program *snort* IDS itu sendiri. *Winpcap* (*windows Packet Capture Library*) adalah *driver* untuk penangkap paket-paket yang hilir-mudik dalam jaringan. Secara funsional artinya *WinPcap* menangkap paket-paket dari kabel jaringan dan melemparnya ke program *snort*.

III. METODE PENELITIAN

3.1. Waktu dan Tempat Penelitian

Waktu penelitian dilaksanakan pada bulan Juni 2014 sampai dengan September 2014, sedangkan tempat penelitian dilakukan di Unit Pelaksana Teknis (UPT) Pusat Komputer Universitas Dehasen Bengkulu dengan pertimbangan bahwa UPT. Puskom Unived merupakan unit yang melaksanakan pelayanan terhadap jaringan komputer dan *internet* serta pelayanan sistem informasi pada Universitas Dehasen.

3.2. Pendekatan dan Metode Penelitian

Dalam penelitian ini dilakukaan beberapa tahap untuk membangun *Intrusion Detection System*. Tahapannya dimulai dari pemasangan perangkat keras IDS dan dilanjutkan dengan proses penginstalan *software snort* dan aplikasi pendukung agar *snort* berjalan secara optimal.

Konfigurasi terhadap perangkat *snort* dan melakukan *update rule snort* merupakan tahap setelah penginstalan. Tahap akhir adalah pengujian IDS terhadap ancaman yang terjadi *web server*. Salah satu pengujian dilakukan dengan cara *scanning* terhadap *web server* yaitu *tool* NMAP *(network mapper)*. Nmap digunakan untuk mengetahui *host, service* serta sistem operasi yang digunakan oleh sistem yang akan dimasuki.

Metode penelitian yang digunakan adalah metode studi eksperimen dengan cara melakukan percobaan-percobaan *rule snort* yang *update* untuk mendeteksi ancaman pada *web server*.

3.3. Metode Pengumpulan Data

Untuk mengumpulkan bahan yang diperlukan dalam penyusunan penelitian ini, menggunakan tiga macam metode yaitu:

1. Metode Pustaka

Menggunakan buku-buku, jurnal dan informasi dari internet yang dapat dijadikan sebagai bahan referensi dalam peneyelesaian penelitian ini.

2. Metode Wawancara

Pengumpulan data dengan melakukan tanya jawab langsung kepada pihak yang berkompeten dengan penelitian ini. Wawancara dalam penelitian ini pada pihak administrator jaringan Universitas Dehasen Bengkulu.

3. Metode Studi Laboratorium

Pengumpulan data dengan metode ini yaitu dengan melakukan pengujian terhadap peralatan dan bahan yang digunakan dan menelaah data terhadap ancaman-ancaman yang terjadi waktu penelitian.

3.4. Metode Analisis

Data yang diperoleh berdasarkan hasil pengamatan setelah diimplementasikan sistem yang dibuat. Hasilnya IDS dalam bentuk peringatan pada *log-file* maupun secara *real-time* sebagai pertimbangan dalam meningkatkan keamanan jaringan Universitas Dehasen Bengkulu oleh administrator jaringan.

IV. HASIL DAN PEMBAHASAN

4.1. Spesifikasi Analisa Kebutuhan

Network Intrusion Detection System (NIDS) ditempatkan pada jaringan yang terhubung dengan beberapa *web server*. IDS memberi laporan pada administrator jaringan secara spesifik berupa IP *address source*, *IP addres destination*, waktu penyusupan, paket data yang digunakan dan lain sebagainya.

Intrusion Detection Sistem yang dibangun ini membutuhkan beberapa perangkat keras dan perangkat lunak agar dapat berjalan sebagaimana mestinya.

a. Perangkat Keras

Perangkat keras yang dibutuhkan sistem ini adalah sebuah Personal Komputer dengan spesifikasi :

1. *Processor dual core*
2. *RAM 1 Gb*
3. *Harddisk 320 Gb*
4. *Network Interface Card*
5. *Monitor LED 15,6*
6. *Kabel UTP Cat 5 20 meter*
7. *Crimping Tool* dan *LAN Tester*

b. Perangkat Lunak

Kebutuhan perangakat lunak dalam membangun sistem IDS ini adalah :

1. Sistem Operasi *Windows*
2. *Snort dan winpcap*

4.2. Perancangan Infrastruktur Jaringan dengan Sistem IDS

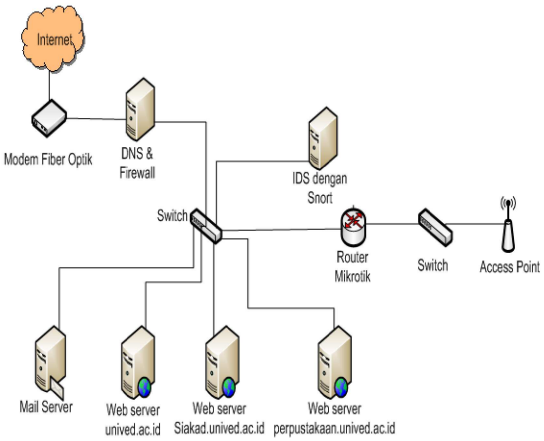
Keamanan data pada sistem informasi webserver saat ini mengandalkan teknologi *firewall*. *Firewall* dijaringan melakukan pemeriksa paket data yang masuk atau keluar sesuai kebijakan yang berlaku.

Firewall dan DNS (*Domain Name System*) pada jaringan universitas dehasen diletakkan dibagian depan koneksi dengan internet, kedua sistem ini di komputerkan dengan sistem operasi *linux centos*.

Semua paket data yang masuk dari internet kedalam jaringan Universitas melalui aturan kebijakan keamanan *firewall*, dan *firewall* akan meneruskan paket data yang dianggap aman menurut *firewall*. Paket data diteruskan ke switch untuk dikirimkan ke masing-masing web server sesuai dengan tujuan ke *web server* yang terkoneksi dengan *switch*.

IDS yang dibangun dipasang pada *switch* yang terhubung pada semua *web server* untuk meberikan peringatan ancaman serangan. Kemungkinan attacker masuk dengan melakukan panetrasi melalui *port* yang terbuka pada *firewall*, sehingga *firewall* tidak mendeteksi adanya serangan pada web server.

Peringatan yang dilaporkan IDS ketika ada ancaman sangat membantu administrator jaringan dalam menambahkan kebijakan baru untuk memperkuat kemanan terhadap *web server* Universitas Dehasen. Gambar 5.1 dibawah ini infrastruktur jaringan web server pada universitas setelah menggunakan IDS.



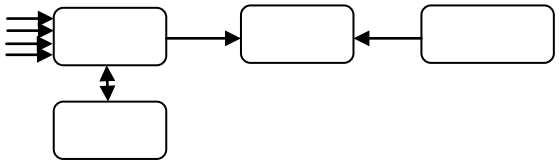
Gambar 1. Infrastruktur jaringan dengan IDS

Dari Gambar 1 terlihat bahwa keamanan *web server mail, unived.ac.id, siakad.unived.ac.id* dan *perpustakaan.unived.ac.id* mengandalkan *firewall* sebagai pemberi kebijakan aman dari ancaman atas informasi yang telah diberikan sistem IDS. Harus di perbaharui terus *rule* keamanan pada IDS sesuai dengan perkembangan keamanan yang berlaku saat ini, kebijakan baru yang ditambahkan pada *firewall* berdasarkan informasi dari IDS.

4.3. Perancangan Sistem IDS

IDS bekerja dengan memiliki penerapan TCP/IP khusus yang dapat mengumpulkan paket data untuk dianalisis. Analisis yang dilakukan sistem adalah dengan melihat aktifitas paket data yang tidak normal pada jaringan. IDS mencocokan data yang tidak normal untuk dikirimkan dalam bentuk peringatan.

Diagram blok yang diterapkan pada IDS ini seperti terlihat pada Gambar 2 dibawah ini.



Gambar 2. Blok diagram sistem

Paket data setelah di saring oleh *firewall* selajutnya masuk menuju IDS *engine* untuk dilakukan *scanning ip header*, *header layer transport, header level layer* aplikasi dan paket *playload*. Kemudian hasil *scanning* tersebut disamakan dengan rule IDS, jika sesuai dengan pola aturan pada rule. Sistem menampilkannya pada layar monitor dan menuliskan aktifitas kejadiannya pada *log*. Sebaliknya jika tidak sesuai dengan pola *rule*, sistem menghentikan proses scanning dan paket diteruskan kealamat tujuan.

Peringatan yang ditampilkan secara *real-time* atau tercatat pada *log file*, bagi administrator jaringan *alert* tersebut berguna untuk menambah aturan untuk diterapkan pada *firewall* dalam memfilter paket data.

4.4. *Implementasi Intrusion Detection System*

Pada bagian ini menjelaskan bagaimana menerapkan *software snort* sebagai sistem pendeteksi ancaman terhadap penyusup pada jaringan web server. Beberapa langkah yang dilakukan adalah:

4.4.1. Proses Komputerasi Software

Ada beberapa software pendukung yang perlu dikomputerkan agar snort bisa digunakan.

a. *Instal WinPcap*

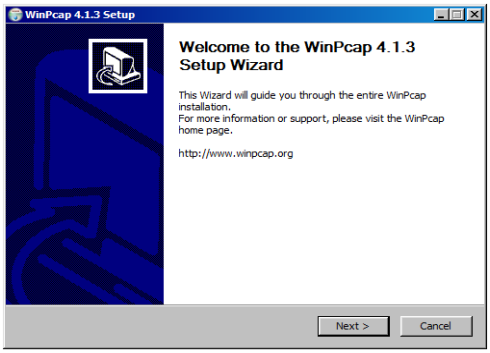
Sistem dasar *snort* membutuhkan beberapa dua komponen yaitu komponen *WinPcap* dan program *snort* IDS. Program *WinPcap* merupakan *tools* standar yang digunakan untuk mengakses *link-layer network* pada *platform* kerja *windows*. *WinPcap* mengizinkan aplikasi untuk mengambil dan mentransmisikan paket–paket pada jaringan yang aktif. Aplikasi *WinPcap* dapat di download dari situs resminya yaitu *http://www.winpcap.org*.

File WinPcap berukuran 894 Kb, file ini bernama WinPcap_4_1_3.exe. Fungsi dari file ini dikomputerkan adalah :

1. Untuk menangkap daftar *adapter* NIC yang beroperasi dan sekaligus mengambil informasi tentang *adapter–adapter* tersebut.
2. Mengawasai paket-paket menggunakan salah satu *adapter* yang dipilih.
3. Menyimpan paket-paket ke dalam *Hard-drive* (atau lebih penting lagi kedalam program *snort*).

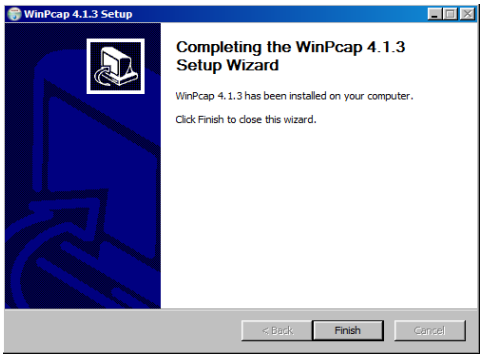
Berikut cara pengkomputeran aplikasi WinPcap:

Setelah filenya berhasil *didownload* dilanjutkan dengan mengeksekusi file hasil *download* yang berekstensi exe. Kemudian akan tampil terlihat pada Gambar 3 dibawah ini :



Gambar 3. Tampilan awal

Kemudian klik tombol **Komputer**, dan proses *WinPcap* selesai dikomputerkan

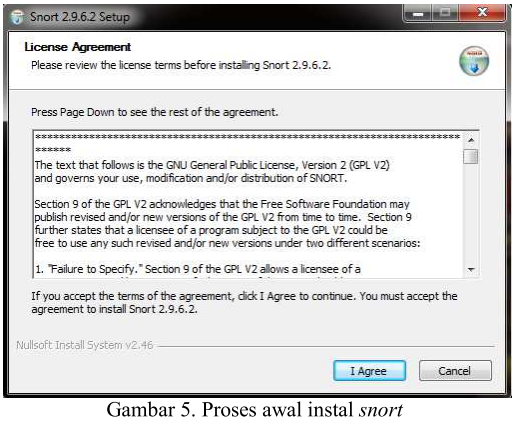


Gambar 4. Informasi WinPcap sukses dikomputerkan

b. *Instal Snort*

Buat folder IDS pada sistem *windows C:\IDS\Snort* tempat mengumpulkan file-file *snort*. File komputer snort dapat di download pada *www.snort.org*.

File komputer *snort* hasil download dengan nama Snort_2_9_6_0_Komputerer.exe dengan ukuran file 2485 Kb. Memulai menginstal klik ganda file instalasi tersebut.



Gambar 5. Proses awal instal *snort*

4.4.2. Proses Konfigurasi Snort

Preprocessor merupakan suatu saringan yang mengidentifikasikan berbagai hal yang harus diperiksa seperti *Detection Engine*. Pada dasarnya *preprocessors* berfungsi mengambil paket yang mempunyai potensi yang berbahaya yang kemudian dikirim ke *detection engine* untuk dikenali polanya.

Konfigurasi agar snort dapat berfungsi sebagai NIDS (*Network Intrusion Detection System*), pengaturannya berada pada direktori *c:\IDS\Snor\etc* dengan nama file snort.conf. File tersebut dapat dilihat menggunakan aplikasi *wordPad* yang sudah ada pada sistem windows. Beberapa bagian yang dikonfigurasi sebagai berikut.

a. *Network setting*

Network dan variable baris ini yang pertama dikonfigurasi, tujuannya untuk memonitor semua *host* yang terhubung pada jaringan dari IP address 192.168.1.0 hingga 192.168.1.255 dan *subnetmasknya* /24 atau 255.255.255.0.

```
############################## Step #1:
Set the network variables.  For more
information, see README.variables
##############################
# Setup the network addresses you are
protecting
var HOME_NET 192.168.1.0/24
```

Agar *snort* mendeteksi serangan dan memberikan peringatan saat serangan terjadi, snort perlu mengetahui dimana *rulebase* yang digunakan, berikut perubahan konfigurasinya :

```
var RULE_PATH C:\IDS\Snort\rules
# var SO_RULE_PATH ../so_rules

var WHITE_LIST_PATH C:\IDS\Snort\rules
var BLACK_LIST_PATH C:\IDS\Snort\rules

config logdir : C:\IDS\Snort\log

var                    PREPROC_RULE_PATH
c:\IDS\Snort\preproc_rules\preprocessor.
rules
```

b. Mengarahkan direktori *dynamicpreprocessor*

```
############################## Step #4:
Configure dynamic loaded libraries.
# For more information, see Snort
Manual, Configuring Snort - Dynamic
Modules
##############################

# path to dynamic preprocessor libraries
dynamicpreprocessor            directory
C:\IDS\Snort\lib
dynamicpreprocessor                 file
C:\IDS\Snort\lib\snort_dynamicpreprocess
or\sf_dce2.dll
dynamicpreprocessor                 file
C:\IDS\Snort\lib\snort_dynamicpreprocess
or\sf_dns.dll
dynamicpreprocessor                 file
C:\IDS\Snort\lib\snort_dynamicpreprocess
or\sf_ftptelnet.dll
```

```
dynamicpreprocessor file C:\IDS\Snort\lib\snort_dynamicpreprocessor\sf_sdf.dll
dynamicpreprocessor file C:\IDS\Snort\lib\snort_dynamicpreprocessor\sf_smtp.dll
dynamicpreprocessor file C:\IDS\Snort\lib\snort_dynamicpreprocessor\sf_ssh.dll
dynamicpreprocessor file C:\IDS\Snort\lib\snort_dynamicpreprocessor\sf_ssl.dll

# path to base preprocessor engine
dynamicengine C:\IDS\Snort\lib\snort_dynamicengine\sf_engine.dll

# path to dynamic rules libraries
#       dynamicdetection       directory C:\IDS\Snort\lib\snort_dynamicengine\sf_engine.dll
```

c. *Update Rule*

Alert rule perlu di *update* agar berbagai bentuk ancaman yang digunakan *attacker* saat ini dapat terdeteksi, ditambahkan beberapa baris *teks alert* seperti dibawah ini :

```
alert icmp any any -> any any (msg:"ICMP testing"; sid:1000001;)
alert udp any any -> any any (msg:"UDP testing "; sid:1000002;)

# alert tcp any any -> any any (msg:"TCP testing "; sid:1000003;)
alert tcp any any -> any any (msg:"SCAN nmap XMAS"; sid:1000004;)
alert tcp $EXTERNAL_NET 10101 -> $HOME_NET any (msg:"SCAN myscan"; flow:stateless; ack:0; flags:S; ttl:>220; reference:arachnids,439; classtype:attempted-recon; sid:613; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 113 (msg:"SCAN ident version request"; flow:to_server,established; content:"VERSION|0A|"; depth:16; reference:arachnids,303; classtype:attempted-recon; sid:616; rev:4;)
alert tcp $EXTERNAL_NET any -> $HOME_NET 80 (msg:"SCAN cybercop os probe"; flow:stateless; dsize:0; flags:SF12; reference:arachnids,146; classtype:attempted-recon; sid:619; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN FIN"; flow:stateless; flags:F,12; reference:arachnids,27; classtype:attempted-recon; sid:621; rev:7;)

# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN ipEye SYN scan"; flow:stateless; flags:S; seq:1958810375; reference:arachnids,236; classtype:attempted-recon; sid:622; rev:8;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN NULL"; flow:stateless; ack:0; flags:0; seq:0; reference:arachnids,4; classtype:attempted-recon; sid:623; rev:6;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN SYN FIN"; flow:stateless; flags:SF,12; reference:arachnids,198; classtype:attempted-recon; sid:624; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN XMAS"; flow:stateless; flags:SRAFPU,12; reference:arachnids,144; classtype:attempted-recon; sid:625; rev:7;)
alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN nmap XMAS"; flow:stateless; flags:FPU,12; reference:arachnids,30; classtype:attempted-recon; sid:1228; rev:7;)

# alert tcp $EXTERNAL_NET any -> $HOME_NET any (msg:"SCAN synscan portscan"; flow:stateless; flags:SF; id:39426; reference:arachnids,441; classtype:attempted-recon; sid:630; rev:7;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET
any (msg:"SCAN cybercop os PA12
attempt"; flow:stateless; flags:PA12;
content:"AAAAAAAAAAAAAAAA"; depth:16;
reference:arachnids,149;
classtype:attempted-recon; sid:626;
rev:8;)

alert tcp $EXTERNAL_NET any -> $HOME_NET
any (msg:"SCAN cybercop os SFU12 probe";
flow:stateless; ack:0; flags:SFU12;
content:"AAAAAAAAAAAAAAAA"; depth:16;
reference:arachnids,150;
classtype:attempted-recon; sid:627;
rev:8;)

alert udp $EXTERNAL_NET any -> $HOME_NET
10080:10081 (msg:"SCAN Amanda client
version request"; content:"Amanda";
nocase; classtype:attempted-recon;
sid:634; rev:2;)
alert udp $EXTERNAL_NET any -> $HOME_NET
49 (msg:"SCAN XTACACS logout";
content:"|80 07 00 00 07 00 00 04 00 00
00 00 00|"; reference:arachnids,408;
classtype:bad-unknown; sid:635; rev:3;)
alert udp $EXTERNAL_NET any -> $HOME_NET
7 (msg:"SCAN cybercop udp bomb";
content:"cybercop";

reference:arachnids,363; classtype:bad-
unknown; sid:636; rev:1;)
alert udp $EXTERNAL_NET any -> $HOME_NET
any (msg:"SCAN Webtrends Scanner UDP
Probe"; content:"|0A|help|0A|quite|0A|";
reference:arachnids,308;
classtype:attempted-recon; sid:637;
rev:3;)

alert tcp $EXTERNAL_NET any -> $HOME_NET
22 (msg:"SCAN SSH Version map attempt";
flow:to_server,established;
content:"Version_Mapper"; nocase;
classtype:network-scan; sid:1638;
rev:5;)
alert udp $EXTERNAL_NET any -> $HOME_NET
1900 (msg:"SCAN UPnP service discover
attempt"; content:"M-SEARCH "; depth:9;
content:"ssdp|3A|discover";
classtype:network-scan; sid:1917;
rev:6;)
```

d. *Setting Output Alert*

Setting output berupa *alert* agar pesan adanya ancaman tersimpan pada *file log*. File tersebut diberi nama dengan alert.ids konfigurasinya pada baris dibawah ini :

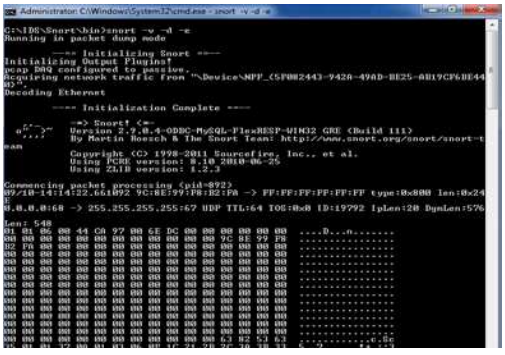
```
# pcap
output alert_fast: alert.ids
output log_tcpdump: tcpdump.log
# metadata reference data. do not
modify these lines
include
c:\IDS\Snort\etc\classification.config
include
c:\IDS\Snort\etc\reference.config
```

4.4.3. Mengetes Hasil Instalasi

Setelah selesai mengkonfigurasi file *snort.conf*, selanjutnya dilakukan pengetesan dengan menjalankan *snort* berdasarkan beberapa perintah seperti dibawah ini :

a. Pengetesan *Sniffer*

Pengetesan ini bertujuan untuk menampilkan paket-paket TCP/IP pada jaringan. Dengan perintah *sniffing* ini menampilkan *header* dan isi paket data secara *real-time* pada jaringan, perintah untuk menjalankan *sniffing* ini dengan mengetikan pada prompt *C:\IDS\Snort\bin\snort –v –d -e*.



Gambar 6. Hasil *Capture* Dengan Perintah Paket *Sniffer*

Kombinasi dari perintah –v, -d dan –e akan menghasilkan beberapa keluaran yang berbeda, yaitu :

-v, untuk melihat header TCP/IP paket yang lewat

-d, untuk melihat isi paket

-e, untuk melihat *header link layer* paket seperti *Ethernet header*.

b. Pengetesan *Packet Logger*

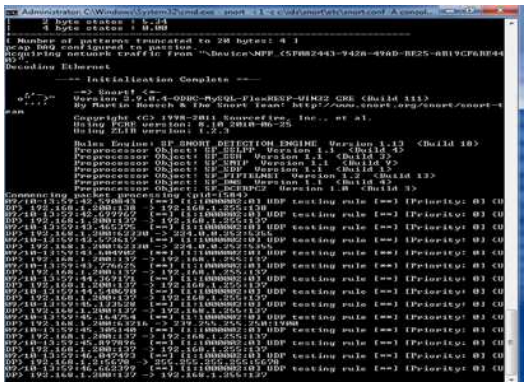
Untuk mengetes kemampuan *logging* dari snort yaitu mencatat semua log yang ditemukan kedalam direktori bernama log diberikan perintah *c:\IDS\Snort\bin>snort –dev –l c:\IDS\Snort\log*.

Gambar 7. Hasil *Capture* Dengan Perintah Paket *Sniffer*

Dengan perintah tersebut *snort* mengumpulkan paket yang berjalan dan menyimpannya kedalam direktori *snort.*

c. Pengetesan Fungsi IDS

Untuk mengetest apakah IDS telah berfungsi sesuai dengan konfigurasi, diberikan perintah pada *command prompt C:\IDS\Snort\bin>snort –i 1 –c c:\ids\snort\etc\snort.conf –A console –l c:\ids\snort\log –K ascii*, hasilnya seperti Gambar 8 dibawah ini :



Gambar 8. Hasil Pengetesan IDS Secara *Real-time*

4.5. Pengujian IDS

Pengujian ini dilakukan terhadap perangkat IDS yang sudah dibangun untuk membuktikan bahwa *snort* sebagai IDS dapat melakukan kegiatannya sesuai dengan konfigurasi yang telah dilakukan sebagaimana telah dibahas pada bagian sebelumnya.

Metode pengujian yang dilakukan tes fungsi IDS dan metode waktu respon IDS, metode tes fungsi IDS yaitu melihat sistem ini berjalan sesuai dengan fungsinya sebagai alat yang memberikan informasi secara *real time* kepada administrator adanya ancaman pada jaringan di *web server*. Selain dengan melihat alert secara *real time* juga dapat dilihat pada *log file* yang tersimpan di direktori log aplikasi snort.

Metode waktu respon IDS adalah waktu respon yaitu waktu yang dibutuhkan snort dalam merespon ancaman ketika ada penyusup pada web server.

4.6. Analisa Pengujian

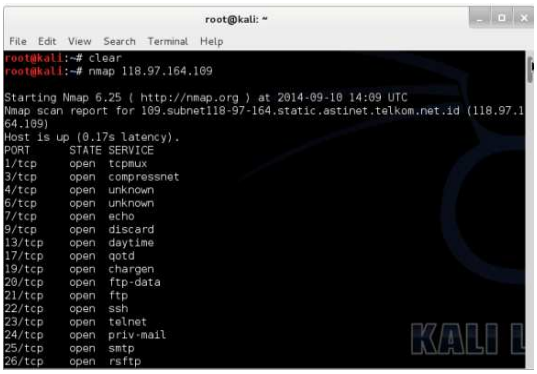
Analisa terhadap pengujian IDS untuk mengetahui kehandalan dari *snort* dilakukan dengan dua metode.

a. Metode Test Fungsi IDS

Untuk menguji *snort* dalam merespon ancaman dilakukan pengujian dengan menggunakan tool nmap. *Nmap* dikenal dengan istilah port scanner yang merupakan *tool* yang digunakan attacker dalam mencari informasi *port* untuk mencari celah kelemahan pada web server.

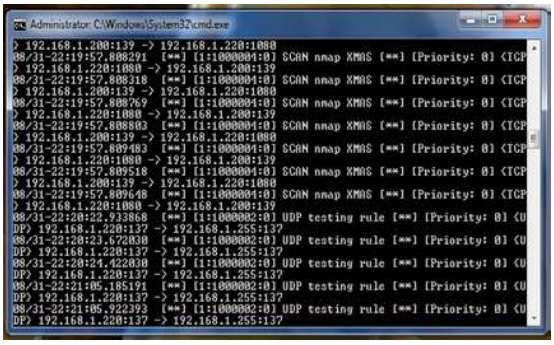
Dalam pengujian ini dilakukan *scanning port* terhadap salah satu *web server* universitas dehasen, pengujian dilakukan pada komputer yang menggunakan sistem operasi Kali. *Tool nmap* dilakukan melalui *root* dengan perintah *nmap*

ip_address unived, hasil dari perintah ini seperti Gambar 9 dibawah ini.



Gambar 5.9. Pengujan Dengan *Tool* Nmap Terhadap *Web Server*

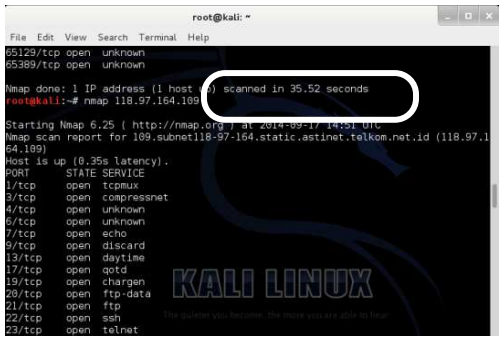
Berdasarkan hasil *scanning* yang dilakukan tersebut sistem *snort* langsung merespon secara real time dan memberikan alert yaitu adanya tindakan scanning dilakukan dengan nmap. *Alert* yang ditampilkan adalah *Ip address* pengirim serta *port number* pengirim dan *ip addres* tujuan serta *port number*-nya. Dan juga tanggal dan waktu dilakukan scanning terhadap ip tersebut. Tampilannya seperti Gambar 10 dibawah ini.



Gambar 5.10.Alert pada snort berupa *scaning* dengan *nmap*

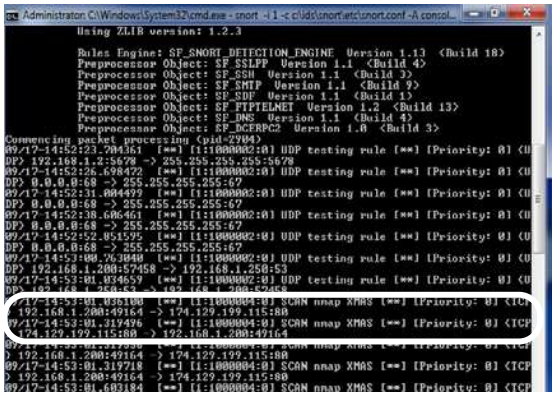
b. Metode Waktu Respon IDS

Analisa pengujian metode ini adalah mengamati waktu yang dibutuhkan IDS dalam merespon adanya tindakan penyusup terhadap keamanan *web server*. Pengujiannya masih menggunakan *tool nmap*, disini fokus unuk melihat perbandingan waktu dimulai scanning dari komputer penyusup.



Gambar 11. Pengujian waktu *scaning port*

Pada Gambar 11 penyusup mulai melakukan scaning pada waktu 14:51, Sedangkan hasil respon snort terhadap adanya *scanning* pada waktu 14:53 seperti pada Gambar 12.



Gambar 12. Tampilan *alert* respon *snort* terhadap *scaning nmap*

Hasil dari perbandingan waktu respon dengan waktu *scaning* IDS adalah 14:53 – 14:51 dengan sisa waktu 2 menit. Ini menyatakan bahwa respon dari IDS ini kurang cepat dalam menanggapi ancaman. Diperlukan alternative lain agar fungsi snort dalam merespon adanya tindakan penyusupan lebih cepat

V. KESIMPULAN

Dari hasil penelitian ini dapat diambil beberapa kesimpulan, yaitu :

a. Aplikasi Snort dapat berfungsi sebagai sistem pendeteksi *intrusi – intrusi* pada jaringan hal ini terbukti dari kemampuan IDS dalam mendeteksi penyusup yang melakukan *scanning port*.

b. Opsi – opsi *rule snort* yang *update* agar lebih banyak mendeteksi ancaman–ancaman yang berlaku saat ini.
c. Berdasarkan perbandingan dari analisa pengujian terhadap kehandalan IDS dalam merespon ancaman dalam waktu 2 menit menampilkan *alert* pada *screen*.

REFERENSI

[1] Sofana Iwan, 2011, *Teori & Modul Praktikum Jaringan Komputer*, Modula, Bandung, 362 Halaman.
[2] Kurniawan Rulianto, 2009, *Word Press untuk Orang Awam*, Maxikom, Palembang, 168 Halaman.
[3] Raharjo Budi, 2005, *Keamanan Sistem Informasi Berbasis Internet*, Insan Infonesia PT. Indosic, Jakarta, 158 Halaman.
[4] Thomas Tom, 2004, *Network Security First-Step*, Andi, Yogyakarta, 512 Halaman.
[5] Ariyus Dony, 2007, *Intrusion Detection System Sistem Pendeteksi Penyusup Pada Jaringan Komputer*, Andi, Yogyakarta, 290 Halaman
[6] Rafiudin Rahmat, 2010, *Mengganyang Hacker dengan SNORT*, Andi, Yogyakarta, 224 Halaman.