MODUL

MEMBUAT DESAIN SISTEM KEAMANAN JARINGAN

Disusun Oleh:

ABDUL ROHMAN

SMK MUHAMMADIYAH 5 BABAT

Jl. Rumah Sakit No. 15-17 Telp (0322) 451313

e-mail:smkm5babat@yahoo.com

web-site:http://www.smkmuh5babat.com

MATERI UNIT KOMPETENSI

4.1 Tujuan Instruksional Umum

- Siswa mampu menjelaskan bagaimana melakukan perancangan sistem keamanan jaringan
- Siswa dapat mengevaluasi dan melakukan audit terhadap kebutuhan pengendalian sistem keamanan jaringan
- Siswa mampu menyelesaikan masalah-masalah yang terjadi dalam sistem jaringan komputer
- Siswa mampu mendesain pengendalian keamanan jaringan untuk diterapkan dalam sistem jaringan komputer

4.2 Tujuan Instruksional Khusus

- Siswa dapat menjelasan tentang deskripsi kebutuhan yang diperlukan dalam melakukan perancangan pengendalian sistem keamanan jaringan
- Siswa dapat mengidentifikasikan pengendalian yang diperlukan dalam sistem keamanan jaringan dan lebih mengenal perangkat keamanan jaringan
- Siswa lebih mengenal penyebab dan masalah yang terjadi pada sistem keamanan jaringan, sehingga diharapkan mampu mengatasinya
- Siswa dapat mengontrol, menyelesaikan, dan memelihara keamanan jaringan
- Siswa dapat melakukan perancangan keamanan jaringan pada suatu sistem jaringan komputer

4.3 Evaluasi Kebutuhan Pengendalian Sistem Keamanan Jaringan

4.3.1 Batasan Bisnis

Hal-hal yang menjadi batasan bisnis dalam pendesainan sistem keamanan jaringan adalah sebagai berikut:

- Kondisi sistem keamanan jaringan yang sedang berjalan saat ini disuatu kantor/instansi yang terkait, sehingga perancang sistem keamanan diperlukan untuk membuat dokumentasi sistem keamanan jaringan tersebut.
- Suatu kantor/instansi yang terkait memiliki rencana untuk mengembangkan dan meningkatkan sistem jaringan yang sedang berjalan, sehingga pengembang diminta untuk melakukan perancangan sistem keamanan jaringan. Dengan demikian dokumen desain tersebut dapat digunakan sebagai referensi untuk pengembangan dan peningkatan jaringan pada masa yang akan datang.

4.3.2 Biaya dan Sumber Daya

Biaya dalam perancangan sistem keamanan jaringan dapat dianggarkan. Dana dapat disediakan oleh suatu instansi yang terkait apabila ada proposal yang benar dan tepat.

Sumber daya yang dibutuhkan dalam perancangan sistem keamanan jaringan diperlukan kesiapan dan ketersediaan dalam bidang berikut ini:

- *Hardware* : fasilitas perangkat keras yang diperlukan dalam sistem keamanan jaringan
- *Software* : fasilitas perangkat lunak yang diperlukan untuk diinstal pada perangkat jaringan
- *Brainware* : Sumber daya manusia yang akan mengoperasikan dan menggunakan sistem keamanan jaringan

4.3.3 *Time line*

Waktu yang dibutuhkan untuk instalasi adalah tidak lebih dari satu minggu. Hal ini dimaksudkan agar tidak mengganggu kegiatan operasional sehari-hari yang menggunakan *internet* atau sistem jaringan tersebut.

4.3.4 Kebutuhan Staf

Dalam instansi yang terkait, terdapat komputer yang semuanya terhubung ke LAN suatu instansi. Untuk memenuhi kebutuhan pengguna mengenai layanan jaringan, diperlukan staf jaringan minimal dua orang. Dimana staf tersebut merupakan administrator yang akan memanajemen sistem jaringan secara menyeluruh dan yang lain akan menjadi *technical support* yang yang membantu administrator untuk memanajemen jaringan serta mengatasi masalah yang terjadi. Sehingga apabila ada terjadi masalah yang berhubungan dengan sistem jaringan, dapat diatasi dengan cepat.

4.3.5 Kebijakan Manajemen

Access Right

Pembagian hak akses yang ada sesuai dengan kebijakan dari pihak manajemen suatu instansi terkait adalah sebagai berikut:

- Administrator : Bertanggung jawab penuh terhadap sistem jaringan serta memiliki *full access* untuk semua service yang ada pada sistem jaringan. Administrator juga memiliki akses untuk menambah atau mengurangi service dan *account* pada jaringan.
- Pengguna : Memiliki hak akses ke setiap komputer masing-masing dan ke *service* yang ada di jaringan sesuai dengan yang telah ditentukan oleh administrator.

Email

Setiap pegawai yang ada di suatu instansi tersebut memiliki *account* untuk menggunakan layanan *email* yang tersedia di *server* LAN instansi tersebut. Yang berhak untuk menambah atau mengurangi *account* baru untuk penggunaan *email* adalah administrator. Sedangkan pengguna lainnya hanya boleh *login* menggunakan layanan *email* dengan menggunakan *account* yang telah diberikan. Namun meskipun

administrator memiliki *full access* untuk semua *services* yang ada pada jaringan tersebut, administrator tidak berhak untuk menyalahgunakan *account* dari masing-masing pengguna untuk menggunakan *layanan email.*

File Server

File server yang disediakan di *server* dapat digunakan setiap pengguna jaringan yang ada di suatu instansi terkait. Setiap pengguna yang ingin memasuki *file server* harus menggunakan *account* masing-masing pengguna. Sementara *file* yang dapat disimpan pada *file server* merupakan *file* yang penting dan berguna serta yang digunakan untuk bekerja. Pada *file server* juga tersedia *file* yang dapat digunakan bersama dan untuk menggunakan folder ini telah disediakan *account* bersama. Sedangkan *file* pribadi hendaknya disimpan di komputer masing-masing pengguna. Hal ini dilakukan supaya penggunaan *file server* lebih efisien .

Akses *Internet*

Setiap pengguna komputer yang ada di suatu kantor/instansi yang terkait memiliki hak akses untuk terhubung ke *internet*. Komputer yang ada di setiap ruangan juga sudah diset agar dapat terhubung ke *internet*. Waktu yang ditentukan untuk terhubung ke *internet* adalah tidak terbatas. Hal ini dilakukan agar setiap pengguna dapat mengeksplorasi source yang ada di *internet* kapan saja pada saat dibutuhkan.

4.3.6 Kebutuhan Sekuriti

Dalam sistem jaringan komputer yang terdiri dari banyak pengguna, diperlukan sekuriti baik untuk *hardware, software*, maupun pengguna. Berikut ini akan dijelaskan mengenai kebutuhan sekuriti yang diperlukan dalam sistem jaringan.

Tipe Sekuriti

Beberapa tipe sekuriti yang digunakan untuk keamanan dalam sistem jaringan di suatu instansi adalah sebagai berikut:

- Untuk layanan *email* dan *web service* menggunakan jenis sekuriti SSL.
- Untuk setiap *password* yang digunakan menggunakan jenis sekuriti MD5.

Kebutuhan Pengaksesan Data dari Luar

Pengguna dalam sistem jaringan terdiri dari 2 (dua) yaitu yang bersifat internal dan eksternal. Pengguna internal adalah pengguna yang berada di dalam LAN suatu

instansi. Sedangkan pengguna eksternal adalah pengguna yang berada diluar suatu instansi yang butuh untuk meng-*update* data yang ada di dalam sistem jaringan suatu instansi yang terkait tersebut.

Kebutuhan Autentikasi

Setiap komputer yang digunakan oleh setiap pengguna diberi otentifikasi yaitu berupa penamaan *hardware* dan pemberian IP *Address*. Hal ini dilakukan untuk mempermudah proses manajemen setiap perangkat yang ada serta menghindari kebebasan pengguna mengganti perangkat yang telah diberikan dengan perangkat pengguna lainnya.

Kebutuhan Keamanan *Host*

Untuk menjaga keamanan setiap komputer pengguna, maka sebelum menggunakan komputer pengguna harus *login* terlebih dahulu. Sehingga penggunaan setiap komputer teratur dan terkontrol serta tidak sesuka hati setiap pengguna. Dimana tanpa menggunakan *account* yang telah ditentukan untuk setiap komputer, pengguna tidak dapat menggunakan komputer tersebut.

4.3.7 Kebutuhan Manajemen

Kebutuhan manajemen yang diperlukan untuk memanajemen sistem jaringan di suatu instansi adalah sebagai berikut:

- *Configuration Management*

  Digunakan untuk layanan *inventory* dan *topology*, manajemen perubahan, penamaan dan pengalamatan, manajemen *asset* dan kabel, serta proses *backup*.
- *Performance Management*

  Untuk mengukur performansi manajemen suatu jaringan seperti *throughput*, *utilization*, *error rate* dan *respon time*.
- *Fault Management*

  Untuk menentukan permasalahan yang terjadi pada jaringan, mendiagnosis jaringan, melakukan *backup*, serta untuk perbaikan atau perbaikan ulang.
- *Accounting* Management

Untuk mengetahui Track utilisation of *network* resources, Granting and removal of *network access*, serta *Licensing* & *billing*

- *Security* Management

  Dapat digunakan untuk mengontrol pengaksesan jaringan dan untuk keperluan *auditing*.

4.3.8 Kebutuhan Aplikasi

Aplikasi

Pada *server* sistem jaringan suatu instansi, perlu disediakan sebuah *server* khusus untuk *server* aplikasi yaitu *web server*. Aplikasi yang dipakai bersama oleh seluruh pengguna komputer di suatu instansi ditempatkan pada *web server*. Dengan demikian semua pengguna yang ingin menggunakan aplikasi tersebut dapat mengaksesnya dari PC masing-masing apabila sudah terhubung ke *server*. Jenis aplikasi yang ditempatkan pada *web server* tersebut adalah aplikasi berbasis *web*. Semua aplikasi ini dapat diakses dalam lingkungan LAN suatu instansi tersebut.

Protokol

Protokol dalam sebuah jaringan komputer adalah kumpulan peraturan yang mendefenisikan bagaimana cara informasi ditransmisikan melalui jaringan. Ada empat macam protokol jaringan, yaitu IPX/SPX, TCP/IP, UDP dan Apple Talk. Protokol yang digunakan untuk desain jaringan ini adalah protokol yang paling luas penggunaannya, yaitu **protokol TCP/IP**. Alasan pemilihan protokol ini adalah karena protokol ini merupakan protokol transportasi yang paling fleksibel dan dapat digunakan pada area yang luas.

Pengguna

Jumlah pengguna yang akan menggunakan aplikasi yang disediakan dan protokol yang ditentukan adalah ±100 pengguna.

Penggunaan Aplikasi

Aplikasi yang tersedia dalam sistem jaringan suatu instansi dapat digunakan setiap saat baik dari *web* internal maupun dari *web* eksternal. Hal ini dilakukan untuk mempermudah pengguna menggunakan aplikasi kapan saja dibutuhkan.

4.3.9 Karakteristik Trafik Jaringan

Karakteristik trafik jaringan yang baik menunjukkan sistem jaringan yang baik. Ciri karakteristik trafik jaringan yang baik adalah tidak pernah putus dan tidak terlalu tinggi karena hal ini menunjukkan trafik jaringan yang berat.

Karakteristik Trafik *Load*

Karakteristik *traffic load* jaringan yang baik adalah *download* lebih tinggi dari upload. Hal ini dianjurkan karena diasumsikan setiap pengguna *internet* lebih banyak men-*download* data daripada meng-*upload* data. Pada umumnya, perbandingan *upload* dan *download* adalah 1:3.

Tools

Tools yang digunakan untuk melakukan *monitoring* adalah PRTG (untuk sistem operasi windows, untuk sistem operasi linux dapat menggunakan MRTG). PRTG akan menghasilkan halaman HTML yang berisi gambar yang menyediakan visualisasi secara langsung mengenai keadaan trafik jaringan, dan dapat memonitor 50 atau lebih interface pada jaringan. Selain itu PRTG juga memungkinkan administrator jaringan untuk memonitor variabel SNMP sesuai dengan pilihannya.

Untuk dapat memonitor sebuah *Router, Switch*, *server*, *workstation* dan sebagainya, komponen yang harus ada yaitu agen SNMP. Pada jaringan LAN Kantor disuatu instansi, yang menjadi agen SNMP yaitu *Switch, Router* dan beberapa *server*. Pada perangkat-perangkat tersebut, jika belum memiliki agen SNMP sendiri, dapat diinstal SNMP v.3 sebagai agen SNMP-nya. Sedangkan pada perangkat yang berperan sebagai station yaitu *server web,* diinstal PRTG yang dapat melakukan pemantauan *troughput, traffic uplink* dan *downlink*, transmisi data dan kondisi *server* dengan mengumpulkan data-data mengenai hal-hal tersebut dari agen-agen SNMP yang terdapat pada jaringan LAN suatu instansi tersebut.

4.3.10 Kebutuhan Performansi

Performansi adalah salah satu unsur pokok yang perlu diperhatikan dalam sebuah sistem jaringan. Yang perlu diperhatikan dalam manajemen performasi adalah *server*, *network, workstation,* dan *application*.

Desain sistem untuk performasi yang lebih baik adalah sebagai berikut:

- Lebih mengutamakan kecepatan CPU daripada kecepatan jaringan sehingga tidak menimbulkan efek kemacetan jaringan
- Mengurangi jumlah paket untuk mengurangi *overhead software*.
- Menambah jumlah bandwith untuk menghindari penundaan yang terlalu lama, meningkatkan kecepatan pemrosesan, serta mengurangi masalah kemacetan.
- Untuk mengontrol *timeout*, jangan menset *timeout* terlalu lama atau terlalu cepat
- Melakukan pencegahan lebih baik daripada perbaikan untuk menjaga kualitas yang baik baik *hardware* maupun *software*.

Response time

Sistem jaringan yang baik memiliki *respon time* yang cepat terhadap *request* ke suatu *services* di jaringan. Dimana setiap *host* yang mengakses jaringan dapat memperoleh *services* dari jaringan dengan cepat.

Accuracy

Keakuratan (*accuracy*) merupakan persentase dari penggunaan trafik yang secara benar di transmisikan pada sistem, yang berhubungan dengan trafik, termasuk *error* yang terjadi saat transmisi. Dalam hal ini keakuratan juga berhubungan dengan penggunaan aplikasi jaringan dan jaringan itu sendiri. Semakin banyak aplikasi jaringan yang digunakan maka akan semakin tinggi keakuratan dari trafik jaringan yang dibutuhkan agar tidak terjadi *error* saat transmisi data dari aplikasi jaringan tersebut.

Availability

Availability (ketersediaan) dalam jaringan merupakan jumlah waktu operasi jaringan yang tersedia, baik ketersediaan dari jumlah layanan kepada *end user* (pengguna) maupun kepada *server*. Jika *delay* pengiriman paket yang terjadi dalam suatu jaringan terlalu panjang walaupun waktu operasi dari jaringan dapat melayani, maka jaringan tetap saja secara virtual dikatakan tidak tersedia. Untuk performansi jaringan, ketersediaan (*availabilty*) layanan jaringan harus diperhatikan untuk menghindari gangguan dalam jaringan.

Penggunaan Jaringan Maksimum

Penggunaan jaringan maksimum merupakan persentase total kapasitas *bandwidth* dari segmen jaringan yang dapat digunakan sebelum suatu jaringan mengalami gangguan. Melakukan pembatasan pada penggunaan jaringan penting dilakukan untuk mencegah kerusakan atau gangguan pada jaringan, sehingga jaringan mengalami performansi yang baik.

Penggunaan maksimum jaringan dapat diukur dari hal-hal berikut:

- Pengiriman paket yang ada (*actual packets*/sec) berbanding pengiriman paket maksimum (vs max packets/sec)
- Persentase dari penggunaan *bandwidth* yang ada berbanding jumlah bandwidth maksimum yang tersedia
- Jumlah *bandwidth* nyata (*Throughput*) bps yang diterima berbanding dengan jumlah maksimum *Throughput* bps yang mungkin.

Throughput

Throughput adalah pengukuran dari kapasitas transmisi, yaitu jumlah dari data yang berhasil di transfer antar node per unit waktu (yang umumnya diukur berdasarkan detik). *Throughput* disebut juga *bandwidth* aktual yang terukur pada suatu ukuran waktu tertentu dalam suatu hari menggunakan rute *internet* yang spesifik ketika sedang men-*download* suatu *file*. *Throughput* dapat diukur dengan membandingkan keefektifan dari komputer yang sedang menjalankan program aplikasi yang banyak di-*download* dari *internet*.

Latency

Latency adalah waktu yang diperlukan untuk mentransmisikan sebuah frame hingga frame tersebut siap untuk ditransmisikan dari titik asal ke titik awal transmisi. *Latency* dapat mempengaruhi performansi suatu jaringan dalam hal transmisi data. Semakin tinggi *latency* proses pengiriman data akan semakin lambat, sebaliknya *latency* yang kecil akan mempercepat proses pengiriman data.

4.4 Identifikasi Pengendalian Pada Sistem Keamanan Jaringan

4.4.1 Evaluasi Kebutuhan Pengendalian Sistem Keamanan Jaringan

Keamanan Jaringan adalah proses untuk melindungi sistem dalam jaringan dengan mencegah dan mendeteksi penggunaan yang tidak berhak dalam jaringan.

Keamanan itu tidak dapat muncul begitu saja, tetapi harus direncanakan. Misalkan, jika kita membangun sebuah rumah, maka pintu rumah kita harus dilengkapi dengan kunci pintu. Jika kita terlupa memasukkan kunci pintu pada budget perencanaan rumah, maka kita akan dikagetkan bahwa ternyata harus keluar dana untuk menjaga keamanan. Kalau rumah kita hanya memiliki satu atau dua pintu, mungkin dampak dari budget tidak seberapa. Bayangkan bila kita mendesain sebuah hotel dengan 200 kamar dan lupa mem-budget-kan kunci pintu, maka dampaknya akan sangat besar.

Demikian pula di sisi pengamanan sebuah sistem dalam jaringan. Jika tidak kita *budget*-kan di awal, kita akan dikagetkan dengan kebutuhan akan adanya perangkat pengamanan (*firewall, Intrusion Detection Sistem,* anti virus*, Dissaster Recovery Center*, dan seterusnya).

Pengelolaan terhadap sistem pengendalian keamanan dapat dilihat dari sisi pengelolaan resiko (*risk management*). Ada tiga komponen yang memberikan kontribusi kepada *Risk*, yaitu *Asset, Vulnerabilities*, dan *Threat*s.

Tabel 1 Tabel Kontribusi terhadap Risk

Komponen	Contoh dan Keterangan
Asset (Aset)	• *Hardware* • *Software* • Dokumentasi • Data

	• Komunikasi • Lingkungan • Manusia
Threats (ancaman)	• Pemakai (*user*s) • Teroris • Kecelakaan (accident) • *Crackers* • Penjahat criminal • Nasib (*Acts of God*) • Intel luar negeri (*foreign intelligence*)
Vulnerabilities (kelemahan)	• *Software* bugs • *Hardware* bugs • Radiasi (dari layer, transmisi) • *Tapping, crosstalk* • *unauthorized user*s • cetakan, *hardcopy* atau p*rint out* • keteledoran *(oversight*) • *cracker* via telepon • *storage media*

Untuk menanggulangi resiko (*risk*) tersebut dilakukan dengan beberapa usaha yang disebut dengan "*countermeasures*" yang dapat berupa:

- usaha untuk mengurangi *Threat*
- usaha untuk mengurangi *Vulnerability*
- usaha untuk mengurangi impak (*impact*)
- mendeteksi kejadian yang tidak bersahabat (*hostile event*)
- kembali (*recover*) dari kejadian

4.4.2 Klasifikasi Kejahatan Komputer

Kejahatan komputer dapat digolongkan kepada yang sangat berbahaya sampai ke yang hanya mengesalkan (*annoying*). Menurut David Icove berdasarkan lubang keamanan, keamanan dapat diklasifikasikan menjadi empat, yaitu:

1. Keamanan yang bersifat fisik (*physical security*)

Termasuk akses orang ke gedung, peralatan, dan media yang digunakan. Beberapa bekas penjahat komputer (*crackers*) mengatakan bahwa mereka sering pergi ke tempat sampah untuk mencari berkas-berkas yang mungkin memiliki informasi tentang keamanan. Misalnya pernah diketemukan coretan *password* atau manual yang dibuang tanpa dihancurkan. Wiretapping atau hal-hal yang berhubungan dengan akses ke kabel atau komputer yang digunakan juga dapat dimasukkan ke dalam kelas ini. *Denial of service*, yaitu akibat yang ditimbulkan sehingga servis tidak dapat diterima oleh pemakai juga dapat dimasukkan ke dalam kelas ini. *Denial of service* dapat dilakukan misalnya dengan mematikan peralata natau membanjiri saluran komunikasi dengan pesan-pesan (yang dapat berisi apa saja karena yang diutamakan adalah banyaknya jumlah pesan).

2. Keamanan yang berhubungan dengan orang (personel)

Termasuk identifikasi, dan profil resiko dari orang yang mempunyai akses (pekerja). Seringkali kelemahan keamanan sistem informasi bergantung kepada manusia (pemakai dan pengelola). Ada sebuah teknik yang dikenal dengan istilah "*social engineering*" yang sering digunakan oleh kriminal untuk berpura-pura sebagai orang yang berhak mengakses informasi. Misalnya kriminal ini berpura-pura sebagai pemakai yang lupa *password*-nya dan minta agar diganti menjadi kata lain.

3. Keamanan dari data dan media serta teknik komunikasi

Yang termasuk di dalam kelas ini adalah kelemahan yang digunakan untuk mengelola data. Contohnya seorang kriminal yang menjalankan virus atau trojan horse untuk mengumpulkan informasi (seperti *password*) yang semestinya tidak berhak diakses.

4. Keamanan dalam operasi

Termasuk prosedur yang digunakan untuk mengatur dan mengelola sistem keamanan, dan juga termasuk prosedur setelah serangan (*post attack recovery*).

4.4.3 Aspek/servis dari keamanan

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

1. *Privacy | Confidentiality*

Inti utama aspek privacy atau *confidentiality* adalah usaha untuk menjaga informasi dari orang yang tidak berhak mengakses. *Privacy* lebih kearah data-data yang sifatnya *private* sedangkan *confidentiality* biasanya berhubungan dengan data yang diberikan ke pihak lain untuk keperluan tertentu (misalnya sebagai bagian dari pendaftaran sebuah servis) dan hanya diperbolehkan untuk keperluan tertentu tersebut. Contoh hal yang berhubungan dengan *privacy* adalah e-*mail* seorang pemakai (*user*) tidak boleh dibaca oleh administrator, ini merupakan hal yang sangat penting. Contoh *confidential information* adalah data-data yang sifatnya pribadi (seperti nama, tempat tanggal lahir, *social security number*, agama, status perkawinan, penyakit yang pernah diderita, nomor kartu kredit, dan sebagainya) merupakan data-data yang ingin diproteksi penggunaan dan penyebarannya. Contoh lain dari *confidentiality* adalah daftar pelanggan dari sebuah *Internet Service Provider* (ISP). Untuk mendapatkan kartu kredit, biasanya ditanyakan data-data pribadi.

2. *Integrity*

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

Salah satu contoh kasus trojan horse adalah distribusi paket program TCP *Wrapper* (yaitu program populer yang dapat digunakan untuk mengatur dan membatasi akses TCP/IP) yang dimodifikasi oleh orang yang tidak bertanggung jawab. Jika anda memasang program yang berisi *trojan horse* tersebut, maka ketika anda merakit (*compile*) program tersebut, dia akan mengirimkan e-*mail* kepada orang tertentu yang kemudian memperbolehkan dia masuk ke sistem. Informasi ini berasal dari CERT Advisory, "CA- 99-01 Trojan-TCP-*Wrappers*" yang didistribusikan 21 Januari 1999. Contoh serangan lain adalah yang disebut "*man in the middle attack*" dimana seseorang menempatkan diri di tengah pembicaraan dan menyamar sebagai orang lain.

3. *Authentication*

Aspek ini berhubungan dengan metoda untuk menyatakan bahwa informasi betul-betul asli, orang yang mengakses atau memberikan informasi adalah betul-betul orang yang dimaksud, atau *server* yang kita hubungi adalah betul-betul *server* yang asli. Masalah pertama, membuktikan keaslian dokumen, dapat dilakukan dengan teknologiwatermarking dan digital *signature*. Watermarking juga dapat digunakan untuk menjaga *"intelectual property",* yaitu dengan menandai dokumen atau hasil karya dengan "tanda tangan" pembuat. Masalah kedua biasanya berhubungan dengan *access control*, yaitu berkaitan dengan pembatasan orang yang dapat mengakses informasi. Dalam hal ini pengguna harus menunjukkan bukti bahwa memang dia adalah pengguna yang sah, misalnya dengan menggunakan *password*, *biometric* (ciri-ciri khas orang), dan sejenisnya. Ada tiga hal yang dapat ditanyakan kepada orang untuk menguji siapa dia:

- *What you have* (misalnya kartu ATM)
- *What you know* (misalnya PIN atau *password*)
- *What you are* (misalnya sidik jari, *biometric*).

Penggunaan teknologi *smart card*, saat ini kelihatannya dapat meningkatkan keamanan aspek ini. Secara umum, proteksi authentication dapat menggunakan *digital certificates*. *Authentication* biasanya diarahkan kepada orang (pengguna), namun tidak pernah ditujukan kepada *server* atau mesin.

4. *Availability*

Aspek *availability* atau ketersediaan berhubungan dengan ketersediaan informasi ketika dibutuhkan. Sistem informasi yang diserang atau dijebol dapat menghambat atau meniadakan akses ke informasi. Contoh hambatan adalah serangan yang sering disebut dengan *"denial of service attack"* (DoS *attack*), dimana *server* dikirimi permintaan (biasanya palsu) yang bertubi-tubi atau permintaan yang diluar perkiraan sehingga tidak dapat melayani permintaan lain atau bahkan sampai *down, hang*, *crash*. Contoh lain adalah adanya *mailbomb*, dimana seorang pemakai dikirimi e-mail bertubi-tubi (katakan ribuan e-*mail*) dengan ukuran yang besar sehingga sang pemakai tidak dapat membuka e-*mail* atau kesulitan mengakses e-*mail* (apalagi jika akses dilakukan melalui saluran telepon).

5. *Access Control*

Aspek ini berhubungan dengan cara pengaturan akses kepada informasi. Hal ini biasanya berhubungan dengan klasifikasi data (*public, private, confidential, top secret*) & *user* (*guest*, admin, top manager, dsb.), mekanisme *authentication* dan juga *privacy*. *Access control* seringkali dilakukan dengan menggunakan kombinasi *userid/password* atau dengan menggunakan mekanisme lain (seperti kartu, *biometrics*).

6. *Non-repudiation*

Aspek ini menjaga agar seseorang tidak dapat menyangkal telah melakukan sebuah transaksi. Sebagai contoh, seseorang yang mengirimkan *email* untuk memesan barang tidak dapat menyangkal bahwa dia telah mengirimkan *email* tersebut. Aspek ini sangat penting dalam hal *electronic commerce*. Penggunaan *digital signature*, *certifiates*, dan teknologi kriptografi secara umum dapat menjaga aspek ini. Akan tetapi hal ini masih harus didukung oleh hukum sehingga status dari *digital signature* itu jelas legal.

4.4.4 Penyebab dan masalah dalam sistem keamanan jaringan

Penyebab dan masalah keamanan jaringan yang akan terbagi empat bagian yaitu serangan yang berasal dari luar, serangan dari dalam, *malware* atau virus dan juga kesalahan konfigurasi.

4.4.4.1 Jenis-Jenis Serangan

Serangan Dari Luar Jaringan:

- DOS (*Denial of Service*)

  DoS merupakan serangan yang dilancarkan melalui paket-paket jaringan tertentu, biasanya paket-paket sederhana dengan jumlah yang sangat besar dengan maksud mengacaukan keadaan jaringan target.

- IP *Spoofing*

  IP *Spoofing* juga dikenal sebagai *Source Address Spoofing*, yaitu pemalsuan alamat IP *attacker*, sehingga sasaran menganggap alamat IP *attacker* adalah alamat IP dari *host* di dalam jaringan bukan dari luar jaringan

- *Malware*

  Malware merupakan serangan yang dilakukan ketika *attacker* menaruh program-program penghancur, seperti virus, *worm* dan *trojan* pada sistem sasaran. Program-program penghancur ini sering juga disebut *malware*. Program-program ini mempunyai kemampuan untuk merusak sistem, pemusnahan File, pencurian pasword sampai dengan membuka *backdoor*.

- FTP *Attack*

  Salah satu serangan yang dilakukan terhadap *File Transfer Protocol* adalah serangan *buffer overflow* yang diakibatkan oleh perintah malformed. Tujuan menyerang FTP *server* ini rata-rata adalah untuk mendapatkan command shell ataupun untuk melakukan *Denial Of Service*. Serangan *Denial Of Service* akhirnya dapat menyebabkan seorang *user* atau *attacker* untuk mengambil *resource* di dalam jaringan tanpa adanya autorisasi, sedangkan *command shell* dapat membuat seorang *attacker* mendapatkan akses ke sistem *server* dan

File data yang akhirnya seorang *attacker* bisa membuat *anonymous root-acces* yang mempunyai hak penuh terhadap sistem bahkan jaringan yang diserang

- *Sniffer*

  Adalah suatu usaha untuk menangkap setiap data yang lewat dari suatu jaringan, dapat berupa *password* dan *user* dari pengguna jaringan.

Serangan Dari Dalam Jaringan

- *Password Attack*

  Password Attack adalah usaha penerobosan suatu sistem jaringan dengan cara memperoleh pasword dari jaringan tersebut. *Password* merupakan sesuatu yang umum jika bicara tentang kemanan. Kadang seorang *user* tidak perduli dengan nomor pin yang mereka miliki, seperti bertransaksi *online* di warnet, bahkan bertransaksi online dirumah pun sangat berbahaya jika tidak dilengkapi dengan *software security* seperti SSL dan PGP.

- Merusak *file server*

  ProtoKol-protokol untuk tranportasi data tulang punggung dari *internet* adalah tingkat TCP (TCP Level) yang mempunyai kemampuan dengan mekanisme untuk baca/tulis (*read/write*) antara jaringan dan *host*. *Attacker* bisa dengan mudah mendapatkan jejak informasi dari mekanisme ini untuk mendapatkan akses ke direktori *file*. Tergantung pada OS (*operating system*) yang digunakan, attacker bisa meng extrack informasi tentang jaringan, sharing privileges, nama dan lokasi dari *user* dan groups, dan spesifikasi dari aplikasi atau banner (nama dan versi *software*). Sistem yang dikonfigurasi atau diamankan secara minimal akan dengan mudah membeberkan informasi ini bahkan melalui *firewall* sekalipun. Pada sistem UNIX, informasi ini dibawa oleh NFS (Jaringan *File System*) di *port* 2049. Sistem Windows menyediakan data ini pada SMB (*server messaging block*) dan Netbios pada *port* 135 – 139 (NT) dan port 445 pada win2k.

- *Deface web server*

  Kerawanan yang terdapat dalam HTTPD ataupun *web server* ada lima macam:

- *buffer overflows,*
- *httpd*
- *bypasses,*
- *cross scripting,*
- *web kode vulnerabilities*, dan
- URL *floods*.

HTTPD *Buffer Overflow* bisa terjadi karena attacker menambahkan *error s* pada *port* yang digunakan untuk *web traffic* dengan cara memasukan banyak karakter dan *string* untuk menemukan tempat *overflow* yang sesuai. Ketika tempat untuk *overflow* ditemukan, seorang *attacker* akan memasukkan *string* yang akan menjadi perintah yang dapat dieksekusi. *Bufer-overfl*ow dapat memberikan *attacker* akses ke *command prompt*. Beberapa *feature* dari HTTPD bisa digunakan untuk meciptakan HTTPD *bypass*, memberi akses ke *server* menggunakan fungsi *Logging*. Dengan cara ini, sebuah halaman *web* bisa diakses dan diganti tanpa dicatat oleh *web server*. Cara ini sering digunakan oleh para *cracker, hacktivis* dan *cyber vandals* untuk men*deface website*. Sedangkan kerawanan pada *script web* bisa terjadi pada semua bahasa pemrograman *web* dan semua ekstensi aplikasi. Termasuk VB, Visual C++, ASP, TCL, Perl, PHP, XML, CGI dan Coldfusion. Pada dasarnya, *attacker* akan mengexploitasi kelemahan dari sebuah aplikasi, seperti CGI *script* yang tidak memeriksa input atau kerawanan pada IIS RDS pada showkode.asp yang mengizinkan menjalankan perintah secara *remote* (*remote command priviledges*). Melalui *cross scripting* dan *cross-site scripting* seorang *attacker* bisa mengexploitasi pertukaran *cookies* antara *browser* dan *webserver*. Fasilitas ini dapat mengaktifkan *script* untuk merubah tampilan *web*. *Script* ini bisa menjalankan *malware*, membaca infomasi penting dan mengexpose data sensitive seperti nomor *credit card* dan *pasword*. Pada akhirnya *attacker* dapat menjalankan *denial of service* dengan URL *flood*, yang dilakukan dengan cara mengulang dan terus mengulang permintaan terhadap *port* 80 httpd yang melalui batas TTL (*time to live*).

4.4.4.2 Sumber Lubang Keamanan Jaringan

Meski sebuah sistem jaringan sudah dirancang memiliki perangkat pengamanan, dalam operasi masalah keamanan harus selalu dimonitor. Hal ini disebabkan oleh beberapa hal, antara lain:

- Ditemukannya lubang keamanan (*security hole*) yang baru. Perangkat lunak dan perangkat keras biasanya sangat kompleks sehingga tidak mungkin untuk diuji seratus persen. Kadang-kadang ada lubang, keamanan yang ditimbulkan oleh kecerobohan implementasi.
- Kesalahan konfigurasi. Kadang-kadang karena lalai atau alpa,konfigurasi sebuah sistem kurang benar sehingga menimbulkan lubang keamanan. Misalnya mode (permission atau kepemilikan) dari berkas yang menyimpan *pasword* (/etc/passwd di sistem UNIX) secara tidak sengaja diubah sehingga dapat diubah atau ditulis oleh orang-orang yang tidak berhak.
- Penambahan perangkat baru (*hardware* dan/atau *software*) yang menyebabkan menurunnya tingkat *security* atau berubahnya metoda untuk mengoperasikan sistem. Operator dan administrator harus belajar lagi. Dalam masa belajar ini banyak hal yang jauh dari sempurna, misalnya *server* atau *software* masih menggunakan konfigurasi awal dari vendor (dengan *password* yang sama).

Hal-hal diatas dapat menyebabkan *security hole* (lubang) dalam jaringan.

Sumber lubang keamanan

Lubang keamanan (*security hole*) dapat terjadi karena beberapa hal:

- Salah disain (*design flaw*)
- Salah implementasi
- Salah konfigurasi
- Salah penggunaan program penyerang.

- **Salah Disain**

Lubang keamanan yang ditimbulkan oleh salah disain umumnya jarang terjadi. Akan tetapi apabila terjadi sangat sulit untuk diperbaiki. Akibat disain yang salah, maka biarpun dia diimplementasikan dengan baik, kelemahan dari sistem akan tetap ada.

Contoh sistem yang lemah disainnya adalah lubang keamanan yang dapat dikategorikan kedalam kesalahan disain adalah disain urutan nomor (*sequence numbering*) dari paket TCP/IP. Kesalahan ini dapat dieksploitasi sehingga timbul masalah yang dikenal dengan nama "*IP Spoofing*", yaitu sebuah *host* memalsukan diri seolah-olah menjadi *host* lain dengan membuat paket palsu setelah mengamati urutan paket dari *host* yang hendak diserang. Bahkan dengan mengamati cara mengurutkan nomor *packet* bisa dikenali sistem yang digunakan. Mekanisme ini digunakan oleh program **nmap** dan **queso** untuk mendeteksi *operating system* (OS) dari sebuah sistem, yang disebut *fingerprinting*.

- **Implementasi kurang baik**

Lubang keamanan yang disebabkan oleh kesalahan implementasi sering terjadi. Banyak program yang diimplementasikan secara terburu-buru sehingga kurang cermat dalam pengkodean. Akibatnya cek atau testing yang harus dilakukan menjadi tidak dilakukan. Sebagai contoh, seringkali batas (*bound*) dari sebuah "*array*" tidak dicek sehingga terjadi yang disebut *out-of-bound array* atau *buffer overflow* yang dapat dieksploitasi (misalnya *overwrite* ke variabel berikutnya).

- **Salah konfigurasi**

Meskipun program sudah diimplementasikan dengan baik, masih dapat terjadi lubang keamanan karena salah konfigurasi. Contoh masalah yang disebabkan oleh salah konfigurasi adalah berkas yang semestinya tidak dapat diubah oleh pemakai secara tidak sengaja menjadi writeable. Apabila berkas tersebut merupakan berkas yang penting, seperti berkas yang digunakan untuk menyimpan *password*, maka efeknya menjadi lubang keamanan. Kadangkala sebuah komputer dijual dengan konfigurasi yang sangat lemah. Ada masanya *workstation* Unix di perguruan tinggi didistribusikan dengan berkas /etc/aliases (berguna untuk mengarahkan e-mail), /etc/utmp (berguna untuk mencatat siapa saja yang sedang menggunakan sistem) yang dapat diubah oleh siapa saja.

Contoh lain dari salah konfigurasi adalah adanya program yang secara tidak sengaja diset menjadi "*setuid root*" sehingga ketika dijalankan pemakai memiliki akses seperti

super *user* (*root*) yang dapat melakukan apa saja. Salah menggunakan program atau sistem. Salah penggunaan program dapat juga mengakibatkan terjadinya lubang keamanan. Kesalahan menggunakan program yang dijalankan dengan menggunakan *account root* (*super user*) dapat berakibat fatal. Sering terjadi cerita horor dari sistem administrator baru yang teledor dalam menjalankan perintah "rm -rf" di sistem UNIX (yang menghapus berkas atau direktori beserta sub direktori di dalamnya). Akibatnya seluruh berkas di sistem menjadi hilang mengakibatkan *Denial of Service* (DoS). Apabila sistem yang digunakan ini digunakan bersama-sama, maka akibatnya dapat lebih fatal lagi. Untuk itu perlu berhati-hati dalam menjalan program, terutama apabila dilakukan dengan menggunakan *account administrator* seperti *root* tersebut.

Kesalahan yang sama juga sering terjadi di sistem yang berbasis MS-DOS. Karena sudah mengantuk, misalnya, ingin melihat daftar berkas di sebuah direktori dengan memberikan perintah "dir *.*" ternyata salah memberikan perintah menjadi "del *.*" (yang juga menghapus seluruh *file* di direktori tersebut).

- **Penggunaan program penyerang**

Salah satu cara untuk mengetahui kelemahan sistem informasi anda adalah dengan menyerang diri sendiri dengan paket-paket program penyerang (*attack*) yang dapat diperoleh di *internet*. Dengan menggunakan program ini anda dapat mengetahui apakah sistem anda rentan dan dapat dieksploitasi oleh orang lain. Perlu diingat bahwa jangan menggunakan program-program tersebut untuk menyerang sistem lain (sistem yang tidak anda kelola). Ini tidak etis dan anda dapat diseret ke pengadilan.

Selain program penyerang yang sifatnya agresif melumpuhkan sistem yang dituju, ada juga program penyerang yang sifatnya melakukan pencurian atau penyadapan data. Untuk penyadapan data, biasanya dikenal dengan istilah "*sniffer*". Meskipun data tidak dicuri secara fisik (dalam artian menjadi hilang), *sniffer* ini sangat berbahaya karena dia dapat digunakan untuk menyadap *password* dan informasi yang sensitif. Ini merupakan serangan terhadap aspek *privacy*. Contoh program penyadap (*sniffer*) antara lain:

- *Pcapture*, berjalan pada sistem operasi Unix
- *sniffit*, berjalan pada sistem operasi Unix
- *tcpdump*, berjalan pada sistem operasi Unix
- *WebXRay*, berjalan pada sistem operasi Windows

4.5 Analisis Sistem Keamanan Jaringan

Menganalisis keamanan jaringan perlu dilakukan untuk mengetahui bagaimana status keamanan jaringan itu. Analisis awal terhadap status keamanan jaringan adalah sbb:

1. Vulnerability

Vulnerability adalah suatu aktivitas menganalisis jaringan untuk mengetahui bagian dari sistem yang cenderung/sering untuk diserang (kelemahan-kelemahan pada sistem jaringan). Hal ini akan sangat membantu peningkatan keamanan jaringan dengan mengetahui dan mencatat sistem yang cenderung diserang. *Vulnerability* dapat dilakukan dengan menggunakan beberapa aplikasi yang telah di sebutkan pada bagain Bab 2 dalam dokumen ini tentang Pemantauan Jaringan.

2. Threat

Tujuan analisis ini adalah mengetahui dan mempelajari kemungkinan ancaman atau serangan yang datang dari luar maupun dari dalam jaringan yang dapat merusak pertahanan kemanan jaringan seperti:

- *Destruction* : Usaha untuk merusak sistem pada jaringan, seperti *Trojan horse*, *Logic bom*, *Trap door*, *Virus*, *Worm* dan *Zombie*
- *Denial* : Upaya untuk melumpuhkan kerja suatu *service* dalam jaringan
- *Theft*: Upaya untuk mencuri informasi-informsi penting dalam jaringan.
- *Modification*: Upaya untuk merubah data penting dalam jaringan.
- *Fraud*: Upaya penipuan terhadap suatu sistem informasi seperti *carding*,pemalsuan data, dll.

3. Impact

Menganalisis pengaruh-pengaruh apa saja yang diakibatkan oleh serangan yang terjadi dalam jaringan, seperti *destruction*, *Denial*.

4. Frequency

Menganalisis dan mencatat tingkat keseringan (terjadinya) suatu serangan dalam jaringan dalam kurun waktu tertentu. Contohnya mencatat frekuensi *host* dalam jaringan terkena virus/serangan lain dalam waktu 2 minggu.

5. *Recommended Countermeasures*

Setelah menganalisis dan mencatat beberapa objek diatas, masalah-masalah yang terjadi dalam jaringan dapat dengan mudah diselesaikan dan langkah-langkah pencegahannya. kemudian hasilnya akan menjadi suatu pegangan yang berguna untuk peningkatan kemanan jaringan selanjutnya.

4.5.1 Kontrol & Penyelesaian Masalah Keamanan Jaringan

1. Kontrol

Adapun kontrol-kontrol yang dilakukan untuk mengatasi masalah keamanan jaringan adalah sebagai berikut:

- *Preventive* : pencegahan, misalnya dengan pemisahan tugas staff administrator, sekuriti dan data *entry*
- *Detective* : pendeteksian, misalnya dengan pengecekan ulang, monitoring, dan auditing.
- *Corective*: memperbaiki dan memperkecil dampak ancaman, misalnya :*update* anti virus, melakukan prosedur *backup* dan restorasinya.

2. Penyelesaian masalah

Beberapa cara untuk menyelesaikan masalah keamanan jaringan:

- *Least previlage*
  Orang atau *user* hanya diberikan akses tidak lebih dari yang dibutuhkan
- *Defense in Depth*
  Pertahanan yang berlapis
- *Diversity of Defence*
  Menggunakan beberapa jenis sistem yang berbeda untuk pertahanan
- *Choke point*
  Keluar masuk pada satu gerbang saja

- *Weakest link*

  "sebuah rantai hanya sekuat mata rantai yang paling lemah"
- *Fail-Safe Stance*

  Kalau sebuah perangkat rusak, maka settingnya akan di-set ke yang paling aman secara otomatis
- *Universal participation*

  Semua harus ikut serta
- *Simplicity*

  Harus sederhana agar sistem keamanannya dapat dipahami dengan baik

4.5.2 Audit dan Pemeliharaan Keamanan Jaringan

- *Preventing*

  Preventing dilakukan untuk pencegahan terhadap serangan yang menembusa jaringan. *Tool* yang biasa digunakan untuk melakuakan *preventing* ini adalah *Firewall*
- *Scanning Virus*

  Untuk menghindari kerusakan sistem yang fatal yang disebabkan oleh virus yang ada, maka perlu diadakan *scanning virus* dengan menggunakan anti virus. Setiap saat virus akan berkembang sehingga kita perlu mengupdate antivirus yang kita punya.
- *Monitoring*

  Monitoring dilakukan guna melihat *traffic* yang terjadi pada jaringan. Dengan *monitoring* kita bisa mengetahui apakah terjadi *traffic* yang tidak seperti biasanya, karena apabila ada serangan pada sistem maka biasanya *traffic* akan langsung melonjak tingkat kesibukannya.

  Untuk melakukan *monitoring* kita bisa menggunakan MRTG, Cacti, NTOP
- *Detecting*

  Detecting dilakukan untuk mendeteksi apakah ada usaha ataupun serangan yang bertujuan merusak sistem jaringan. *Tool* yang bisa digunakan untuk melakukan deteksi ini yaitu IDS.
- *Backup*

Mengapa kita perlu mengadakan *backup* ? Apabila suatu saat terjadi *error* pada sistem kita yang memang sudah fatal maka kita diwajibkan untuk melakukan *configurasi* ulang atau *restore*. Untuk menghindari configurasi ulang yang membutuhkan waktu yang tidak singkat maka diadakan *backup* secara berkala (rutin). Sehingga apabila terjadi *error* tadi maka kita hanya perlu me-*restrore* kekeadaan semula dengan menggunakan *backup* tadi.

4.5.3 Perangkat Keamanan Jaringan Yang Umum Digunakan

Ada dua jenis perangkat yang digunakan dalam keamanan jaringan yaitu:

1. Perangkat Keras

- *Firewall*

Secara umum *firewall* biasanya menjalankan fungsi :

- Analisis dan *filter packet*
  Data yang dikomunikasikan lewat protocol di *internet*, dibagi atas paket-paket. *Firewall* dapat menganalilsa paket ini kemudian memberlakukannya sesuai kondisi tertentu.
- *Blocking* dan isi *protocol*
  Firewall dapat melakukan bloking terhadap isi paket, misalnya berisi applet Java, ActiveX, VBScript,dan Cookie
- Autentikasi Koneksi dan enkripsi
  Firewall pada umumnya memiliki kemampuan untuk menjalankan enkripsi dalam identitas *user*, integritas dari suatu session dan melapisi transfer data dari intipan pihak lain. Enkripsi yang dimaksud antara lain DES, Triple DES, SSL, IPSEC, SHA. MD5.

2. Perangkat Lunak

- MRTG (*Multi Router Traffic Grapher Software*).

MRTG akan mengenerate halaman HTML yang menampilkan gambar dalam format PNG dari traffic pada jaringan. MRTG bekerja di sistem operasi UNIX dan Windows.

- *Proxy*

Proxy digunakan untuk membatasi akses *internet* pada lingkup suatu jaringan keamanan. Hal ini dilakukan untuk meminimalisasi terjadinya penyebaran virus pada

jaringan dimana virus itu tanpa kita sadari dapat masuk pada saat kita melakukan *browsing*.

- Anti Virus

Anti virus kita gunakan untuk mendeteksi apakah ada virus pada komputer kita. Karena virus terus bermunculan dan yang diserang makin bervariasi maka kita perlu mengadakan update anti virus sehingga sistem kita lebih terjamin keamanannya dari virus. Sebab ada anti virus yang tidak bisa mendeteksi jenis virus tertentu.

4.6 Mendesain Sistem Keamanan Jaringan

Pada umunya, pengamanan dapat dikategorikan menjadi dua jenis yaitu pencegahan (*preventif*) dan perbaikan (*recovery*). Usaha pencegahan dilakukan supaya sistem informasi tidak memiliki lubang keamanan. Sementara itu usaha-usaha untuk memperbaiki dilakukan setelah lubang keamanan dieksploitasi. Pengamanan sistem informasi dapat dilakukan melalui beberapa layer yang berbeda. Misalnya pada layar "*transport*" menggunakan "*Secure Socket Layer*" (SSL). Metoda ini umum digunakan untuk *server web*. Secara fisik, sistem anda dapat juga diamankan dengan menggunakan "*firewall*" yang memisahkan sistem anda dengan *internet*. Penggunaan teknik enkripsi dapat dilakukan di tingkat aplikasi sehingga data-data anda atau e-mail anda tidak dapat dibaca oleh orang yang tidak berhak.

4.6.1 Keamanan *Host Server*

Seorang administrator yang baik akan menjaga keamanan jaringan dengan baik. Selain itu seorang *administrator* yang baik juga perlu mengamankan *host server* dalam jaringan. Di bawah ini diuraikan cara yang baik yang dilakukan untuk mengamankan *host server* dalam jaringan.

4.6.1.1 Administrasi *Account*

Di dalam masalah keamanan, *server* administrasi adalah masalah yang sangat penting, kenapa? Seorang *user* bisa saja mengobrak-abrik pertahanan *server* walaupun seberapa hebatnya keamanan *sever*. Dalam hal keamanan kita tidak boleh percaya dengan *user* manapun walaupun itu adalah teman sendiri, karena suatu saat si *user* ini akan bisa membobol keamanan jaringan. Sebagai contoh seorang *administrator* berteman dengan si A yang merupakan *user* dalam sistem.

Administrator tersebut adalah *superuser* atau yang memegang sebagai *root*. Karena *administrator* tersebut sangat percaya dengan si A maka dia memberikan *password root* kepada si A. Suatu ketika *administrator* tersebut menyakiti perasaan si A karena suatu hal. Karena si A sakit hati maka si A ingin membalasnya dengan cara mengobrak-abrik *server* yang ditangani oleh *administrator* itu. Padahal kemananan jaringan tersebut terkenal sangat kuat dan sekarang hancur karena hal sekecil ini. Nah, ini adalah gambaran agar kita tidak percaya kepada orang yang sangat kita percaya sekalipun.

Oleh karena itu sebaiknya *superuser* dan *group administrator* tidak diberikan kepada sembarang orang. Masalah lainnya yang umum adalah masalah *user non administrator*. Untuk *user* yang sudah tidak digunakan lagi lebih baik dihapus. Ini digunakan untuk memperkecil kemungkinan penyerang yang masuk ke dalam sistem dan untuk memudahkan mengontrol *user* yang masih aktif.

4.6.1.2 *Administrasi Pasword*

Mendengar kata *pasword* sudah pasti berhubungan dengan sesuatu yang sangat rahasia, yang bahayanya lagi kita lebih suka mengetahui rahasia orang lain. *Administrasi pasword* sangat dibutuhkan untuk menghindari celah keamanan yang memungkinkan untuk dibobol oleh orang yang tidak bertanggung jawab. Masalah yang sering ditemukan adalah *user* yang tidak memiliki *password*, kebanyakan si *user* malas untuk menghafalkan *password* untuk *account*-nya. Ini sangatlah berbahaya, karena penyerang bisa saja memanfaatkan *user* yang tidak ber-*password* untuk sarana masuk kedalam sistem. Oleh karena itu tugas seorang *administrator* yang baik adalah mengecek tiap-tiap *user*. Jika ditemukan ada *user* yang tidak memiliki *pasword* secepatnya diberitahu untuk membuat *password*. Jika teguran tersebut tidak dihiraukan sebaiknya *administrator* menghapus account-nya.

Password sangatlah penting, maka dari itu pastikan *password* pada sistem anda tidak boleh diakses oleh *user* lain. Alangkah lebih baik jika pasword super *user* diganti secara berkala. Sebagai contoh: minggu ini menggunakan *password* : ”p45c4l”, kemudian minggu depan sudah harus ganti dengan *password* lain, misal: ”4k3upm4n”. Hal ini sangat penting, untuk menghindari pengaksesan oleh *user* lain yang mengetahui *password* lama.

Saran untuk pembuatan *password*:

1. Buatlah *password* sesulit mungkin tapi mudah untuk dihafal, kalau bisa gunakan kombinasi antara huruf dan karakter ini sangat ampuh untuk mempersulit si penyerang.
2. Menset batas berlakunya *password*.
3. Menggunakan *password* secara berkala.

4.6.1.3 Administrasi Akses

Administrasi akses yang dimaksudkan adalah administrasi pada direktori maupun *file* penting yang perlu dijaga agar tidak dapat diakses oleh *user* lain. Usahakan selalu *file* atau direktori anda tidak bisa diakses oleh orang lain sekalipun itu orang yang sangat anda percaya.

4.6.1.4 Administrasi Layanan

Kebanyakan penyerang melakukan penyerangan melalui fasilitas yang satu ini. *Server* memiliki banyak *port* yang terbuka ketika layanan suatu layanan dibuka. Jika sistem anda menggunakan layanan *Web Server* dan *Mail Server* maka sebaiknya cukup kedua layanan ini saja yang dibuka. Makin banyak *port* yang terbuka maka makin besar kemungkinan *server* diserang. Pada umumnya penyerang akan melakukan *scanning* sebelum melakukan penyerangan.

Hal penting lainnya adalah memastikan bahwa *program server* yang dijalankan benar-benar aman, dengan kata lain sebaiknya *administrator* harus rajin-rajin meng-*update program server*. Ini dikarenakan program *server* terkadang memiliki *bug* yang suatu saat bisa diekslpoitasi oleh penyerang untuk memperoleh akses. Menggunakan aplikasi dalam *server* yang memiliki fasilitas enkripsi untuk *transfer* data. Misalnya SSH (*Secure Shell*) untuk telnet, dan Apache + SSL untuk www.

4.6.1.5 Administrasi *Log File*

Setiap kegiatan pada sistem pada umumnya sudah otomatis terekam pada *Log File*. Tugas dari administrator adalah memeriksa *Log File* sesering mungkin untuk melihat

setiap kegiatan-kegiatan yang terjadi. Jika ditemukan kegiatan-kegiatan yang mencurigakan, misalnya upaya *login* berulang-ulang maka itu adalah upaya penyeran untuk masuk ke dalam sistem. Beberapa program dapat memonitor jaringan dan mendeteksi kalau ada hal-hal yang mencurigakan.

4.6.2 Mengatur akses (*Access Control*)

Salah satu cara yang umum digunakan untuk mengamankan informasi adalah dengan mengatur akses ke informasi melalui mekanisme "*authentication*" dan "*access control*". Implementasi dari mekanisme ini antara lain dengan menggunakan "*userid*" dan "*password*". Informasi yang diberikan ini dibandingkan dengan *userid* dan *password* yang berada di sistem. Apabila keduanya valid, pemakai yang bersangkutan diperbolehkan menggunakan sistem. Apabila ada yang salah, pemakai tidak dapat menggunakan sistem. Informasi tentang kesalahan ini biasanya dicatat dalam berkas *Log*. Besarnya informasi yang dicatat bergantung kepada konfigurasi dari sistem setempat. Misalnya, ada yang menuliskan informasi apabila pemakai memasukkan userid dan *password* yang salah sebanyak tiga kali. Ada juga yang langsung menuliskan informasi ke dalam berkas *Log* meskipun baru satu kali salah. Informasi tentang waktu kejadian juga dicatat. Selain itu asal hubungan (*connection*) juga dicatat sehingga administrator dapat memeriksa keabsahan hubungan.

Setelah proses *authentication*, pemakai diberikan akses sesuai dengan level yang dimilikinya melalui sebuah *access control*. *Access control* ini biasanya dilakukan dengan mengelompokkan pemakai dalam "*group*". Ada *group* yang berstatus pemakai biasa, ada tamu, dan ada juga administrator atau *super user* yang memiliki kemampuan lebih dari *group* lainnya. Pengelompokan ini disesuaikan dengan kebutuhan dari penggunaan sistem anda. Di lingkungan kampus mungkin ada kelompok mahasiswa, staf, karyawan, dan administrator. Sementara itu di lingkungan bisnis mungkin ada kelompok *finance*, *engineer*, *marketing*, dan seterusnya.

4.6.3 Menutup servis yang tidak digunakan

Seringkali pada suatu sistem (perangkat keras dan/atau perangkat lunak) terdapat servis yang dijalankan sebagai default. Sebagai contoh, pada sistem UNIX servis-

servis berikut sering dipasang dari vendornya: finger, telnet, ftp, smtp, pop, echo, dan seterusnya. Servis tersebut tidak semuanya dibutuhkan. Untuk mengamankan sistem, servis yang tidak diperlukan di *server* (komputer) tersebut sebaiknya dimatikan. Sudah banyak kasus yang menunjukkan abuse dari servis tersebut, atau ada lubang keamanan dalam servis tersebut akan tetapi sang administrator tidak menyadari bahwa servis tersebut dijalankan di komputernya.

Servis-servis di sistem UNIX ada yang dijalankan dari "inetd" dan ada yang dijalankan sebagai daemon. Untuk mematikan servis yang dijalankan dengan menggunakan fasilitas inet, periksa berkas /etc/inetd.conf, matikan servis yang tidak digunakan (dengan memberikan tanda komentar #) dan memberitahu inetd untuk membaca berkas konfigurasinya (dengan memberikan signal HUP kepada PID dari proses inetd). Contoh:

```
unix# ps -aux | grep inetd 105 inetd
unix# kill -HUP 105
```

4.6.4 Memasang Proteksi

Untuk lebih meningkatkan keamanan sistem informasi, proteksi dapat ditambahkan. Proteksi ini dapat berupa filter (secara umum) dan yang lebih spesifik adalah *firewall*. *Firewall* dapat digunakan untuk memfilter e-mail, informasi, akses, atau bahkan dalam level packet. Sebagai contoh, di sistem UNIX ada paket program "tcpwrapper" yang dapat digunakan untuk membatasi akses kepada servis atau aplikasi tertentu. Misalnya, servis untuk "telnet" dapat dibatasi untuk untuk sistem yang memiliki nomor IP tertentu, atau memiliki domain tertentu. Sementara *firewall* dapat digunakan untuk melakukan filter secara umum. Untuk mengetahui apakah server anda menggunakan tcpwrapper atau tidak, periksa isi berkas /etc/inetd.conf. Biasanya tcpwrapper dirakit menjadi "tcpd". Apabila servis di *server* anda (misalnya telnet atau ftp) dijalankan melalui tcpd, maka *server* anda menggunakan tcpwrapper. Biasanya, konfigurasi tcpwrapper (tcpd) diletakkan di berkas /etc/hosts.allow dan /etc/hosts.deny.

4.6.5 *Firewall*

Firewall merupakan sebuah perangkat yang diletakkan antara *internet* dengan jaringan internal. Informasi yang keluar atau masuk harus melalui *firewall* ini. Tujuan utama dari *firewall* adalah untuk menjaga (*prevent*) agar akses (ke dalam maupun ke luar) dari orang yang tidak berwenang (*unauthorized access*) tidak dapat dilakukan. Konfigurasi dari *firewall* bergantung kepada kebijaksanaan (*policy*) dari organisasi yang bersangkutan, yang dapat dibagi menjadi dua jenis:

- apa-apa yang tidak diperbolehkan secara eksplisit dianggap tidak diperbolehkan (*prohibitted*).
- apa-apa yang tidak dilarang secara eksplisit dianggap diperbolehkan (*permitted*).

Firewall bekerja dengan mengamati paket IP (*Internet Protocol*) yang melewatinya. Berdasarkan konfigurasi dari *firewall* maka akses dapat diatur berdasarkan IP *address*, *port*, dan arah informasi. Detail dari konfigurasi bergantung kepada masing-masing *firewall*. *Firewall* dapat berupa sebuah perangkat keras yang sudah dilengkapi dengan perangkat lunak tertentu, sehingga pemakai (administrator) tinggal melakukan konfigurasi dari *firewall* tersebut. *Firewall* juga dapat berupa perangkat lunak yang ditambahkan kepada sebuah *server* (baik UNIX maupun Windows NT), yang dikonfigurasi menjadi *firewall*. Dalam hal ini, sebetulnya perangkat komputer dengan prosesor Intel 80486 sudah cukup untuk menjadi *firewall* yang sederhana. *Firewall* biasanya melakukan dua fungsi; fungsi (IP) *filtering* dan fungsi *proxy*. Keduanya dapat dilakukan pada sebuah perangkat komputer (*device*) atau dilakukan secara terpisah. Beberapa perangkat lunak berbasis UNIX yang dapat digunakan untuk melakukan IP *filtering* antara lain:

- ipfwadm: merupakan standar dari sistem Linux yang dapat diaktifkan pada level kernel
- ipchains: versi baru dari Linux kernel *packet filtering* yang diharapkan dapat menggantikan fungsi ipfwadm

Fungsi *proxy* dapat dilakukan oleh berbagai *software* tergantung kepada jenis *proxy* yang dibutuhkan, misalnya *web proxy,* rlogin *proxy, ftp proxy* dan seterusnya. Di sisi

client sering kalai dibutuhkan software tertentu agar dapat menggunakan *proxy server* ini, seperti misalnya dengan menggunakan SOCKS. Beberapa perangkat lunak berbasis UNIX untuk *proxy* antara lain:

- Socks: *proxy server* oleh NEC *Network Sistems Labs*
- Squid: *web proxy server*

Satu hal yang perlu diingat bahwa adanya *firewall* bukan menjadi jaminan bahwa jaringan dapat diamankan seratus persen. *Firewall* tersebut sendiri dapat memiliki masalah. Sebagai contoh, *Firewall* Gauntlet yang dibuat oleh *Network Associates* Inc. (NAI) mengalami masalah1 sehingga dapat melewatkan koneksi dari luar yang seharusnya tidak boleh lewat. Padahal Gauntlet didengung-dengungkan oleh NAI sebagai "*The World's Most Secure Firewall*". Inti yang ingin kami sampaikan adalah bahwa meskipun sudah menggunakan *firewall*, keamanan harus tetap dipantau secara berkala.

4.6.6 Pemantau adanya serangan

Sistem pemantau (*monitoring* sistem) digunakan untuk mengetahui adanya tamu tak diundang (*intruder*) atau adanya serangan (*attack*). Nama lain dari sistem ini adalah "*intruder detection system*" (IDS). Sistem ini dapat memberitahu administrator melalui *email* maupun melalui mekanisme lain seperti melalui pager. Ada berbagai cara untuk memantau adanya *intruder*. Ada yang sifatnya aktif dan pasif. IDS cara yang pasif misalnya dengan memonitor *LogFile*. Contoh *software* IDS yang digunakaan di sistem operasi Linux antara lain:

- *Autobuse*, mendeteksi probing dengan memonitor LogFile.
- *Courtney* dan *portsentry*, mendeteksi *probing* (*port scanning*) dengan memonitor packet yang lalu lalang. *Portsentry* bahkan dapat memasukkan IP penyerang dalam filter tcpwrapper (langsung dimasukkan kedalam berkas /etc/hosts.deny)
- *Shadow* dari SANS
- *Snort*, mendeteksi pola (*pattern*) pada paket yang lewat dan mengirimkan *alert* jika pola tersebut terdeteksi. Pola-pola atau rules disimpan dalam berkas yang disebut *library* yang dapat dikonfigurasi sesuai dengan kebutuhan.

4.6.7 Pemantau integritas sistem

Pemantau integritas sistem dijalankan secara berkala untuk menguji integratitas sistem. Salah satu contoh program yang umum digunakan di sistem UNIX adalah program Tripwire. Program paket Tripwire dapat digunakan untuk memantau adanya perubahan pada berkas. Pada mulanya, tripwire dijalankan dan membuat database mengenai berkas-berkas atau direktori yang ingin kita amati beserta "*signature*" dari berkas tersebut.

Signature berisi informasi mengenai besarnya berkas, kapan dibuatnya, pemiliknya, hasil *checksum* atau *hash* (misalnya dengan menggunakan program MD5), dan sebagainya. Apabila ada perubahan pada berkas tersebut, maka keluaran dari *hash function* akan berbeda dengan yang ada di database sehingga ketahuan adanya perubahan.

4.6.8 *Audit*: Mengamati Berkas *Log*

Segala (sebagian besar) kegiatan penggunaan sistem dapat dicatat dalam berkas yang biasanya disebut "*LogFile"* atau "*Log*" saja. Berkas *Log* ini sangat berguna untuk mengamati penyimpangan yang terjadi. Kegagalan untuk masuk ke sistem (*Login*), misalnya, tersimpan di dalam berkas log. Untuk itu para administrator diwajibkan untuk rajin memelihara dan menganalisis berkas log yang dimilikinya. Letak dan isi dari berkas log bergantung kepada *operating system* yang digunakan. Di sistem berbasis UNIX, biasanya berkas ini berada di direktori /var/adm atau /var/log.

4.6.9 *Backup* secara rutin

Seringkali tamu tak diundang (*intruder*) masuk ke dalam sistem dan merusak sistem dengan menghapus berkas-berkas yang dapat ditemui. Jika *intruder* ini berhasil menjebol sistem dan masuk sebagai *super user* (*administrator*), maka ada kemungkinan dia dapat menghapus seluruh berkas. Untuk itu, adanya *backup* yang dilakukan secara rutin merupakan sebuah hal yang esensial. Bayangkan apabila yang dihapus oleh tamu ini adalah berkas penelitian, tugas akhir, skripsi, yang telah dikerjakan bertahun-tahun. Untuk sistem yang sangat esensial, secara berkala perlu dibuat backup yang letaknya berjauhan secara fisik. Hal ini dilakukan untuk

menghindari hilangnya data akibat bencana seperti kebakaran, banjir, dan lain sebagainya. Apabila data-data di-*backup* akan tetapi diletakkan pada lokasi yang sama, kemungkinan data akan hilang jika tempat yang bersangkutan mengalami bencana seperti kebakaran.

4.6.10 Penggunaan Enkripsi untuk meningkatkan keamanan

Salah satu mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang anda kirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Banyak servis di *internet* yang masih menggunakan "*plain text*" untuk authentication, seperti penggunaan pasangan *userid* dan *password*. Informasi ini dapat dilihat dengan mudah oleh program penyadap atau pengintersepsi (*sniffer*). Contoh servis yang menggunakan *plain text* antara lain:

- akses jarak jauh dengan menggunakan telnet dan rlogin
- transfer *file* dengan menggunakan FTP
- akses *email* melalui POP3 dan IMAP4
- pengiriman *email* melalui SMTP
- akses *web* melalui HTTP

4.6.11 Telnet atau shell

Telnet atau *remote login* digunakan untuk mengakses sebuah "*remote site* " atau komputer melalui sebuah jaringan komputer. Akses ini dilakukan dengan menggunakan hubungan TCP/IP dengan menggunakan *userid* dan *password*. Informasi tentang *userid* dan *password* ini dikirimkan melalui jaringan komputer secara terbuka. Akibatnya ada kemungkinan seorang yang nakal melakukan "*sniffing*" dan mengumpulkan informasi tentang pasangan *userid* dan *password* ini. Untuk menghindari hal ini, enkripsi dapat digunakan untuk melindungi adanya sniffing. Paket yang dikirimkan dienkripsi dengan algoritma DES atau Blowish (dengan menggunakan kunci *session* yang dipertukarkan via RSA atau Diffie-Hellman) sehingga tidak dapat dibaca oleh orang yang tidak berhak. Salah satu implementasi mekanisme ini adalah SSH (*Secure Shell*). Ada beberapa implementasi SSH ini, antara lain:

- ssh untuk UNIX (dalam bentuk *source code*, gratis, mengimplementasikan protokol SSH versi 1 dan versi 2)
- SSH untuk Windows95 dari Data Fellows (komersial, ssh versi 1 dan versi 2)
- TTSSH, yaitu skrip yang dibuat untuk Tera Term Pro (gratis, untuk Windows 95, ssh versi 1)
- SecureCRT untuk Windows95 (*shareware* / komersial)

putty (SSH untuk Windows yang gratis, ssh versi 1). Selain menyediakan ssh, paket putty juga dilengkapi dengan pscp yang mengimplementasikan *secure copy* sebagai pengganti FTP.

4.6.12 Keamanan *Workstation* Dalam Jaringan

Beberapa masalah yang sering terjadi pada *workstation* dalam jaringan adalah sbb:

1. Penambahan *User Account*

Penambahan *user account* di suatu komputer yang bukan merupakan hak seorang *user* dapat dilakukan dengan beberapa cara sbb:

- Pemanfaatan SQL *Server* 2000

Beberapa komputer dalam jaringan telah diinstal aplikasi SQL *Server* 2000. Seorang pengguna mungkin tidak sadar bahwa melalui SQL *Server* 2000 seorang *user* lain dalam jaringan dapat menambahkan *account user* dengan *privilage* sebagai administrator/*Super User*. Hal ini dilakukan dengan menjalankan beberapa baris perintah pada editor SQL pada komputer tujuan setelah melakukan koneksi sebelumnya melalui komputer lain. Pada saat melakukan koneksi, database SQL akan meminta *password* agar dapat mengkases database yang ada di dalamnya, biasanya default *user* adalah ”su” dan *password*-nya adalah ”su”.

Perintah yang dijalankan adalah:

Net user user_name user_pasword /add
Net localgroup administrators user_name /add

Dengan menjalankan kedua perintah diatas, maka sebuah *user* dengan *privilage* administrator akan ditambahkan ke komputer tujuan.

Penanggulangan.

Sebenarnya sangat sederhana untuk mencegah terjadinya hal diatas, yaitu merubah *passowrd user* su pada SQL *Sever* atau merubah *user* dan *passowd* defaultnya.

- Menggunakan *software* tertentu.

Salah satu contoh *softwar*e yang dapat menambah *user account* dengan *privilage* sesuka hati kita adalah KAHT. Program ini berjalan pada Windows 2000/XP. Saat aplikasi ini dijalankan pada *Command Prompt* di Windows, dia akan melacak alamat-alamat ip yang telah ditentukan sebelumnya. Bila alamat IP yang dilacak membuka *port* 135, maka secara otomatis direktori kerja kita akan berada pada direktori komputer yang diserang dengan *privilage* sebagai *administrator*. Dengan demikian pengguna akan secara leluasa menambahkan sebuah *user account*, bahkan merusak sistem komputer tersebut.

Penanggulangan.

Hal yang dilakukan untuk mengatasi hal diatas adalah menginstal SP (*Service Pack*) 2 pada PC Windows.

2. Virus

- Langkah-Langkah untuk pencegahan

  Untuk pencegahan anda dapat melakukan beberapa langkah-langkah berikut :

  - Gunakan *antivirus* yang benar-benar dipercayai dengan update terbaru. Tidak perduli apapun merknya asalkan selalu di-*update*, dan *auto-protect* dinyalakan maka komputer (*workstation*) akan terlindungi.
  - Selalu *scanning* semua media penyimpanan eksternal yang akan digunakan, mungkin hal ini agak merepotkan tetapi jika *auto-protect* antivirus anda bekerja maka prosedur ini dapat dilewatkan.
  - Jika anda terhubung langsung ke *internet* cobalah untuk mengkombinasikan antivirus anda dengan *Firewall, Anti-spamming*, dsb.
  - Selalu waspada terhadap *fle* yang mencurigakan, contoh: File dengan 2 buah *exstension* atau *file* executable yang terlihat mencurigakan.
  - Untuk *software freeware* + *shareware*, ada baiknya anda mengambilnya dari situs resminya.

- Semampunya hindari membeli barang bajakan, gunakan *software open source*.

- Langkah-Langkah apabila telah terinfeksi
  - Deteksi dan tentukan dimanakah kira-kira sumber virus tersebut apakah di disket, jaringan, *email* dsb.
  - Jika anda terhubung ke jaringan maka ada baiknya anda mengisolasi komputer anda dulu (baik dengan melepas kabel atau mendisable sambungan *internet* dari *control panel*).
  - Identifikasi dan klasifikasikan jenis virus apa yang menyerang pc anda, dengan cara:

    * Gejala yang timbul, misal : pesan, *file* yang *corrupt* atau hilang dsb

    * *Scan* dengan antivirus anda, jika anda terkena saat auto-protect berjalan berarti virus definition di dalam komputer anda tidak memiliki data virus ini, cobalah update secara manual atau men-*download virus definition*-nya untuk kemudian anda instal. Jika virus tersebut memblok usaha anda untuk mengupdate, maka upayakan untuk menggunakan media lain (komputer) dengan antivirus yang memiliki update terbaru.
  - Bersihkan virus tersebut. Setelah anda berhasil mendeteksi dan mengenalinya maka usahakan segera untuk mencari *removal* atau cara-cara untuk memusnahkannya di situs-situs yang memberikan informasi perkembangan virus tersebut. Hal ini perlu dilakukan apabila antivirus dengan update terbaru anda tidak berhasil memusnahkannya.
  - Langkah terburuk. Jika semua hal diatas tidak berhasil adalah memformat ulang komputer anda.

4.6.13 *Monitoring*/ Pendeteksian Jaringan

Seorang penyusup/*attacker* yang mencoba masuk ke dalam jaringan dapat dideteksi dengan berbagai program/aplikasi. Monitoring/pendeteksian ini sangat penting untuk menjaga sistem jaringan tetap berada pada kondisi aman dan terkendali.

Berikut adalah beberapa perangkat lunak bantu yang bisa digunakan untuk pendeteksi penyusup :

1. ***Portsentry*.**

   Sebuah program bantu yang cukup "ringan" dan tidak begitu sulit untuk mengkonfigurasikan dan menggunakannya. Cocok untuk sistem jaringan kecil.

2. ***Snort*.**

   Program bantu ini berfungsi memeriksa data-data yang masuk dan melaporkan ke administrator apabila ada "gerak-gerik" yang mencurigakan. Bekerja dengan prinsip program sniffer yaitu mengawasi paket-paket yang melewati jaringan.

3. **LIDS (Linux *Intrussion Detection System*)**

   merupakan salah satu *tools* IDS yang sangat baik dalam melindungi sistem. Ketika lids aktif, maka bahkan *root* sekalipun mempunyai akses yang sangat terbatas sekali dalam mengkonfigurasikan sistem.

4. ***Carnivore*.**

   Sebenarnya *tools* ini lebih bisa dianggap sebagai sniffer daripada IDS. Dikembangkan di amerika, kini *Carnivore* oleh FBI dipasang di semua server yang berfungsi sebagai tulang-punggung (*backbone*) *internet* yang ada di Amerika. Sehingga secara tidak langsung Amerika telah menyadap semua data yang lewat dari seluruh penjuru dunia. Perlu diketahui bahwa hampir semua server utama atau backbone yang ada di dunia ini berlokasi di Amerika Serikat.

   Dan masih banyak tools untuk IDS lainnya yang dapat digunakan untuk lebih meningkatkan keamanan sistem.

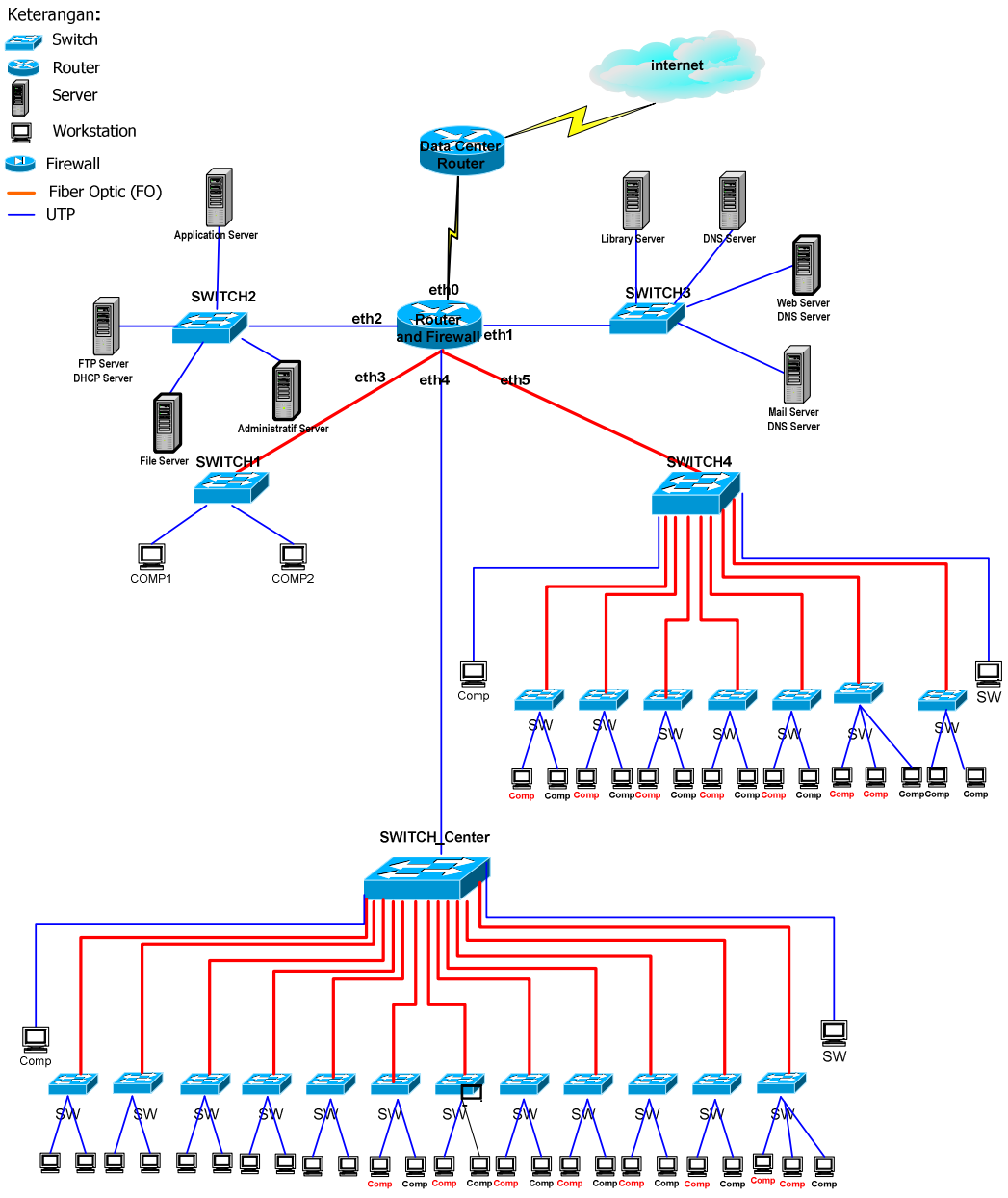
Beberapa informasi tentang *security* yang dapat diperoleh di *internet*:

- http://www.linuxsecurity.com
- http://securityfocus.com
- http://www.cert.org
- http://atrittion.org
- http://packetstorm.securify.com
- http://www.karet.org
- http://www.securitylinux.net
- http://www.securityportal.com

4.6.14 Topologi Jaringan

Topologi jaringan yang paling umum digunakan dalam mendesain jaringan adalah topologi *Extended Star*. Sesuai namanya topologi ini merupakan perluasan atau gabungan dari beberapa topologi star. Topologi ini digunakan untuk memberi gambaran jaringan yang akan dibangun dan mempermudah perluasan jaringan di waktu yang akan datang.

Secara fisik, desain topologi jaringan yang akan dapat dilihat seperti gambar di bawah ini:



Gambar 1 Topologi Jaringan

DAFTAR PUSTAKA

- Overview of Network Security. Budi Rahardjo, 2002.
- *Website*:
  - http://en.wikipedia.org/wiki/Secure_Sockets_Layer
  - http://en.wikipedia.org/wiki/Intrusion-detection_system
  - http://en.wikipedia.org/wiki/Intrusion_prevention_system
  - http://pangea.standord.edu/computerinfo/network/security
  - http://www.tasscc.org/presentations/tec_2004/Donaho-Jaeger.ppt.
  - http:// ilkom.del.ac.id
  - http://www.cisco.com/application/vnd.mspowerpoint/en/us/guest/products/ps5477/c116
  - http://ilkom.del.ac.id
  - http://en.wikipedia.org/
  - http://www.microsoft.com/
  - http://technet2.microsoft.com/