BAB 2

ANALISIS DAN PERANCANGAN

2.1 Analisis

2.1.1 *Intrusion Detection System*

Intrusion Detection System (disingkat IDS) adalah sebuah aplikasi perangkat lunak atau perangkat keras yang dapat mendeteksi aktivitas yang mencurigakan dalam sebuah sistem atau jaringan. IDS dapat melakukan inspeksi terhadap lalu lintas *inbound* dan *outbound* dalam sebuah sistem atau jaringan, melakukan analisis dan mencari bukti dari percobaan intrusi (penyusupan).

Ada dua jenis IDS, yakni:

- *Network-based 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 umumnya diletakkan pada server-server kritis di jaringan, seperti halnya *firewall*, *web server*, atau server yang terkoneksi ke Internet.

2.1.1.1 Snort

Snort merupakan *software* yang bersifat *opensource* GNU *General Public License* [GNU89], sehingga boleh digunakan dengan bebas secara gratis, dan kode sumber *(source code)* untuk Snort juga bisa didapatkan dan dimodifikasi sendiri .

Snort dikembangkan oleh Marty Roesch. Awalnya dikembangkan di akhir 1998-an sebagai *sniffer* dengan konsistensi *output*.

Snort adalah sebuah *software* ringkas yang sangat berguna untuk mengamati aktivitas dalam suatu jaringan komputer. Snort dapat digunakan sebagai suatu *Network Intrusion Detection System (NIDS)* yang berskala ringan *(lightweight),* dan *software* ini menggunakan sistem peraturan-peraturan *(rules system)* yang relatif mudah dipelajari untuk melakukan deteksi dan pencatatan *(logging)* terhadap berbagai macam serangan terhadap jaringan komputer.

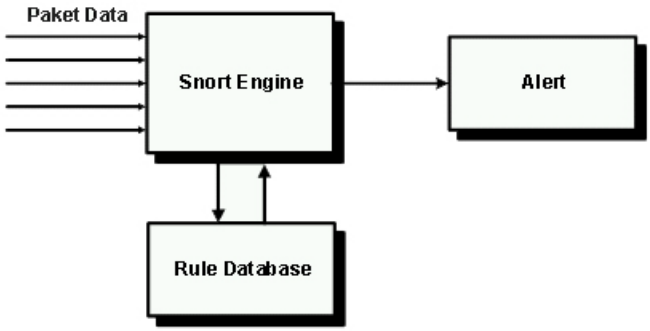
Dengan membuat berbagai *rules* untuk mendeteksi ciri-ciri khas (*signature*) dari berbagai macam serangan, maka Snort dapat mendeteksi dan melakukan *logging* terhadap serangan-serangan tersebut. Snort pada awalnya dibuat untuk sistem operasi berdasarkan Unix, tetapi versi Windows juga sudah dibuat sehingga sekarang ini Snort bersifat *cross-platform*.

Snort dapat berjalan dalam tiga mode, yaitu *sniffer*, *packet logger* dan *intrusion detection system*.

- Mode *Sniffer*, disini snort berfungsi untuk melihat paket yang lewat di jaringan.
- Mode *Packet Logger*, disini snort mencatat semua paket yang lewat di jaringan untuk dianalisa di kemudian hari.
- Mode *intrusion detection*, 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.

2.1.1.2 Prinsip Kerja Snort

Pada prinsipnya Snort bekerja dengan mengambil paket data di jaringan kemudian diolah oleh preprosesor sehingga tinggal data-data yang diperlukan saja kemudian menganalisa dengan mencocokkan dengan *rule*. Apabila sesuai dengan *rule* maka *alert* akan dihasilkan dan paket data dicatat ke dalam *log*.



Gambar 2.1. Prinsip kerja Snort

2.1.1.3. *Rule* Snort

Rule Snort merupakan *database* yang berisi pola-pola serangan berupa *signature* jenis-jenis serangan. *Rule* Snort harus di-*update* secara rutin agar ketika ada suatu teknik serangan yang baru, serangan tersebut dapat terdeteksi. *Rule* Snort dapat didownload di www.snort.org.

Rule Snort terdiri dari dua bagian utama yaitu *rule header* dan *rule option.*

Rule header berisi informasi tentang aturan tindakan apa yang akan dilakukan *rule* dan juga berisi kriteria untuk pencocokan aturan terhadap paket data. Pada *rule header* berisi :

- *action,* adalah aksi yang diambil ketika suatu serangan cocok dengan *rule*. *Action* dalam *rule* Snort diantaranya :
    - *pass,* digunakan agar Snort melewatkan paket.
    - *log,* digunakan untuk melakukan pencatatan *(logging)*
    - *alert,* digunakan untuk membuat *alert* kemudian melakukan

pencatatan.

- *activate,* digunakan untuk membuat *alert* kemudian mengaktifkan *rule* lain.
- *dynamic,* untuk diaktifkan oleh *rule activate*

- *protocol,* protokol paket data yang dideteksi. Protokol yang didukung antara lain : TCP, UDP, ICMP, IP.
- *IP address,* penulisannya dalam bentuk Classless Inter-Domain Routing (CIDR). *IP address* dalam *rule* Snort terdiri dari :
  - *source IP* alamat IP host yang melakukan intrusi.
  - *destenation ip,* alamat IP host tujuan intursi..

  Contoh penulisan :
  - *any* digunakan untuk mendeteksi alamat IP berapapun
  - 10.0.2.1/32 digunakan untuk mendeteksi host 10.0.2.1
  - 192.168.0.0/24 digunakan untuk mendeteksi jaringan 192.168.0.0/24
  - !10.0.2.1/32 digunakan untuk mendeteksi selain host 10.0.2.1
- *port,* terdiri dari :
  - *source port, port* host yang melakukan intrusi.
  - *destenation port, port* host tujuan.

  Contoh penulisan :
  - *any* digunakan untuk mendeteksi *port* berapapun.
  - !80 digunakan untuk mendeteksi selain *port* 80.
  - :6000 digunakan untuk mendeteksi *port* 1 sampai 6000.

- 500: digunakan untuk mendeteksi *port* 500 ke atas.
- 10:1024 digunakan untuk mendeteksi *port* 10 sampai 1024.

- *direction,* menyatakan arah dari aliran paket.

  Contoh penulisan :

  - -> memeriksa paket dengan arah aliran ke host tujuan.
  - <- memeriksa paket dengan arah aliran menuju host sumber intrusi.
  - <> memeriksa paket dengan dua arah aliran, memeriksa paket dengan arah aliran menuju host sumber intrusi atau sebaliknya.

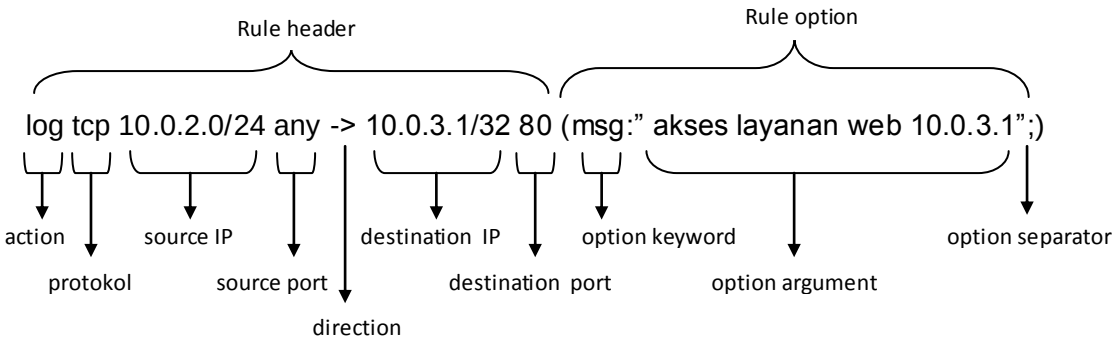
Rule option berisi pesan peringatan dan informasi tentang bagian mana dari paket yang digunakan untuk menghasilkan *alert*. Bagian ini juga berisi kriteria tambahan untuk pencocokan *rule* terhadap paket data. Pada rule option berisi :

- *option keyword,* digunakan untuk mengidentifikasi paket untuk menghasilkan alert. Beberapa contoh *option keyword* yaitu :
  - *msg*, untuk memberikan pesan berupa teks tertentu.
  - *flow*, untuk memeriksa arah aliran paket atau status koneksi.
  - *sid*, untuk mengenali rule secara unik.
  - *flag*, untuk mendetesi *flag* tertentu
  - dll.
- *option argument,* adalah argumen atau nilai yang akan di-cek dari *option keyword.*

Contoh penggunaan rule option :

- *msg:"Serangan DoS"* akan memberikan pesan "Serangan DoS".
- *flow:established* akan memeriksa hanya pada koneksi yang terbentuk.
- *flag:F* akan memeriksa hanya pada paket yang terdapat *flag* FIN

Berikut adalah contoh struktur *rule* Snort :



Gambar 2.2. Contoh struktur *rule* Snort.

Pada *rule* diatas akan melakukan *logging* dengan pesan *"*akses layanan web 10.0.3.1*"* jika ada paket data menggunakan protocol TCP yang berasal dari jaringan 10.0.2.0/24 port berapapun menuju host 10.0.3.1 port 80.

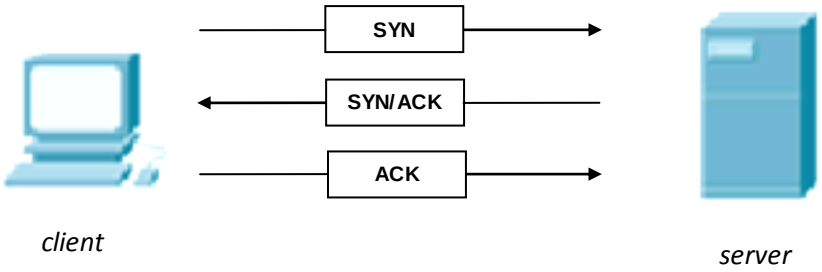
2.1.2 *Flag* TCP

Sebuah segmen TCP dapat memiliki *flag* khusus yang mengindikasikan segmen yang bersangkutan. Jenis-jenis *flag* TCP yaitu : SYN, ACK, URG, RST, PSH dan FIN. Berikut adalah penjelasan masing-masing *flag* :

Nama flag	Keterangan
URG	Mengindikasikan bahwa beberapa bagian dari segmen TCP mengandung data yang sangat penting.
ACK	Mengindikasikan field Acknowledgment, mengandung oktet selanjutnya yang diharapkan dalam koneksi.
PSH	Mengindikasikan bahwa isi dari TCP Receive buffer harus diserahkan kepada protokol lapisan aplikasi..
RST	Mengindikasikan bahwa koneksi yang dibuat akan digagalkan.
SYN	Mengindikasikan bahwa segmen TCP yang bersangkutan mengandung Initial Sequence Number (ISN).
FIN	Menandakan bahwa pengirim segmen TCP telah selesai dalam mengirimkan data dalam sebuah koneksi TCP.

2.1.3 *Three Way Handshake*

Pada saat komputer yang menggunakan protokol TCP/IP ingin melakukan komunikasi, terdapat tahapan awal yang penting yang dinamakan *three way handshake* atau jabatan tangan tiga kali. Berikut adalah ilustrasinya :



Gambar 2.3. Ilustrasi *three way handshake*

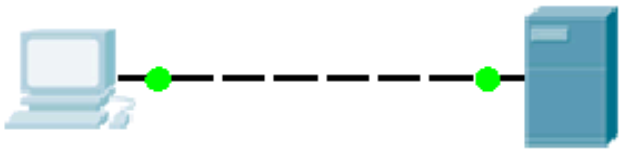
Berikut adalah tahapan-tahapannya :

- Komputer klien akan mengirimkan paket SYN. Paket ini berguna untuk inisiasi koneksi baru antar komputer yang melakukan komunikasi dan menentukan *sequence number* awal yang disepakati kedua pihak. *Sequence number* ini ditentukan secara acak oleh klien.
- Komputer server mengirimkan balasan dengan paket SYN/ACK. Paket ini mempunyai terdapat dua flag yaitu flag SYN dan ACK. Pada paket SYN terdapat *sequence number* yang ditentukan secara acak oleh server. Pada paket ACK memberikan tanda ke klien tentang paket berikutnya yang diharapkan datang dari klien.
- Komputer klien mengirimkan paket ACK sebagai konfirmasi pemberitahuan bahwa paket yang dikirimkan telah diterima.

Setelah proses *three way handshake* selesai, maka koneksi antara klien dan server telah terbentuk (*established*). Selanjutnya kedua komputer dapat melakukan transmisi data.

2.2. Diagram

Berikut adalah rancangan server dengan klien yang akan dibangun. Pada server menggunakan sistem operasi Ubuntu Server10.04 yang diinstal Snort-2.9.2.3. Sedangkan pada klien yang akan bertindak sebagai *intruder*, diinstal sistem operasi Backtrack5 r1.

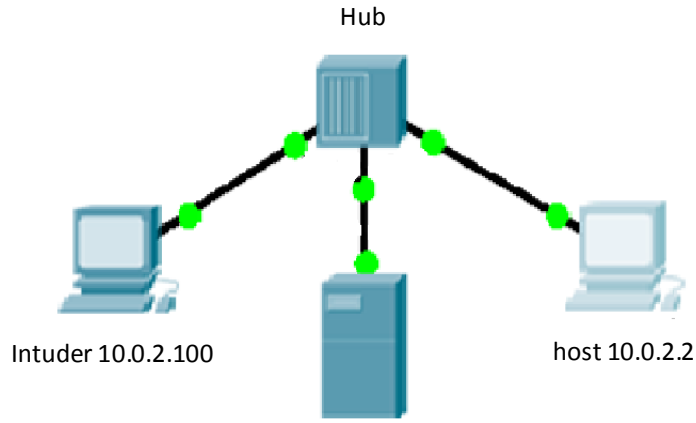


Intruder : Backtrack5 r1

Ubuntu Server10.04 dengan Snort-2.9.2.3

Gambar 2.4. Rancangan jaringan

Berikut adalah rancangan jaringan untuk kasus mendeteksi serangan ke beberapa host tertentu.



Gambar 2.5. Rancangan jaringan kasus mendeteksi serangan ke beberapa host tertentu.