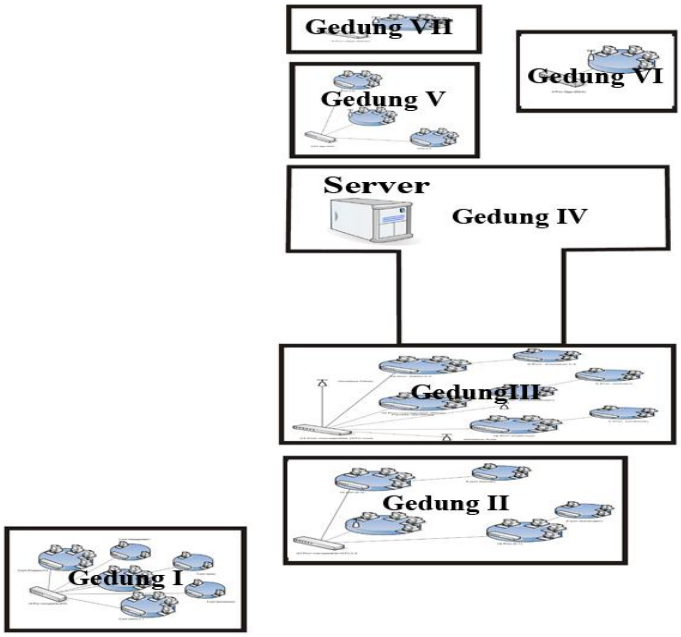
BAB III

PEMBAHASAN

3.1 Analisis Fisik Bangunan

Dari hasil analilis yang telah dilakukan di Balai Besar Bahan dan Barang Teknik (B4T) memiliki denah bangungan sebagai berikut



Gambar 8 : Denah Bangunan B4T

3.2 Analisis Perangkat Keras

Dari hasil analisis yang telah dilakukan khususnya pada perangkat keras yang digunakan di Balai Besar Bahan dan Barang Teknik (B4T) sudah memenuhi standar menjalankan IDS Snort. Adapun Spesifikasi nya sebagai berikut :

- PC sever IDS menggunakan OS Ubuntu server 10.4
    - Prosesor Intel Pentium 4 3 Ghz
    - Memory 1 GB
    - Hardisk 80 GB
    - VGA 64 (onboard)
    - DVD-ROM 52 x
    - Ethernet Card
    - Monitor color 14"
- Spesifikasi Perangkat keras jaringan.
    - Router mikrotik
    - Switch
    - Kabel UTP cat 5
    - Konektor RJ-45

3.3 Analisis Perangkat Lunak

Perangkat lunak yang ada di B4T saat ini sudah memenuhi standar menjalankan IDS Snort. Seperti menggunakan Ubuntu Server 10.4 LTE untuk server IDS atau windows XP untuk client. Untuk mengolah datatabase digunakan mysql server,webmin dan base.

3.4 Analisis Pemakai

Adapun profile pengguna IDS Snort ini adalah :

1. Administrator

   Sistem deteksi penyusup ini digunakan oleh administrator computer sebagai sarana utama mendeteksi adanaya penyusup.

2. Operator

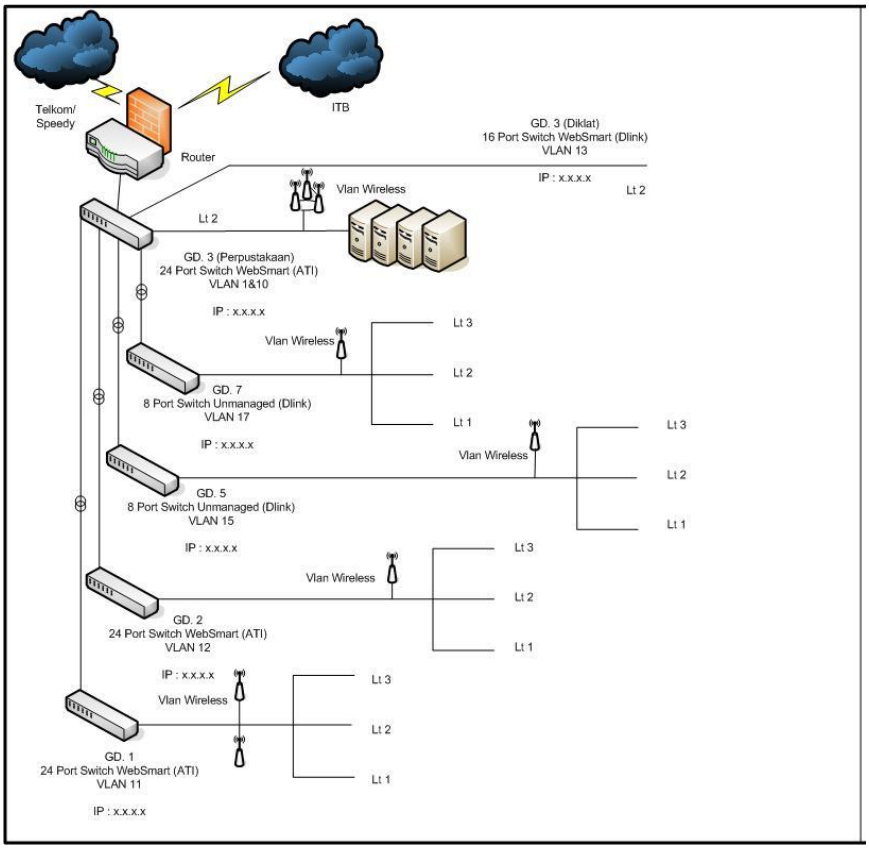
   Operator adalah staff yang diberikan wewenang / hak oleh administrator untuk menggantikan tugas dan tanggung jawab administrator jika diperlukan.

3.5 Analisis Jaringan

Balai Besar Bahan dan Barang Teknik (B4T) memiliki sebuah jaringan dimana jaringan tersebut berfungsi untuk komunikasi maupun pengiriman data. Namun adanya permasalahan dalam keamanan jaringan komputer yang datang dari luar maupun dari dalam baik itu berupa virus atau pada jaringan komputer di B4T

3.5.1 Skema Jaringan Awal

Berikut gambar skema jaringan awal pada instansi Balai Besar Bahan dan Barang Teknik (B4T) :



Gambar 9 : Skema Jaringan Awal

Keterangan gambar :

a) Modem Speedy

Internet Service Provider yang digunakan pada instansi B4T adalah Speedy, sebanyak 2 buah modem.

b) MikroTik RouterOS™

B4T menggunakan MikroTik RouterOS™ sistem operasi dan perangkat lunak yang dapat digunakan untuk menjadikan komputer manjadi router network.

c) *Proxy*

B4T menggunakan *Proxy Server* berbasis Windows Server.

d) *SWITCH*

Hub ini berfungsi untuk menerima sinyal dari satu komputer dan mentransmisikannya ke komputer yang lain.

e) Client

Client ini merupakan komputer yang terhubung ke HUB yang memiliki akses lokal maupun internet.

Analisa permasalahan yang terjadi adalah B4T belum memiliki keamanan jaringan sehingga rentan dating serangan dari luar maupun dari dalam baik itu berupa virus atau pada jaringan komputer. Hal ini menyebabkan akses keamanan jaringan di B4T rentan terkena serangan virus dan serangan-serangan dari luar .

3.5.2 Pembuatan IDS (*Instrusion Detection System*)

IDS merupakan solusi yang tepat untuk mengatasi permasalahan keamanan jaringan yang ada di B4T ini. Karena IDS memiliki beberapa kemampuan dalam mengatasi permasalahan mengenai keamanan jaringan. Dengan memanfaatkan sebuah PC, hal tersebut akan dimaanfaatkan untuk pembuatan Server yang terinstal IDS.

3.6 Pengembangan Keamanan

Dengan adanya tujuan untuk mengatasi permasalahan pada keamanan jaringan di B4T, pengembangan keamanan pun dibuat sebagai solusi keamanan jaringan untuk menjalankan sesuai kebutuhan yang ada pada jaringan tersebut IDS sendiri menggunakan aplikasi SNORT.

3.6.1 Kebutuhan Perangkat Lunak IDS (*Instrusion Detection System*)

IDS adalah sebuah aplikasi perangkat atau perangkat lunak yang memonitor aktivitas jaringan dan / atau sistem untuk kegiatan berbahaya atau pelanggaran kebijakan dan menghasilkan laporan-laporan ke Management Station.

IDS dibagi menjadi 2 jenis :

-*Network intrusion detection system* (NIDS)

Semua lalu lintas yang mengalir ke sebuah jaringan akan dianalisis untuk mencari apakah ada percobaan serangan atau penyusupan ke dalam sistem jaringan. NIDS umumnya terletak di dalam segmen jaringan penting di mana server berada atau terdapat pada "pintu masuk" jaringan.

-*Host-based Intrusion Detection System* (HIDS)

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

Penerapan IDS sendiri menggunakan aplikasi SNORT yang berfungsi sebagai software atau aplikasi atau juga sebuah tool *security*.

IDS sudah banyak mendukung berbagai macam driver hardware. Beberapa hal yang perlu diperhatikan dalam instalasi antara lain :

Table 1. Spesifikasi perangkat keras

No	Hardware	Unit	Spesifikasi
1	Processor	1 unit	Pentium3
2	RAM	1 unit	512
3	Hard Disk	1 unit	40 GB
4	LAN Card	1 unit	

Table 2. Spesifikasi perangkat lunak

No	Software	Spesifikasi	Keterangan
1	Sistem Operasi	Ubuntu Server 10.04 LTS	
2	Paket Snort	Snort 2.8.5	
3	Paket rules	Snortrules-pr-2.4	
4	Paket BASE	BASE 1.4.5	

3.7 Tinjauan Fungsional IDS

3.7.1 SNORT

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

terhadap ruleset yang didefinisikan oleh pengguna. Maka program akan melakukan tindakan tertentu berdasarkan apa yang telah diidentifikasi.

Kelebihan Snort dibanding IDS lainnya adalah:

- Snort adalah Network IDS dengan 3 modus: *sniffer*, *packet logger*, dan *network intrusion detection*.
- Snort dapat dijalankan di *background* sebagai sebuah daemon.
- Cepat dalam meganalisa paket. Apakah paket merupakan paket 'aneh' atau normal.
- *Multiplatform*, dapat digunakan pada jaringan yang memilki *platform* berbeda dan sistem operasi mana saja.
- *Open source.*
- Merupakan IDS yang paling banyak dipakai di dunia.

Kelemahan:

- Snort hanya menganggap suatu paket atau aksi adalah sebuah intrusi saat paket atau aksi tersebut tidak sesuai dengan *rules*–nya. Jadi jika ada intrusi yang tidak dijabarkan pada *rules*, maka intrusi tadi akan diloloskan oleh Snort.
- Snort kadang meloloskan intrusi jika suatu intrusi terlalu besar atau melebihi beban *scanning* biasa.

Tujuan *Snort* adalah:

- Untuk mendeteksi instruksi – instruksi jaringan, mendeteksi penyusupan yang memasuki jaringan, penyerangan terhadap jaringan, pemindaian dan berbagai bentuk ancaman lain di jaringan.
- Jika terdapat pola serangan baru dan termasuk dalam kategori serangan berbahaya maka tambahkan rule baru pada IDS Snort,

sehingga sistem yang di manage akan benar-benar menjadi sistem yang secure (aman).

- Cara kerja Snort

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

3.7.2 Snort *Rules*

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

alert tcp $EXTERNAL NET alert tcp $EXTERNAL NET any -> $HTTP SERVERS $HTTP PORTS (msg:"WEB-IIS unicode directory traversal attempt"; flow:to server, established; content:¨/..%c0%af../"; nocase; classtype:web-application-attack; reference:cve, CVE-2000-0884; sid:981; rev:6;)

Rule di atas terdiri dari 2 bagian: header dan option. Bagian ”alert tcp $EXTERNAL NET any -¿ $HTTP SERVERS $HTTP PORTS” adalah header dan selebihnya merupakan option. Dari rule-rule seperti di ataslah IDS Snort menghukumi apakah sebuah paket data dianggap sebagai penyusupan/serangan atau bukan, paket data dibandingkan dengan rule

IDS, jika terdapat dalam rule, maka paket data tersebut dianggap sebagai penyusupan/serangan dan demikian juga sebaliknya jika tidak ada dalam rule maka dianggap bukan penyusupan/serangan. Setelah menginstal Snort *engine* dan rulesnya, maka langkah selanjutnya adalah mengkonfigurasi Snort *engine*. Snort *engine* dimodifikasi sesuai kebutuhan dan spesifikasi jaringan yang akan dipindai oleh Snort.

- Cara kerja Snort *rules*

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

3.7.3 BASE (*Basic Analysis and Security Engine*)

Kelebihan:

- Program berbasis web yang memungkinkan implementasi antar *platform*.
- Merupakan perintis antarmuka GUI untuk Snort dan paling banyak digunakan oleh pengguna IDS. BASE merupakan rekomendasi dari Snort.org sendiri.
- Multi language, antarmuka memilki beberapa bahasa selain bahasa Inggris.
- Layanan peringatan yang real time.
- *Open source*
- Dapat diimplementasikan pada IDS manapun selain Snort.

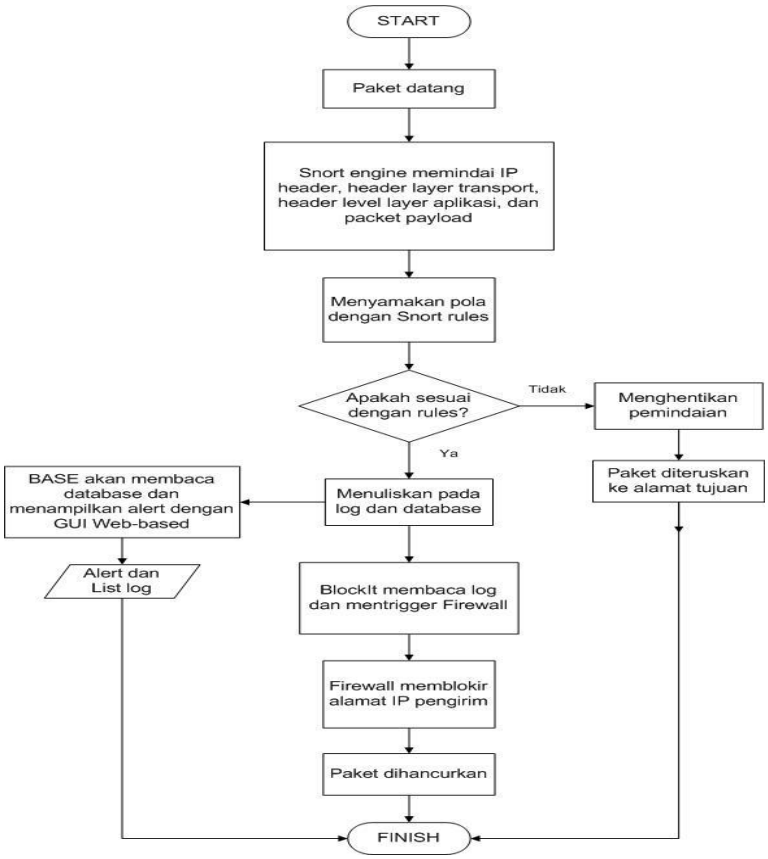
Kekurangan:

- Tampilan hanya berupa list log dan alert. Jika ingin ditampikan sebagai grafik untuk melihat pergerkan traffic jaringan dibutuhkan *add-ons* tambahan.
- Untuk versi dibawah 1.4.5 masih terdapat bug dimana BASE dapat disisipkan SQLinjection.

- Cara kerja BASE

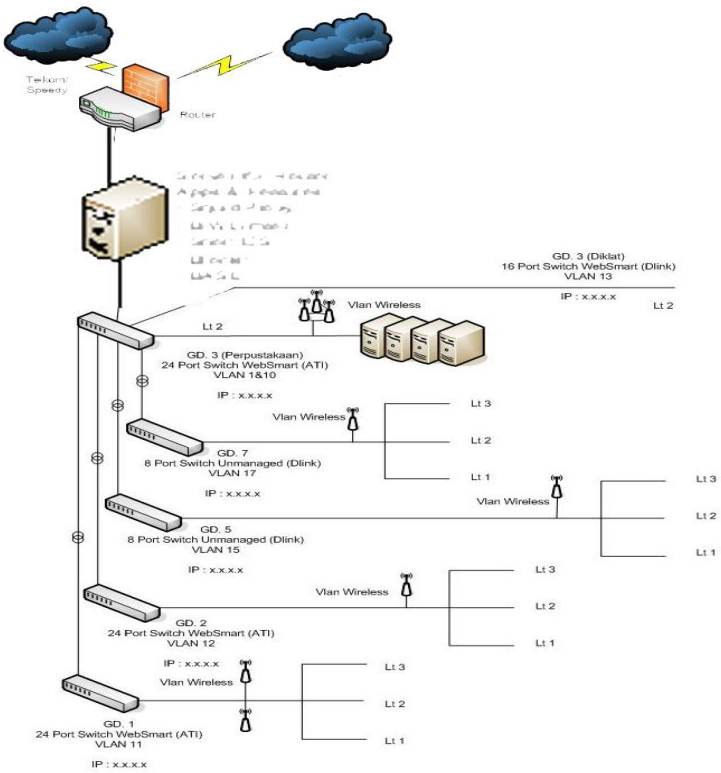
  BASE membaca *log* yang ditulis Snort pada *database*. Secara otomatis BASE akan menampilaknnya sebagai *alert* dan *list log* dengan antarmuka yang mudah dipahami dan dilihat oleh administrator.

3.8 Alur Kerja IDS



Gambar 10 : Diagram Alur IDS Bekerja

3.9 Topologi Jaringan Dengan IDS



Gambar 11 : Topologi Jaringan Dengan IDS

3.10 Implementasi dan Pengujian

3.10.1 Implementasi

Berikut adalah tahapan pembangunan sistem IDS pada B4T :

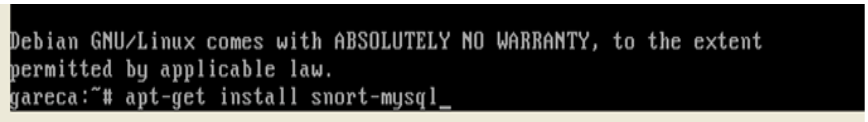
1. Instalasi paket Snort.
2. Konfigurasi Snort.
3. Instalasi paket-paket *dependency* yang dibutuhkan sebelum komponen utama sistem terinstal.
4. Instalasi paket Snort rules.
5. Instalasi paket BASE.

1. Instalasi paket Snort

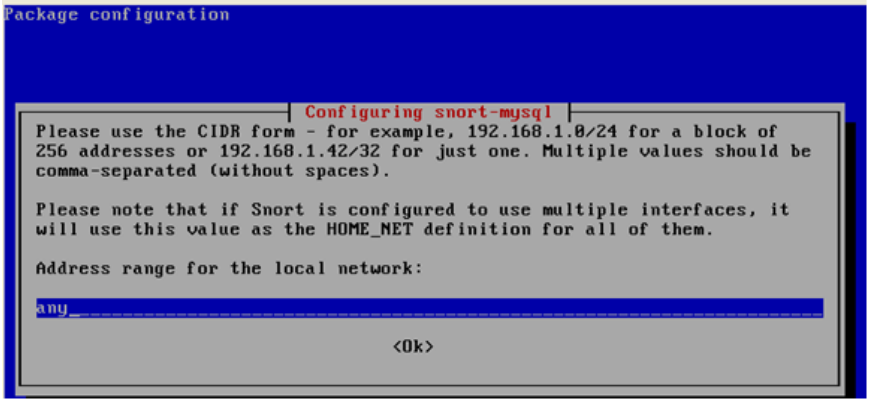
Secara umum teknik instalasi snort di Linux sangat mudah. Bagi pemula mungkin ada baiknya menggunakan file RPM yang jauh lebih mudah instalasinya. Pada kesempatan ini saya menggunakan file tar.gz yang sedikit lebih sulit, walaupun sebetulnya tidak terlalu sulit juga.

Beberapa langkah persiapan yang perlu dilakukan, yaitu :

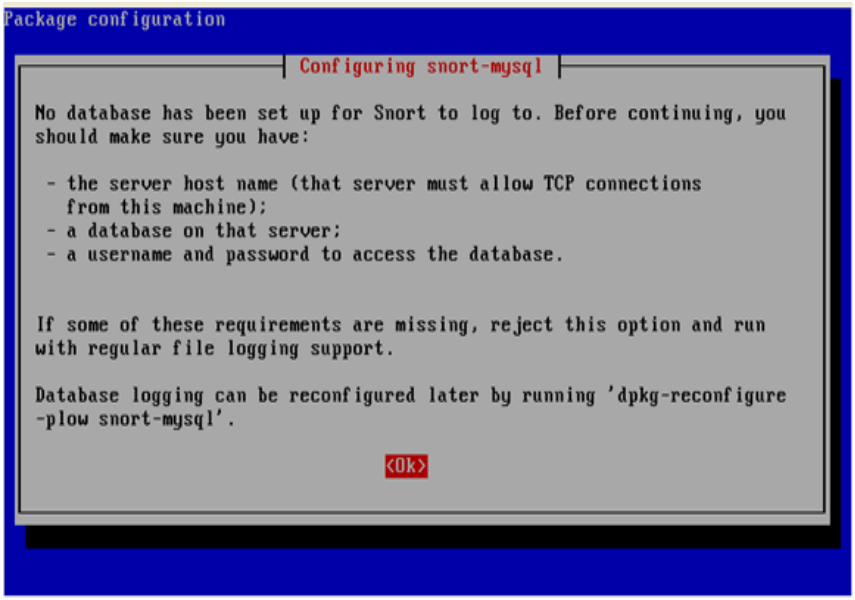
1. Apt-get install snort-mysql



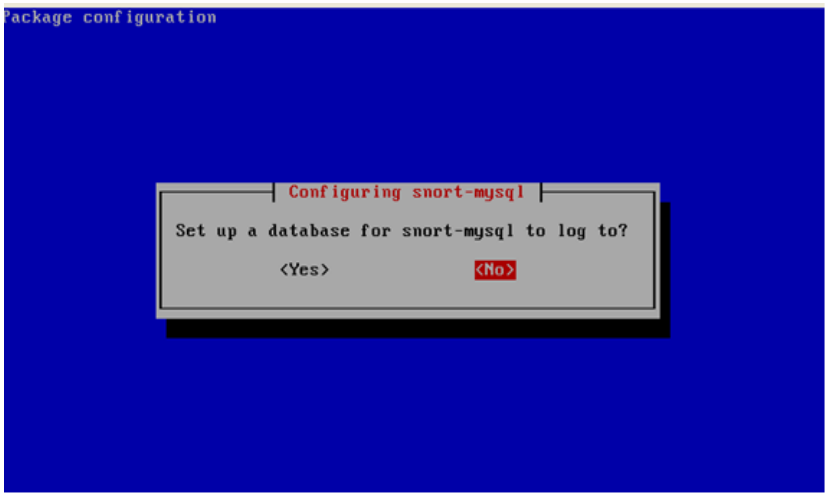
2. Muncul pop-up, pada textfield masukkan any, lalu tekan OK



3. Tekan OK lagi



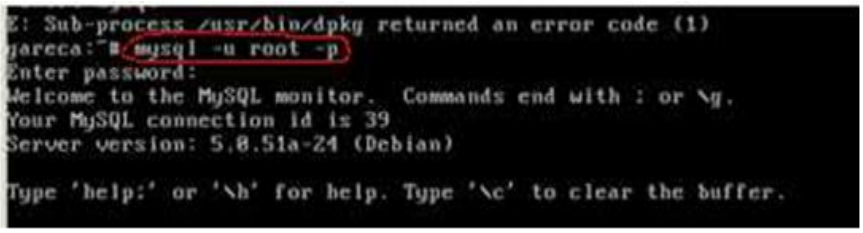
4. Set up a database, pilih No



Untuk sementara akan terjadi error, tapi tidak masalah.

Sekarang kita akan masuk ke MySql

1. Ketikkan : mysql –u root –p lalu masukkan password user root sebagai contoh: ukm12345*



2. Ketikkan : create database snort;



3. Ketikkan : grant all on snort.* to snortuser@localhost identified by 'snortpassword';



4. Ketikkan : flush privileges; <- perintah untuk reload privileges



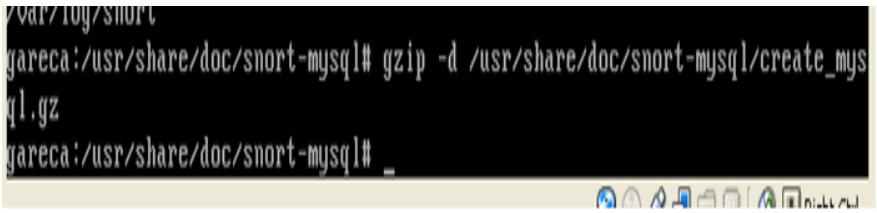
5. Lalu exit

Sekarang kita akan mengimpor table ke dalam mysql,

1. Ketikkan : cd /usr/share/doc/snort-mysql

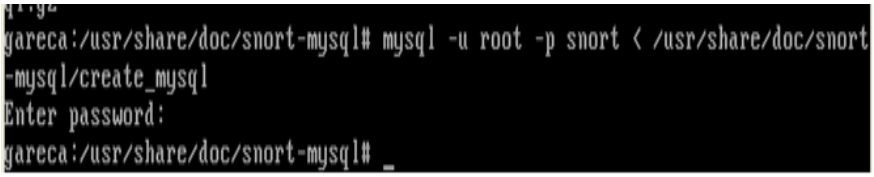


2. Ketikkan : gzip –d /usr/share/doc/snort-mysql/create_mysql.gz



3. Ketikkan : mysql -u root -p snort < /usr/share/doc/snort mysql/create_mysql

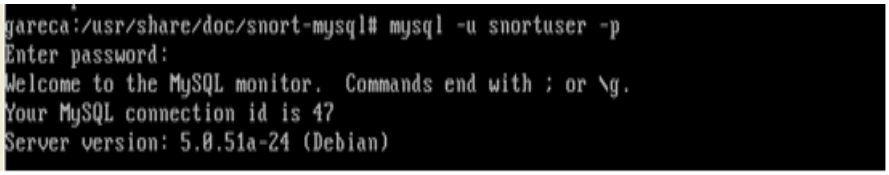
   Lalu masukkan password seperti password yang diatas. Contoh ukm12345*



Untuk melihat apakah table sudah diimpor, kita masuk ke mysql

1. Ketikkan : mysql –u snortuser –p (untuk password masukkan snortpassword) <- masuk mysql

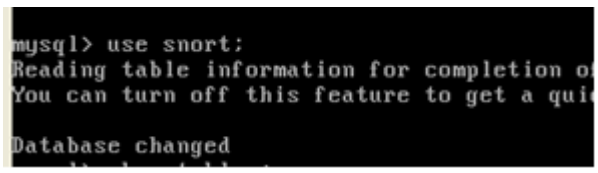
   menggunakan user yang bernama snortuser dengan password snortpassword



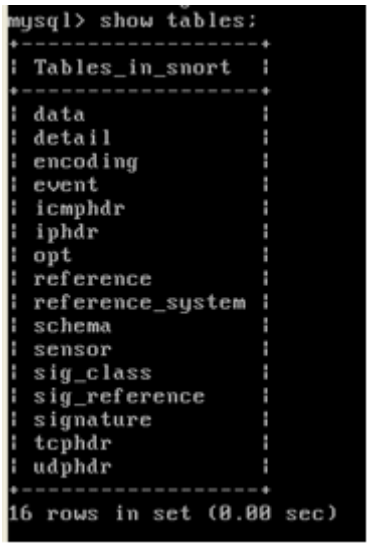
2. Ketikkan : show databases;



3. Ketikkan : use snort;



4. Ketikkan : show tables;



5. Ketikkan exit; untuk keluar

2. Konfigurasi Snort

1. Ketikkan : pico /etc/snort/snort.conf



2. Untuk lebih mudah gunakan search yaitu CTRL+W lalu ketikkan dbstart
3. Diantara (#DBSTART#) dan (#DBBEND#), ketikkan : output database: log, mysql, user=snortuser password=snortpassword dbname=snort host=localhost

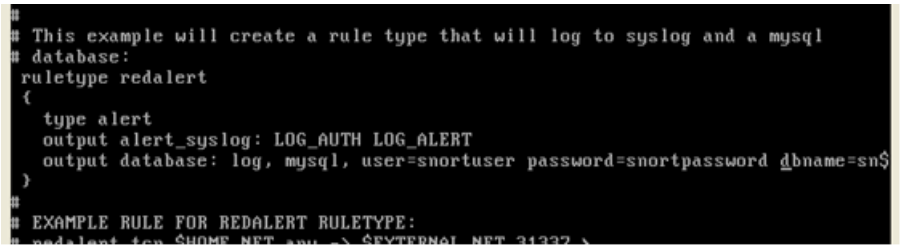


4. Masih di file yang sama, search kata-> redalert

5. Hilangkan tanda comment (#) dari bagian ruletype redalert{}
6. Ganti bagian output database dari ruletype redalert{}

   dengan : output database: log, mysql, user=snortuser password=snortpassword dbname=snort host=localhost

   langkah 4,5,6 seperti gambar dibawah ini.



7. Save dan exit
8. Lalu kita jalankan Snort dengan perintah : snort –u snort –c /etc/snort/snort.conf



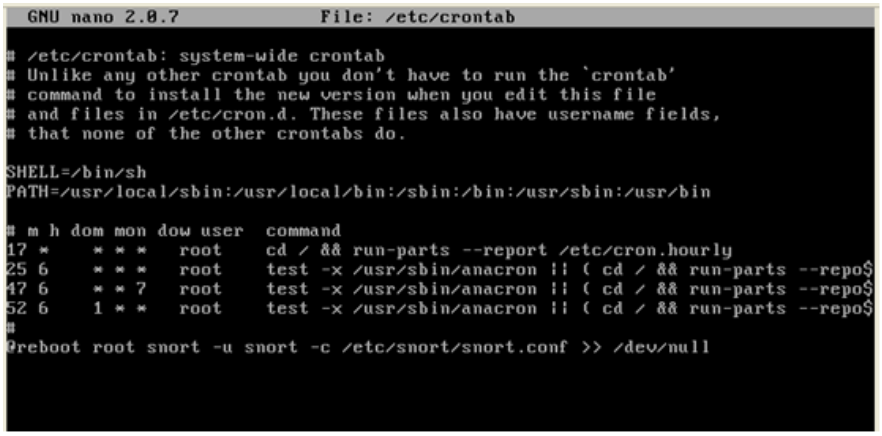
9. CTRL+c untuk mengakhiri
10. Buka file crontab dan edit

    Ketikkan : pico /etc/crontab

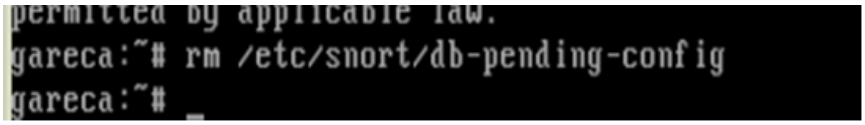


11. Tambahkan perintah dibawah ini di baris paling akhir.

    ketikkan : @reboot root snort -u snort -c /etc/snort/snort.conf >> /dev/null



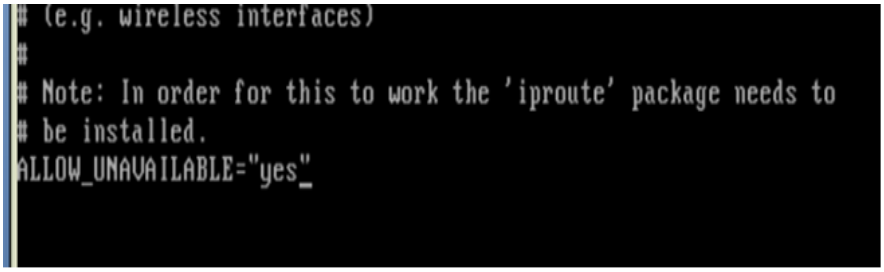
12. reboot lagi, maka pesan error tetap akan tampil
13. ketikkan : rm /etc/snort/db-pending-config



14. ketikkan : pico /etc/default/snort



pada bagian paling bawah, ubah nilai bagian ALLOW_UNAVAILABLE dari no menjadi yes

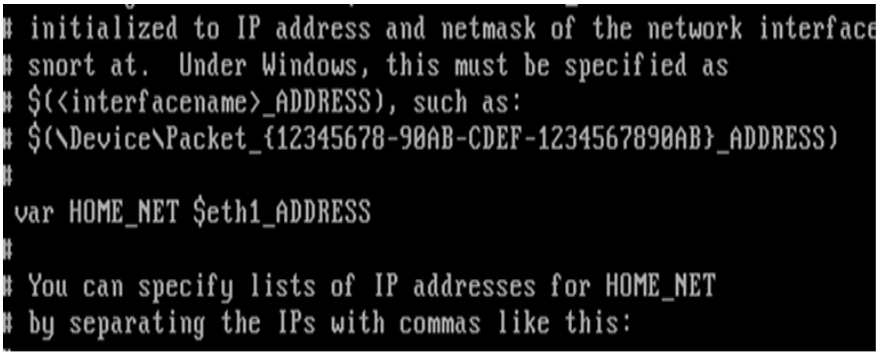


15. save dan exit
16. ketikkan : pico /etc/snort/snort.conf

cari bagian eth0 di (var HOME_NET $eth0_ADDRESS)

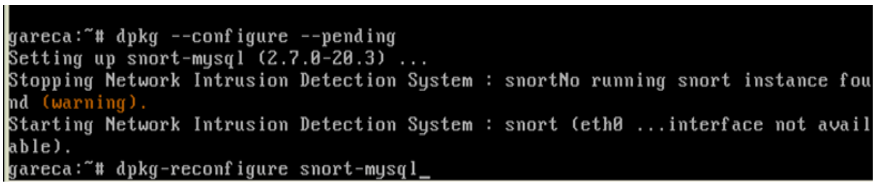
hilangkan tanda comment (#), ganti eth0 jadi eth1

hasil akhir seperti gambar dibawah ini

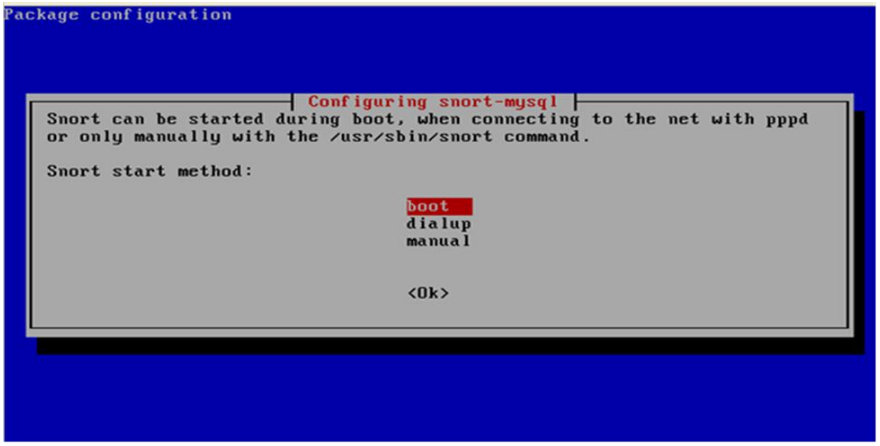


17. ketikkan : dpkg --configure –pending

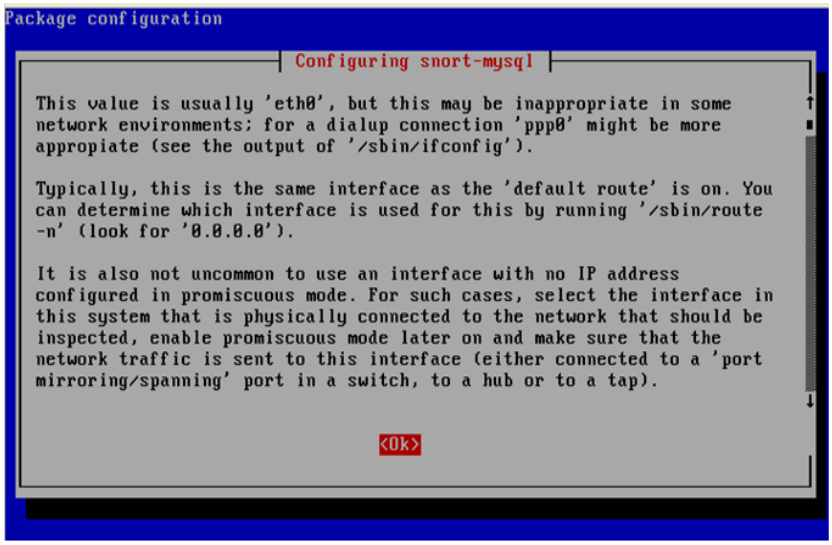
18. ketikkan : dpkg-reconfigure snort-mysql



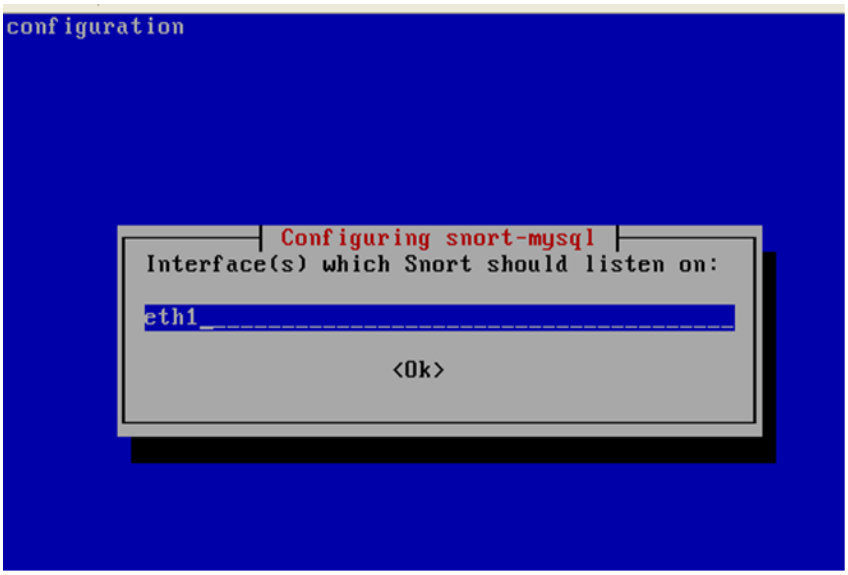
19. Pilih boot, lalu OK



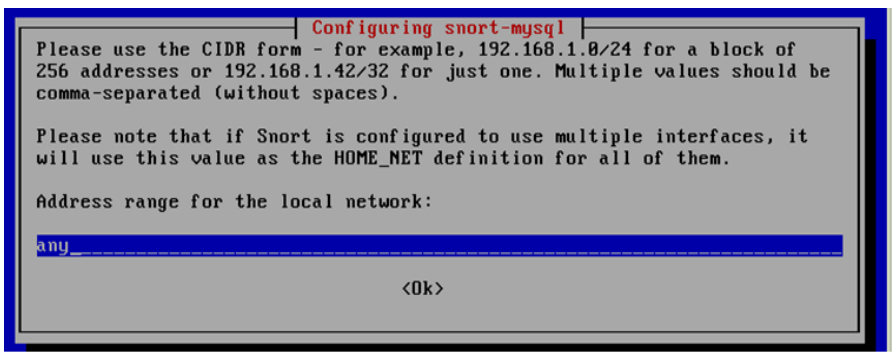
20. Pilih OK



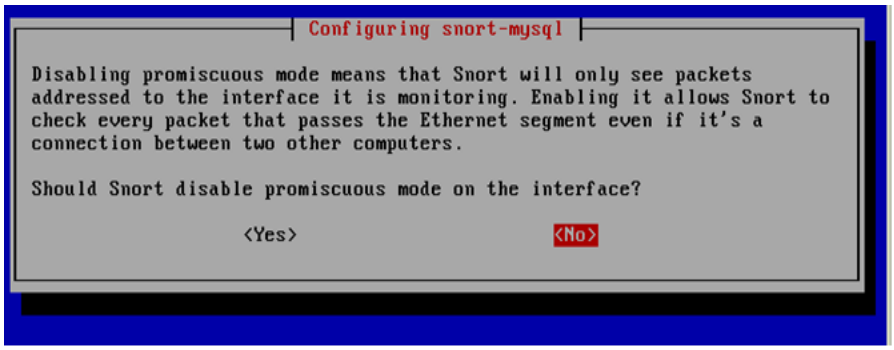
21. Ganti eth0 jadi eth1



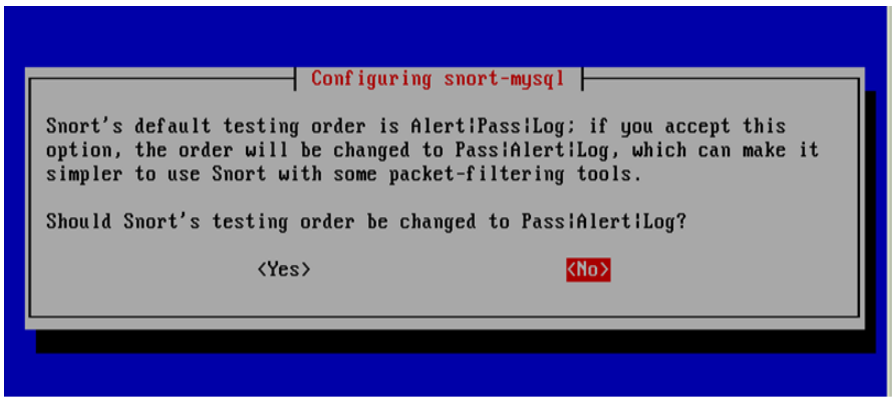
22. Biarkan any



23. Disable pilih NO



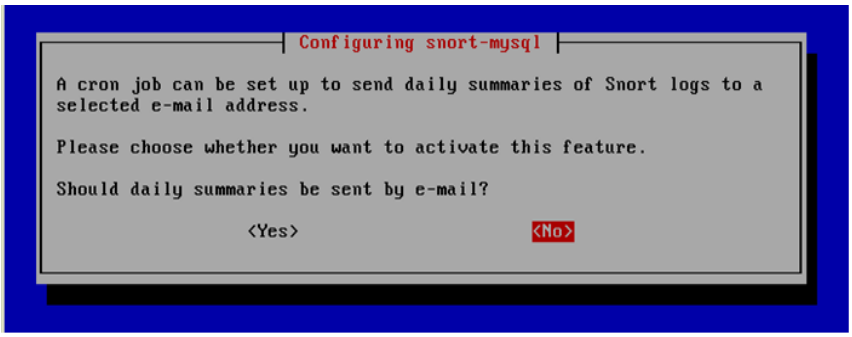
24. Testing pilih NO



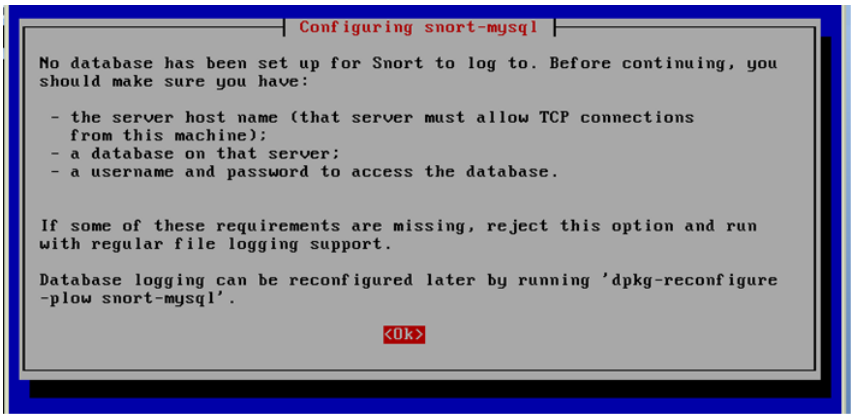
25. Biarkan kosong, OK



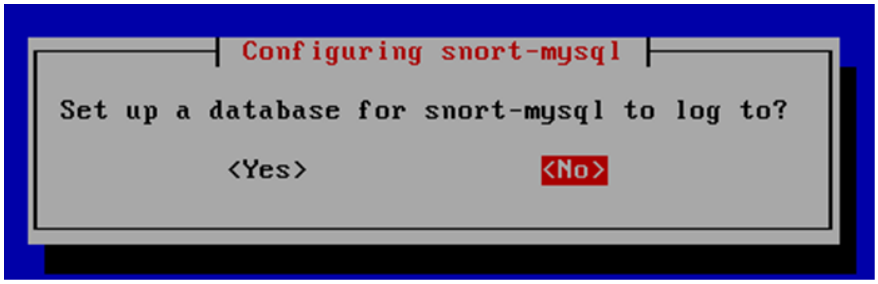
26. Email, pilih NO



27. OK



28. Set up database, pilih NO

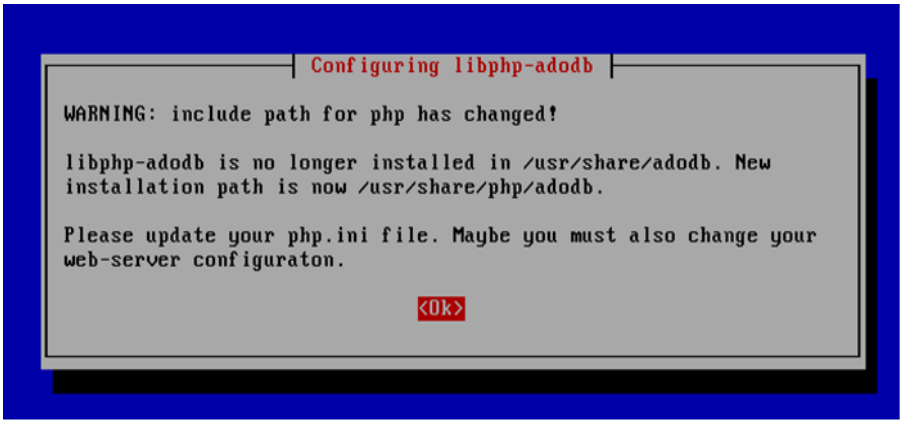


- Menginstal ACIDBASE

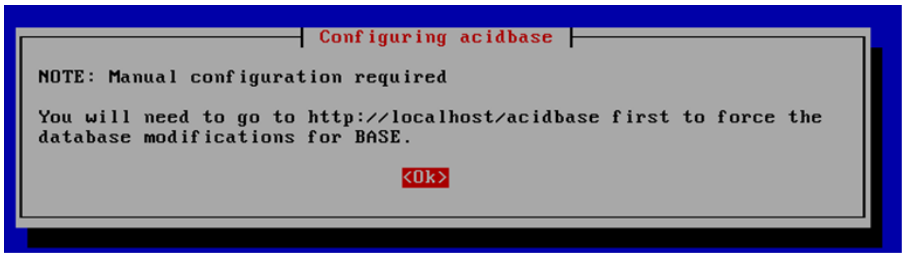
1. Ketikkan : apt-get install acidbase



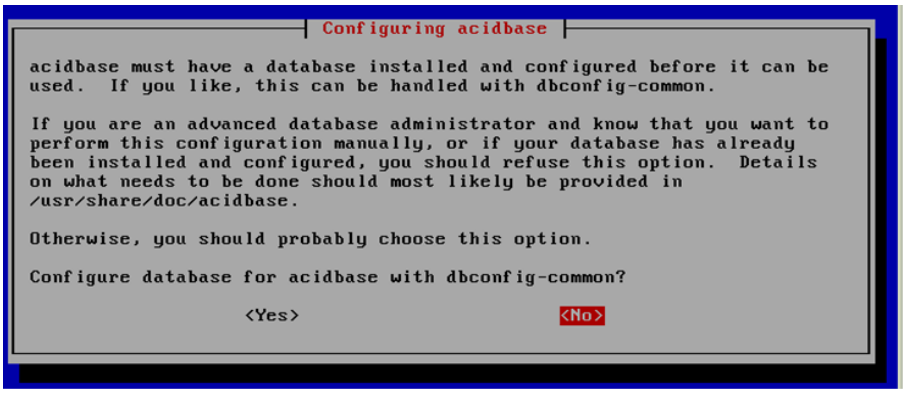
2. Muncul pop up, OK



3. Pilih OK



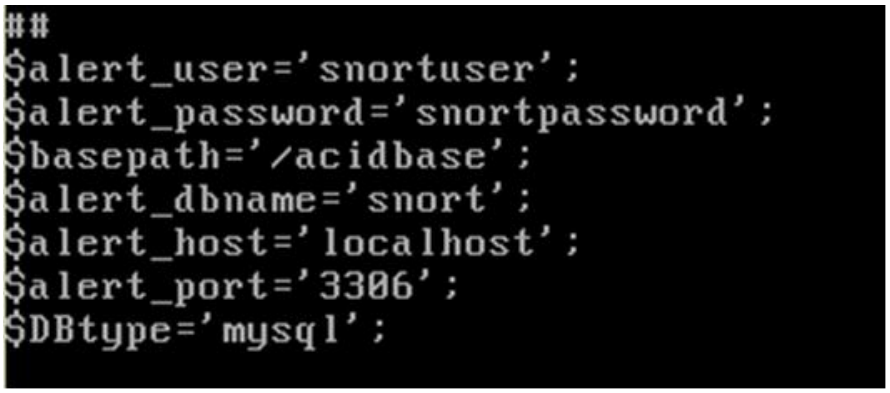
4. Create database, Pilih NO



5. Ketikkan : pico /etc/acidbase/database.php



6. Isi bagian ‘’ seperti di bawah ini:



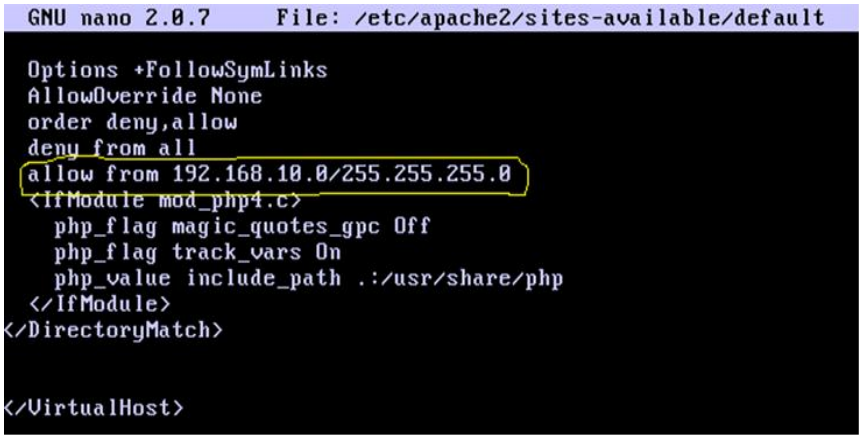
7. Save and exit
8. Ketikkan: pico /etc/apache2/sites-available/default taruh kursor diatas </VirtualHost>, lalu CTRL+R, ketikkan: /etc/acidbase/apache.conf





9. edit bagian allow_from 127.0.0.0/255.0.0.0 menjadi 192.168.10.0/255.255.255.0

   Create



3. Instalasi Paket *Dependency*

Table 3. Daftar paket dependency dan fungsinya

Nama Paket	Perintah Instalasi	Fungsi
Nmap	$ sudo apt–get install nmap	Eksplorasi jaringan atau audit keamanan. Mendukung ping *scanning* (*host* yang *up*), *port scanning*, deteksi versi protokol, dan TCP/IP *finger printing*.
Nbtscan	$ sudo apt–get install nbtscan	Pemindai IP *network* untuk informasi nama

		NetBIOS. Respon berupa list alamat IP, NetBIOS *computer name*, *username* untuk *login*, dan MAC *Address*.
Apache2	$ sudo apt–get install apache2	Paket untuk membangun web server.
Php5	$ sudo apt–get install php5	Merupakan *metapackage*, yang jika diisntal, menjamin bahwa OS telah memiliki setidaknya salah satu dari tiga versi *server-side interpreter* PHP5 diinstal.
Php5-mysql	$ sudo apt–get install php5-mysql	Menyediakan modul koneksi *database* MySQL langsung dari skrip PHP.
Php5-gd	$ sudo apt–get	Menyediakan modul

	install php5-gd	khusus untuk penanganan grafis langsung dari skrip PHP.
Libphp-adodb	$ sudo apt–get install libphp-adodb	Lib yang digunakan untuk koneksi *database* dengan skrip PHP.
Libpcap0.8	$ sudo apt–get install libpcap0.8-dev	Berfungsi untuk meng-*capture* paket.
Libpcre3	$ sudo apt–get install libpcre3-dev	Satu set fungsi-fungsi yang mengimplementasikan *regular expression*.
Mysql-server	$ sudo apt–get install mysql-server	Menginstal *server database* MySQL.
Libmysqlclientl6	$ sudo apt–get install libmysqlclientl6-dev	Menginstal *database client* MySQL.

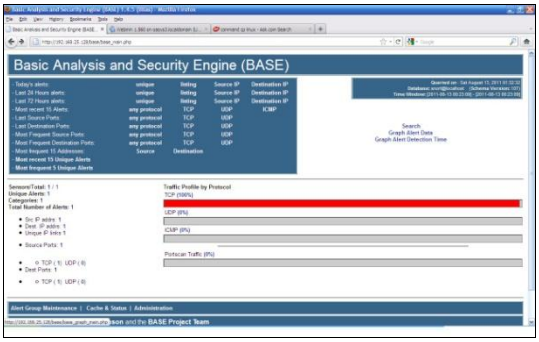
Libdnet 1.11	$ sudo make S sudo make install	Menyediakan penyederhanaan antarmuka portable ke beberapa rutinitas jaringan level rendah.
Libnet 1.0.2a	$ sudo make S sudo make install	Sebuah API *generic networking* yang menyediakan akses menuju beberapa protokol.

4. Instalasi Snort Rules

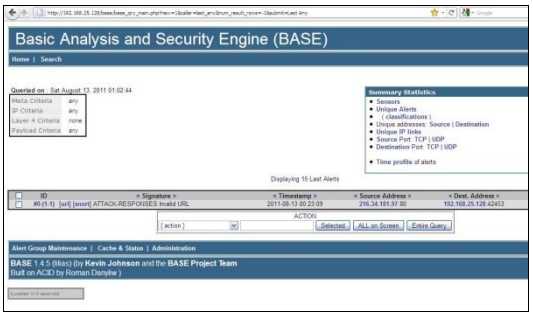
Instalasi paket Snort *rules* ini dilakukan setelah menginstal Snort *engine*. Paket ini merupakan kumpulan *rules* Snort secara *default*.

5. Instal Paket BASE

Instalasi paket BASE merupakan langkah terakhir dalam pembangunan sistem ini. BASE akan membaca database yang ditulis oleh Snort, maka BASE harus dikonfigurasikan dengan database Snort. Karena BASE merupakan program web-based, maka aplikasi ini ditanam adalam direktori /var/www/.



Gambar 13. Tampilan BASE yang telah di instal



Gambar 14. Log Snort pada BASE

3.10.2 Mengoperasikan Snort

Secara umum snort dapat di operasikan dalam tiga (3) buah mode, yaitu

- Sniffer mode, untuk melihat paket yang lewat di jaringan.
- Packet logger mode, untuk mencatat semua paket yang lewat di jaringan untuk di analisa di kemudian hari.
- Intrusion Detection mode, pada mode ini snort akan berfungsi untuk mendeteksi serangan yang dilakukan melalui jaringan komputer. Untuk menggunakan mode IDS ini di perlukan setup dari berbagai rules / aturan yang akan membedakan sebuah paket normal dengan paket yang membawa serangan.

3.10.3 Sniffer Mode

Untuk menjalankan snort pada sniffer mode tidaklah sukar, beberapa contoh perintah-nya terdapat di bawah ini,

./snort –v

./snort –vd

./snort –vde

./snort –v –d –e dengan menambahkan beberapa switch –v, -d, -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.

Contoh hasil sniffing paket di jaringan menggunakan perintah /usr/local/bin/snort –v dapat dilihat berikut ini,

```
[root@gate onno]# /usr/local/bin/snort –v

Log directory = /var/log/snort

Initializing Network Interface eth0\


  --== Initializing Snort ==--

Checking PID path...

PATH_VARRUN is set to /var/run/ on this operating system

PID stat checked out ok, PID set to /var/run/
```

```
Writing PID file to "/var/run/"

Decoding Ethernet on interface eth0

--== Initialization Complete ==--

-*> Snort! <*-

Version 1.8.3 (Build 88)

By Martin Roesch (roesch@sourcefire.com, www.snort.org)

04/18-11:32:00.261488 192.168.120.232:2757 -> 192.168.120.255:8859

UDP TTL:64 TOS:0x0 ID:1735 IpLen:20 DgmLen:38

Len: 18

=+=+=+
```

Datagram Protocol (UDP). Kalimat pertama berisi header IP, beberapa informasi yang penting yang dapat dihat di atas dari kalimat pertama adalah,

04/18-11:32:20.261518 192.168.120.232:2759 -> 192.168.120.255:8859

04 = versi protokol IP yang digunakan

192.168.120.232 = IP address sumber paket

192.168.120.255 = IP address tujuan

2759 = port sumber

8859 = port tujuan

Kalimat ke dua berisi protokol UDP, yang berisi antara lain informasi

UDP TTL:64 TOS:0x0 ID:1737 IpLen:20 DgmLen:38

UDP = memberitahukan bahwa dia protocol UDP.

TTL:64 = Time To Live 64, paket maksimal melalui 64 router.

ID = nomor / identitas

IPLen:20 = panjang byte protokol IP 20 byte

DgmLen = panjang byte seluruh paket 38 byte.

Tentunya hasil akan lain jika kita melihat paket TCP, seperti beberapa contoh berikut,

04/18-11:32:20.573898 192.168.120.114:1707 -> 202.159.32.71:110

TCP TTL:64 TOS:0x0 ID:411 IpLen:20 DgmLen:60 DF

******S* Seq: 0x4E70BB7C Ack: 0x0 Win: 0x16D0 TcpLen: 40

TCP Options (5) => MSS: 1460 SackOK TS: 6798055 0 NOP WS: 0

=+

04/18-11:32:20.581556 202.159.32.71:110 -> 192.168.120.114:1707

TCP TTL:58 TOS:0x0 ID:24510 IpLen:20 DgmLen:60 DF

***A**S* Seq: 0x423A85B3 Ack: 0x4E70BB7D Win: 0x7D78 TcpLen: 40

TCP Options (5) => MSS: 1460 SackOK TS: 163052552 6798055 NOP WS: 0

=+

04/18-11:32:20.581928 192.168.120.114:1707 -> 202.159.32.71:110

TCP TTL:64 TOS:0x0 ID:412 IpLen:20 DgmLen:52 DF

A* Seq: 0x4E70BB7D Ack: 0x423A85B4 Win: 0x16D0 TcpLen: 32

TCP Options (3) => NOP NOP TS: 6798056 163052552

=+

Sama seperti sebelumnya maka line pertama di atas adalah milik Internet Protocol (IP), kita dapat melihat bahwa terjadi komunikasi antara IP 202.159.32.71 port 110 dengan IP 192.168.120.114 port 1707.

Di bawah kalimat milik protocol IP, ada satu kalimat yang berisi informasi umum dari paket yang dikirim merupakan gabungan informasi milik protokol IP & TCP. Beberapa informasi seperti,

TCP TTL:64 TOS:0x0 ID:412 IpLen:20 DgmLen:52 DF

TCP TTL :64= ada 64 router yang masih bisa di lewati.

TOS:0x0 = Type of Service dari protocol IP, 0x0 adalah servis normal.

IpLen:20 = Panjang protocol IP 20 byte.

DgmLen:52 = Panjang seluruh paket 52 byte.

Selanjutnya ada dua (2) kalimat yang semuanya milik Transmission Control Protocol (TCP).

A* Seq: 0x4E70BB7D Ack: 0x423A85B4 Win: 0x16D0 TcpLen: 32

TCP Options (3) => NOP NOP TS: 6798056 163052552

Yang menarik adalah melihat state / kondisi sambungan, terlihat dari jenis paket yang dikirim, seperti

A* = packet acknowledge

******S* = paket sinkronisasi hubungan

***A**S* = paket acknowledge sinkronisasi hubungan

sisanya adalah nomor urut paket data yang dikirim *Sequence number* (Seq), dan *Acknowledge number* (Ack) yang menunjukan sejauh ini nomor paket mana yang sudah diterima dengan baik.

Jenis paket lain yang kadang-kadang terlihat di layar adalah paket *Address Resolution Protocol* (ARP). Contoh adalah,

04/18-11:32:25.451498 ARP who-has 192.168.120.114 tell 192.168.120.1

04/18-11:32:25.451671 ARP reply 192.168.120.114 is-at 0:0:F0:64:96:AE

ARP digunakan untuk menanyakan address dari *hardware*, seperti ethernet *card address*, atau callsign amatir radio di AX.25. Dalam contoh di

atas 192.168.120.1 menanyakan ke jaringan berapa ethernet address dari 192.168.120.114. Dalam kalimat selanjutnya 192.168.120.114 menjawab bahwa dia menggunakan ethernet card dengan address 00:00:f0:64:96:ae.

Tentunya masih banyak lagi jenis protokol yang akan berseliweran di kabel ethernet atau media komunikasi data anda, untuk mempelajari berbagai protocol ini secara detail sangat disarankan untuk membaca keterangan lengkap dari masing-masing protocol yang dapat di download dari situs milik Internet Engineering Task Force IETF http://www.ietf.org.

Untuk mematikan program snort dapat menekan tombol Control-C (^C). Akan tampak pada layar berbagai statistik yang sangat berguna untuk melihat kondisi jaringan

Snort analyzed 255 out of 255 packets, dropping 0(0.000%) packets

Breakdown by protocol: Action Stats:

TCP: 211 (82.745%) ALERTS: 0

UDP: 27 (10.588%) LOGGED: 0

ICMP: 0 (0.000%) PASSED: 0

ARP: 2 (0.784%)

IPv6: 0 (0.000%)

IPX: 0 (0.000%)

OTHER: 15 (5.882%)

DISCARD: 0 (0.000%)

Fragmentation Stats:

Fragmented IP Packets: 0 (0.000%)

Fragment Trackers: 0

Rebuilt IP Packets: 0

Frag elements used: 0

Discarded(incomplete): 0

Discarded(timeout): 0

Frag2 memory faults: 0

TCP Stream Reassembly Stats:

TCP Packets Used: 0 (0.000%)

Stream Trackers: 0

Stream flushes: 0

Segments used: 0

Stream4 Memory Faults: 0

Snort received signal 2, exiting

[root@gate onno]#

3.10.4 Packet Logger Mode

Tentunya cukup melelahkan untuk melihat paket yang lewat sedemikian cepat di layar terutama jika kita menggunakan ethernet berkecepatan 100Mbps, layar anda akan *scrolling* dengan cepat sekali susah untuk melihat paket yang di inginkan. Cara paling sederhana untuk mengatasi hal ini adalah menyimpan dulu semua paket yang lewat ke

sebuah file untuk di lihat kemudian. Beberapa perintah yang mungkin dapat digunakan untuk mencatat paket yang ada adalah

./snort –dev –l ./log

./snort –dev –l ./log –h 192.168.0.0/24

./snort –dev –l ./log –b

perintah yang paling penting untuk me-log paket yang lewat adalah

-l ./log

yang menentukan bahwa paket yang lewat akan di log / di catat ke file ./log. Beberapa perintah tambahan dapat digunakan seperti –h 192.168.0.0/24 yang menunjukan bahwa yang di catat hanya packet dari host mana saja, dan –b yang memberitahukan agar file yang di log dalam format binary, bukan ASCII.

Untuk membaca file log dapat dilakukan dengan menjalankan snort dengan di tambahkan perintah –r nama file log-nya, seperti,

./snort –dv –r packet.log

./snort –dvr packet.log icmp

3.10.5 Intrusion Detection Mode

Mode operasi snort yang paling rumit adalah sebagai pendeteksi penyusup (*intrusion detection*) di jaringan yang kita gunakan. Ciri khas mode operasi untuk pendeteksi penyusup adaah dengan menambahkan perintah ke snort untuk membaca file konfigurasi –c nama-file-konfigurasi.conf. Isi file konfigurasi ini lumayan banyak, tapi sebagian besar telah di set secara baik dalam contoh snort.conf yang dibawa oleh source snort.

Beberapa contoh perintah untuk mengaktifkan snort untuk melakukan pendeteksian penyusup, seperti

```
./snort –dev –l ./log –h 192.168.0.0/24 –c snort.conf
```

```
./snort –d –h 192.168.0.0/24 –l ./log –c snort.conf
```

Untuk melakukan deteksi penyusup secara prinsip snort harus melakukan logging paket yang lewat dapat menggunakan perintah –l nama-file-logging, atau membiarkan snort menggunakan *default file* logging-nya di directory /var/log/snort. Kemudian menganalisa catatan / logging paket yang ada sesuai dengan isi perintah snort.conf.

Ada beberapa tambahan perintah yang akan membuat proses deteksi menjadi lebih effisien, mekanisme pemberitahuan alert di Linux dapat di set dengan perintah –A sebagai berikut,

-*A fast*, *mode alert* yang cepat berisi waktu, berita, IP & port tujuan.

-*A full*, *mode alert* dengan informasi lengkap.

-*A unsock*, *mode alert* ke unix socket.

-*A none*, mematikan mode alert.

Untuk mengirimkan alert ke syslog UNIX kita bisa menambahkan switch –s, seperti tampak pada beberapa contoh di bawah ini.

./snort –c snort.conf –l ./log –s –h 192.168.0.0/24

./snort –c snort.conf –s –h 192.168.0.0/24

Untuk mengirimkan alert binary ke workstation windows, dapat digunakan perintah di bawah ini,

./snort –c snort.conf –b –M WORKSTATIONS

Agar snort beroperasi secara langsung setiap kali *workstation* / server di boot, kita dapat menambahkan ke file /etc/rc.d/rc.local perintah di bawah ini :

/usr/local/bin/snort –d –h 192.168.0.0/24 –c /root/snort/snort.conf –A full –s –D

Atau

/usr/local/bin/snort –d –c /root/snort/snort.conf –A full –s –D

dimana –D adalah switch yang menset agar snort bekerja sebagai Daemon (bekerja dibelakang layar).

3.10.6 Setup snort.conf

Secara umum ada beberapa hal yang perlu di set pada snort.conf, yaitu:

- Set konfigurasi dari jaringan kita.
- Konfigurasi pemrosesan sebelum di lakukan proses deteksi penyusup.
- Konfigurasi output.
- Konfigurasi rule untuk melakukan deteksi penyusup.

Secara umum kita terutama perlu menset konfigurasi jaringan saja, sedang setting lainnya dapat dibiarkan menggunakan *default* yang ada. Khususnya konfigurasi rule jika ingin gampang kita ambil saja contoh file *.rules yang ada di snort.tar.gz.

Konfigurasi jaringan yang perlu dilakukan sebetulnya tidak banyak, hanya mengisi

```
var HOME_NET 192.168.0.0/24
```

memberitahukan snort IP jaringan lokal-nya adalah 192.168.0.0/24 (satu kelas C). Sisanya dapat di diamkan saja menggunakan nilai default-nya.

Bagian konfigurasi output dapat kita mainkan sedikit untuk menset kemana alert & informasi adanya portscan di kirim, secara default akan dimasukan ke /var/log/snort. Sedikit modifikasi perlu dilakukan jika kita menginginkan untuk menggunakan ACID untuk menganalisa alert yang ada. Output perlu dimasukan ke database, seperti MySQL.

Rules biasanya terdapat pada file *.rules. Untuk mengedit sendiri rules agak lumayan, kita membutuhkan pengetahuan yang dalam tentang protokol, payload serangan dll. Untuk pemula sebaiknya menggunakan contoh *.rules yang di sediakan oleh snort yang dapat langsung dipakai melalui perintah include pada snort.conf.

Beberapa contoh rules dari serangan / eksploit dapat dilihat berikut ini:

alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32 overflow /bin/sh"; flags:A+; content:"/bin/sh"; reference:bugtraq,2347; reference:cve,CVE-2001-0144; classtype:shellcode-detect; sid:1324; rev:1;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 22 (msg:"EXPLOIT ssh CRC32 overflow NOOP"; flags:A+; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90|"; reference:bugtraq,2347; reference:cve,CVE-2001-0144; classtype:shellcode-detect; sid:1326; rev:1;)

Detail berbagai parameter rules terdapat di SnortUsersManual.pdf yang juga di sediakan bersama source snort.tar.gz.

3.11 Hasil Deteksi Penyusup

Default snort hasil deteksi penyusup atau paket yang mencurigakan akan disimpan pada folder /var/log/snort. Catatan kemungkinan serangan portscan akan disimpan pada file /var/log/snort/portscan.log, sedang untuk alert akan di letakan pada folder-folder berdasarkan alamat IP sumber serangan karena saya menggunakan mode alert –A full.

Contoh cuplikan isi portscan.log dapat dilihat berikut ini:

Apr 4 19:00:21 202.159.32.71:110 -> 192.168.120.114:2724 NOACK 1*U*P*S*

Apr 4 20:47:43 168.143.117.4:80 -> 192.168.120.114:2916 NOACK 1*U*P*S*

Apr 5 06:04:04 216.136.171.200:80 -> 192.168.120.114:3500 VECNA 1*U*P***

Apr 5 17:28:20 198.6.49.225:80 -> 192.168.120.114:1239 NOACK 1*U*P*S*

Apr 6 09:35:56 202.153.120.155:80 -> 192.168.120.114:3628 NOACK 1*U*P*S*

Apr 6 17:44:06 205.166.76.243:80 -> 192.168.120.114:1413 INVALIDACK *2*A*R*F

Apr 6 19:55:03 213.244.183.211:80 -> 192.168.120.114:43946 NOACK 1*U*P*S*

Apr 7 16:07:57 202.159.32.71:110 -> 192.168.120.114:1655 INVALIDACK *2*A*R*F

Apr 7 17:00:17 202.158.2.4:110 -> 192.168.120.114:1954 INVALIDACK *2*A*R*F

Apr 8 07:35:42 192.168.120.1:53 -> 192.168.120.114:1046 UDP

Apr 8 10:23:10 192.168.120.1:53 -> 192.168.120.114:1030 UDP

Apr 8 10:23:49 192.168.120.1:53 -> 192.168.120.114:1030 UDP

Apr 20 12:03:51 192.168.120.1:53 -> 192.168.120.114:1077 UDP

Apr 21 01:00:11 202.158.2.5:110 -> 192.168.120.114:1234 INVALIDACK *2*A*R*F

Apr 21 09:17:01 66.218.66.246:80 -> 192.168.120.114:42666 NOACK 1*U*P*S*

Apr 21 11:00:28 202.159.32.71:110 -> 192.168.120.114:1800 INVALIDACK *2*A*R*F

Secara umum kita akan dapat membaca tanggal & jam serangan, IP address & port sumber, IP address & port tujuan, protokol yang digunakan, kesalahan yang terjadi beserta pointer pada protokol TCP-nya seperti,

U = *Urgent*

A = *Acknowledge*

P = *Push*

F = *Final*

Detail dari berbagai pointer TCP tersebut dapat diketahui dengan membaca detail protokol TCP.

Contoh alert dari dapat dilihat berikut ini:

[**] INFO - Possible Squid Scan [**]

04/20-14:06:49.953376 192.168.0.33:1040 -> 192.168.0.1:3128

TCP TTL:128 TOS:0x0 ID:393 IpLen:20 DgmLen:48 DF

******S* Seq: 0x60591B9 Ack: 0x0 Win: 0x4000 TcpLen: 28

TCP Options (4) => MSS: 1460 NOP NOP SackOK

Tampak pada contoh alert di atas, ada usaha untuk menscan keberadaan Sqiud proxy server pada 192.168.0.1 port 3128 dari workstation 192.168.0.33 port 1040. Workstation 192.168.0.33 mengirimkan paket Sinkronisasi TCP terlihat dari ******S*.

3.12 Pengujian

3.12.1 Tujuan Pengujian

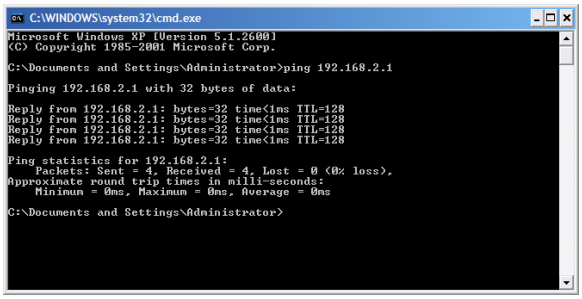
Setelah proses pembangunan komponen selesai, kemudian masuk ketahap selanjutnya yaitu pengujian sistem. Pengujian ini bertujuan untuk memastikan bahwa sistem yang dibangun dapat memenuhi tujuannya. Pengujian dilakukan antara *Client-Server*. Sesuai dengan skenario, pengujian difokuskan pada intrusi dari jaringan internal. Karena intrusi paling sering terjadi dari *client* dalam jaringan sendiri. Pengujian dilakukan berdasarkan serangan yang sering terjadi pada perusahaan.

1. Serangan ICMP flooding

   Penyerang: *Client* (IP Address: 192.168.26.1)Tujuan : *Server* #3 (IP Address: 192.168..26.132)

   Penyerang melakukan *ping* dengan *byte* besar agar memenuhi *bandwidth* dan membanjiri jaringan dengan request pengiriman paket besar-besaran sekaligus. Ping yang diberikan adalah 1001 bytes. Dimana ping normal hanya berukuran 56 bytes. Dengan demikian Snort akan menganggap ping tersebut sebagai suatu intrusi.

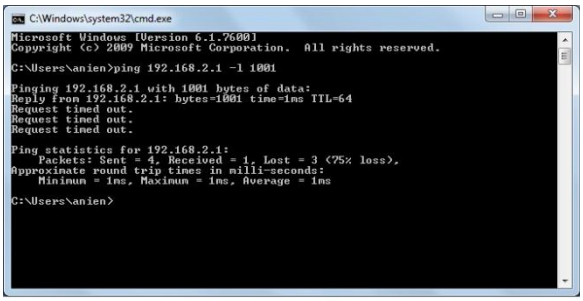
Berikut ping dalam keadaan normal (32 bytes) dari komputer client:



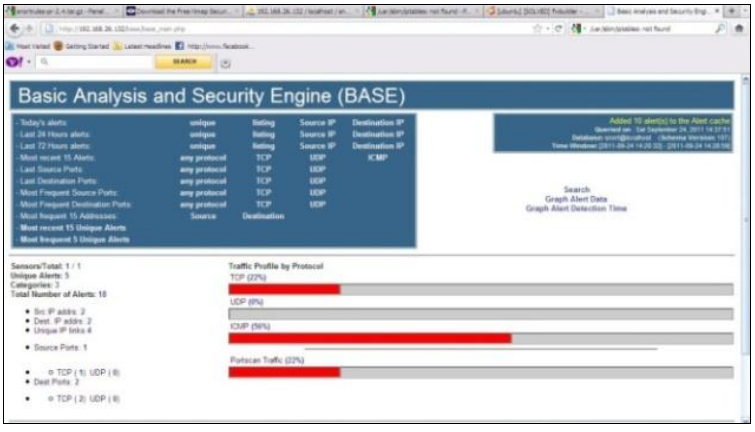
Gambar 15. Ping normal

Ping dalam jumlah kecil akan dibaca oleh Snort bukan sebagai intrusi karena tidak sesuai dengan pola yang ada pada rules. Snort akan meloloskan paket ping ke alamat tujuannya.

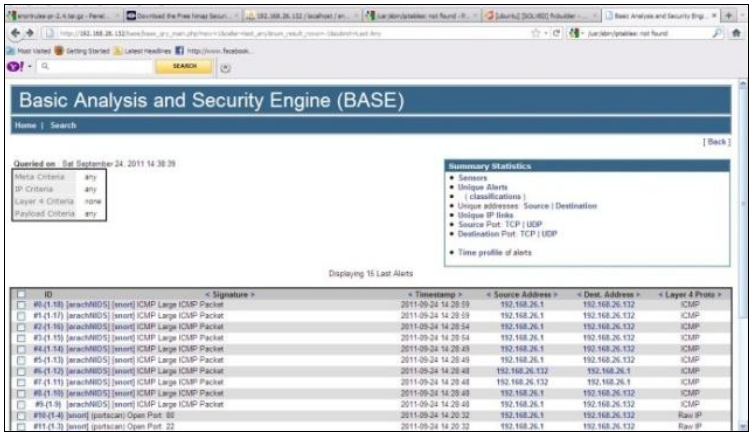
Berikut ping keadaan intrusi dengan 1001 bytes dari komputer client:



Gambar 16. Ping intrusi dengan 1001 bytes



Gambar 17. Alert pada BASE



Gambar 18. List alert BASE

```
root@Oasys-Server:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
BLOCKIT    all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain BLOCKIT (1 references)
target     prot opt source               destination
DROP       all  --  192.168.26.1         anywhere
root@Oasys-Server:~#
```

Gambar 19. Tampilan pada Firewall server

Hasil:

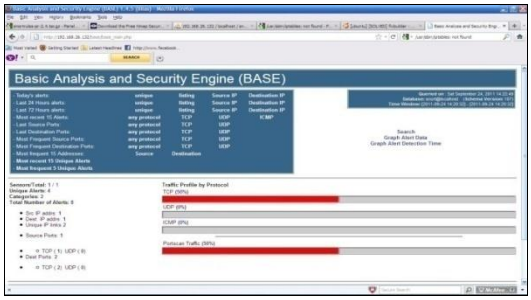
Snort berhasil mendeteksi intrusi *flooding* dan mencatat *alert* dan *log* ke database. BASE menerima alert tersebut dan menyajikannya dalam GUI. BlockIt berhasil mentrigger Firewall untuk melakukan *dropping* paket dan *blocking* IP Address penyerang.

1. Port scanning

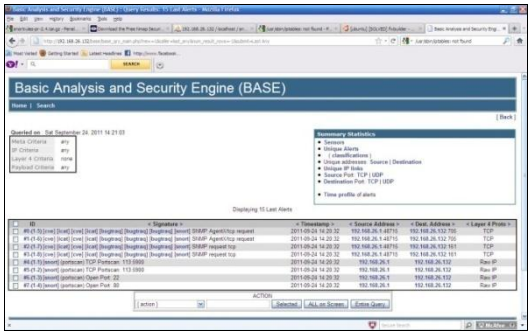
   Penyerang: *Client* (IP Address: 192.168.26.1)

Tujuan : *Server* #3 (IP Address: 192.168..26.132)

Segala jenis intrusi berawal dari mengetahui alamat dan *port* tujuan yang akan disusupi. *Client* akan mencari *port* yang dapat dia susupi dengan cara men-*scanning* port Server. Pada pengujian ini digunakan tools Nmap-5.51. Nmap berfungsi untuk memindai port yang ada pada server.



Gambar 20. Tampilan alert pada BASE



Gambar 21. List alert pada BASE

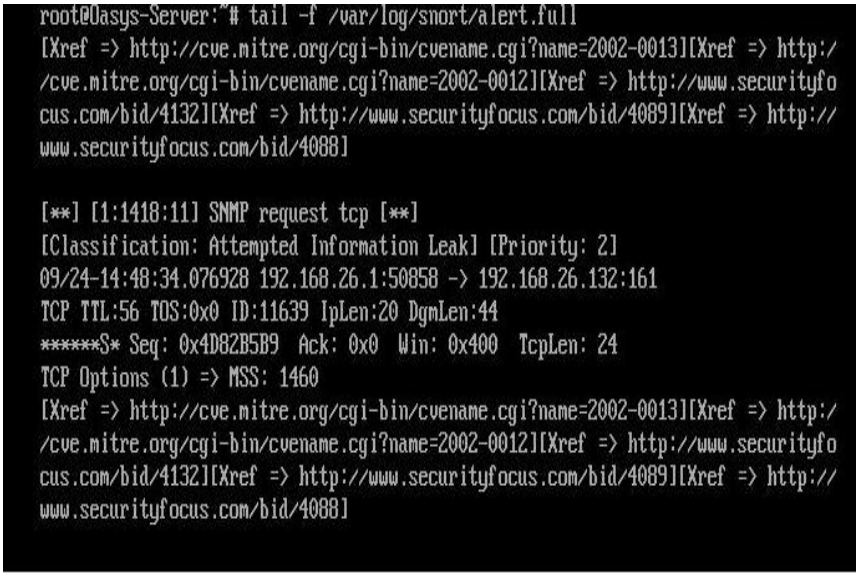
```
root@Oasys-Server:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
BLOCKIT    all  --  anywhere             anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain BLOCKIT (1 references)
target     prot opt source               destination
DROP       all  --  192.168.26.1         anywhere
root@Oasys-Server:~# _
```

Gambar 22. Tampilan pada Firewall server



Gambar 23. Tampilan pada log Snort

Hasil:

Snort berhasil mendeteksi intrusi *port scanning* dan mencatat *alert* dan *log* ke *database*. BASE menerima *alert* tersebut dan menyajikannya dalam GUI. BlockIt berhasil mentrigger Firewall untuk melakukan *dropping* paket dan *blocking* IP Address penyerang.