Pemillihan IDS (Intrusion Detection System) sebagai Sistem Keamanan Jaringan Server di Politeknik Batam

TUGAS AKHIR

Oleh :

Heru Suparsin 3310801036

Mariaty H 3310801125

Disusun untuk memenuhi syarat kelulusan Program Diploma III

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK BATAM

BATAM

2011

LEMBAR PENGESAHAN

Batam, 11 Februari 2011

Pembimbing,

Nur Cahyono K, S.Si

NIK. 106044

LEMBAR PERNYATAAN

Dengan ini, saya:

Nim : 3310801036

Nama: Heru Suparsin

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Pemillihan IDS (Intrusion Detection System) sebagai Sistem Keamanan Jaringan Server di Politeknik Batam

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 11 Februari 2011

Heru Suparsin
3310801036

LEMBAR PERNYATAAN

Dengan ini, saya:

Nim : 3310801125

Nama: Mariaty H

adalah mahasiswa Teknik Informatika Politeknik Batam yang menyatakan bahwa tugas akhir dengan judul:

Pemillihan IDS (Intrusion Detection System) sebagai Sistem Keamanan Jaringan Server di Politeknik Batam

disusun dengan:

1. tidak melakukan plagiat terhadap naskah karya orang lain
2. tidak melakukan pemalsuan data
3. tidak menggunakan karya orang lain tanpa menyebut sumber asli atau tanpa ijin pemilik

Jika kemudian terbukti terjadi pelanggaran terhadap pernyataan di atas, maka saya bersedia menerima sanksi apapun termasuk pencabutan gelar akademik.

Lembar pernyataan ini juga memberikan hak kepada Politeknik Batam untuk mempergunakan, mendistribusikan ataupun memproduksi ulang seluruh hasil Tugas Akhir ini.

Batam, 11 Februari 2011

Mariaty H
3310801125

KATA PENGANTAR

Puji dan syukur kehadirat Tuhan Yang Maha Esa atas berkat dan karuniaNya, penulis dapat menyelesaikan Tugas Akhir sesuai dengan waktu yang telah ditentukan. Penelitian terhadap pemillihan IDS (Intrusion Detection System) sebagai sistem keamanan jaringan server di Politeknik Batam dibuat dengan tujuan untuk mengetahui software IDS yang cocok untuk Politeknik Batam, kelebihan dan kelemahan antara software IDS yang satu dengan yang lain, serta dapat menjadi referensi instalasi *Intrusion Detection System*. Dalam kesempatan ini pula penulis mengucapkan terima kasih kepada:

1. Bapak Ir. Priyono Eko Sanyoto selaku direktur Politeknik Batam
2. Bapak Uuf Brajawidagda, MT selaku koordinator Tugas Akhir
3. Bapak Nur cahyono, S.Si selaku pemberi ide/konsep dalam pencarian judul Tugas Akhir dan dosen pembimbing
4. Dosen program studi Teknik Informatika atas bimbingannya
5. Keluarga yang telah memberikan doa serta dukungan
6. Semua pihak yang telah memberikan doa dan dukungannya

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan laporan ini. Oleh karena itu penulis sangat mengharapkan bantuan dari beberapa pihak baik berupa kritik maupun saran guna untuk penyempurnaan selanjutnya. Akhir kata penulis mengucapkan terima kasih, semoga penulisan laporan ini dapat bermanfaat bagi pembaca yang ingin mengembangkan sebuah penelitian yang serupa.

Batam, Februari 2011

Penulis

ABSTRAKSI

Pemillihan IDS (Intrusion Detection System) sebagai Sistem Keamanan Jaringan Server di Politeknik Batam

Intrusion Detection System adalah sistem dirancang untuk mengumpulkan informasi tentang aktivitas berbahaya dalam jaringan, menganalisis informasi, dan memberikan peringatan jika terdapat intrusi. Tujuan dari tugas akhir ini adalah untuk menentukan *software intrusion detection system* yang cocok digunakan di Politeknik Batam sebagai sistem keamanan jaringan server. *Software* intrusion detection system yang dibandingkan dalam tugas akhir ini adalah snort dan base, suricata, dan ossec.

Kata Kunci: *Intrusion Detection System*

ABSTRACT

IDS Election (Intrusion Detection System) as a Network Security System Server in Politeknik Batam

Intrusion detection system is a system designed to collect information about malicious activity on the network, analyse information, and provide warnings if there is intrusion. The purpose of this final project is to determine the software intrusion detection system suitable for use at the Politeknik Batam as a network security system server. Software of intrusion detection system compared in this final project is snort and base, suricata, and ossec.

Key words: Intrusion detection system.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

Bab I Pendahuluan

Pada bab ini akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan dan sistematika penulisan.

I.1 Latar Belakang

Perkembangan jaringan komputer terus mengalami peningkatan, baik dari skalabilitas, jumlah *node* dan teknologi yang digunakan. Oleh karena itu, sangat dibutuhkan pengelolaan jaringan yang baik. Administrator jaringan yang bertindak sebagai pengelola jaringan memiliki banyak permasalahan dalam melakukan tugasnya, diantaranya yang berkaitan dengan keamanan jaringan komputer. Tujuan utama dari keamanan sistem adalah memberikan jalur yang aman antar entitas dalam melakukan pertukaran informasi dan untuk menyediakan perlindungan data.

Intrusion (penyusupan) merupakan kegiatan yang berusaha merusak atau menyalahgunakan sistem. Hal ini yang dapat mengakibatkan kerusakan sebagian sistem maupun secara keseluruhan.

Proses pengelolaan jaringan dapat dilakukan dengan memanfaatkan *Intrusion Detection System* (IDS). *Intrusion Detection System* (IDS) merupakan usaha mengidentifikasi adanya penyusup yang dapat melakukan perusakan atau penyalahgunaan sistem. Untuk melakukan pemilihan *software* IDS yang tepat, dapat dilakukan melalui proses perbandingan antara berbagai *software* IDS yang ada. Berdasarkan proses perbandingan tersebut maka diketahui kelemahan dan kelebihan dari masing-masing *software*, serta membantu dalam menentukan pilihan *software* yang tepat sebelum diterapkan pada sistem keamanan jaringan di Politeknik Batam.

I.2 Rumusan Masalah

Rumusan masalah dalam analisis ini adalah:

1. Apa kriteria yang dipakai dalam proses pemilihan IDS.
2. IDS apakah yang cocok digunakan sebagai sistem keamanan *server* Politeknik Batam.

I.3 Batasan Masalah

Batasan masalah dalam analisis ini adalah :

1. Proses perbandingan hanya akan dilakukan pada beberapa *software Intrusion Detection System*, tidak dilakukan pada semua *software software Intrusion Detection System*.
2. Proses pengujian menggunakan *software Intrusion Detection System open source*, bukan komersial (berbayar).
3. Proses pengujian tidak dilakukan pada semua bentuk topologi jaringan.

I.4 Tujuan

Tujuan dari analisis ini adalah:

1. Mengetahui IDS yang sesuai untuk sistem keamanan jaringan Politeknik Batam.
2. Referensi instalasi *Intrusion Detection System*.
3. Mengetahui kelebihan dan kelemahan *software-software* IDS.

I.5 Sistematika Penulisan

Laporan ini terdiri dari Bab Pendahuluan, Landasan Teori, Pemilihan dan Perancangan Pengujian *Intrusion Detection System*, Implementasi dan Pembahasan, Kesimpulan dan Saran serta Lampiran yang berhubungan dengan proses instalasi.

Bab 1 Pendahuluan berisi penjelasan mengenai latar belakang masalah dalam proses penelitian, perumusan masalah, batasan masalah dalam proses penelitian, tujuan penelitian, dan sistematika penulisan yang memberikan gambaran isi laporan tugas akhir.

Bab 2 Landasan Teori berisi mengenai studi literatur yang digunakan sebagai referensi dalam proses penelitian yakni definisi *Intrusion Detection System*, tujuan penggunaan *Intrusion Detection System*, jenis-jenis *Intrusion Detection System*, cara kerja *Intrusion Detection System*, *my sql*, *firewall*, jenis-jenis *firewall*, jenis-jenis serangan.

Bab 3 Pemilihan dan Perancangan Pengujian *Intrusion Detection System* dan berisi mengenai proses pemilihan, *software-software Intrusion Detection System*, perancangan pengujian, skema jaringan, lingkungan pengujian, kriteria evaluasi, *tools* serangan yang digunakan.

Bab 4 Implementasi dan pengujian berisi mengenai proses implementasi *software-software* IDS, evaluasi *software* IDS berdasarkan kriteria yang digunakan, perbandingan *software-software* IDS.

Bab 5 Kesimpulan, Saran dan Solusi berisi mengenai penyimpulan hasil dari proses

implementasi dan pengujian pada bab sebelumnya, saran sebagai bahan pertimbangan untuk pengembangan penelitian selanjutnya.

Bab II Landasan Teori

Pada bab ini akan dijelaskan mengenai definisi *intrusion detection system*, tujuan penggunaan IDS, *mysql*, *firewall*, jenis serangan, dan skema jaringan di Politeknik Batam.

I.1 Definisi Intrusion Detection System (IDS)

Intrusion Detection System (IDS) adalah sistem komputer (dapat berupa *software* dan *hardware*) yang berusaha melakukan deteksi penyusupan. *Intrusion Detection System* akan memberikan pemberitahuan saat mendeteksi sesuatu yang dianggap sebagai tindakan illegal. IDS tidak melakukan pencegahan terjadinya penyusupan.[1]

I.1.1 Tujuan Penggunaan IDS

IDS merupakan *software* atau *hardware* yang melakukan otomasi proses *monitoring* kejadian yang muncul di sistem komputer atau jaringan dan menganalisanya untuk menemukan permasalahan keamanan. IDS adalah pemberi sinyal pertama jika seorang penyusup mencoba membobol sistem keamanan komputer. Secara umum penyusupan dapat berarti serangan atau ancaman terhadap keamanan dan integritas data, serta tindakan atau percobaan untuk melewati sebuah sistem keamanan yang dilakukan oleh seseorang dari internet atau dari dalam sistem.[2]

IDS dibuat bukan untuk menggantikan fungsi *firewall* karena kegunaannya berbeda. Sebuah sistem *firewall* tidak dapat mengetahui apakah sebuah serangan sedang terjadi atau tidak, tapi IDS dapat mengetahuinya. Dengan meningkatnya jumlah serangan pada jaringan, IDS merupakan sesuatu yang diperlukan pada infrastruktur keamanan di kebanyakan organisasi. Secara singkat, fungsi IDS adalah pemberi peringatan kepada administrator atas serangan yang terjadi pada sistem.

I.1.2 Jenis-Jenis Intrusion Detection System (IDS)

1. NIDS (*Network Intrusion Detection System*)

   Penempatannya dilakukan pada tempat atau titik yang strategis atau sebuah titik di dalam sebuah jaringan untuk melakukan pengawasan terhadap lalulintas yang menuju atau berasal dari jaringan.

---

[1] http://comes.umy.ac.id/file.php/1/etc/Membangun_Sistem_Intrusion_Detection.pdf

[2] http://yudiagusta.files.wordpress.com/2009/11/288-296-snsi07-050-analisis-perancangan-perangkat-lunak-intrusion-detection-system-_ids_-pada-jaringan-komputer-berbasis-teknologi-mobile.pdf

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

   Memantau aktivitas sebuah *host* jaringan apakah terjadi sebuah percobaan serangan atau penyusupan ke dalamnya atau tidak.

I.1.3 Cara Kerja IDS

Berdasarkan cara kerja IDS dalam menganalisis apakah paket data dianggap sebagai penyusupan atau bukan, IDS dibagi 2 yaitu:

1. Knowledge-*based* atau *misuse detection*

   Cara kerjanya adalah menyadap paket data kemudian membandingkannya dengan *database rule* IDS (berisi *signature-signature* paket serangan). Jika paket data mempunyai pola dengan salah satu pola di *database rule* IDS, maka data tersebut dianggap sebagai serangan.

2. Behavior-*based* atau anomaly *based*

   Cara kerjanya adalah dengan mengamati adanya kejanggalan-kejanggalan pada sistem, sebagai contoh adanya penggunaan memori yang meningkat secara terus-menerus atau ada koneksi paralel dari 1 IP dalam jumlah banyak dan dalam waktu yang bersamaan.

   Berdasarkan kemampuan mendeteksi penyusupan pada jaringan, IDS dibagi 2 yaitu:

   1. Host *based Intrusion Detection System*

      Hanya mampu mendeteksi penyusupan pada *host* tempat implementasi IDS.

   2. *Network based Intrusion Detection System*

      Mampu mendeteksi seluruh *host* yang berada satu jaringan dengan *host* implementasi IDS tersebut.

I.2 MySQL

SQL (*Structured Query Language*) adalah bahasa standar yang digunakan untuk mengakses *server database*. Semenjak tahun 70-an SQL telah dikembangkan oleh IBM, yang kemudian diikuti dengan adanya oracle, Informix, dan Sybase. Dengan menggunakan SQL, proses akses *database* menjadi lebih *user-friendly* dibandingkan dengan misalnya dBase ataupun clipper yang masih menggunakan perintah-perintah pemrograman murni.

MySQL adalah sebuah *server database* SQL *multiuser* dan *multi-threaded*. SQL sendiri adalah salah satu bahasa *database* yang paling popular didunia. Implementasi program *server database* ini adalah program daemon mysqld dan beberapa program lain.

MySQL dibuat oleh TcX dan telah dipercaya mengelola sistem dengan 40 buah *database* berisi 10.000 tabel dan 500 diantaranya memiliki 7 juta baris (kira-kira 100 gigabyte data). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan. Walaupun memiliki kemampuan yang cukup baik, MySQL untuk sistem operasi UNIX bersifat *freeware,* dan terdapat versi *shareware* untuk sistem operasi windows.

Sebagaimana *database* sistem yang lain, dalam SQL dikenal juga hierarki *server* dengan *database-database*. Tiap-tiap *database* memiliki tabel-tabel.Setiap tabel memiliki *field-field*.

Keunggulan MySQl antara lain:

1. MySQL merupakan program yang *multi-threaded,* sehingga dapat dipasang pada *server* yang memiliki *multi-CPU*.
2. Didukung program-program umum seperti C, C++, Java, Perl, PHP, Python, TCL APIs dls.
3. Bekerja pada berbagai *platform* (tersedia berbagai versi untuk berbagai sistem operasi).
4. Memiliki jenis kolom yang cukup banyak sehingga memudahkan konfigurasi sistem *database*.
5. Memiliki sistem keamanan yang cukup baik dengan verifikasi *host*.
6. Mendukung ODBC untuk sistem operasi windows.
7. Mendukung *record* yang memiliki kolom dengan panjang tetap atau panjang bervariasi.
8. MySQL merupakan *software free.*
9. MySQL dan PHP saling terintegrasi. Maksudnya adalah pembuatan *database* dengan menggunakan sintak PHP dapat dibuat. Sedangkan *input* yang di masukkan melalui aplikasi *web* yang menggunakan *script server-side* seperti PHP dapat berlangsung di masukkan ke *database* MySQL yang ada di *server* dan tentunya *web* tersebut berada disebuah *web server*.

I.3 Firewall

Firewall adalah sebuah sistem atau perangkat yang mengizinkan lalu lintas jaringan yang dianggap aman untuk melaluinya dan mencegah lalu lintas jaringan yang tidak aman. Umumnya, *firewall* diterapkan dalam sebuah mesin yang terdedikasi, yang berjalan pada *gateway* antara jaringan lokal dan jaringan lainnya. *Firewall* umumnya digunakan untuk mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan lokal dari pihak

luar[3]. Saat ini, istilah *firewall* menjadi istilah lazim yang merujuk pada sistem yang mengatur komunikasi antar dua jaringan yang berbeda.

Secara fundamental , *firewall* dapat berfungsi sebagai berikut:

1. Mengatur dan mengontrol lalu lintas jaringan.
2. Melakukan autentikasi terhadap akses.
3. Melindungi sumber daya dalam jaringan lokal.

I.3.1 Jenis-Jenis firewall

Firewall terbagi menjadi dua jenis, yaitu sebagai berikut:

1. *Personal Firewall* : didesain untuk melindungi sebuah komputer yang terhubung ke jaringan dari akses yang tidak dikehendaki. *Firewall* jenis ini akhir-akhir ini berevolusi menjadi sebuah kumpulan program yang bertujuan untuk mengamankan komputer secara total dengan ditambahkannya beberapa fitur pengaman tambahan seperti perangkat proteksi terhadap virus, *anti-spyware*, *anti-spam*,dll. Bahkan beberapa produk *firewall* lainnya dilengkai dengan fungsi pendeteksian gangguan keamanan jaringan (*Intrusion Detection System*). Contoh *firewall* dari jenis ini adalah Microsoft Windows *Firewall*. *Personal firewall* secara umum hanya memiliki dua fitur utama, yaitu *Packet Filter Firewall* dan *Stateful Firewall*.
2. *Network Firewall:* didesain untuk melindungi jaringan secara keseluruhan dari berbagai serangan. Ada dua bentuk yaitu sebuah perangkat yang terdedikasi atau sebuah perangkat lunak yang diinstalasikan dalam sebuah *server*. *Network Firewall* secara umum mempunyai beberapa fitur utama yaitu *Packet Filter Firewall*, *Stateful Firewall*, *Circuit Level Gateway*, *Application Level Gateway*, dan NAT *Firewall*. Network *Firewall* umumnya bersifat transparan (tidak terlihat) dari pengguna dan menggunakan teknologi *routing* untuk menentukan paket mana yang dizinkan, dan paket mana yang ditolak.[4]

I.4 Jenis serangan

I.4.1 Denial of Service

Denial of service merupakan jenis serangan terhadap sebuah komputer atau *server* dengan cara menghabiskan *resources* (sumber) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak

[3] http://akuyola.wordpress.com/2009/01/03/firewall/
[4] http://community.gunadarma.ac.id/blog/view/id_9032/title_jenis-jenis-firewall/

langsung mencegah pengguna lain mendapatkan layanan dari *server*/komputer yang diserang tersebut.[5]

Beberapa cara yang dilakukan oleh penyerang dalam melakukan *denial of service*, yakni sebagai berikut:

1. *Traffic flooding* merupakan teknik yang digunakan dengan membanjiri *traffic network* dengan data sehingga *traffic network* yang datang dari pengguna yang terdaftar, tidak dapat masuk ke dalam sistem jaringan.
2. *Request flooding* dilakukan dengan membanjiri jaringan dengan banyak *request* terhadap sebuah layanan jaringan yang disediakan oleh sebuah *host* sehingga *request* yang datang dari pengguna yang terdaftar tidak dapat dilayani oleh layanan tersebut.
3. Mengubah informasi konfigurasi sistem atau bahkan perusakan fisik terhadap komponen dan *server* yang dapat mengganggu komunikasi antara *host* dengan kliennya.

I.4.1.1 Cara Kerja Serangan DDoS

Cara kerja DDoS dalam melakukan serangan kepada situs yang diinginkan. Secara sederhana serangan DDoS dapat dilakukan dengan menggunakan perintah *ping*. Proses *ping* ini ditujukan kepada situs yang akan menjadi korban. Jika perintah ini hanya dilakukan oleh sebuah komputer, perintah ini mungkin tidak akan menimbulkan efek bagi komputer korban. Akan tetapi, jika perintah ini dilakukan oleh banyak komputer kepada satu situs maka perintah ini dapat memperlambat kerja komputer korban.

Satu komputer mengirimkan data sebesar 32 bytes/detik ke situs yang dituju. Jika ada 10.000 komputer yang melakukan perintah tersebut secara bersamaan, itu artinya ada kiriman data sebesar 312 Mega Bytes/detik yang diterima oleh situs yang dituju tadi. Dan *server* dari situs yang dituju tadi pun harus merespon kiriman yang dikirim 10.000 komputer secara bersamaan, Jika 312 MB/detik data yang harus diproses oleh *server*, dalam 1 menit saja, *server* harus memproses kiriman data sebesar 312 MB x 60 detik = 18.720 MB . Dapat dipastikan situs yang diserang dengan metode ini akan mengalami overload/kelebihan data, dan tidak sanggup memproses kiriman data yang datang. Komputer-komputer lain yang ikut melakukan serangan tersebut disebut komputer *zombie*, dimana sudah terinfeksi semacam *adware*. Jadi si penyerang hanya memerintahkan komputer utamanya untuk mengirimkan perintah ke komputer *zombie* yang sudah terinfeksi agar melakukan *ping* ke situs yang dituju.

[5] http://www.knowledg-e.co.cc/2009/10/serangan-dos-inggris-denial-of-service.html

I.4.1.2 Efek Dari Serangan DDoS

Efek dari serangan DDoS sangat mengganggu pengguna *internet* yang ingin mengunjungi situs yang telah diserang menggunakan DDoS. Situs yang terserang DDoS sulit untuk diakses bahkan mungkin tidak dapat diakses. Kesulitan pengaksesan sebuah situs di internet dapat saja merugikan sebagian orang yang bisnisnya sangat tergantung pada layanan *internet*. Secara umum korban serangan DDoS ini hanya sadar bahwa serangan seperti ini hanya merupakan gangguan yang memerlukan *restart system*. Dengan serangan DDoS ini juga dapat merupakan pengalihan *point of view* dari si *hacker* untuk mendapatkan informasi penting yang ada. Pada dasarnya serangan DDoS ini merupakan rangkaian rencana kerja yang telah disusun oleh *hacker* dalam mencapai tujuannya yang telah ditargetkan.

I.4.2 Scanning

Scanning merupakan aktivitas yang dilakukan untuk mendapatkan informasi target.

Adapun informasi yang ditemukan oleh penyerang antara lain *IP address*, sistem operasi, arsitektur sistem, *service* running di tiap komputer.

Scanning dapat dibagi menjadi tiga jenis yaitu:

1. *Port Scanning* merupakan *scanning* yang bertujuan untuk menemukan *port-port* yang terbuka dari suatu *host*.
2. *Network Scanning* merupakan *scanning* yang bertujuan untuk menemukan *host* atau komputer yang aktif pada suatu jaringan.
3. *Vulnerability Scanning* merupakan *scanning* yang bertujuan menemukan kelemahan dari suatu sistem.

Tujuan dari scanning antara lain:

1. Untuk mendeteksi *live* sistem yang berjalan di jaringan.
2. Untuk menemukan *port* yang aktif/*running*.
3. Untuk menemukan sistem operasi yang berjalan di sistem target.
4. Untuk menemukan *service* yang berjalan.
5. Untuk menemukan *IP address* sistem target.

Terdapat metodologi ataupun langkah-langkah yang dilakukan dalam melakukan *scanning* antara lain:[6]

1. Discover/ reconnaissance

[6] http://syah69.blogspot.com/2008/04/metodologi-hacking.html

Reconnaissance dikenal juga dengan sebutan *footprinting*, yang bertujuan untuk mendapatkan informasi awal, seperti alamat IP, DNS *server*, *domain* , tabel *routing*, sistem operasi, dsb. Intinya adalah mendapatkan informasi detail sebanyak-banyaknya sebagai persiapan untuk melakukan langkah selanjutnya. Seluruh informasi tersebut tidak selalu diambil secara diam-diam.

2. Scanning

   Setelah mengenali sistem secara keseluruhan, penyerang mulai mencari jalur penyusupan yang lebih spesifik. Jalur tersebut dapat berupa *port*. *Port* yang umum digunakan sistem antara lain *port* 80 untuk HTTP, *port* 21 untuk FTP, *port* 1433 untuk Microsoft SQL *Server*, *port* 3389 untuk terminal *service*, dsb.

3. Enumeration

   Langkah selanjutnya yang dilakukan untuk mengambil informasi yang lebih detail. Informasi tersebut dapat berupa *user-user*, *sharing-folder*, *service* yang berjalan termasuk versinya.

4. Penetration

   Pada tahap ini, penyerang mengambil alih sistem setelah memperoleh informasi-informasi yang dibutuhkan. Bisa jadi penyerang masuk tidak dengan hak *administrator*, tetapi mampu menyerang *resource* sehingga akhirnya mendapatkan hak akses *administrator*. Dapat dikatakan, jika penyerang sampai masuk ke tahap ini, berarti telah melewati pintu terpenting pertahanan sistem. Terkadang jebolnya pintu keamanan ini diakibatkan oleh kelalaian sistem itu sendiri. Sebagai contohnya adalah penggunaan *password* yang lemah dan mudah ditebak, kesalahan pemrograman yang mengakibatkan terbukanya serangan dari luar. Karena itu, selain melakukan konfigurasi sistem dan jaringan yang baik, pengamanan dari sisi pemrograman juga sangat vital.

5. Elevation

   Setelah mampu mengakses sistem, maka penyerang mengubah status *privilege*nya setara dengan *user* yang memiliki hak penuh terhadap sistem.

6. Pilfer

   Dengan memperoleh control penuh dari sistem, penyerang leluasa untuk melakukan apa yang dikehendakinya, seperti mengambil data yang baik dalam bentuk *text file*, *database*, dokumen, *e-mail*, dsb.

7. Expansion

Tidak hanya menyusup pada suatu sistem, penyerang dapat memperluas penyusupannya dengan memasuki sistem atau jaringan yang lain. Dalam tahap ini, seorang penyerang melakukan kembali proses *reconnaissance, scanning*, dan *enumeration* dengan target sistem yang lain.

8. Housekeeping

   Dengan melakukan proses yang sering disebut dengan *covering track*, penyerang berusaha menghapus jejaknya dengan bersih.

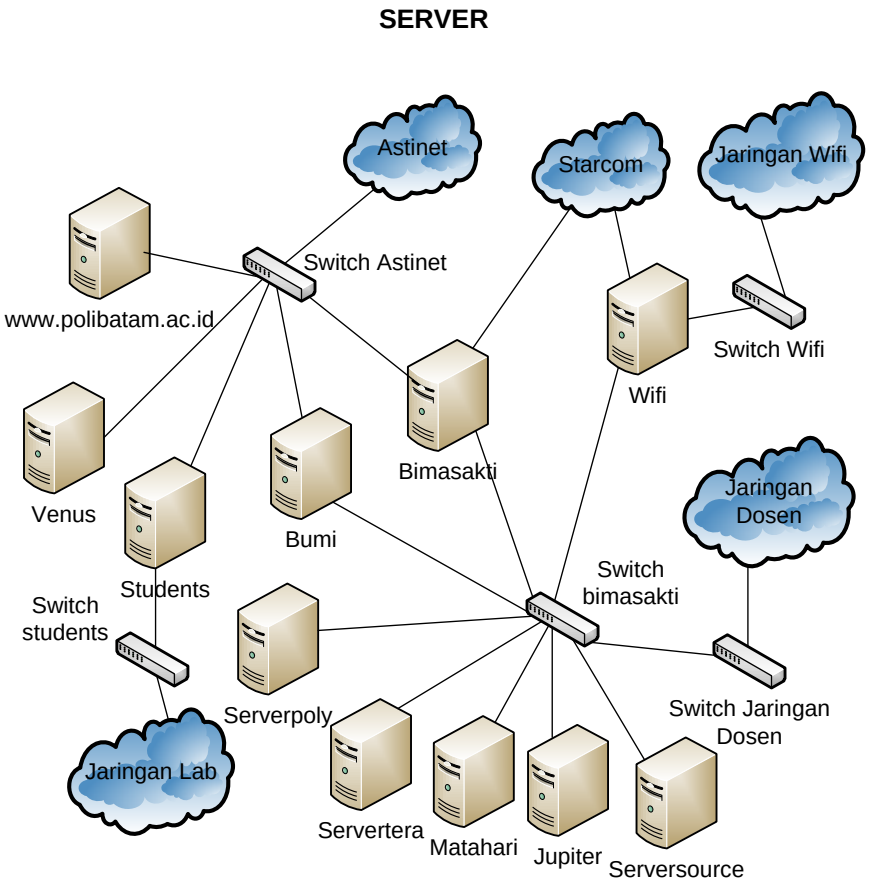
Klasifikasi *scanning* antara lain:

1. TCP *connect/full open scan*
   a. Bentuk *scanning* yang paling banyak digunakan.
   b. *connect()* sistem *call* disediakan oleh sistem operasi.
   c. Jika *port* terbuka maka *connect()* akan berhasil.
2. *Half open scan*
   a. Seringkali dianggap sebagai *half open scan* karena bukan benar-benar *open full* TCP koneksi.
   b. Pertama sebuah SYN paket dikirim ke sebuah *port* dari mesin sebagai permintaan koneksi dan jawaban ditunggu.
   c. Jika *port* mengirimkan balik sebuah SYN/ACK paket kemudian diduga *port* telah terbuka.
   d. Keuntungan kunci dari scan ini adalah lebih sedikit site lognya.
3. FIN *stealth scan*
   a. FIN paket dapat berjalan melalui beberapa program dimana dapat mendeteksi SYN paket yang dikirin ke *port* terlarang.
   b. Hal ini disebabkan *port* yang tertutup cenderung melaporkan FIN paket ketika *port* terbuka mengabaikan paket.
4. FTP *bounce scan*
   a. Ini merupakan bagian dari *port* scanning dimana membuat penggunaan Bounce Attack menjadi lebih mudah di FTP *server*.
   b. Kemudahan ini memperbolehkan seseorang meminta ke FTP *server* untuk membuka koneksi sebagai bagian ketiga dari sebuah bagian *port*.
   c. Penyerangan ini hamper mirip dengan ip spoofing.
   d. Penyerangan tipe ini susah dilacak, mengijinkan akses ke lokal dan menghindari *firewall*.
5. SYN/FIN Scanning menggunakan IP Fragment

a. Hal ini bukan metode baru tapi sebuah modifikasi dari metode awal.
b. Ketika *header* TCP dibagi menjadi beberapa paket maka paket filter tidak dapat mendeteksi yang dilakukan paket tersebut.

I.5 Skema Jaringan Politeknik Batam

Berikut adalah skema jaringan di Politeknik Batam yang menggambarkan susunan *server* yang terhubung dengan *switch* dan *provider* seperti Astinet dan Starcom. Skema jaringan ini diperlukan sebagai gambaran penerapan IDS di Politeknik Batam.



Gambar I.4.2.1 Skema Jaringan Politeknik Batam

Berikut keterangan untuk setiap server pada jaringan Politeknik Batam:

Tabel I.4.2.1 Deskripsi Server Politeknik Batam

Server	**Keterangan**
www.polibatam.ac.id	Berisi *web* polibatam.ac.id
Venus	Berisi mail.polibatam.ac.id
Students atau Neptunus	Berisi invideo.polibatam.ac.id
Bumi	Berisi learning.polibatam.ac.id, akademik.polibatam.ac.id, dan digilib.polibatam.ac.id

Server	**Keterangan**
Bimasakti	Sebagai router dari semua switch yang ada, tempat pengaturan atau setting-an *rules* firewall, nat, proxy, dan untuk melihat kecepatan *bandwidth client*
Wifi	Router wifi Politeknik Batam
Serverpoly	Berisi data akademik/dosen/tps
Servertera	Berisi data film
Matahari	Berisi sdm.polibatam.ac.id, akad.polibatam.ac.id, dan intranet.polibatam.ac.id
Jupiter	Berisi siper.polibatam.ac.id
Serversource	Berisi aplikasi atau *software*

Bab III Pemilihan dan Perancangan Pengujian IDS

Pada bab ini akan dijelaskan mengenai urutan proses yang ditempuh dalam proses pemilihan software IDS, software IDS, software-software yang digunakan dalam proses perbandingan, serta perancangan pengujian yang meliputi serangan yang dibangun, lingkungan pengujian dan kriteria evaluasi.

III.1 Proses Pemilihan

Ada beberapa langkah dalam melakukan proses pemilihan IDS yaitu:

1. Penentuan software berdasarkan kriteria, proses ini dapat membantu admin dalam menentukan IDS yang sesuai dengan kebutuhan keamanan jaringan.
2. Pemilihan software IDS digunakan dalam proses perbandingan dan pengujian.
3. Keunggulan software merupakan kriteria yang dijadikan panduan dalam pemilihan IDS.
4. Implementasi, yaitu melakukan penerapan IDS yang sesuai dengan kebutuhan jaringan di Politeknik Batam.

III.2 Software IDS

Berikut ini software-software IDS (*Intrusion Detection System*):

1. Real Secure Server
2. Symantec Client Security
3. Kane Security Monitor
4. *Snort* dan Base
5. OSSEC
6. Suricata

Deskripsi masing-masing software IDS dijelaskan pada tabel 3.2.1.

Tabel II.4.2.1 Deskripsi Software-Software IDS

NO	NAMA IDS	DESKRIPSI	FITUR	JENIS	Open source/ komersial
1	Real Secure Server	Real Secure merupakan aplikasi intrusi real-time untuk melindungi *server* dari perkembangan spectrum threat ketika menjaga data dan aplikasi.	Otomasi, real time intrusion detection dan prevention dengan analisis event, log OS, daftar kelua-masuk , blocking malicious.	*Software*	Komersial
2	Symantec Client Security	Symantec Client Security adalah produk perusahaan keamanan Symantec dan solusi. Symantec Enterprise Security produk dikombinasikan dengan teknologi kelas dunia dan memiliki layanan penuh dan tim tanggap darurat di seluruh dunia untuk membantu perusahaan beroperasi kepercayaan lebih aman.	Konfigurasi *client Auto-Protect*, konfigurasi *client* dan *server update*, konfigurasi hanya untuk *Client* administrator, membuat general *rules* pada *Symantec Client Firewall* Administrator, membuat *rules* pada *Symantec Client Firewall* Administrator, konfigurasi *quarantine*.	*Software*	komersial
3	Kane Security Monitor	Kane Security Monitor adalah aplikasi yang memantau aktivitas jaringan. Jika terdapat aktivitas yang mencurigakan, Kane Security Monitor akan memberikan	*Custome reporting*, *Password Cracking*, *IP Addressing.*	*Software*	komersial
4	*Snort* dan Base	*Snort* merupakan sebuah aplikasi jaringan yang opensource, yang bekerja untuk melakukan intrusion detection system dan prevention yang dikembangkan oleh Sourcefire. Base adalah aplikasi yang ditulis dengan menggunakan bahasa php, berfungsi menganalisis log intrusi dan menampilkan informasi *database* dalam bentuk web.	Snort : Deteksi threat, new signature, record packet-packet, record traffic, monitor koneksi DSL. Base: Mengenerate graph dan alert berdasarkan sensor, waktu, *rule* dan protocol, menampilkan summary *log* dari semua alert dan link untuk graph	*Software*	Open source
6	OSSEC	OSSEC adalah aplikasi open source yang berbasis Host-based Intrusion Detection System.	Menampilkan analisis *log*, mengecek integritas file, memonitor keamanan jaringan, deteksi rootkit, peringatan real-time, aktif respon.	Software	Open source
7	Suricata	Suricata adalah perangkat lunak pendeteksi dan pencegah intrusi, open source yang merupakan generasi berikutnya dari perangkat-perangkat IDS/IPS yang ada saat ini.	Deteksi threat, new signature, record packet-packet, record traffic, monitor koneksi DSL.	Software	Open Sourece

Berdasarkan table tersebut, software intrusion detection system yang digunakan sebagai bahan perbandingan ada tiga yaitu snort dan base, OSSEC, suricata. Proses pemilihan software dilakukan pada software yang *open source*.

III.2.1 Snort dan Base

Snort merupakan sebuah aplikasi yang berfungsi untuk mendeteksi penyusup dan mampu menganalisa paket yang melintasi jaringan secara langsung dan melakukan pencatatan ke dalam penyimpanan data serta mampu berbagai serangan yang berasal dari luar jaringan.

Snort merupakan software open source yang dikembangkan oleh Marty Roesch. Snort dapat digunakan pada sistem operasi linux, windows, BSD, solaris, dll. Snort dapat bekerja sebagai IDS berbasis jaringan yang menggunakan metode deteksi *rule* based, menganalisa paket data apakah sesuai dengan jenis serangan yang sudah ada di *database*. Snort memanfaatkan perangkat tcpdump untuk mengambil dan menganalisis paket data. Snort dapat berjalan dalam tiga mode antar lain:

1. Paket sniffer

   Dalam mode ini, snort bertindak sebagai software sniffer yang dapat melihat semua paket yang lewat dalam jaringan komputer dimana snort diletakkan. Dalam mode ini, berbagai paket hanya ditampilkan di layar monitor secara *real time*.

2. Paket logger

   Dalam *mode* ini, selain melihat semua paket yang lewat di jaringan komputer, snort dapat mencatat juga melakukan *logging* terhadap berbagai paket tersebut ke *disk*. Dengan kata lain, snort membuat *copy* dari paket-paket yang lewat dan menyimpan *copy* tersebut di *disk* sehingga pengguna snort dapat melakukan analisis terhadap lalu lintas jaringan atau untuk keperluan lainnya.

3. Intrusion Detection Mode

Dalam *mode* yang paling rumit dan fleksibel ini, snort bertindak sebagai NIDS yang dapat mendeteksi dan melakukan *logging* terhadap berbagai macam serangan terhadap jaringan komputer berdasarkan *rule* sistem yang telah

ditetapkan oleh pengguna. *Rule* sistem di snort akan mendeteksi serangan-serangan tersebut berdasarkan ciri-ciri khas (*signature*) dari serangan tersebut.
Komponen-komponen snort meliputi:

1. *Rule* snort merupakan *database* yang berisi pola-pola serangan berupa *signature* jenis-jenis serangan. Diperlukan *update rule* snort secara rutin, agar ketika ada suatu teknik serangan yang baru, serangan tersebut dapat terdeteksi.
2. *Snort engine* merupakan program yang berjalan sebagai daemon proses yang selalu bekerja untuk membaca paket data dan kemudian membandingkannya dengan *rule* snort.
3. *Alert* merupakan catatan serangan pada IDS. Jika paket data yang lewat merupakan serangan, maka *snort engine* akan mengirimkan *alert* berupa *log* file.

Fitur-fitur snort antara lain:

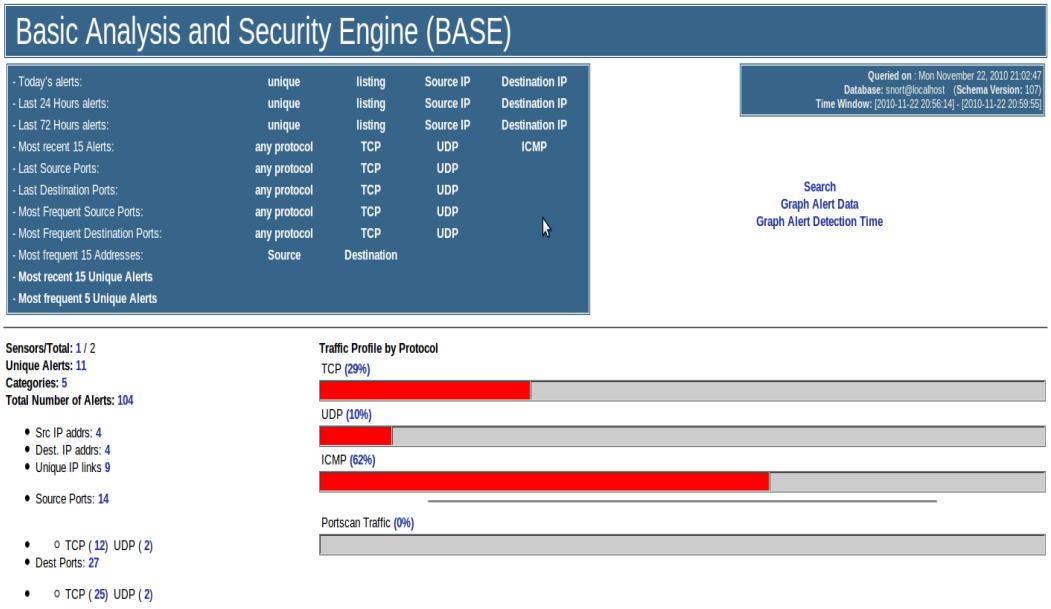
1. Snort memiliki *rule*. *Rule* sistem ini digunakan sebagai signature terhadap serangan.
2. Snort memiliki sebuah *database* untuk berbagai macam *rules*, dan *database* ini secara aktif dikembangkan oleh komunitas snort sehingga tipe-tipe serangan yang baru dapat dikenali dan dicatat.
3. Snort dapat melakukan *logging* langsung ke *database*, misalnya ke MySQL, PostGRE SOL, dan MS SQL.

Base adalah aplikasi yang ditulis dengan menggunakan bahasa php, berfungsi menganalisis *log* intrusi dan menampilkan informasi *database* dalam bentuk web. Dalam analisis ini snort diintegrasikan dengan base dan mysql agar tampilan yang dihasilkan berupa tampilan web dan semua pencatatan yang dilakukan disimpan dalam *database*.
Fitur-fitur base sebagai berikut:

1. Search
    - Meta criteria

        - Sensor.
        - Alert group.
        - Signature.
        - Classification.
        - Priority.
        - Alert time.
    - IP criteria.
    - Payload criteria.
2. Graph alert data.
3. Graph alert detection time.
4. Alert group maintenance
    - List all.
    - Create.
    - View.
    - Edit.
    - Delete.
    - Clear.
5. Cache and status
    - Php build.
    - *Database*.
    - Alert information cache.
    - IP address cache.
6. Administration
    - User management.
    - List users.
    - Create a user.
    - Role management.
    - List roles.
    - Create a role.
7. Summary statistic

- Sensor.
- Unique alerts (classification).
- Unique addresses (source dan destination).
- Unique ip links
- Source *port* (tcp dan udp)
- Destination *port* (tcp dan udp).
- Time profile of alerts.



Gambar III.2.1.1. Snort dan Base

Spesifikasi komputer yang dibutuhkan untuk membangun snort dan base antara lain:

1. Komputer minimal Pentium III.
2. Memori 512 MB.
3. Harddisk 40 GB (bisa lebih dari itu).

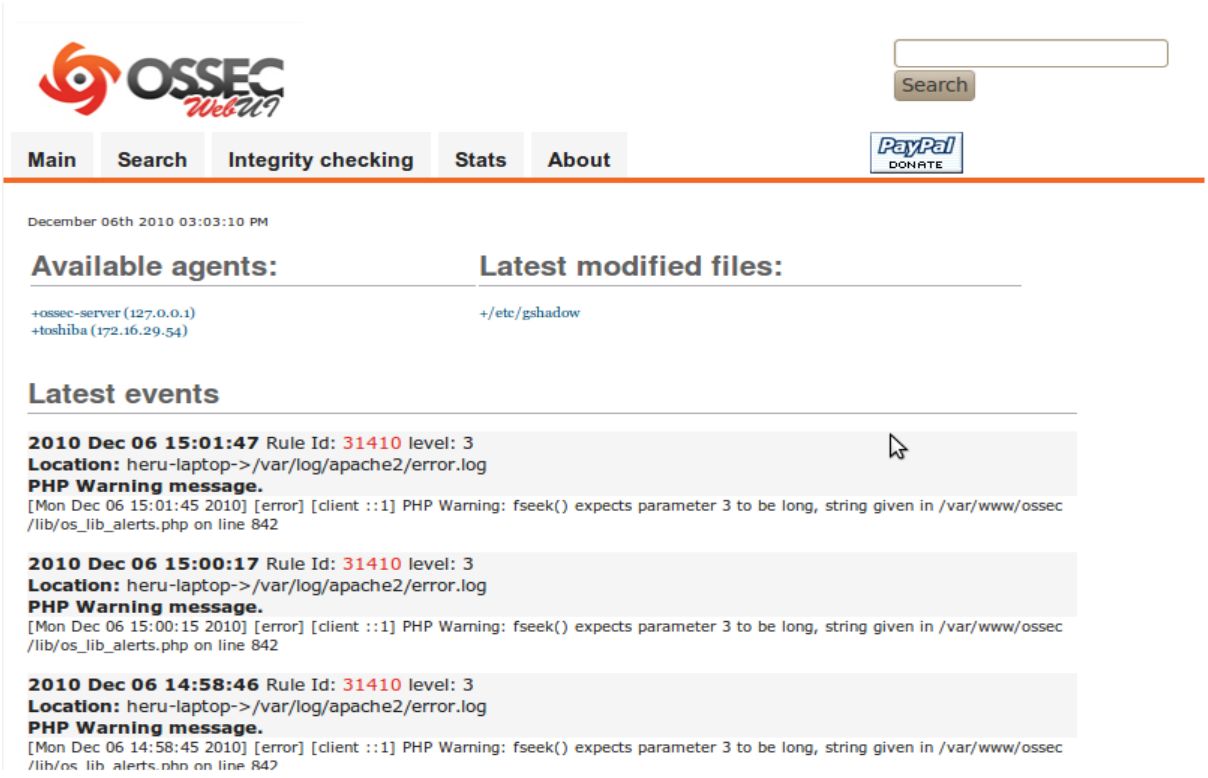
III.2.2 OSSEC

OSSEC merupakan aplikasi HIDS yang bersifat *open source*. Aplikasi ini dapat melakukan analisis *log*, memeriksa integritas *file*, pemantauan lalu lintas jaringan, deteksi *rootkit*.

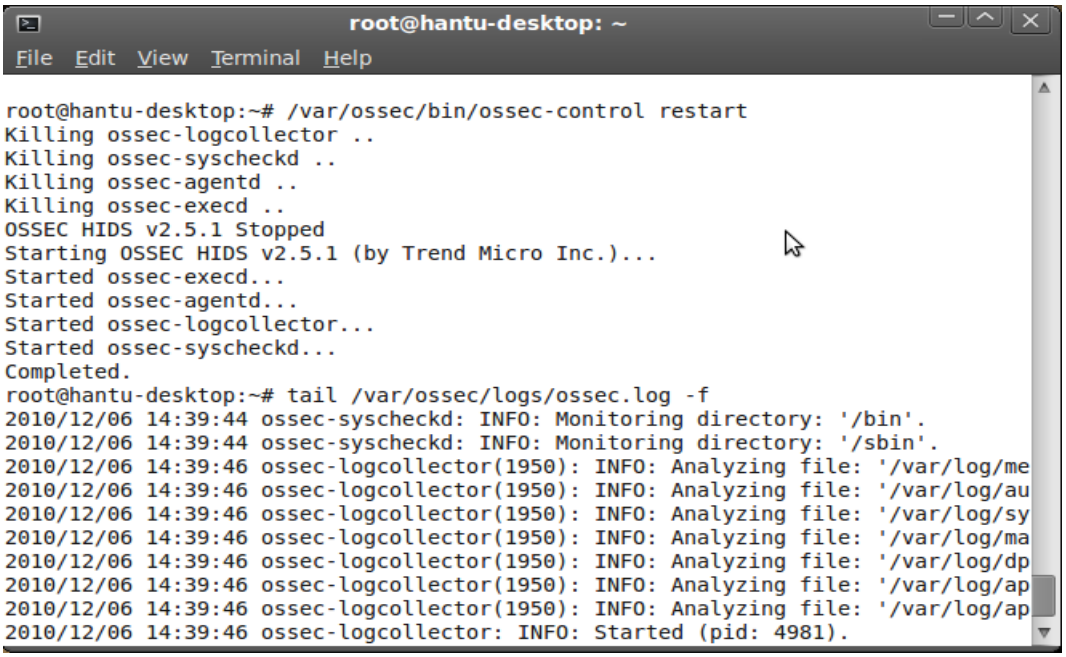
OSSEc memiliki beberapa bagian yaitu *server*, *agent*. *Server* bertugas untuk memantau seluruh jaringan yang terhubung ke server dan menerima informasi dari *agent*.

Server merupakan pusat OSSEC. *Server* mempunyai *database* yang dapat digunakan untuk memeriksa integritas *file*, *log*, aktivitas jaringan. Selain itu, fitur lain yang dimiliki oleh *server* adalah *rule* (*signature*) dan pilihan konfigurasi, sehingga mempermudah dalam memantau sejumlah *agent*.

Agent merupakan kumpulan program yang diinstall pada sistem yang dipantau. *Agent* akan mengumpulkan informasi mengenai aktivitas jaringan yang terjadi pada *agent* tersebut secara *real time* dan mengirimkan informasi tersebut ke *server* untuk dianalisis.



Gambar III.2.2.1. OSSEC *Server*



Gambar III.2.2.2. OSSEC Agent

Berikut fitur-fitur OSSEC *server*:

1. Menu search yang berfungsi untuk mencari aktivitas jaringan berdasarkan tanggal dan waktu.
2. Integrity Checking
3. Stats
4. About
5. Menampilkan agent-agent yang terhubung dengan ossec *server* yang sedang aktif.

Pada OSSEC agent tampilan berbasis *console*. Agar ossec *agent* terhubung dan telah terdaftar pada *server*, maka ossec *agent* harus menjalankan ossec dengan perintah /var/ossec/bin/ossec-control start.

Spesifikasi komputer yang dibutuhkan untuk membangun OSSEC *server* antara lain:

1. Komputer minimal Pentium III, memiliki 2 NIC.
2. Memori minimal 512 MB.
3. Harddisk 40 GB (bisa lebih dari itu).

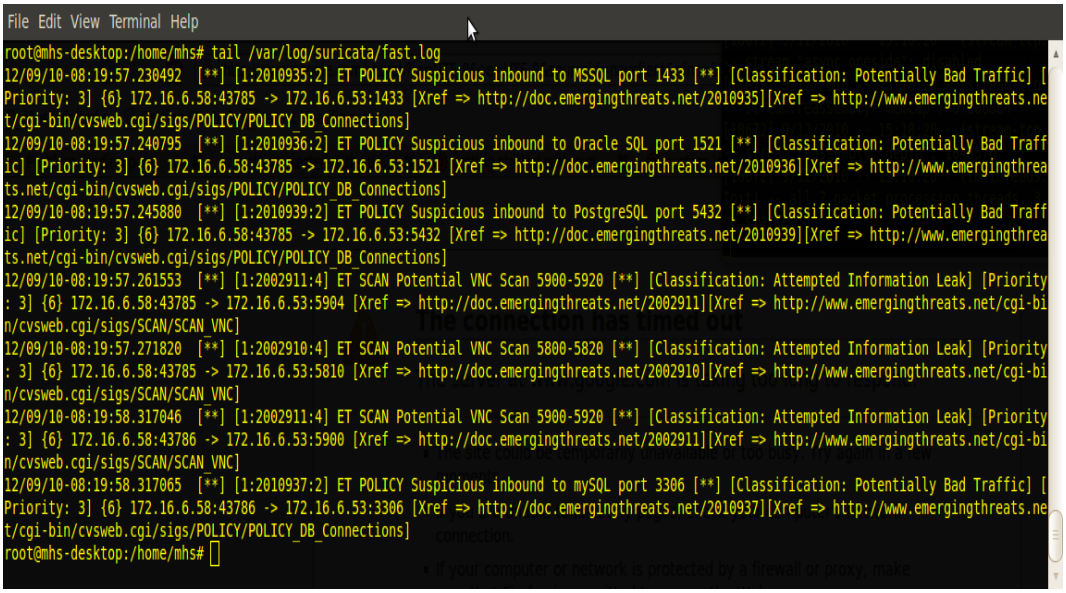
Spesifikasi komputer yang dibutuhkan untuk membangun OSSEC agent antara lain:

1. Komputer minimal Pentium III, memiliki 2 NIC.
2. Memori minimal 512 MB.
3. Harddisk 40 GB (bisa lebih dari itu).

III.2.3 Suricata

Suricata merupakan perangkat lunak pendeteksi dan pencegah intrusi (*Intrusion Detection and Prevention System*) bersifat *open source* yang merupakan generasi berikutnya dari perangkat-perangkat IDS/IPS yang ada saat ini. Suricata dirilis oleh OISF.

Suricata dapat menggunakan *rule-rule* yang biasa digunakan oleh perangkat lunak snort.



Gambar III.2.3.1 Suricata

Report yang ditampilkan pada suricata berbasis *console*.

Spesifikasi komputer yang dibutuhkan untuk membangun suricata antara lain:

1. Komputer minimal Pentium III.
2. Memori 512 MB.
3. Harddisk 40 GB (bias lebih dari itu).

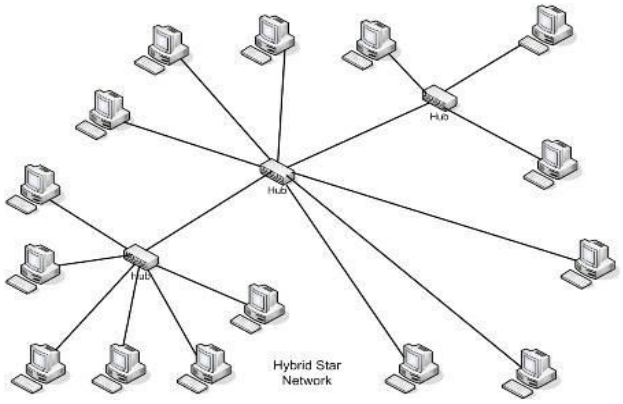
III.3 Perancangan Pengujian

Perancangan pengujian dibagi dalam tiga bagian yaitu skema jaringan, lingkungan pengujian, kriteria evaluasi, dan jenis serangan yang akan dibangun sebagai pengujian terhadap IDS.

III.3.1 Skema Jaringan

Intrusion Detection System dapat diterapkan pada berbagai jenis topologi. Salah satu topologi yang dapat digunakan adalah topologi *hybrid star*. Topologi *hybrid* star merupakan penggabungan beberapa topologi bintang (star). Kelebihan dari topologi ini sebagai berikut:

1. Tidak langsung terhubung satu sama lain tetapi melalui perangkat pusat pengendali yang disebut hub.
2. Kabel yang dibutuhkan hanya sebanyak komputer dalam jaringan dan I/O cukup hanya satu di setiap komputer, kabel link dan *port* I/O lebih sedikit dan biaya lebih sedikit.
3. Memiliki sifat robustness yaitu jika satu link rusak maka kerusakan hanya terjadi pada komputer yang berada pada link tersebut.



Gambar III.3.1.1. Topologi *Hybrid*

III.3.2 Lingkungan Pengujian

Di bawah ini merupakan spesifikasi lingkungan pengujian *intrusion detection system*.

Tabel III.3.2.1 Spesifikasi PC Server Snort dan Base

No	Spesifikasi Hardware	
1	Prosesor	Intel (R) Pentium (R) 4 CPU 2.60 GHz
2	Memory	512 MB RAM
3	Harddisk	40 GB
3	Program Utilities	Web browser (Mozilla Firefox)
4	Sistem Operasi	Linux

Tabel III.3.2.2 Spesifikasi PC Server OSSEC

No	Spesifikasi Hardware	
1	Prosesor	Intel (R) Core 2 Duo CPU 2.00 GHz
2	Memory	2 GB RAM
3	Harddisk	200 GB
3	Program Utilities	Web browser (Mozilla Firefox)
4	Sistem Operasi	Linux

Tabel III.3.2.3 Spesifikasi PC OSSEC Agent

No	Spesifikasi Hardware	
1	Prosesor	Intel (R) Pentium (R) 4 CPU 2.60 GHz
2	Memory	512 MB RAM
3	Harddisk	40 GB
4	Sistem Operasi	Linux

Tabel III.3.2.4 Spesifikasi PC Suricata

No	Spesifikasi Hardware	
1	Prosesor	Intel (R) Pentium (R) 4 CPU 2.60 GHz
2	Memory	512 MB RAM
3	Harddisk	40 GB
3	Program Utilities	Web browser (Mozilla Firefox)
4	Sistem Operasi	Linux

Tabel III.3.2.5 Komponen Pendukung

No	Nama	Deskripsi
1	Kabel UTP RJ-45 tipe straight	Untuk menghubungkan perangkat yang berbeda.

III.3.3 Kriteria Evaluasi

Dalam melakukan pemilihan software IDS dan IPS diperlukan kriteria. Kriteria yang digunakan dilihat dari pengukuran hal-hal tertentu, walaupun kriteria yang ditetapkan berbeda antara satu pihak dan yang lainnya. Penetapan kriteria menurut Ranum (*Experiences Benchmarking Intrusion Detection System*, 2001)[7] yaitu:

1. Kemampuan mendeteksi serangan

   Tingkat kemampuan IDS untuk mengenali kerentanan serangan, mendeteksi serangan yang tidak dikenali, dan adanya relevansi serangan yaitu memberikan peringatan yang berbeda terhadap serangan yang berbeda.

2. Stability, reliability and security

   Kemampuan IDS beroperasi secara konsisten, melaporkan aktivitas jaringan secara *realtime* dan adanya *database* yang dapat menyimpan semua data aktivitas jaringan.

3. Penyediaan informasi

Menyediakan informasi yang jelas dan akurat mengenai peringatan (signature). Dapat mengidentifikasi sasaran dan sumber serangan.

Dapat mengidentifikasi potensi kerusakan serangan.

4. Scalability

   Adanya infrastruktur dukungan.

5. Vendor support

   - Signature update

     Kemampuan untuk melakukan update rule yang berfungsi untuk menambah atau memperbaharui *rule intrusion detetction system*.

III.3.4 Jenis Serangan

Berikut dijelaskan mengenai jenis serangan yang digunakan sebagai bahan pengujian *intrusion detection system*.

III.3.4.1 Port Scanning

Port Scanning merupakan *scanning* yang bertujuan untuk menemukan *port-port* yang terbuka dari suatu *host*. Tools yang digunakan dalam pengujian *port scanning* adalah nmap.

Nmap merupakan *utility* yang digunakan untuk melakukan *scanning*. Nmap adalah sebuah utility yang melakukan eksplorasi dan audit terhadap sistem keamanan computer. Nmap gratis dan tersedia dalam bentuk *open source code*.

Cara kerja nmap dengan menggunakan paket-paket IP untuk menentukan *host* atau komputer mana saja yang statusnya aktif dalam jaringan tersebut, layanan apa saja yang dijalankan host tersebut, sistem operasi apa saja yang digunakan oleh *host* tersebut, dan jenis *firewall* apa saja yang digunakan *host-host* tersebut di dalam jaringan yang di*scan* oleh nmap pada saat itu.

---

[7] http://www.sans.org/security-resources/idfaq/eval_ids.php

Nmap memiliki tiga fungsi utama yaitu: deteksi sistem operasi, *scan* terhadap *port*, *ping scan*. Secara *default*, nmap melakukan sebuah proses *ping scan* untuk ping ke setiap host untuk memastikan *host* tersebut aktif atau tidak.

III.3.4.2 Ping Flood

Ping flood merupakan salah satu DDoS yang digunakan untuk melakukan serangan secara langsung. Membutuhkan akses root untuk melakukan ini pada sistem Linux. Implementasinya sederhana saja, yaitu dengan mengirimkan paket data secara besar-besaran.

ping -fs 65000 [ip_target]

III.3.4.3 DDoS Attack

DDoS attack merupakan jenis serangan terhadap sebuah komputer atau *server* dengan cara menghabiskan *resources* (sumber) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya. *Tools* yang digunakan dalam pengujian ddos attack adalah ettercap.

Ettercap merupakan sebuah program yang ditujukan sebagai sebuah sniffer. Ettercap dapat digunakan untuk melakukan serangan *denial of service* terhadap *server*.

Bab IV Implementasi dan Pengujian

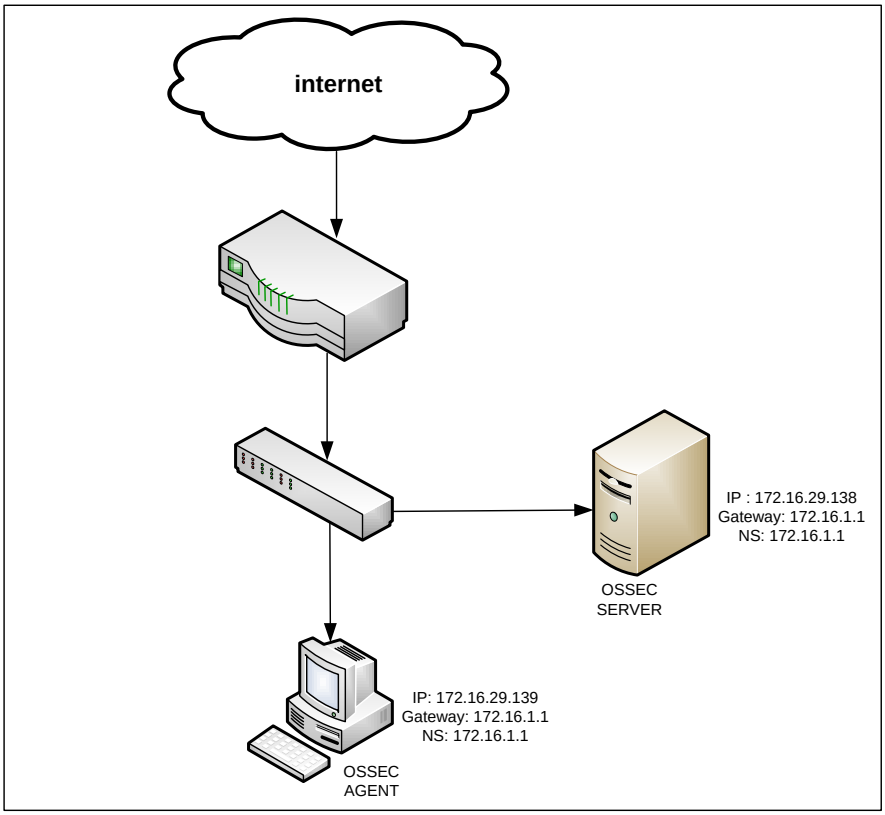
Pada bab ini akan dijelaskan mengenai proses implementasi (OSSEC, suricata, snort dan base), pengujian (OSSEC, suricata, snort dan base) terhadap serangan dengan menggunakan *tools* nmap, nikto, ping flood, dan ettercap.

IV.1 Implementasi

Berikut dijabarkan mengenai implementasi yang dilakukan pada snort dan base, OSSEC, dan suricata.

IV.1.1 Implementasi OSSEC

Di bawah ini adalah topologi jaringan OSSEC pada saat melakukan implementasi OSSEC.



Gambar IV.1.1.1 Topologi Jaringan OSSEC

OSSEC mempunyai dua bagian yaitu bagian *server* dan agent. Pada tahapan awal untuk memonitor aktivitas *server* dengan memonitor log yaitu dengan menginstall

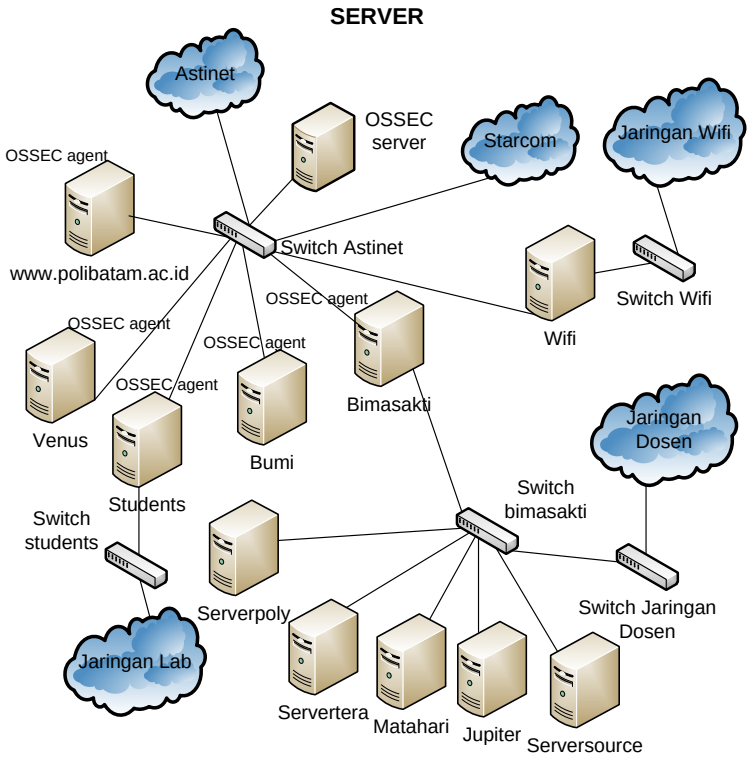
OSSEC HIDS. Versi OSSEC yang digunakan baik *server* maupun agent adalah 2.5.1 .Proses konfigurasi OSSEC akan dijabarkan pada lampiran.

Evaluasi yang dilakukan terhadap OSSEC sesuai dengan kriteria yang telah ditetapkan dijelaskan pada table berikut:

Tabel IV.1.1.1 Evaluasi OSSEC

NO	Kriteria	Keterangan
1	Scalability	Dukungan komersial, pengguna milis, pengembangan milis, Bugzilla.
Vendor Support		
2	Signature Update	Ada.Dapat dilakukan secara otomatis dan manual.

Penerapan OSSEC di jaringan Politeknik Batam dapat dilakukan seperti gambar berikut:

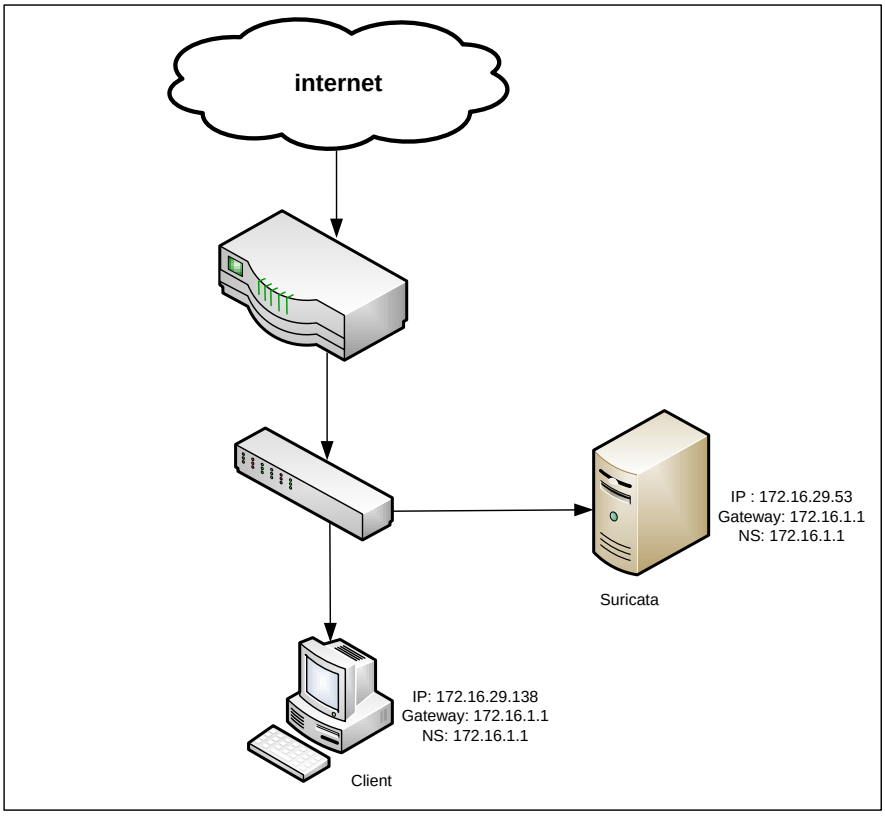


Gambar IV.1.1.2 Gambaran Penerapan OSSEC di Jaringan Politeknik Batam

Sebagai gambaran penerapannya, pada *switch* Astinet ditambah satu *server* yang berisi OSSEC *server* yang bertindak sebagai *server*. Pada *server* www.polibatam.ac.id, venus, students, bumi, dan bimasakti diinstall ossec *agent*. *Server* www.polibatam.ac.id, venus, students, bumi, dan bimasakti bertindak sebagai agent. Server yang bertindak sebagai agent didaftarkan pada server yang bertindak sebagai server. Sehingga semua aktivitas yang terjadi pada *agent* akan dikirim ke OSSEC *server*.

IV.1.2 Implementasi Suricata

Di bawah ini adalah topologi jaringan suricata pada saat melakukan implementasi suricata.



Gambar IV.1.2.1 Topologi Jaringan Suricata

Sebelum melakukan instalasi, ada beberapa paket yang harus sudah terinstal pada sistem linux. Paket tersebut berfungsi sebagai pendukung suricata dalam sistem linux. Paket-paket tersebut sebagai berikut:

- build-essential
- checkinstall

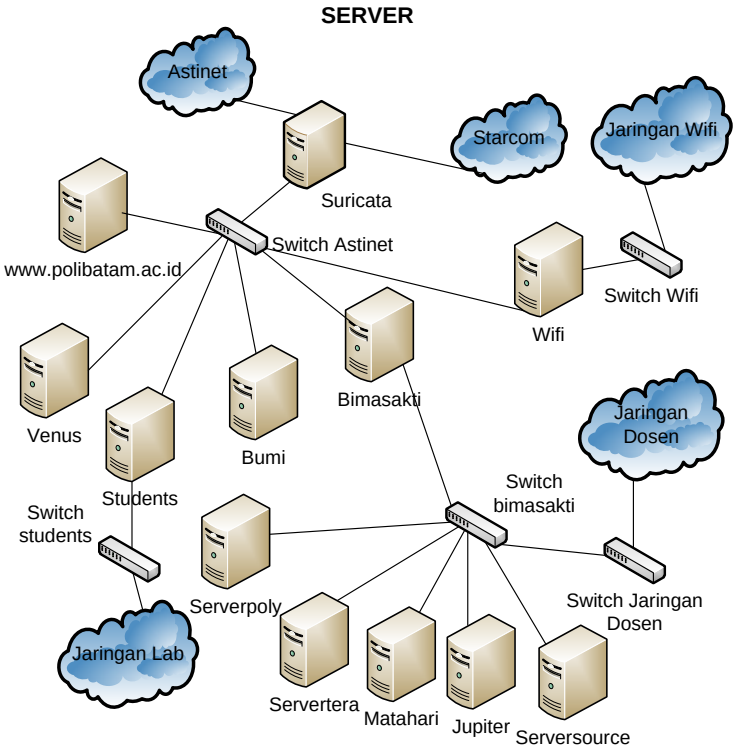
- libpcre3-dev
- libpcap-dev
- libyaml-dev
- zliblg-dev
- libnet1libnet1-dev
- libcap-ng-dev
- libhtp1
- libnetfilter-queue-dev
- libnetfilter-queue1
- libnfnetlink-dev
- libnfnetlink0

Proses konfigurasi suricata akan dijabarkan pada lampiran.

Tabel IV.1.2.1 Evaluasi Suricata

NO	Kriteria	Keterangan
1	Scalability	Mailing list
Vendor Support		
2	Signature Update	Ada.Dilakukan secara manual.

Penerapan suricata di jaringan Politeknik Batam dapat dilakukan seperti gambar berikut:

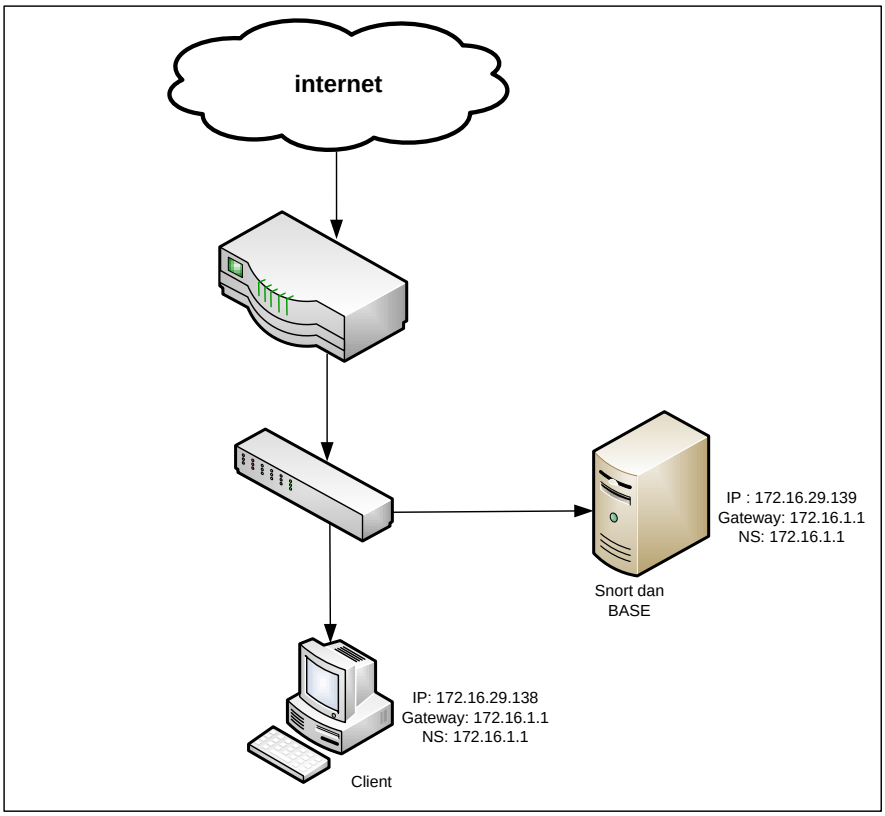


Gambar IV.1.2.2 Gambaran Penerapan Suricata di Jaringan Politeknik Batam

Sebagai gambaran penerapannya, *server* suricata ditambahkan sebelum *switch* Astinet. Setiap paket yang masuk ke *switch* Astinet terlebih dahulu harus melewati *server* suricata, sehingga semua aktivitas yang terjadi dapat dianalisis oleh suricata.

IV.1.3 Implementasi Snort dan BASE

Di bawah ini adalah topologi jaringan snort dan base pada saat melakukan implementasi snort dan base.



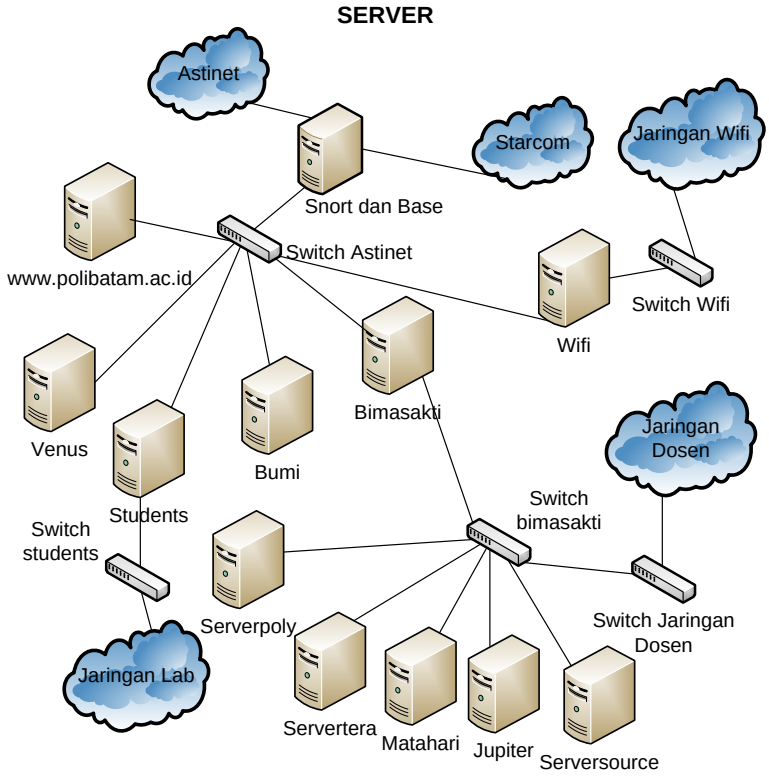
Gambar IV.1.3.1. Topologi Jaringan Snort dan BASE

Dalam analisis ini base diintegrasikan dengan *snort*. Proses konfigurasi base akan dijabarkan pada lampiran.

Tabel IV.1.3.1 Evaluasi Snort dan Base

NO	Kriteria	Keterangan
1	Scalability	Forum,mailing list, laporan bug.
Vendor Support		
2	Signature Update	Ada.Dilakukan secara manual.

Penerapan snort dan base di jaringan Politeknik Batam dapat dilakukan seperti gambar berikut:



Gambar IV.1.3.2 Gambaran Penerapan Snort Dan Base di Jaringan Politeknik Batam

Sebagai gambaran penerapannya, sama seperti halnya penerapan suricata di Politeknik Batam, server snort dan base ditambahkan sebelum *switch* Astinet. Setiap paket yang masuk ke switch Astinet terlebih dahulu harus melewati server snort dan base, sehingga semua aktivitas yang terjadi dapat dianalisis oleh snort dan base.

IV.2 Pengujian

Berikut dijabarkan pengujian yang dilakukan pada snort *report*, OSSEC, suricata, dan base dengan menggunakan serangan port scanning, vulnerability scanning, ddos attack.

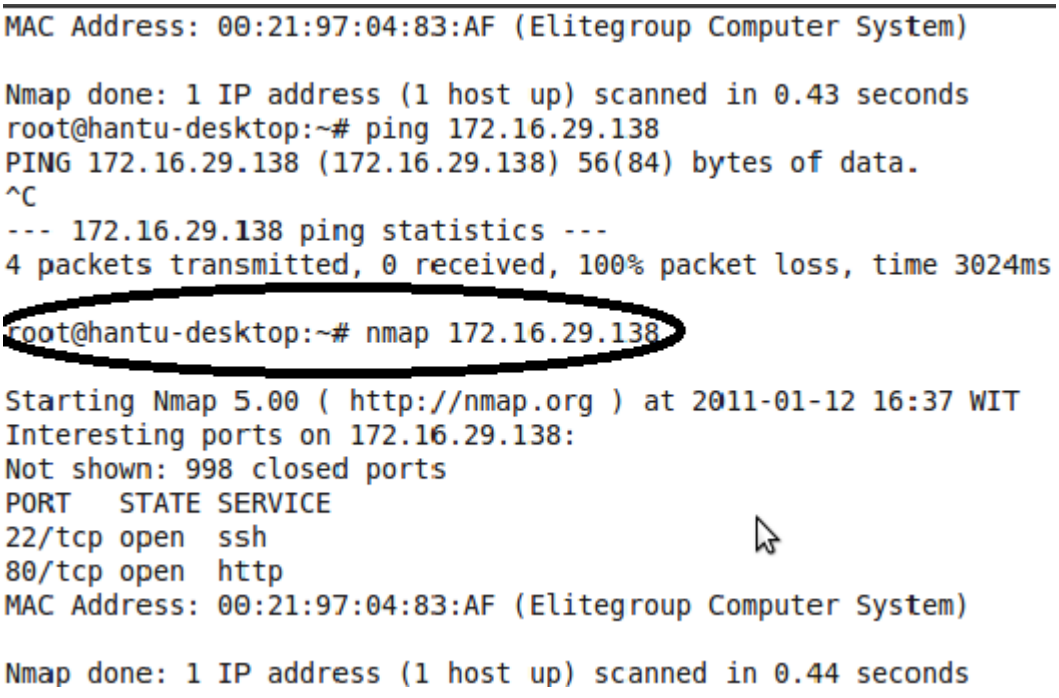
IV.2.1 Pengujian Port Scanning

Utiliti yang dibutuhkan adalah nmap, sehingga paket yang harus sudah diinstall adalah nmap. Tahapan yang dilakukan sebagai berikut:

1. Menginstall paket nmap dengan menggunakan perintah
   apt-get install nmap.

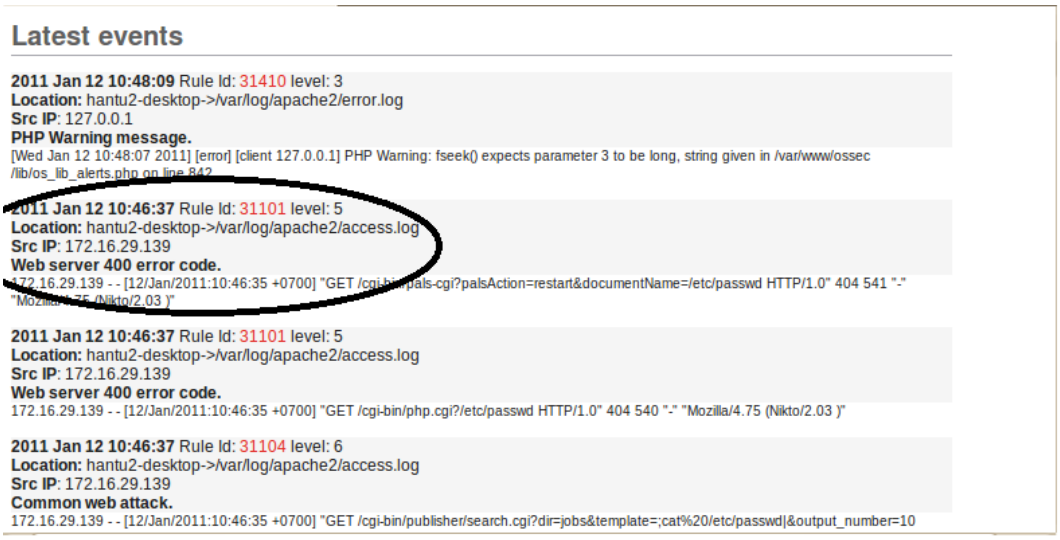
Berikut dijelaskan mengenai pengujian nmap yang dilakukan terhadap snort dan base, OSSEC, dan suricata.

IV.2.1.1 OSSEC



Gambar IV.2.1.1 Pengujian Port Scanning Pada OSSEC

Pengujian dilakukan pada OSSEC agent dengan IP 172.16.29.139, menyerang OSSEC *server* dengan IP 172.16.29.138.

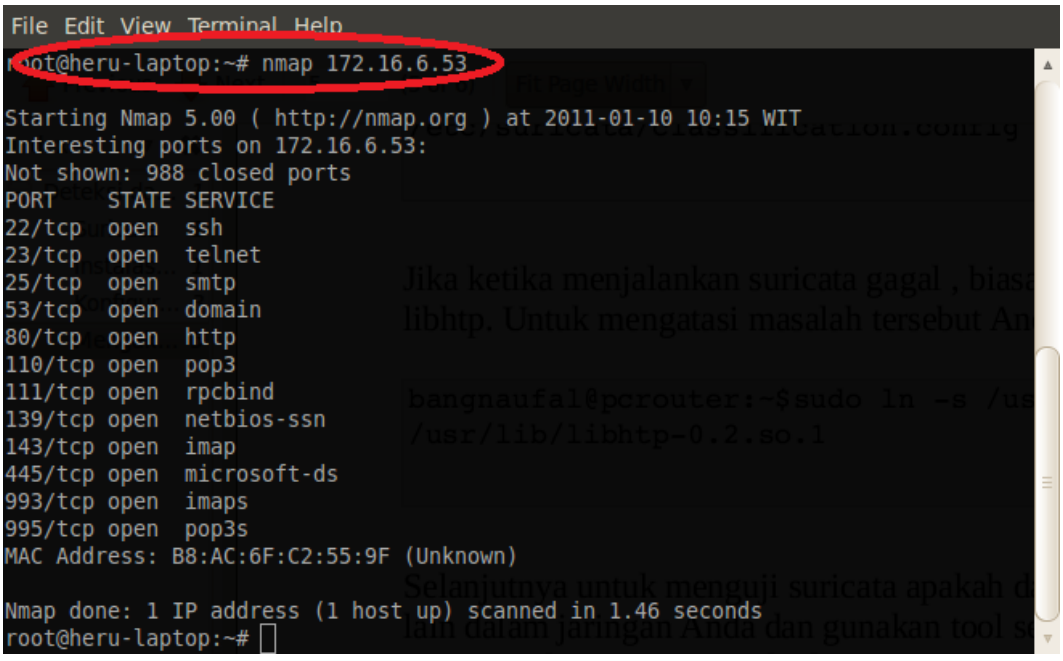


Gambar IV.2.1.2 Pendeteksian Port Scanning Pada OSSEC

Pengujian *port* scanning dengan menggunakan nmap berhasil dideteksi oleh OSSEC *server*.

IV.2.1.2 Suricata

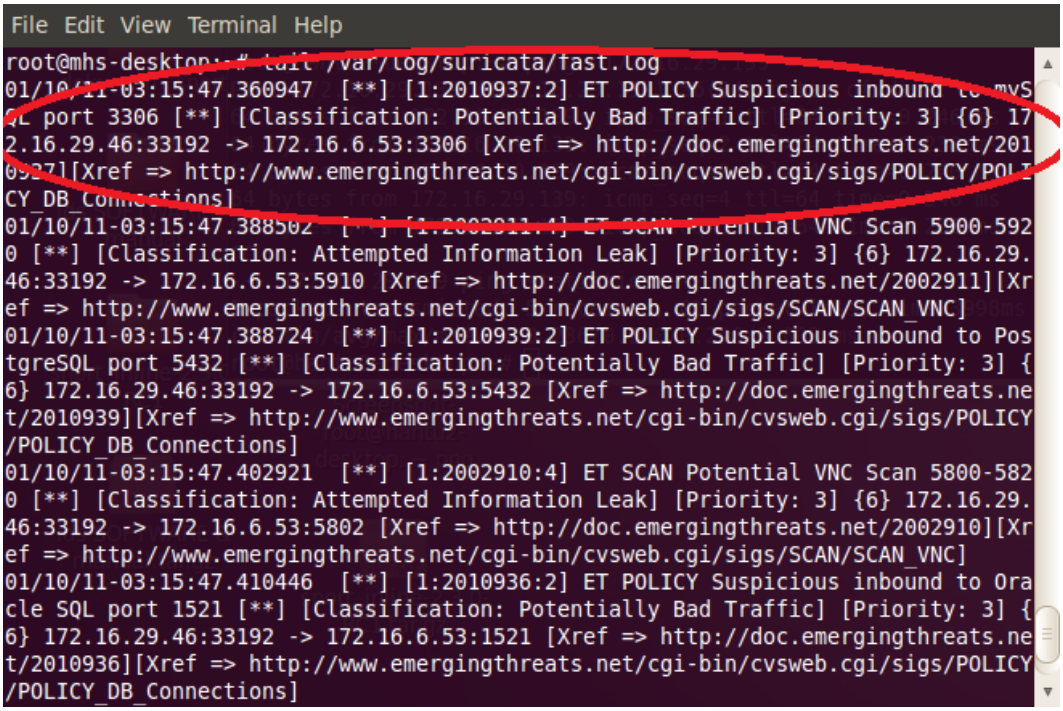
Pengujian *port* scanning menggunakan nmap dilakukan untuk mengetahui apakah aktivitas *port* scanning dapat dideteksi oleh software suricata yang bertindak sebagai IDS.



Gambar IV.2.1.3 Pengujian Port Scanning Untuk Suricata

Pada gambar diatas (tampilan pada komputer yang melakukan *port* scanning) perintah nmap 172.16.6.53 ditujukan pada komputer suricata yang memiliki ip 172.16.6.53.

Hasil dari *port* scanning tersebut adalah diketahui *port* dan service yang sedang aktif pada komputer suricata (172.16.6.53).



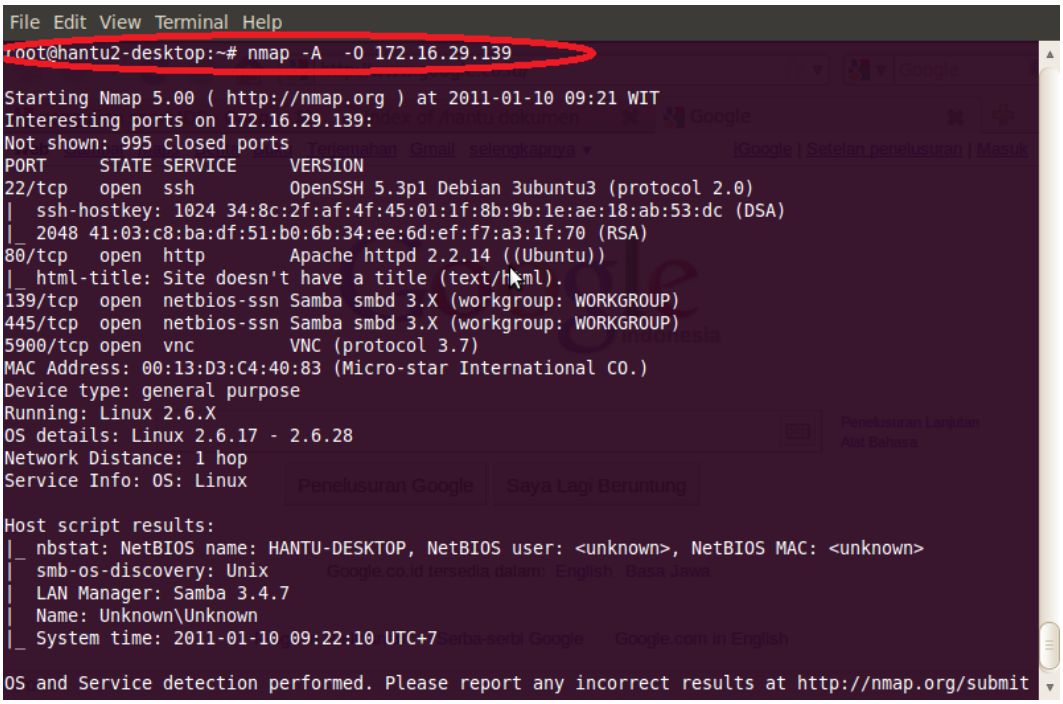
Gambar IV.2.1.4 Pendeteksian Port Scanning Pada Suricata

Gambar di atas merupakan tampilan pada komputer yang menjadi target *port* scanning (ip 172.16.6.53) dan berhasil mendeteksi aktivitas *port* scanning yang dilakukan oleh komputer penyusup dengan ip 172.16.29.46.

IV.2.1.3 Snort dan Base

Tahapan pengujian nmap yang dilakukan pada base (*basic analysis and secure engine*) sebagai berikut:

1. Setting ip (komputer yang tidak memiliki IDS) 172.16.29.138/16.



Gambar IV.2.1.5 Pengujian Port Scanning Pada Snort dan Base

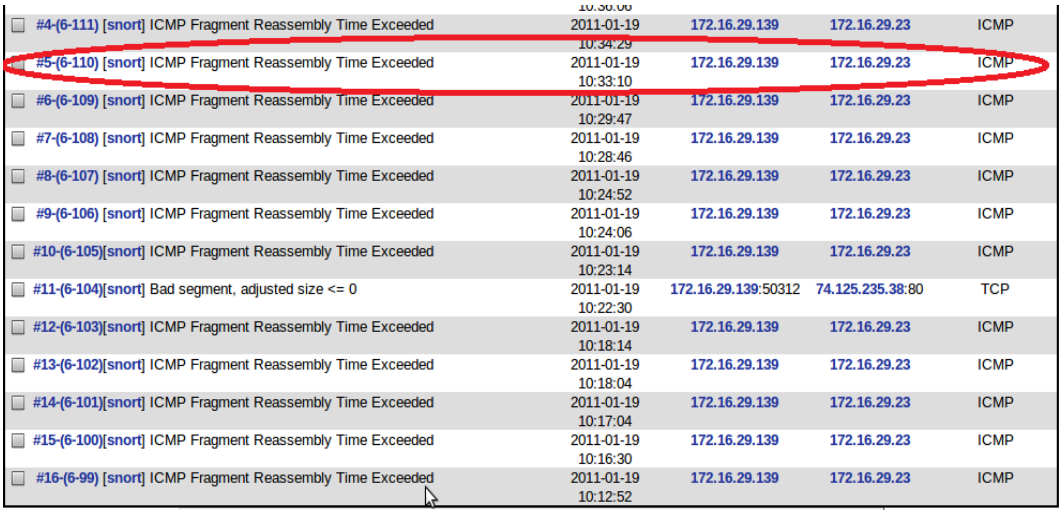
Gambar di atas merupakan tampilan komputer yang melakukan *port* scanning (IP 172.16.29.138). Perintah nmap –A –O 172.16.29.139 ditujukan pada komputer yang memiliki snort dengan ip 172.16.29.139.

Pengujian nmap pada snort yang diintegrasikan dengan base tidak berhasil dideteksi.

IV.2.2 Pengujian Ping Flood

Berikut dijelaskan mengenai pengujian ddos attack dengan menggunakan ping flood dan ettercap yang dilakukan terhadap snort dan base, OSSEC, dan suricata.

IV.2.2.1 Snort dan Base



#4-(6-111) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:34:29	172.16.29.139	172.16.29.23	ICMP
#5-(6-110) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:33:10	172.16.29.139	172.16.29.23	ICMP
#6-(6-109) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:29:47	172.16.29.139	172.16.29.23	ICMP
#7-(6-108) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:28:46	172.16.29.139	172.16.29.23	ICMP
#8-(6-107) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:24:52	172.16.29.139	172.16.29.23	ICMP
#9-(6-106) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:24:06	172.16.29.139	172.16.29.23	ICMP
#10-(6-105)[snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:23:14	172.16.29.139	172.16.29.23	ICMP
#11-(6-104)[snort] Bad segment, adjusted size <= 0	2011-01-19 10:22:30	172.16.29.139:50312	74.125.235.38:80	TCP
#12-(6-103)[snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:18:14	172.16.29.139	172.16.29.23	ICMP
#13-(6-102)[snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:18:04	172.16.29.139	172.16.29.23	ICMP
#14-(6-101)[snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:17:04	172.16.29.139	172.16.29.23	ICMP
#15-(6-100)[snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:16:30	172.16.29.139	172.16.29.23	ICMP
#16-(6-99) [snort] ICMP Fragment Reassembly Time Exceeded	2011-01-19 10:12:52	172.16.29.139	172.16.29.23	ICMP

Gambar IV.2.2.1 Pendeteksian Ping Flood Pada Snort dan Base

Snort dan base berhasil mendeteksi serangan ddos dengan menggunakan ping flood. Komputer penyerang memiliki IP 172.16.29.23 menyerang komputer dengan IP 172.16.29.139.

IV.2.2.2 OSSEC

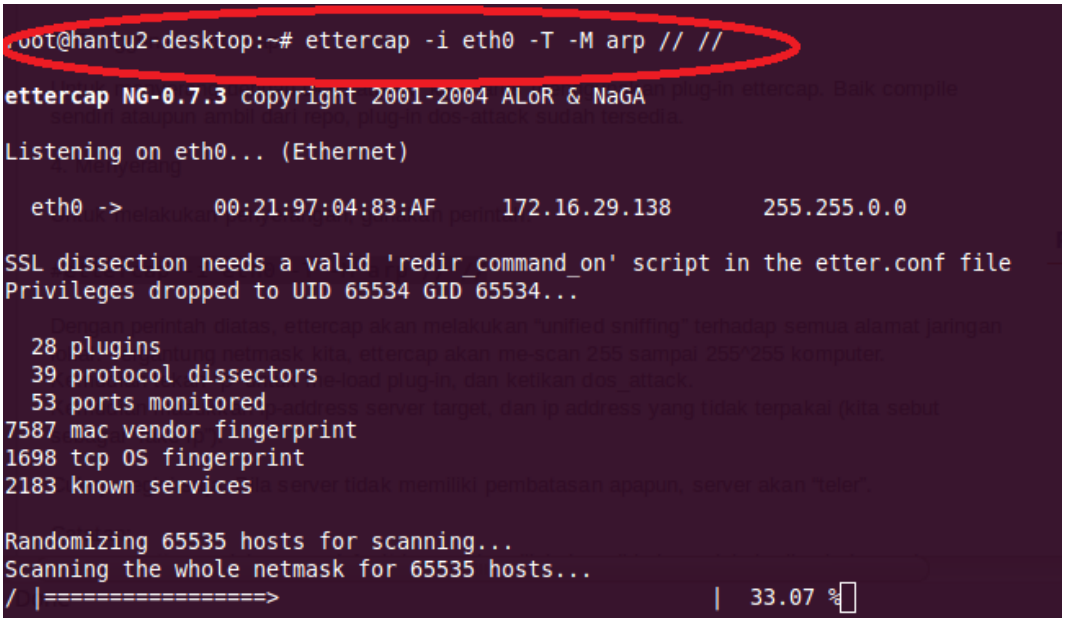
Ossec tidak berhasil mendeteksi serangan ddos dengan menggunakan ping flood.

IV.2.2.3 Suricata

Suricata tidak berhasil mendeteksi serangan ddos dengan menggunakan ping flood.

IV.2.3 Pengujian DDoS Attack

Berikut tahapan penggunaan ettercap:

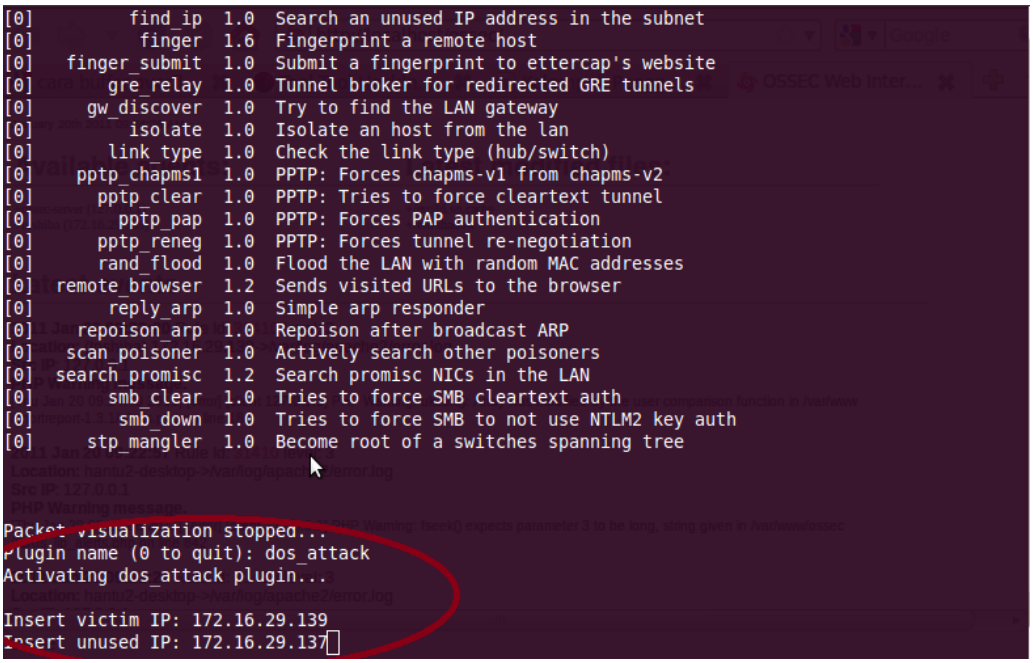


Gambar IV.2.3.1 Penggunaan DDoS Attack

Dengan menggunakan perintah di atas, ettercap akan melakukan *unified sniffing* terhadap semua alamat jaringan lokal.

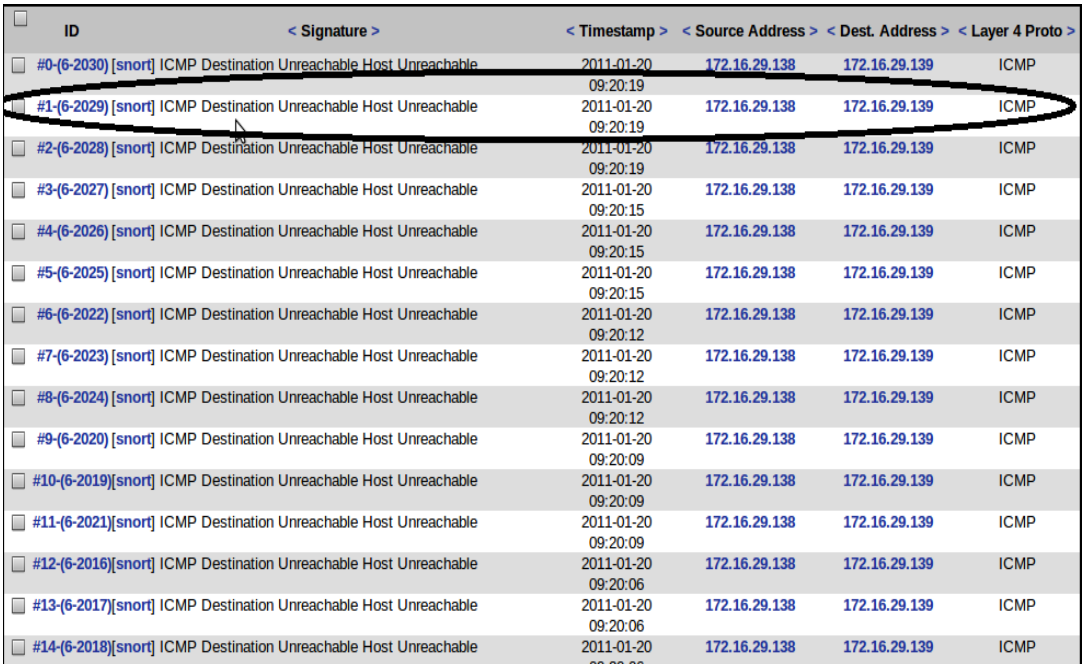
Kemudian ketik p untuk menampilkan *plug-in*. Lalu ketikkan dos_attack.

Kemudian masukkan *ip address server* target da nip yang tidak terpakai (disebut sebagai *fake-ip*), seperti gambar berikut:



Gambar IV.2.3.2 Penyerangan Melalui DDoS Attack Terhadap IDS

IV.2.3.1 Snort dan Base

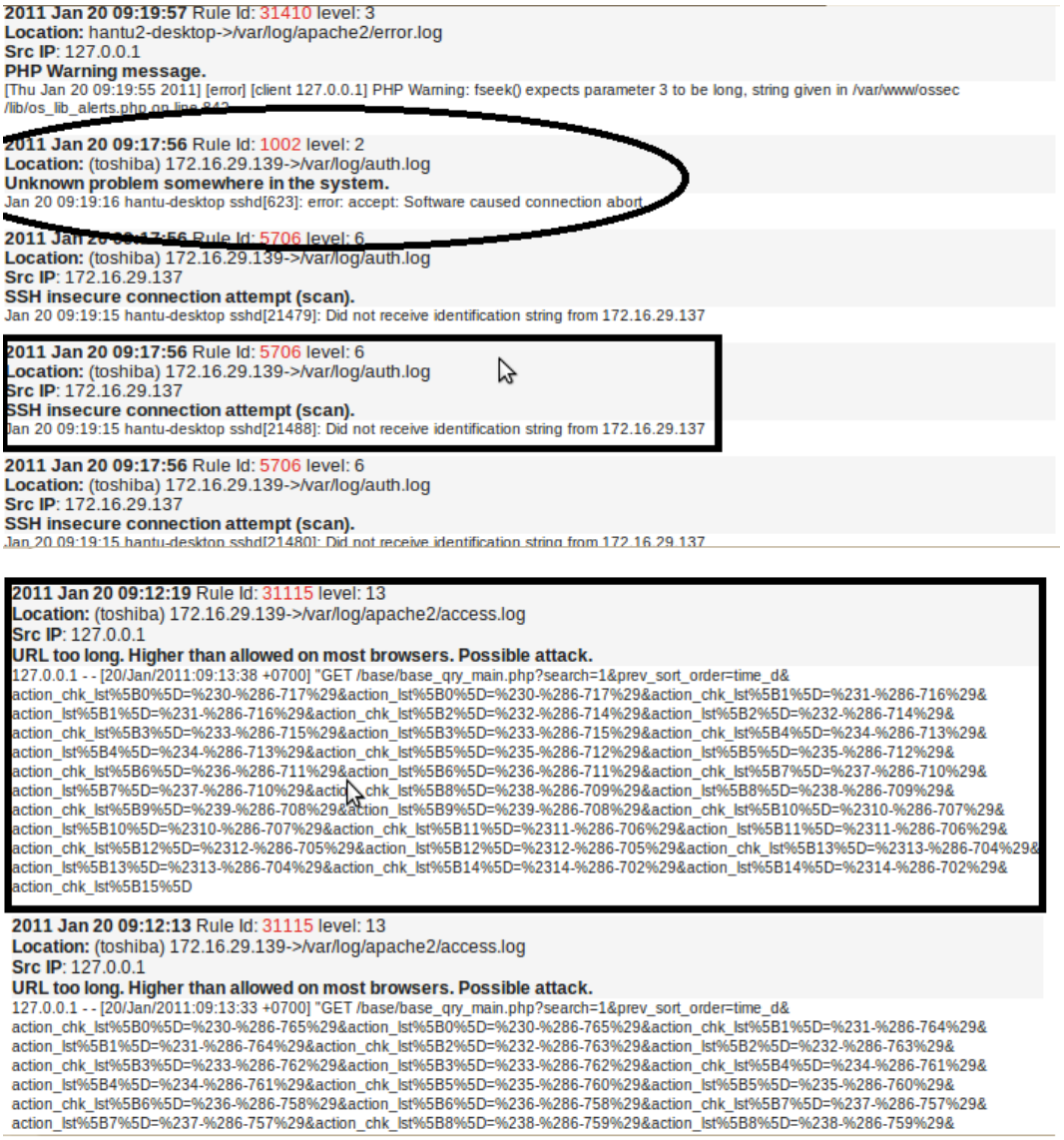


ID	< Signature >	< Timestamp >	< Source Address >	< Dest. Address >	< Layer 4 Proto >
#0-(6-2030)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:19	172.16.29.138	172.16.29.139	ICMP
#1-(6-2029)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:19	172.16.29.138	172.16.29.139	ICMP
#2-(6-2028)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:19	172.16.29.138	172.16.29.139	ICMP
#3-(6-2027)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:15	172.16.29.138	172.16.29.139	ICMP
#4-(6-2026)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:15	172.16.29.138	172.16.29.139	ICMP
#5-(6-2025)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:15	172.16.29.138	172.16.29.139	ICMP
#6-(6-2022)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:12	172.16.29.138	172.16.29.139	ICMP
#7-(6-2023)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:12	172.16.29.138	172.16.29.139	ICMP
#8-(6-2024)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:12	172.16.29.138	172.16.29.139	ICMP
#9-(6-2020)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:09	172.16.29.138	172.16.29.139	ICMP
#10-(6-2019)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:09	172.16.29.138	172.16.29.139	ICMP
#11-(6-2021)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:09	172.16.29.138	172.16.29.139	ICMP
#12-(6-2016)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:06	172.16.29.138	172.16.29.139	ICMP
#13-(6-2017)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20 09:20:06	172.16.29.138	172.16.29.139	ICMP
#14-(6-2018)	[snort] ICMP Destination Unreachable Host Unreachable	2011-01-20	172.16.29.138	172.16.29.139	ICMP

Gambar IV.2.3.3 Pendeteksian DDoS Attack Pada Snort dan Base

Snort dan base berhasil mendeteksi serangan ddos dengan menggunakan ettercap. Komputer penyerang memiliki IP 172.16.29.138 menyerang komputer dengan IP 172.16.29.139.

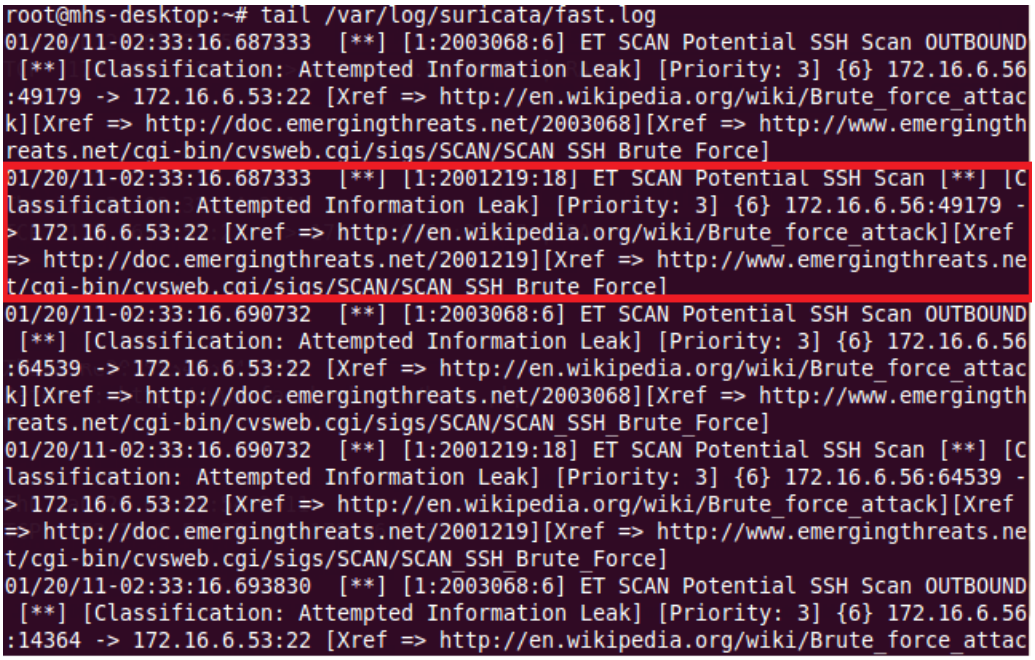
IV.2.3.2 OSSEC



Gambar IV.2.3.4 Pendeteksian DDoS Attack Pada Ossec

OSSEC berhasil mendeteksi serangan ddos dengan menggunakan ettercap. Komputer penyerang memiliki IP 172.16.29.137 (*fake-ip*) menyerang komputer dengan IP 172.16.29.139.

IV.2.3.3 Suricata



Gambar IV.2.3.5 Pendeteksian DDoS Attack Pada Suricata

Suricata berhasil mendeteksi serangan ddos dengan menggunakan ettercap. Komputer penyerang memiliki IP 172.16.6.56 (*fake-ip*) menyerang komputer dengan IP 172.16.29.53.

IV.3 Perbandingan OSSEC, Snort dan Base, Suricata

Kesimpulan berdasarkan kriteria yang ditetapkan sebagai bahan perbandingan antara *software intrusion detection system* dijelaskan pada tabel berikut:

Tabel IV.2.3.1 Kesimpulan

Kriteria Evaluasi	**Intrusion Detection System Open Source**		
	OSSEC	**Snort dan Base**	**Suricata**
Scalability	Dukungan komersial, pengguna milis, pengembangan milis, Bugzilla.	Forum,mailing list, laporan bug	Mailing list.
Vendor Support			
Signature update	Ada.Dapat dilakukan secara otomatis dan manual.	Ada.Dilakukan secara manual.	Ada.Dilakukan secara manual.
Kemampuan mendeteksi serangan	• Nikto (serangan vulnerability scanning) • Nmap (serangan *port* scanning) • Ettercap (serangan *dos attack)* • tidak dapat mendeteksi ping flood (serangan ddos)	• nikto (serangan vulnerability scanning) • ping flood (serangan ddos). • Ettercap (serangan *dos attack)* • tidak dapat mendeteksi nmap (serangan *port* scanning)	• nikto (serangan vulnerability scanning). • Nmap (serangan *port* scanning) • Ettercap (serangan *dos attack)* • tidak dapat mendeteksi ping flood
Stability, reliability and security	Konsisten, realtime, menggunakan mysql.	Konsisten, realtime, menggunakan mysql.	Konsisten, realtime, menggunakan mysql.
Penyediaan informasi	Signature yang diberikan mudah dimengerti dan sangat jelas. *Report* ditampilkan dibagian OSSEC *server*. Cepat dalam mendeteksi intrusi.	Signature yang diberikan cukup mudah dimengerti dan dipahami. Tidak adanya pengklasifikasian intrusi apakah berpotensi bahaya atau tidak. Sehingga dibutuhkan pemahaman yang lebih untuk mengerti pada signature-	Signature yang diberikan mudah dimengerti. Terdapatnya klasifikasi mengenai intrusi yaitu penggolongan apakah aktivitas tersebut berpotensial buruk. Cepat dalam mendeteksi intrusi. Diperlukan ketelitian dalam membaca

Kriteria Evaluasi	Intrusion Detection System Open Source		
	OSSEC	**Snort dan Base**	**Suricata**
		signature yang diberikan.Informasi aktivitas jaringan ditampilkan dalambentuk tabel. Cepat dalam mendeteksi intrusi.	*report* karena berbasis *console* sehingga informasi yang diberikan tidak dalam bentuk tabel.
Kelebihan	• Rule dapat diupdate secara otomatis. Sehingga OSSEC lebih uptodate terhadap jenis-jenis serangan yang baru. • Lebih mudah dalam melakukan instalasi	• Snort dapat diintegrasikan pada beberapa web seperti base. • Snort memiliki kemampuan monitoring pada sisi *server*, sehingga setiap *client* yang terhubung pada *server* snort tidak perlu melakukan instalasi snort sehingga lebih efisien.	• Suricata dapat menggunakan rule pada snort. • Adanya pengklasifikasian dan prioritas aktivitas jaringan, sehingga mempermudah dalam mengenali suatu aktivitas apakah berpotensi buruk.
Kekurangan	OSSEC agent harus diinstall pada setiap komputer dan OSSEC agent harus terhubung ke OSSEC *server* agar dapat mendeteksi intrusi. Hal ini membutuhkan banyak waktu.	• Dibutuhkan pemahaman yang cukup tinggi dalam memahami signature yang diberikan oleh snort. • Update rule tidak dapat dilakukan secara otomatis. Update rule tersedia free dan berbayar.	Belum adanya web yang dapat diintegrasikan dengan suricata sehingga menyulitkan dalam membaca *report* yang ditampilkan pada console, memerlukan ketelitian.

Bab V Kesimpulan, Saran, dan Solusi

V.1 Kesimpulan

Kesimpulan yang dapat diambil berdasarkan hasil perbandingan *software intrusion detection system* adalah:

1. Proses pemilihan *software* IDS dilakukan berdasarkan efisiensi penerapannya:
    - Untuk penerapan jangka panjang suricata, snort dan base baik untuk diterapkan di Politeknik Batam, dengan mengubah semua ip pada server www.polibatam.ac.id, Venus, Students, Bumi, dan Bimasakti. IP publik pada server-server tersebut tidak dibuka. Server snort dan base diletakkan di depan switch Astinet. Sehingga semua paket yang datang dari ISP Astinet dan Starcom dapat dianalisis oleh suricata, snort dan base. Sehingga suricata, snort dan base selain berfungsi sebagai IDS juga sebagai router.
    - Untuk penerapan jangka pendek OSSEC baik untuk diterapkan di Politeknik Batam dengan menggantikan fungsi salah satu server menjadi IDS.
2. Berdasarkan tujuan penelitian, telah diketahui kelebihan dan kelemahan yang terdapat dalam software IDS, serta menyimpulkan hasil evaluasi proses perbandingan software IDS yang dapat digunakan sebagai acuan dalam pemilihan software IDS.
3. Kriteria yang digunakan dalam proses perbandingan adalah:
    - Kemampuan mendeteksi serangan
    - Stability, reliability and security
    - Penyediaan informasi
    - Scalability
    - Vendor support

V.2 Saran

Adapun saran untuk pengembangan dalam proses perbandingan selanjutnya adalah:

1. Software-software yang digunakan dalam proses pemilihan ditambah.
2. Kriteria yang digunakan dalam proses perbandingan dapat ditambahkan.
3. Pengujian dengan berbagai serangan ditambah, sehingga diketahui lebih banyak lagi serangan yang dapat dideteksi.
4. Pemilihan software dapat diterapkan pada IPS (Intrusion Prevention System), sebuah sistem yang dapat mendeteksi dan mencegah serangan.

DAFTAR PUSTAKA

1. http://comes.umy.ac.id, diakses pada tanggal 24 Oktober 2010
2. http://ronajingga.com, diakses pada tanggal 26 Oktober 2010.
3. http://yudiagusta.files.wordpress.com, diakses pada tanggal 3 November 2010.
4. http://akuyola.wordpress.com, diakses pada tanggal 4 November 2010.
5. http://community.gunadarma.ac.id, diakses pada tanggal 12 November 2010.
6. http://jancokitusenihacking.files.wordpress.com, diakses pada tanggal 12 November 2010.
7. http://www.knowledg-e.co.cc, diakses pada tanggal 15 November 2010.
8. http://syah69.blogspot.com, diakses pada tanggal 15 November 2010.
9. http://lists.jammed.com, diakses pada tanggal 18 November 2010.
10. http://jancokitusenihacking.files.wordpress.com, diakses pada tanggal 19 November 2010.
11. http://www.sans.org, diakses pada tanggal 19 Januari 2011.

LAMPIRAN PROSES IMPLEMENTASI

Berikut akan dijabarkan mengenai proses konfigurasi pada empat buah software Intrusion Detection System.

1. OSSEC

1. Setting ip address untuk server 172.16.29.138
2. Download OSSEC server

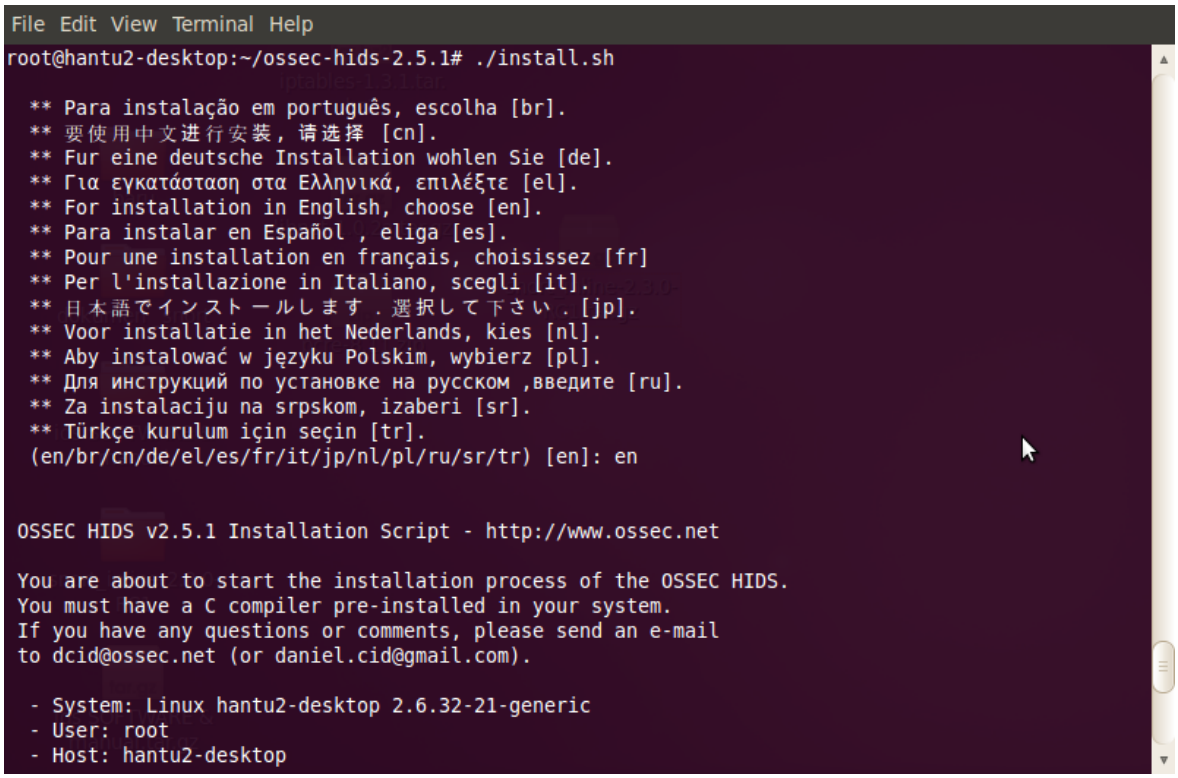
   # wget http://www.ossec.net/files/ossec-hids-latest.tar.gz
3. Mengekstrak file dengan perintah:

   tar –xzf ossec hids
4. Menginstal ossec dengan perintah

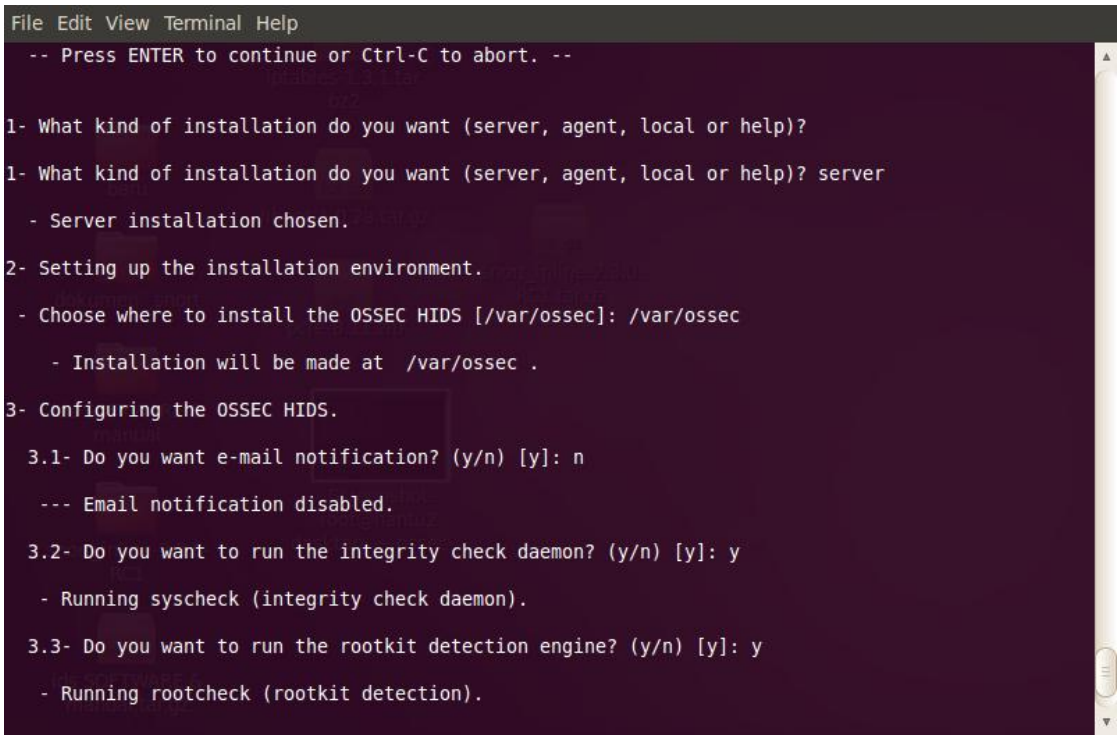
   cd ossec-hids

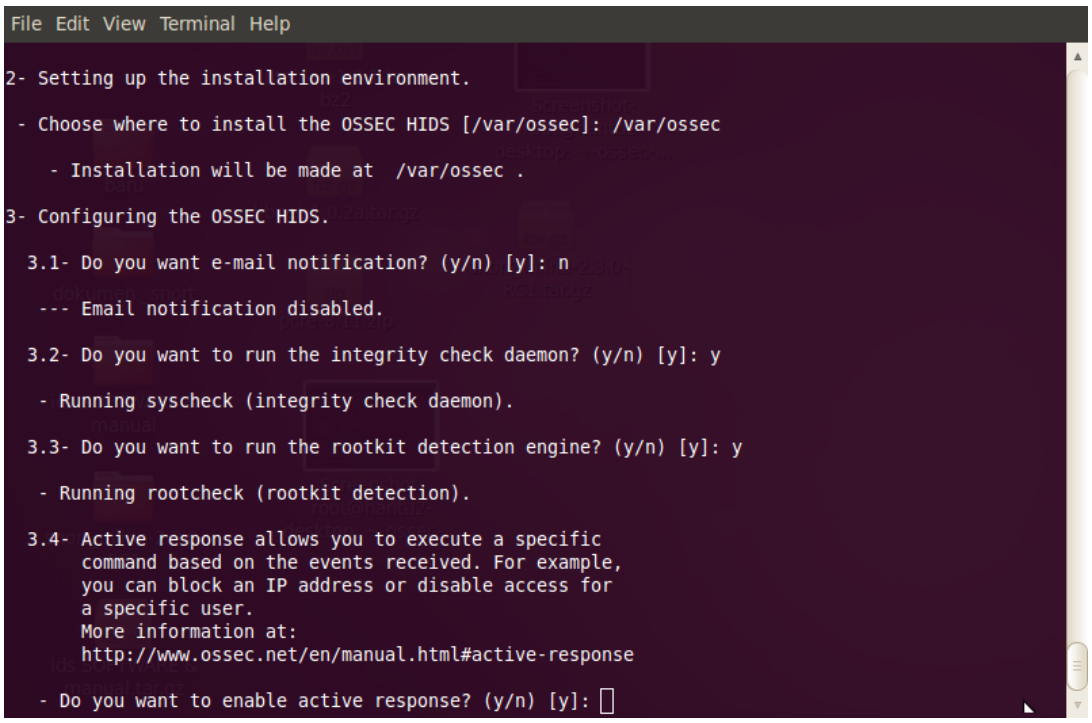
   ./install.sh

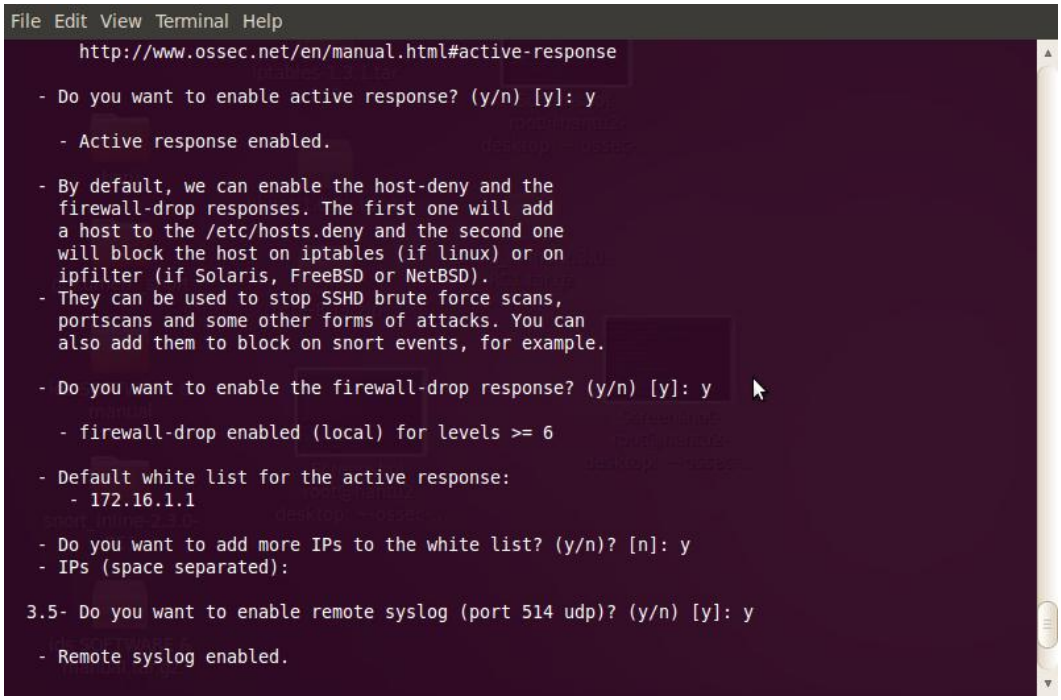
   Maka tampilan yang akan muncul adalah sebagai berikut:

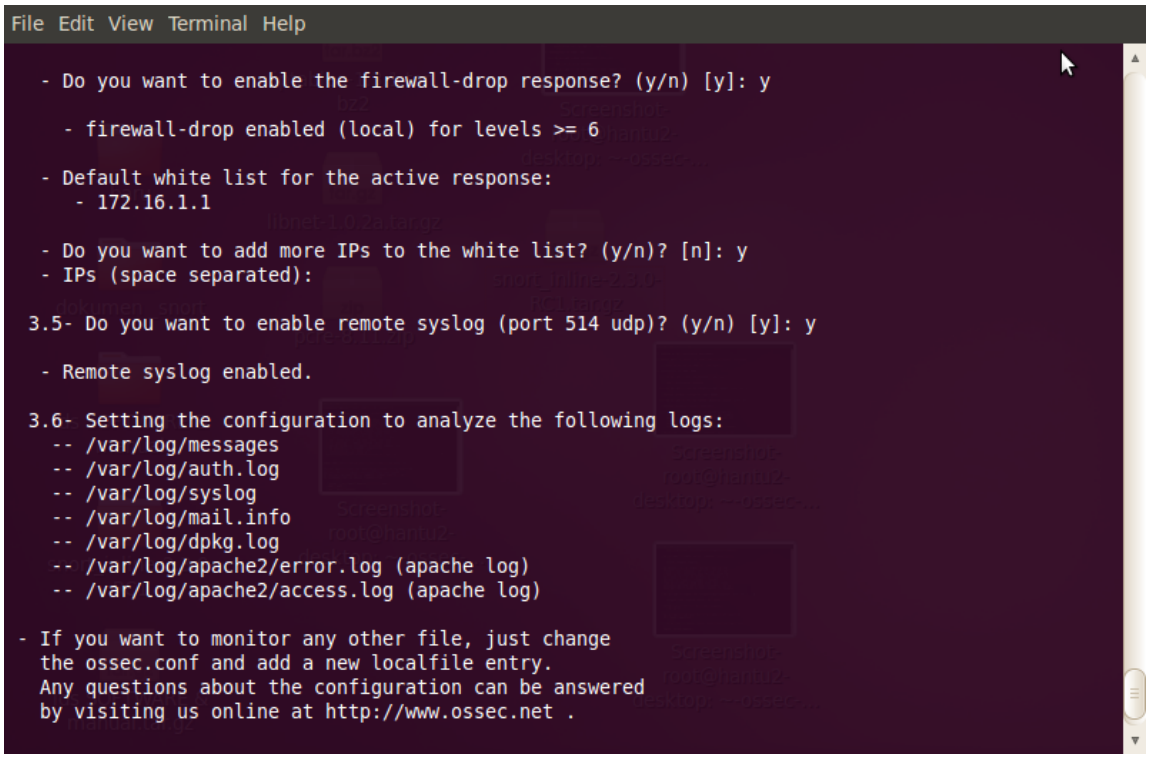


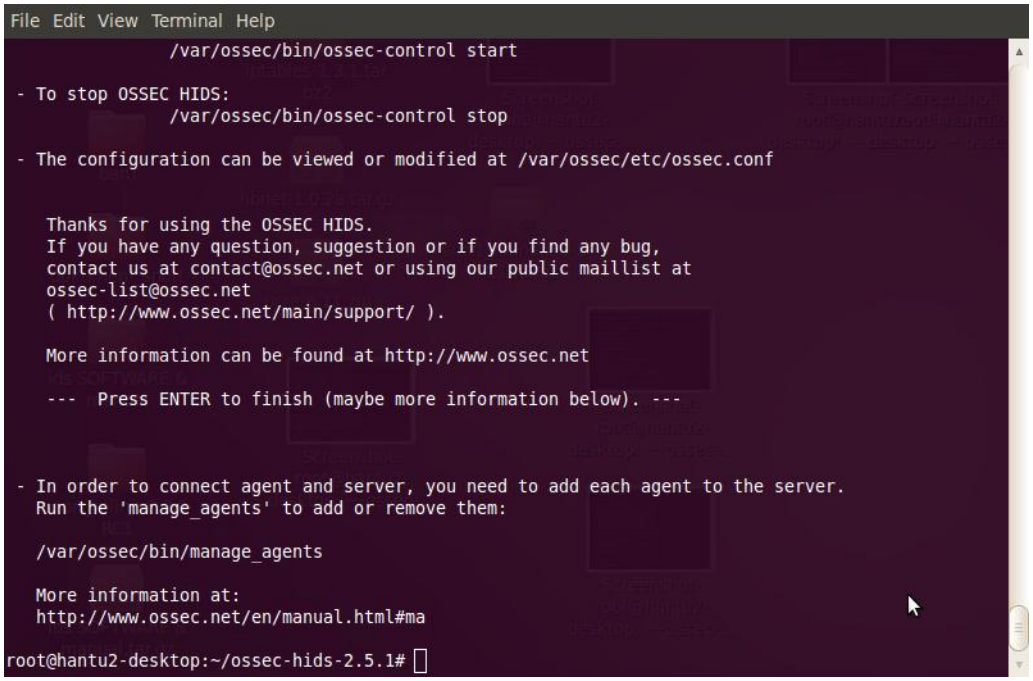
5. Kemudian ikuti perintah proses install.






File Edit View Terminal Help
http://www.ossec.net/en/manual.html#active-response
- Do you want to enable active response? (y/n) [y]: y
- Active response enabled.
- By default, we can enable the host-deny and the
firewall-drop responses. The first one will add
a host to the /etc/hosts.deny and the second one
will block the host on iptables (if linux) or on
ipfilter (if Solaris, FreeBSD or NetBSD).
- They can be used to stop SSHD brute force scans,
portscans and some other forms of attacks. You can
also add them to block on snort events, for example.
- Do you want to enable the firewall-drop response? (y/n) [y]: y
- firewall-drop enabled (local) for levels >= 6
- Default white list for the active response:
- 172.16.1.1
- Do you want to add more IPs to the white list? (y/n)? [n]: y
- IPs (space separated):
3.5- Do you want to enable remote syslog (port 514 udp)? (y/n) [y]: y
- Remote syslog enabled.


File Edit View Terminal Help
- Do you want to enable the firewall-drop response? (y/n) [y]: y
- firewall-drop enabled (local) for levels >= 6
- Default white list for the active response:
- 172.16.1.1
- Do you want to add more IPs to the white list? (y/n)? [n]: y
- IPs (space separated):
3.5- Do you want to enable remote syslog (port 514 udp)? (y/n) [y]: y
- Remote syslog enabled.
3.6- Setting the configuration to analyze the following logs:
-- /var/log/messages
-- /var/log/auth.log
-- /var/log/syslog
-- /var/log/mail.info
-- /var/log/dpkg.log
-- /var/log/apache2/error.log (apache log)
-- /var/log/apache2/access.log (apache log)
- If you want to monitor any other file, just change
the ossec.conf and add a new localfile entry.
Any questions about the configuration can be answered
by visiting us online at http://www.ossec.net .



6. Setelah instalasi selesai, maka untuk menjalankan OSSEC dapat dilakukan dengan cara

   /var/ossec/bin/ossec-control start
7. Untuk melihat apa saja yang dilakukan server dapat dengan cara

   tail /var/ossec/log/ossec.log –f

OSSEC server dapat menerima informasi atau data dari agent menggunakan port udp 1514, sehingga firewall harus tetap dibuka. Untuk memudahkan dalam melakukan monitor aktivitas, maka perlu menginstall web OSSEC. Tahapan instalasi web OSSEC sebagai berikut:

1. Download web OSSEC (http://www.ossec.net/main/downloads)
2. Ekstrak file di /var/www

   tar –xzf ossec-wui
3. Tambahkan user apache dalam group ossec agar dapat membaca data ossec

   vi /etc/group :

   ossec:x:500: menjadi

   ossec:x:500:apache

   apache adalah user webserver.

4. Untuk webserver harus support PHP, apabila belum install maka terlebih dahulu harus diinstall.

   apt-get install httpd php
5. Membuat keamanan web ossec

   cd ossec-wui

   ./setup.sh

   Perintah setup.sh akan membuat file .htaccess dan .htpasswd, dengan tujuan agar setiap yang akses web ossec harus memasukkan password.

Setelah memiliki ossec-hids-server, maka tahapan selanjutnya adalah menginstall ossec agent di server lain agar setiap server dapat mengirmkan informasi ke server ossec-hids. Tahapan instalasi OSSEC agent sebagai berikut:

1. Setting ip address untuk agent 172.16.29.139.
2. Download OSSEC agent

   # wget http://www.ossec.net/files/ossec-hids-latest.tar.gz
3. Ekstrak file

   tar –xzf ossec-hids
4. Menginstall OSSEC agent dengan perintah

   cd ossec-hids

   ./install.sh

Setelah melakukan instalasi OSSEC agent, perlu dilakukan registrasi ossec agent ke ossec server berikut tahapan yang dilakukan:

A. Di server
   1. Pastikan bahwa port UDB 1514 di server tidah ditutup oleh firewall
   2. Lakukan management agent dengan perintah

      /var/ossec/bin/manage-agent

      Hasilnya:

      ****

      * OSSEC HIDS v2.4.1 Agent manager. *

```
* The following options are available:                *
****************************************
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
(Q)uit.
Choose your action: A,E,L,R or Q: a

- Adding a new agent (use '\q' to return to the main menu).
Please provide the following:
* A name for the new agent: mailx.uii.ac.id
* The IP Address of the new agent: 172.16.29.139
* An ID for the new agent[022]:
Agent information:
ID:022
Name:toshiba
IP Address:172.16.29.139

Confirm adding it?(y/n): y
Agent added.

****************************************
* OSSEC HIDS v2.4.1 Agent manager.             *
* The following options are available:                *
****************************************
(A)dd an agent (A).
(E)xtract key for an agent (E).
(L)ist already added agents (L).
(R)emove an agent (R).
```

(Q)uit.

Choose your action: A,E,L,R or Q: e

Available agents:

……

ID: 022, Name: mailx.uii.ac.id, IP: 172.16.29.139

Provide the ID of the agent to extract the key (or '\q' to quit): 022

Agent key information for '022′ is:

MDIyIG1haWx4LnVpQxN2IwODQ2MjE ……

** Press ENTER to return to the main menu.

3. Catat Agent key

B. Di agent

1. Lakukan management agent dengan perintah /var/ossec/bin/manage_agents

   Hasilnya

   ****

   * OSSEC HIDS v2.4.1 Agent manager. *

   * The following options are available: *

   ****

   (I)mport key from the server (I).

   (Q)uit.

   Choose your action: I or Q: i

   * Provide the Key generated by the server.

   * The best approach is to cut and paste it.

   *** OBS: Do not include spaces or new lines.

Paste it here (or '\q' to quit):

MDIyIG1haWx4LnVpQxN2IwODQ2MjE ……

Agent information:

ID:022

Name:mailx.uii.ac.id

IP Address: 172.16.29.139

Confirm adding it?(y/n): y

Added.

** Press ENTER to return to the main menu.

2. Jalankan ossec agent dengan perintah

   /var/ossec/bin/ossec-control start

   Hasilnya:

   Starting OSSEC HIDS v2.4.1 (by Trend Micro Inc.)…

   Started ossec-execd…

   Started ossec-agentd…

   Started ossec-logcollector…

   Started ossec-syscheckd…

   Completed.

2. Suricata

Menginstall semua paket yang dibutuhkan.

- build-essential
  - checkinstall
  - libpcre3-dev
  - libpcap-dev
  - libyaml-dev
  - zliblg-dev

- libnet1libnet1-dev
- libcap-ng-dev
- libhtp1
- libnetfilter-queue-dev
- libnetfilter-queue1
- libnfnetlink-dev
- libnfnetlink0

Kemudian setting IP 172.16.6.53 / 16, gateway 172.16.1.1.

1. membuat group dan user suricata dengan perintah berikut:
   sudo useradd –s /bin/false –c “Suricata user” suricata
2. membuat direktori konfigurasi dan log suricata dengan perintah berikut:
   sudo mkdir /etc/suricata
   sudo mkdir /var/log/suricata
   sudo chown suricata.suricata /var/log/suricata
3. Jika suricata belum di download maka dapat di download di http://openinfosecfoundation.org/downloads/suricata-1.0.1.tar.gz.
   Kemudian lakukan ekstrak file dengan perintah berikut:
   tar –xzvf suricata-1.0.1.tar.gz
4. Setelah melakukan ektrak file, lakukan konfigurasi perangkat lunak suricata agar mendukung Intrusion Prevention System menggunakan Netfilter netlink-queue library (nfqueue) dengan menggunakan perintah:
   cd suricata-1.0.1
   ./configure --enable-nfqueue --enable-debug
   make
5. Jika pada saat kompilasi suricata tidak terdapat kegagalan lanjutkan ke tahap berikutnya yaitu
   sudo checkinstall

6. Hasil eksekusi checkinstall akan menghasilkan packet binary pada deb suricata (suricata_1.0.1-1_i386.deb). kemudian install packet binary tersebut dengan menggunakan perintah:
   sudo dpkg –i suricata_1.0.1-1_i386.deb
7. Setelah instalasi suricata selesai, lakukan konfigurasi suricata terlebih dahulu dengan menggunakan perintah berikut:
   sudo cp suricata.yaml /etc/suricata
   sudo cp classification.config /etc/suricata
8. Download rule (http://www.emergingthreats.net/rules/) kemudian mengekstrak file dengan perintah:
   sudo tar –xzvf emerging.rules.tar.gz –C /etc/suricata/
9. Mengedit file /etc/suricata/suricata.yaml. Ubah bagian section rule-files hapus semua nama file rule yang tidak mengandung kata emerging dan biarkan nama rule yang mengandung kata emerging, sehingga menjadi seperti berikut:
   # Set the default rule path here to search for the files.
   # if not set, it will look at the current working dir
   default-rule-path: /etc/suricata/rules/
   rule-files:
   - emerging-attack_response.rules
   - emerging-dos.rules
   - emerging-exploit.rules
   - emerging-game.rules
   - emerging-inappropriate.rules
   - emerging-malware.rules
   - emerging-p2p.rules
   - emerging-policy.rules
   - emerging-scan.rules
   - emerging-virus.rules
   - emerging-voip.rules

- emerging-web.rules
- emerging-web_client.rules
- emerging-web_server.rules
- emerging-web_specific_apps.rules
- emerging-user_agents.rules
- emerging-current_events.rules

Langkah konfigurasi berikutnya adalah mendefinisikan variabel HOME_NET, variabel ini menunjukan alamat jaringan atau alamat komputer yang akan dipantau oleh suricata. Edit file /etc/suricata/rules sehingga variabel HOME_NET sesuai dengan alamat jaringan atau alamat komputer yang akan dipantau, seperti berikut ini:

HOME_NET: "[172.16.29.0/16,10.0.0.0/8]"

3. Snort dan Base

1. Setting IP 172.16.29.139, gateway 172.16.1.1.
2. Terlebih dahulu membuat folder sebagai tempat hasil download dan kompilasi.

   # mkdir /root/snorttemp

   # cd /root/snorttemp
3. File yang dibutuhkan adalah program snort (versi 2.6.1.1), rule snort (sesuai dengan versi snort yang digunakan), PCRE (versi 6.7), libpca (versi 0.95), base (versi 1.2.7), adodb (versi adodb-493a-for-php)
4. Lakukan pengekstrakan:

   # tar xzvf snort-2.6.1.1.tar.gz

   # tar xzvf snortrules-snapshot-CURRENT.tar.gz

   # tar xzvf pcre-6.7.tar.gz

   # tar xzvf libpcap-0.9.5.tar.gz

   # tar xzvf base-1.2.7.tar.gz

   # tar xzvf adodb493a.tgz
5. Melakukan kompilasi libpcap

cd /root/snorttemp/libpcap-0.9.5

./configure # make && make install

6. Melakukan kompilasi pcre

   # cd /root/snorttemp/pcre-6.7

   # ./configure

   # make && make install

7. Melakukan kompilasi snort

   # cd /root/snorttemp/snort-2.6.1.1

   # ./configure --enable-dynamicplugin --with-mysql

   # make && make install

8. Membuat direktori map snort untuk log dan rulesnya

   # mkdir -p /etc/snort/rules

   # mkdir /var/log/snort

9. Mengcopy seluruh isi ekstrak snortrules ke direktori map snort

   # cp rules/* /etc/snort/rules/

   # cp -rvf so_rules /etc/snort/

   # cp -rvf doc /etc/snort/

10. Kemudian copy semua file snort ke /etc/snort

    # cd snort-2.6.1.1/etc

    # cp * /etc/snort/

11. Mengedit file snort.conf

    # nano /etc/snort.conf

12. Lakukan perubahan pada baris-baris berikut:

    ganti "var HOME_NET any" jadi "var HOME_NET **172.16.29.0/24**".

    ganti "var EXTERNAL_NET any" jadi "var EXTERNAL_NET **!$HOME_NET**".

    ganti "var RULE_PATH ../rules" jadi "var RULE_PATH **/etc/snort/rules**".

    Untuk mysql, perubahan dilakukan pada baris berikut:

output database: log, mysql, user=root password=root dbname=snort host=local$

dan hilangkan tanda "#"

output database: log, mysql, user=root password=root dbname=snort host=local$

Sesuaikan juga username, password dan database yang akan digunakan. Base akan melakukan koneksi database menggunakan username, password dan database tersebut.

13. Menyeting database untuk snort. Dalam analisis ini digunakan phpmyadmin, karena lebih mudah digunakan.
14. Melakukan test konfigurasi snort:

    # snort -c /etc/snort/snort.conf

    Jika tidak terdapat error, berarti konfigurasi berhasil. Untuk membatalkan test tekan Ctrl+C
15. Memindahkan direktori ADODB ke root direktori web server

    # cd /root/snorttemp/

    # mv adodb /var/www/
16. Memindahkan direktori base ke direktori web server yang dapat diakses.

    # mv base-1.2.7 /var/www/html/
17. Kemudian masuk ke direktori base

    # cd /var/www/html/
18. Untuk mempermudah akses, ganti nama base-1.2.7 menjadi base

    # mv base-1.2.7 base

    Kemudian ganti permissionnya:

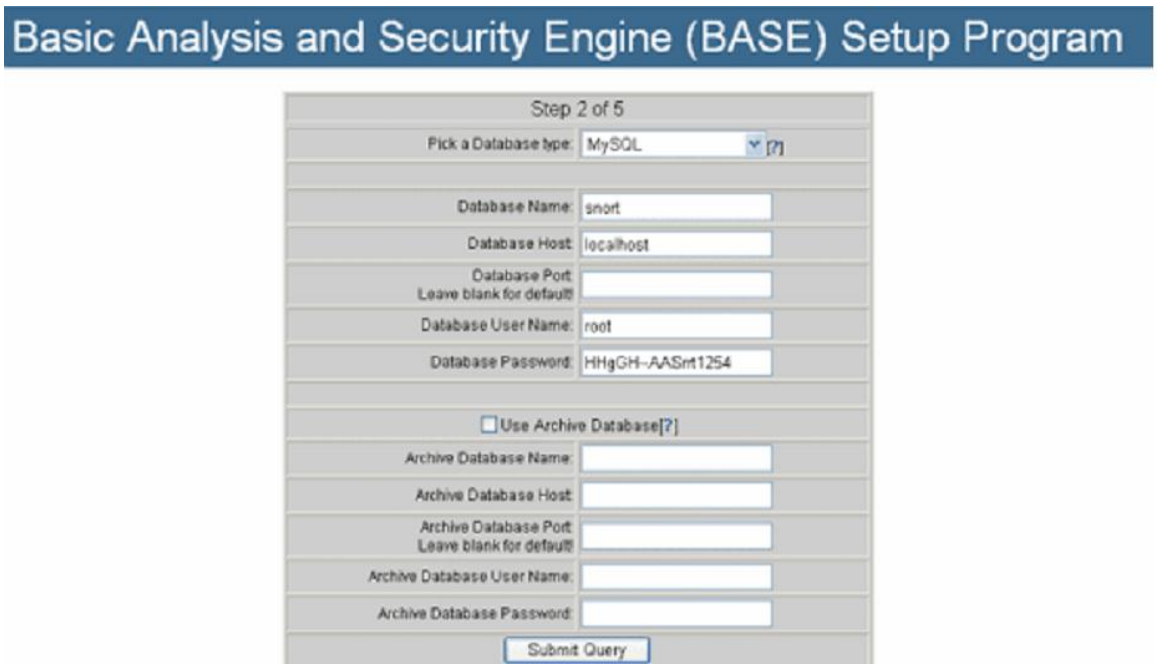
    # chmod 757 base
19. Kemudian buka web browser dan arahkan ke

    http://172.16.29.54/base/setup, mucul tampilan seabagai berikut

20. Setelah itu, klik continue maka muncul tampilan berikut:

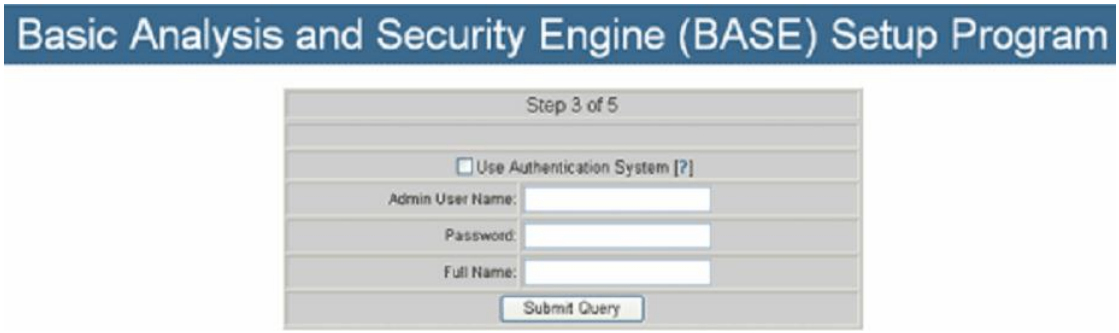
    Masukkan path **ADODB** (/var/www/adodb):

21. Setelah mengklik Submit Query akan muncul tampilan sebagai berikut:



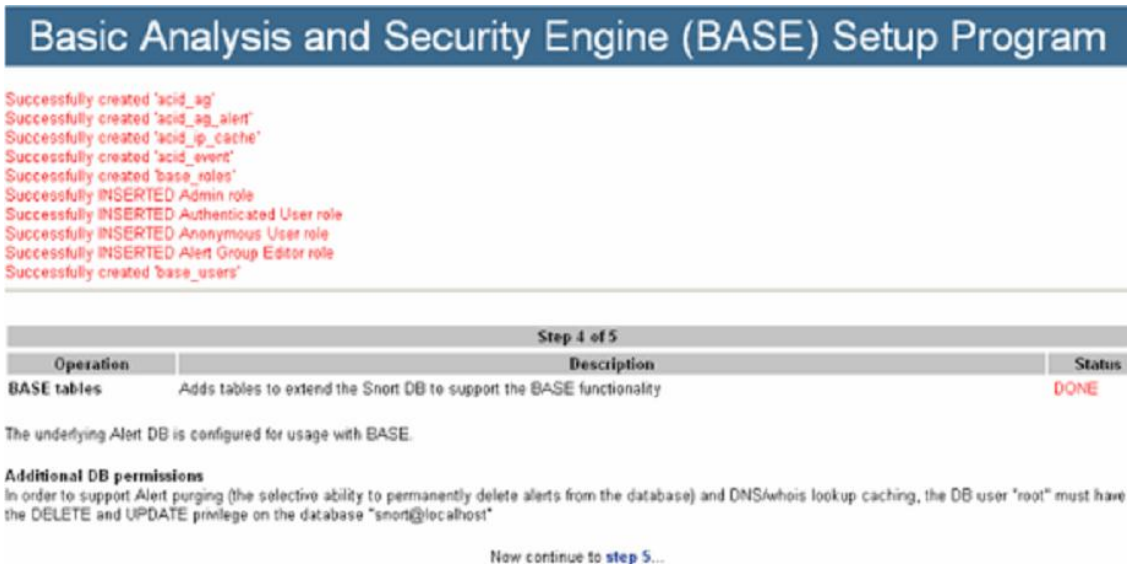
    Tidak perlu mengisi field pada tampilan ini. Kemudian klik Submit Query.

22. Maka tampilan untuk step berikutnya adalah

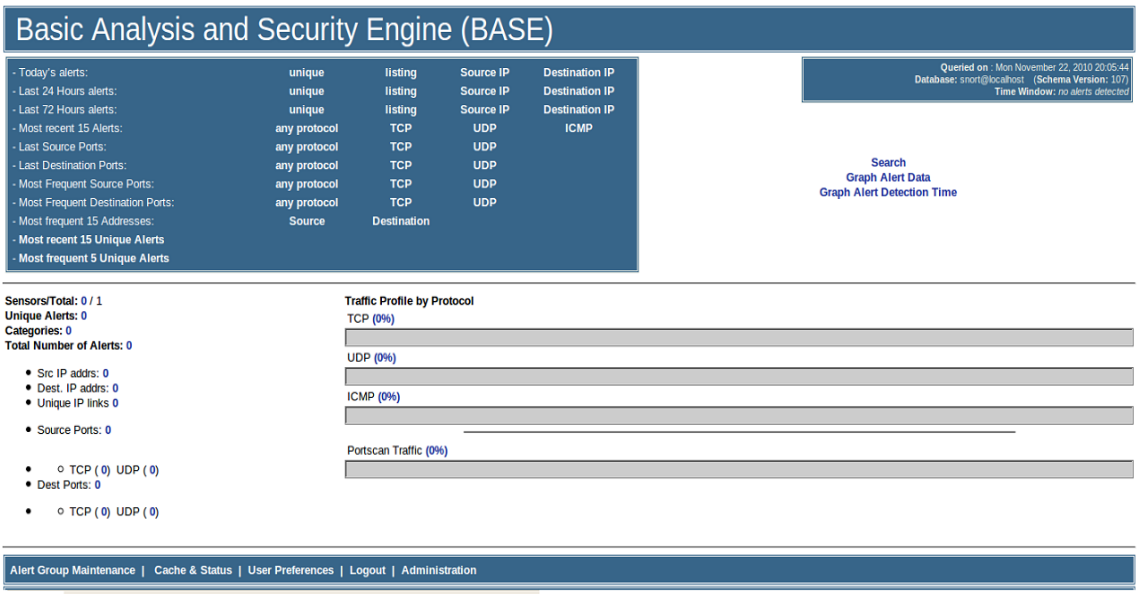


Untuk memberikan tingkat keamanan yang lebih, dapat menggunakan Use Authentication System. Kemudian klik Submit Query.

23. Berikut tampilan pada step keempat:



24. Berikut tampilan pada tahap kelima


Basic Analysis and Security Engine (BASE)
- Today's alerts: unique listing Source IP Destination IP
- Last 24 Hours alerts: unique listing Source IP Destination IP
- Last 72 Hours alerts: unique listing Source IP Destination IP
- Most recent 15 Alerts: any protocol TCP UDP ICMP
- Last Source Ports: any protocol TCP UDP
- Last Destination Ports: any protocol TCP UDP
- Most Frequent Source Ports: any protocol TCP UDP
- Most Frequent Destination Ports: any protocol TCP UDP
- Most frequent 15 Addresses: Source Destination
- Most recent 15 Unique Alerts
- Most frequent 5 Unique Alerts
Queried on : Mon November 22, 2010 20:05:44
Database: snort@localhost (Schema Version: 107)
Time Window: no alerts detected
Search
Graph Alert Data
Graph Alert Detection Time
Sensors/Total: 0 / 1
Unique Alerts: 0
Categories: 0
Total Number of Alerts: 0
Src IP addrs: 0
Dest. IP addrs: 0
Unique IP links 0
Source Ports: 0
TCP (0) UDP (0)
Dest Ports: 0
TCP (0) UDP (0)
Traffic Profile by Protocol
TCP (0%)
UDP (0%)
ICMP (0%)
Portscan Traffic (0%)
Alert Group Maintenance | Cache & Status | User Preferences | Logout | Administration