BAB II

LANDASAN TEORI

A. Jaringan Komputer

Jaringan komputer adalah himpunan "interkoneksi" antara 2 komputer atau lebih yang terhubung dengan media transmisi kabel atau tanpa kabel (*wireless*). Dua unit komputer dikatakan terkoneksi apabila keduanya bisa saling bertukar data/informasi, berbagi *resource* yang dimiliki seperti file, printer, media penyimpanan (harddisk, floopy disk, cd-rom, flash disk, dll). Data yang berupa teks, audio maupun video bergerak melalui media kabel atau tanpa kabel sehingga memungkinkan pengguna komputer dalam jaringan komputer dapat saling bertukar file/data, mencetak pada printer yang sama dan menggunakan *hardware*/*software* yang terhubung dalam jaringan secara bersama-sama.

(Melwin Syafrizal, 2005)

B. Interkoneksi

Interkoneksi adalah koneksi antara satu jaringan penyedia layanan telekomunikasi dengan jaringan penyedia layanan telekomunikasi lainnya atau koneksi antara suatu alat telepon dengan jaringan telepon nasional.

(Tim Penelitian dan Pengembangan Wahana Komputer, 2004)

C. Model Keamanan Sistem Informasi

Terdapat beberapa model dalam mengamankan sistem informasi pada suatu jaringan, antara lain:

1. Mengatur akses

   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 "*password*". Untuk menggunakan sebuah sistem atau komputer, pemakai diharuskan melalui proses *authentication* dengan menuliskan "*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*.

   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". Di lingkungan kampus mungkin ada kelompok mahasiswa, staf, karyawan, dan administrator.

2. Menutup *service* yang tidak digunakan

   Seringkali sistem (perangkat keras dan/atau perangkat lunak) diberikan dengan beberapa *service* dijalankan sebagai *default*. Sebagai contoh, pada sistem UNIX *service-service* berikut sering dipasang dari

vendornya: *telnet, ftp, smtp* dan seterusnya. *Service* tersebut tidak semuanya dibutuhkan. Untuk mengamankan sistem, *service* yang tidak diperlukan di server (komputer) tersebut sebaiknya dimatikan. Sudah banyak kasus yang menunjukkan ada lubang keamanan dalam *service* tersebut akan tetapi sang administrator tidak menyadari bahwa *service* tersebut dijalankan di komputernya.

3. Memasang proteksi

Untuk lebih meningkatkan keamanan sistem informasi, proteksi dapat ditambahkan. Proteksi ini dapat berupa *filter* (secara umum) dan yang lebih spesifik adalah *firewall*. *Filter* dapat digunakan untuk memfilter e-mail, informasi, akses, atau bahkan dalam level packet. *Service* 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.

a. Firewall

1) Pengertian 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, sebuah *firewall* diimplementasikan dalam sebuah mesin terdedikasi yang berjalan pada pintu gerbang (*gateway*) antara jaringan lokal dan jaringan lainnya. Firewall umumnya juga

digunakan untuk mengontrol akses terhadap siapa saja yang memiliki akses terhadap jaringan pribadi dari pihak luar.

2) Jenis-Jenis Firewall

Firewall terbagi menjadi dua jenis, yakni sebagai berikut:

a) Personal Firewall

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 semacam perangkat proteksi terhadap virus, *anti-spyware*, *anti-spam* dan lainnya. Bahkan beberapa produk firewall lainnya dilengkapi dengan fungsi pendeteksian gangguan keamanan jaringan (*Intrusion Detection System*). Contoh dari firewall jenis ini adalah Microsoft Windows Firewall (yang telah terintegrasi dalam sistem operasi Windows XP SP2, Windows Vista dan Windows Server 2003 SP1), Symantec Norton Personal Firewall, Kerio Personal Firewall, dan lain-lain.

b) Network Firewall

Network Firewall didesain untuk melindungi jaringan secara keseluruhan dari berbagai serangan. Umumnya dijumpai dalam dua bentuk yakni sebuah perangkat terdedikasi atau

sebagai sebuah *software* yang diinstalasikan dalam sebuah server. Contoh dari firewall ini adalah Microsoft Internet Security and Acceleration Server (ISA Server), Cisco PIX, Cisco ASA, IPTables dalam sistem operasi GNU/Linux, pf dalam keluarga sistem operasi Unix BSD, serta SunScreen dari Sun Microsystems, Inc. yang dibundel dalam sistem operasi Solaris. Network Firewall umumnya bersifat transparan (tidak terlihat) dari pengguna dan menggunakan teknologi routing untuk menentukan paket mana yang diizinkan dan mana paket yang akan ditolak.

3) Fungsi Firewall

Firewall dapat melakukan hal-hal berikut:

a) Mengatur dan mengontrol lalu lintas jaringan

Firewall harus dapat mengatur dan mengontrol lalu lintas jaringan yang diizinkan untuk mengakses jaringan *private* atau komputer yang dilindungi oleh firewall. Firewall melakukan hal yang demikian, dengan melakukan inspeksi terhadap paket-paket dan memantau koneksi yang sedang dibuat lalu melakukan penapisan (*filtering*) terhadap koneksi berdasarkan hasil inspeksi paket dan koneksi tersebut.

b) Melakukan autentikasi terhadap akses

Protokol TCP/IP dibangun dengan premis bahwa protokol tersebut mendukung komunikasi yang terbuka. Jika dua host saling mengetahui alamat IP satu sama lainnya, maka mereka diizinkan untuk saling berkomunikasi. Pada awal-awal perkembangan Internet, hal ini boleh dianggap sebagai suatu berkah. Tapi saat ini, di saat semakin banyak yang terhubung ke Internet, mungkin kita tidak mau siapa saja yang dapat berkomunikasi dengan sistem yang kita miliki. Karenanya, firewall dilengkapi dengan fungsi autentikasi dengan menggunakan beberapa mekanisme autentikasi, sebagai berikut:

i. Firewall dapat meminta input dari pengguna mengenai nama pengguna (*username*) serta kata kunci (*password*). Metode ini sering disebut sebagai extended authentication atau xauth. Menggunakan xauth pengguna yang mencoba untuk membuat sebuah koneksi akan diminta *input* mengenai nama dan kata kuncinya sebelum akhirnya diizinkan oleh firewall. Umumnya, setelah koneksi diizinkan oleh kebijakan keamanan dalam firewall, firewall pun tidak perlu lagi mengisikan *input password* dan namanya kecuali jika

koneksi terputus dan pengguna mencoba menghubungkan dirinya kembali.

ii. Metode kedua adalah dengan menggunakan sertifikat digital dan kunci publik. Keunggulan metode ini dibandingkan dengan metode pertama adalah proses autentikasi dapat terjadi tanpa intervensi pengguna. Selain itu, metode ini lebih cepat dalam rangka melakukan proses autentikasi. Meskipun demikian, metode ini lebih rumit implementasinya karena membutuhkan banyak komponen seperti halnya implementasi infrastruktur kunci publik.

iii. Metode selanjutnya adalah dengan menggunakan Pre-Shared Key (PSK) atau kunci yang telah diberitahu kepada pengguna. Jika dibandingkan dengan sertifikat digital, PSK lebih mudah diimplementasikan karena lebih sederhana, tetapi PSK juga mengizinkan proses autentikasi terjadi tanpa intervensi pengguna. Dengan menggunakan PSK, setiap host akan diberikan sebuah kunci yang telah ditentukan sebelumnya yang kemudian digunakan untuk proses autentikasi. Kelemahan metode ini adalah kunci PSK jarang sekali diperbarui dan banyak organisasi sering sekali menggunakan kunci yang sama untuk melakukan

koneksi terhadap *host-host* yang berada pada jarak jauh, sehingga hal ini sama saja meruntuhkan proses autentikasi. Agar tercapai sebuah derajat keamanan yang tinggi, umumnya beberapa organisasi juga menggunakan gabungan antara metode PSK dengan xauth atau PSK dengan sertifikat digital.

Dengan mengimplementasikan proses autentikasi, firewall dapat menjamin bahwa koneksi dapat diizinkan atau tidak.

c) Melindungi sumber daya dalam jaringan *private*

Proteksi ini dapat diperoleh dengan menggunakan beberapa peraturan pengaturan akses (*access control*), penggunaan SPI, *application* proxy atau kombinasi dari semuanya untuk mencegah *host* yang dilindungi dapat diakses oleh *host-host* yang mencurigakan atau dari lalu lintas jaringan yang mencurigakan. Meskipun demikian, firewall bukanlah satu-satunya metode proteksi terhadap sumber daya dan mempercayakan proteksi terhadap sumber daya dari ancaman terhadap firewall adalah salah satu kesalahan fatal. Jika sebuah *host* yang menjalankan sistem operasi tertentu yang memiliki lubang keamanan yang belum ditambal dikoneksikan ke Internet, firewall mungkin tidak dapat mencegah dieksploitasinya *host* tersebut oleh

host-host lainnya, khususnya jika exploit tersebut menggunakan lalu lintas yang oleh firewall telah diizinkan (dalam konfigurasinya).

d) Mencatat semua kejadian dan melaporkan kepada administrator

4) Application Level Firewall

Application Level Gateway (atau *Application-Level Firewall* atau sering juga disebut sebagai *Proxy Firewall*), yang umumnya juga merupakan komponen dari sebuah proxy server. Firewall ini tidak mengizinkan paket yang datang untuk melewati firewall secara langsung. Tetapi, aplikasi proxy yang berjalan dalam komputer yang menjalankan firewall akan meneruskan permintaan tersebut kepada layanan yang tersedia dalam jaringan *private* dan kemudian meneruskan respon dari permintaan tersebut kepada komputer yang membuat permintaan pertama kali yang terletak dalam jaringan publik yang tidak aman.

Umumnya, firewall jenis ini akan melakukan autentikasi terlebih dahulu terhadap pengguna sebelum mengizinkan pengguna tersebut untuk mengakses jaringan. Selain itu, firewall ini juga mengimplementasikan mekanisme auditing dan pencatatan (*logging*) sebagai bagian dari kebijakan keamanan yang diterapkannya. Application Level Firewall juga

umumnya mengharuskan beberapa konfigurasi yang diberlakukan pada pengguna untuk mengizinkan mesin *client* agar dapat berfungsi. Sebagai contoh, jika sebuah proxy FTP dikonfigurasikan di atas sebuah application layer gateway, proxy tersebut dapat dikonfigurasikan untuk mengizinkan beberapa perintah FTP dan menolak beberapa perintah lainnya. Jenis ini paling sering diimplementasikan pada proxy SMTP sehingga mereka dapat menerima surat elektronik dari luar (tanpa menampakkan alamat e-mail internal), lalu meneruskan e-mail tersebut kepada e-mail server dalam jaringan. Tetapi, karena adanya pemrosesan yang lebih rumit, firewall jenis ini mengharuskan komputer yang dikonfigurasikan sebagai application gateway memiliki spesifikasi yang tinggi.

Satu hal yang perlu diingat bahwa adanya firewall bukan menjadi jaminan bahwa jaringan dapat diamankan seratus persen. Firewall tersebut sendiri dapat memiliki masalah. (Budi Raharjo, 2004)

(id.wikipedia.org/wiki/Firewall, 2007)

b. Sistem Pemantau

Sistem pemantau (*monitoring system*) 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 e-mail maupun melalui mekanisme lain.

c. Backup

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.

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.

d. Penggunaan Enkripsi

Salah satu mekanisme untuk meningkatkan keamanan adalah dengan menggunakan teknologi enkripsi. Data-data yang anda kirimkan diubah sedemikian rupa sehingga tidak mudah disadap. Banyak *service* 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 pengendus (*sniffer*).

Contoh *service* yang menggunakan *plain text* antara lain:

1) Akses jarak jauh dengan menggunakan telnet dan rlogin.
2) *Transfer* file dengan menggunakan FTP.
3) Akses email melalui POP3 dan IMAP4.
4) Pengiriman e-mail melalui SMTP.
5) Akses web melalui HTTP.

D. Protocol

Protokol adalah sebuah aturan atau standar yang mengatur atau mengijinkan terjadinya hubungan, komunikasi, dan perpindahan data antara dua atau lebih titik komputer. Protokol dapat diterapkan pada perangkat keras, perangkat lunak atau kombinasi dari keduanya. Pada tingkatan yang terendah, protokol mendefinisikan koneksi perangkat keras.

Protokol perlu diutamakan pada penggunaan standar teknis, untuk menspesifikasi bagaimana membangun komputer atau menghubungkan peralatan perangkat keras. Protokol secara umum digunakan pada komunikasi *real-time* dimana standar digunakan untuk mengatur struktur dari informasi untuk penyimpanan jangka panjang.

E. Port

Port dalam arti bahasa Indonesianya adalah lubang. Tetapi kalau dalam komputer, port diasumsikan sebagai pintu *service*. Misalnya di komputer A membuka port 21 maka kita tahu bahwa komputer A membuka diri untuk

sebuah pelayanan ftp (*file transfer protocol*) sehingga komputer lain baik dalam jaringan internal maupun jaringan yang berbeda dapat mengakses komputer tersebut melalui port yang terbuka tersebut.

Berikut ini adalah beberapa daftar standar untuk port:

No	Protocol	Nama Port	Keterangan
21	TCP	ftp	Protokol untuk menangani pentransferan file dari satu komputer ke komputer lainnya. Protokol ini menggunakan dua saluran, yang satu sebagai saluran untuk mengendalikan lalu-lintas data, sedangkan yang satunya lagi untuk dilintasi oleh data itu sendiri. Proses autentikasi berdasar login/password. FTP adalah metode paling sering digunakan di situs-situs web untuk mengirim file (seringkali akses 'ftp only' diberikan kepada para pelanggan webhosting). Siapapun yang berhasil login akan diberikan hak yang sama berdasar id (jadi boleh menulis ke direktori 'home' milik user tsb), atau bahkan di banyak server, boleh membaca/tulis direktori milik pengguna lain. Eksploit-eksploit yang beredar termasuk untuk program-program wu_ftpd, ncftpd, ftpbounce, dll.
80/8080	TCP	www/http	Protokol pentransferan data berformat HTML (*webpage-webpage*) lainnya dari server kepada publik. Port ini sangat populer dijadikan sasaran para cracker (terutama ditargetkan oleh program-program pelacak kelemahan CGI), sebab banyak program CGI yang bisa dieksploit. Mesin-mesin yang menyediakan *service* 'Frontpage Extensions' secara default terkenal berlubang besar, dan di-'patch'. Ada juga serangan dari pengunjung yang mencoba mengakses direktori-direktori yang terlindung password (contohnya

			di situs-situs porno) dengan cara mem-*brute force* proses autentikasi (*Username* dan *Password*), sebab serangan semacam ini tidak direkam oleh *log* server.

F. Internet Protocol (IP)

IP adalah bangunan blok Internet. Fungsinya yaitu :

1. Menentukan datagram yang merupakan unit dasar transmisi data di Internet.
2. Menentukan skema pengalamatan Internet.
3. Memindahkan data diantara lapisan akses network dan lapisan transport *host* ke *host*.
4. Melakukan *routing* datagram ke *host* jauh (*remote host*).
5. Membuat fragmentasi (pemecahan data menjadi serpihan data) dan menyatukan ulang datagram.

Karakteristik IP

1. Merupakan protokol yang tidak harus tersambung (*connectionless protocol*). Artinya IP tidak mengontrol pertukaran informasi dalam menyelenggarakan sambungan antar komputer sebelum ada komunikasi data. Sebaliknya pada protokol yang berorientasi pada sambungan (*connection oriented protocol*) akan mengontrol informasi pertukaran data dengan sistem yang berjauhan (*remote system*) untuk memverifikasi apakah itu sudah siap menerima data sebelum data dikirim kepadanya. Pada saat

sambungan terhubung dengan baik, sistem akan memberi kabar bahwa sambungan sudah terjadi.

2. IP tidak memberikan pengecekan *error* dan perbaikan *error* ke lapisan lainnya. Karena itu IP juga disebut sebagai protokol yang tidak baik (*unreliable protocol*). Tapi bukan berarti IP tidak bisa merupakan protokol seperti itu. IP dapat menyelenggarakan pengiriman data dengan akurat ke dalam jaringan, tetapi IP tidak dapat memastikan apakah data itu sudah diterima dengan baik atau tidak. Untuk keperluan ini dilakukan oleh protokol pada lapisan lainnya.

G. World Wide Web

World Wide Web atau sering dikenal dengan istilah *WWW* atau singkatnya *WEB* dikembangkan pada tahun 1990 di CERN (laboratorium Fisika Partikel) yang bertempat di Swiss. Web adalah suatu ruang informasi di mana sumber-sumber daya yang berguna dapat diakses apabila telah memenuhi tiga standar yaitu the *Uniform Resource Identifier* (URI), the *HyperText Transfer Protocol* (HTTP) dan the *HyperText Markup Language* (HTML).

1. *Uniform Resource Identifier*

   URI adalah rangkaian karakter menurut suatu format standar tertentu yang digunakan untuk menunjukkan alamat suatu sumber seperti dokumen dan gambar di Internet. Tujuan utama dari pengidentifikasian ini adalah untuk membuka interaksi dengan representasi (penyajian) sumber daya yang melalui jaringan komputer khususnya Web. Contoh dari *sintax*

URI antara lain “http”, “ftp”, “mailto” dan lain sebagainya. Sejak 1994, konsep *Uniform Resource Locator* (URL) telah dikembangkan menjadi istilah URI yang lebih umum sifatnya. Walaupun demikian, istilah URL masih tetap digunakan secara luas.

2. *HyperText Transfer Protocol*

HTTP (*HyperText Transfer Protocol*) adalah protokol yang dipergunakan untuk mentransfer dokumen dalam web. Protokol ini adalah protokol ringan, tidak berstatus dan generik yang dapat dipergunakan berbagai macam tipe dokumen.

Pengembangan HTTP dikoordinasi oleh Konsorsium World Wide Web (W3C) dan grup bekerja *Internet Engineering Task Force* (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC 2616, yang menjelaskan HTTP/1.1, versi HTTP yang digunakan umum sekarang ini.

HTTP adalah sebuah protokol meminta/menjawab antara *client* dan server. Sebuah *client* HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu *client* mengirim kode permintaan (*request*), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikuti dengan badan dari data tertentu. Beberapa

kepala (*header*) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban dan sebuah pesan yang diminta atau sebuah pesan *error* atau pesan lainnya.

(id.wikipedia.org/wiki/HTTP, 2007)

Dua hal khusus yang membedakan layanan Web dengan layanan lainnya, yaitu:

a. Informasi di Web dapat ditampilkan dalam bentuk multimedia yang berupa grafik, suara, video disamping tulisan teks.
b. Informasi di Web dapat menghubungkan (*link*) ke informasi atau dokumen atau alamat Internet lainnya lewat *hypertext*. *Hypertext* merupakan teks yang ditampilkan dengan *font* yang berbeda misalnya dengan huruf miring, digarisbawahi dan sebagainya. Untuk menggunakan layanan Web, diperlukan sebuah program web browser seperti Mozilla Firefox, Internet Explorer, Netscape dan lain sebagainya.

(Jogiyanto Hartono, 2002)

3. *HyperText Markup Language*

HTML adalah sebuah bahasa *markup* yang dipergunakan untuk membuat sebuah halaman web dan menampilkan berbagai informasi di dalam sebuah *browser* Internet.

H. Server

Server adalah sebuah sistem komputer yang menyediakan jenis layanan tertentu dalam sebuah jaringan komputer. Server didukung dengan *processor* yang bersifat *scalable* dan RAM yang besar, juga dilengkapi dengan sistem operasi khusus, yang disebut sebagai sistem operasi jaringan atau *network operating system*. Server juga menjalankan perangkat lunak administratif yang mengontrol akses terhadap jaringan dan sumber daya yang terdapat di dalamnya, seperti halnya berkas atau alat pencetak (printer), dan memberikan akses kepada workstation anggota jaringan.

Umumnya, di atas sistem operasi server terdapat aplikasi-aplikasi yang menggunakan arsitektur *client*/*server*. Contoh dari aplikasi ini adalah HTTP Server, DNS Server dan lain sebagainya. Setiap sistem operasi server umumnya membundel layanan-layanan tersebut atau layanan tersebut juga dapat diperoleh dari pihak ketiga. Setiap layanan tersebut akan merespon terhadap *request* dari *client*.

Fungsi server sangat banyak, misalnya untuk situs Internet, ilmu pengetahuan atau sekedar penyimpanan data. Namun yang paling umum adalah untuk mengkoneksikan komputer *client* ke Internet.

(id.wikipedia.org/wiki/Server, 2007)

I. FreeBSD

FreeBSD adalah sistem operasi yang berdasarkan *Berkeley Software Distribution* (BSD) versi 4.4BSD-Lite dan dapat berjalan pada komputer varian Intel (x86 dan Itanium), AMD64 dan Sun UltraSPARC.

FreeBSD memiliki banyak fitur-fitur penting, antara lain:

1. *Preemptive multitasking*

   Penyesuaian prioritas yang dinamis untuk memastikan pembagian yang adil pada sebuah komputer antara pengguna dan aplikasi walaupun pada kondisi beban penuh.

2. *Multi-user fasilities*

   Mengijinkan banyak pengguna untuk menggunakan FreeBSD secara bersama-sama

3. *Strong TCP/IP Networking*

   Men-*support* standar industri seperti SLIP, PPP, NFS, DHCP dan NIS. Hal ini menunjukkan bahwa FreeBSD mudah berkomunikasi dengan sistem lain sebaik server *enterprise*.

4. *Memory Protection*

   Aplikasi atau pengguna yang menggunakan FreeBSD secara bersama-sama tidak saling mengganggu.

5. dan lain sebagainya.

Beberapa situs pemakai FreeBSD sebagai server antara lain:

1. Yahoo! (www.yahoo.com)
2. Apache (www.apache.org)
3. Blue Mountain Arts (www.bluemountain.com)
4. Pair Networks (www.pair.com)
5. Sony Japan (www.sony.co.jp)
6. Netcraft (www.netcraft.com)

7. WeatherNews (www.wni.com)
8. Supervalu (www.supervalu.com)
9. TELEHOUSE America (www.telehouse.com)
10. Sophos Anti-Virus (www.sophos.com).
11. JMA Wired (www.jmawired.com), dan lain-lain.

(The FreeBSD Documentation Project, 1999)

J. Web Server

Web Server merupakan sebuah perangkat lunak maupun sistem komputer yang berfungsi menerima permintaan HTTP dari *client* yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman-halaman web yang umumnya berbentuk dokumen HTML.

Apache HTTP Server

Server HTTP Apache atau Server Web Apache adalah perangkat lunak server web dengan lisensi gratis yang dapat dijalankan di banyak sistem operasi seperti Unix, BSD, Linux, Microsoft Windows dan Novell Netware serta platform lainnya yang berguna untuk melayani dan memfungsikan situs web. Protokol yang digunakan untuk melayani fasilitas web ini menggunakan HTTP.

Apache merupakan server web yang paling banyak digunakan di dunia. Berdasarkan hasil survei Netcraft.com, dari 113.658.468 situs pada bulan April tahun 2007 menunjukkan bahwa Server Web Apache digunakan lebih dari 66.900.000 situs di dunia.

K. Proxy Server

Proxy Server adalah sebuah sistem komputer atau program aplikasi yang melayani permintaan dari *client* dengan meminta layanan ke server lain. *Client* menghubungi proxy server, meminta file, koneksi internet, halaman web atau *resource* lain yang tersedia oleh server lain.

Beberapa kemampuan proxy server antara lain:

1. Connection Sharing

   Pada proxy, pengguna tidak langsung berhubungan dengan jaringan luar atau internet tetapi harus melewati suatu gateway yang bertindak sebagai batas antara jaringan lokal dan jaringan luar. Koneksi dari jaringan lokal ke internet akan menggunakan sambungan yang dimiliki oleh gateway secara bersama-sama (*connection sharing*). Dalam hal ini, gateway adalah juga sebagai proxy server karena menyediakan layanan sebagai perantara antara jaringan lokal dan jaringan luar atau internet.

2. Caching Proxy Server

   Sebuah proxy server dapat melayani permintaan dari *client* tanpa perlu menghubungi komputer server yang dimaksud dengan cara menyimpan data dari server tersebut yang diperoleh dari permintaan *client* sebelumnya. Hal ini disebut dengan *caching*. Caching proxy menyimpan data yang sering diminta oleh *client* ke suatu tempat penyimpanan.

3. Web Proxy

   Proxy yang terfokus pada *traffic* WWW disebut sebagai Web Proxy. Banyak Web Proxy yang berusaha untuk mem-*block content* web yang berbahaya. Operator jaringan juga dapat membuat proxy untuk melewatkan virus-virus komputer dan *content-content* berbahaya dari suatu situs yang diakses.

4. Intercepting proxy server

   Sebuah Intercepting Proxy Server menggabungkan proxy server dengan sebuah *gateway*. Koneksi yang diciptakan oleh *browser client* yang melewati *gateway* diubah melalui proxy tanpa adanya konfigurasi dari sisi *client*.

(en.wikipedia.org/wiki/Proxy_server, 2007)

Keuntungan menggunakan proxy server:

1. Dapat menghemat biaya *bandwidth*.
2. Mempercepat koneksi karena file-file web yang di *request* (selanjutnya disebut *object*) disimpan di dalam *cache* sehingga tidak perlu keluar menuju Internet.
3. Dapat mengatur kecepatan *bandwidth* untuk subnet yang berbeda.
4. Dapat melakukan pembatasan untuk file-file tertentu dan situs-situs tertentu.
5. Dapat melakukan pembatasan *download* untuk file-file tertentu (misalnya file-file mp3, wav, dsb).
6. Dapat melakukan pembatasan waktu-waktu untuk *download*.

7. Dapat melakukan pembatasan siapa saja yang boleh mengakses Internet dengan menggunakan autentikasi.
8. Dapat melakukan pembatasan-pembatasan lainnya.

(Aulia Ahmad, 2006)

Squid Proxy Server

Squid web-cache proxy server adalah *software* proxy server yang bersifat *open-source* yang didesain untuk berjalan di sistem Unix dan keluarganya (termasuk Linux). Squid tidak hanya dapat meng-*cache* objek-objek web saja, namun juga dapat meng-*cache* DNS dan *network lookup* lainnya. Meskipun pada awalnya didesain untuk sistem Unix, namun Squid dapat pula men-*support* Windows NT, namanya menjadi SquidNT.

Konfigurasi Dasar Squid

1. *http_port*

   Port HTTP yang didengarkan oleh Squid. Defaultnya adalah 3128. Biasanya port yang umum untuk sebuah proxy server adalah 8080.

2. *acl src ipaddress/netmask*

   Access Control List untuk alamat network asal. Biasanya digunakan untuk mengidentifikasi subnet yang digunakan *user*. ACL ini bisa berupa alamat *network* dan subnet mask atau alamat IP address tertentu saja.

3. *acl dst ipaddress/netmask*

   ACL untuk alamat *network* yang dituju oleh *user*/*client*.

4. *acl srcdomain .foo.com*

   ACL untuk nama domain asal.

5. *acl dstdomain .foo.com*

   ACL untuk nama domain tujuan.

6. *acl srcdom_regex [-i] xxx*

   ACL domain asal yang difilter oleh sebuah regular expression.

7. *acl dstdom_regex [-i] xxx*

   ACL domain tujuan yang difilter oleh sebuah regular expression.

8. *acl time [singkatan-hari] [h1:m1-h2:m2]*

   ACL untuk mendefinisikan waktu. Singkatan hari didefinisikan sebagai:

   – S: Sunday

   – M: Monday

   – T: Tuesday

   – W: Wednesday

   – H: Thursday

   – F: Friday

   – A: Saturday

9. *acl url_regex [-i] ^http://..*

   URL yang difilter dengan regular expression didefinisikan dalam ACL ini.

10. *acl urllogin [-i]*

    URL yang memakai autentikasi difilter dengan regular expression yang didefinisikan dalam ACL ini.

11. *acl port*

    Definisi port yang dituju oleh *client*.

12. *acl proto*

    Protokol yang digunakan oleh *client*, misalnya FTP, HTTP.

13. *acl method*

    Method yang digunakan oleh *client*, misalnya GET, POST.

14. *acl browser [-i] regex*

    Jika Anda ingin memfilter browser yang digunakan oleh *client*, Anda dapat menggunakan ACL ini dengan menambahkan regular expression di belakangnya.

15. *acl ident username*

    ACL untuk mendefinisikan *user* yang *login* di Squid. Untuk itu Anda harus menerapkan fungsi autentikasi di Squid.

16. *acl proxy_auth username*

    ACL untuk autentikasi user. Gunakan REQUIRED pada *username* untuk menerima *username* yang *valid*.

17. *acl maxconn*

    ACL untuk maksimum koneksi yang digunakan oleh satu *host* yang melakukan koneksi ke internet lewat proxy server.

18. *icp_port*

Port yang digunakan Squid untuk melakukan kerjasama dengan Squid yang lain. Secara default, Squid bekerjasama pada port 3130.

Beberapa jenis autentikasi pada Squid Proxy Server

1. NCSA_Auth

   Program ncsa_auth akan memanfaatkan data user dan password yang dibuat oleh program htpasswd. Program htpasswd merupakan salah satu program yang dipaketkan bersama instalasi web server Apache.

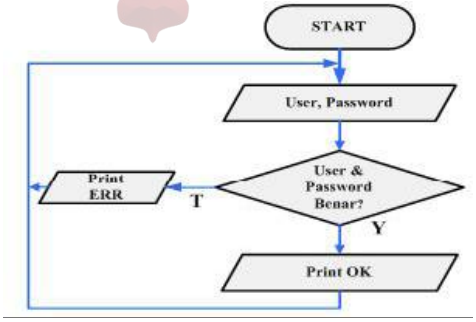
2. Squid_ldap_auth

   Program squid_ldap_auth akan memanfaatkan data-data di server LDAP yang didalamnya menyimpan nama user dan password.

3. MySQL_Auth

   Program mysql_auth merupakan program dalam bahasa C yang dibuat sendiri dengan memanfaatkan cara kerja Squid saat memeriksa nama user dan password yang berhak. Mysql_auth memanfaatkan database MySQL dalam pengelolaan nama user dan password.

Alur program auth



Gambar 2.1. Flowchart Program Auth

Program akan menghasilkan output berupa teks OK jika autentikasi berhasil dan output berupa teks ERR jika autentikasi gagal. Hasil berupa pesan OK akan diterima Squid sebagai tanda dibukanya hak untuk mengakses Internet. Hasil berupa pesan ERR diterima oleh Squid sebagai tanda gagalnya hak untuk mengakses Internet.