PERBANDINGAN CARA KERJA PACKET FILTERING DAN PROXY SERVICES SEBAGAI FIREWALL PADA KEAMANAN JARINGAN

ABSTRAK

Salah satu implementasi perkembangan teknologi informasi adalah pembuatan jaringan kecil Local Area Network, di mana terjadi hubungan antara satu mesin dengan mesin lain di suatu tempat untuk menjadi satu kesatuan jaringan kecil. Penggunaan jaringan dibutuhkan agar informasi dan data dari satu tempat ke tempat lain disampaikan dengan cepat. Keamanan jaringan merupakan masalah yang penting. Data atau file yang akan dikirim biasanya dibungkus dengan beberapa protokol yang telah terdapat pada jaringan itu sendiri, tetapi hanya beberapa protokol. Protokol-protokol yang tersusun ketika mengirimkan data atau file sering dikenal dengan alamat IP atau Internet Protokol, Protokol yang digunakan ini terdapat pada setiap komputer yang tersambung dengan suatu jaringan. Diperlukan suatu sistem untuk mengamankan file-file dan data-data yang tersimpan pada jaringan lokal yang telah dibuat pada suatu tempat dengan memperlihatkan sistem keamanan melalui packet filtering. Sistem keamanan tersebut akan dibandingkan dengan fasilitas keamanan lain yaitu proxy services. Hasil perbandingan menunjukkan bahwa penggunaan proxy services lebih ditujukan kepada komputer pribadi dan packet filtering merupakan pilihan yang paling tepat untuk sebuah jaringan yang dibangun pada dasar jaringan lokal.

Kata Kunci: LAN, Packet Filtering, Protokol, Proxy

Rodiah

Fakultas Teknologi Industri
Universitas Gunadarma
Jl. Margonda Raya No.100 Pondok Cina Depok
email : rodiah@staff.gunadarma.ac.id

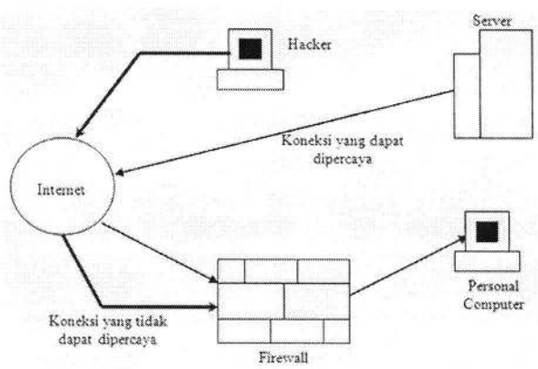
PENDAHULUAN

Keamanan jaringan komputer berfungsi melindungi investasi dan sumber daya dalam suatu jaringan secara efektif. Perencanaan kebijaksanaan keamanan suatu jaringan dilihat dari risiko kemungkinan penyusup berhasil mengakses komputer ke dalam jaringan yang dilindungi, ancaman adanya orang yang ingin memperoleh akses illegal ke dalam suatu jaringan komputer, dan kelemahan yang merupakan gambaran dari seberapa kuat sistem keamanan suatu jaringan komputer terhadap jaringan komputer lain dan kemungkinan bagi seseorang untuk mendapat akses illegal ke dalamnya (, , 2002).

Selain itu harus dilihat terlebih dahulu insiden-insiden yang dapat terjadi, antara lain *probe* (usaha yang tidak lazim untuk memperoleh akses ke dalam suatu sistem atau secara elektronik), *probing ekuivalen* (menggunakan *handle* pintu untuk mencari pintu yang tidak terkunci untuk masuk ke dalam suatu ruangan), dan *scan* (kegiatan *probe* dalam jumlah besar dengan menggunakan *tool* secara otomatis). *Tool* yang dipakai dapat secara otomatis mendeteksi kelemahan pada *host local* maupun *host remote* tanpa memperhatikan jarak, *account compromise* (penggunaan account suatu komputer secara illegal oleh orang yang bukan pemilik account, di mana account mempunyai posisi sebagai user), *root compromise* (sama dengan *account compromise*, tapi bedanya account mempunyai sebagai administrator system), *denial of service* (penolakan karena jaringan tidak berfungsi karena kebanjiran data/ jaringan dipartisi dengan membuat komponen jaringan yang menjadi penghubung jaringan tidak berfungsi/ada virus yang menyebar dan menyebabkan sistem komputer menjadi lambat bahkan lumpuh, *packet sniffer* (suatu *device*, perangkat lunak atau perangkat keras, yang digunakan untuk memperoleh informasi yang melewati jaringan komputer yang menggunakan protokol apa saja; *device* tersebut membuat NIC, dalam hal ini Ethernet, dalam mode *promiscuous*, yaitu keadaan sedang 'mendengarkan' semua trafik termasuk dari *workstation* lain sehingga dapat menangkap semua trafik dalam jaringan), dan *malicious code* yaitu suatu program yang bila dieksekusi akan menyebabkan sesuatu yang tak diinginkan di dalam sistem, termasuk *Trojan Horse*, Virus dan *Worm* dan *Internet infrastructure attacks*. Insiden ini jarang terjadi karena penyerangannya mencakup komponen-komponen pokok dari infrastruktur internet bahkan mencakup sistem yang khusus dari internet (Elizabeth, Simon Cooper, D. Brent, 2000).

Firewall merupakan komponen atau suatu set dari beberapa komponen yang melarang suatu akses antara *network* yang terlindungi dan internet, atau antara set-set dari *network* yang lain. Dengan kata lain Firewall merupakan suatu adaptasi modern dari sistem-sistem keamanan dari abad pertengahan yang telah mengalami perkembangan sesuai dengan kebutuhan dari kebanyakan pengguna. Firewall mempunyai dua komponen penting yaitu *Router* dan *Application Gateway* (Mark Grennan, 2000). *Router* adalah hardware yang mempunyai software sendiri untuk membangun suatu benteng yang menjadi pertahanan untuk jaringan, sedangkan *Appliction Gateway* adalah software khusus yang digunakan untuk mengamati paket yang keluar dan masuk.

Kemampuannya dalam menjalankan keamanan terdiri atas *packet filtering* dan *proxy services*. *Packet filtering* adalah aksi yang dilakukan oleh suatu alat atau software yang secara ketat mengontrol pemilihan aliran dari suatu paket yang berisi informasi yang didapat dari atau berasal dari suatu jaringan. *Proxy services* adalah program yang menangani segala kegiatan dengan menjadikan server eksternal (luar) sebagai perwakilan dari klien internal (dalam). Gambar 1 memperlihatkan secara singkat kegunaan dari sistem keamanan dengan Firewall

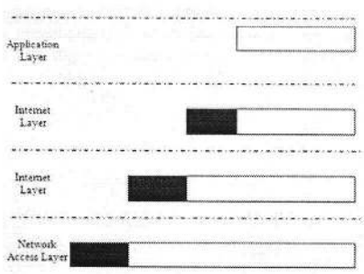


Gambar 1. Sistem Keamanan Dengan Firewall

Sistem keamanan merupakan konsep fundamental dari sudut pandang kebutuhan dan kegunaannya karena perkembangan sistem jaringan yang semakin merambah di masyarakat kita maupun masyarakat secara global. Dewasa ini internet bukan lagi hal yang biasa saja melainkan suatu kebutuhan yang penting. Posisi dari sistem keamanan sangat bergantung pada pengguna. Firewall yang dibahas pada penulisan ini merupakan konsep yang telah berkembang sejak pertama kali ditemukannya internet. Firewall memerlukan 3 buah point penting yang menyusun firewall itu sendiri yaitu protokol, paket dan *bastion host* (Mark Grennan, 2000).

Dalam menyampaikan informasi di suatu jaringan, sebuah paket dikonstruksi sedemikian rupa sehingga layer-layer untuk setiap protokol yang digunakan di dalamnya untuk koneksi tertentu terkunci dan membungkus di sekitar paket seperti lapisan kulit pada bawang. Paket-paket ini terdiri dari bagian kepala dan bagian badan. Bagian kepala berisi protokol-protokol untuk memuat informasi yang masih berhubungan dengan layer yang membungkus paket, sedangkan bagian badannya berisi data pada layer yang akan dikirim. Dengan kata lain, informasi yang ingin kita sampaikan pada komputer lain dalam suatu jaringan, tetapi terkadang bagian ini berisikan keseluruhan paket dari layer yang lain dalam *stack* (Elizabeth at al, 2000).

Gambar 2 memperlihatkan layer-layer yang berfungsi sebagai kepala dalam suatu paket dan ada yang berfungsi sebagai badannya. Layer-layer sangat penting untuk membangun suatu paket. Pada setiap layer, paket mengikutsertakan seluruh informasi yang diturunkan dari layer di atasnya sebagai data dan membuat sendiri bagian kepalanya untuk data itu sehingga informasi yang dari satu unit komputer ke unit komputer yang lain tidak hilang. Proses akan dijelaskan dengan mengamankan data sementara itu membuat bagian kepala yang baru ini disebut *Encapsulation*, seperti yang terlihat pada Gambar 2.



Gambar 2. Skema Data Encapsulation

METODE PENELITIAN

DNS yang sekarang sering digunakan mengedit file yang terdapat pada direktori WINDOWS yaitu HOSTS. Penggunaan DNS menghemat tenaga untuk menduplikasi dan mendistribusikan database kepada semua pemakai yang melakukan login ke dalam domain yang dibuat. Pada software DNS dapat dicari direktori atau folder bertuliskan \RESKIT (Resource Kit) dan menjalankan INSTALL.BAT pada folder tersebut yang, ketika dijalankan, akan langsung menginstall sekaligus menduplikasi file ke direktori C:\WINDOWS\SYSTEM32\ DRIVERS.

Setelah menginstal software DNS dan kegiatan *copy* file ke sistem perlu dilakukan konfigurasi ulang karena adanya server DNS primer dan DNS sekunder. Server DNS primer merupakan server TCP/IP yang mengelola file-file BOOT, CACHE, Netid.REV, DHCP.MDB, DHCP.TMP, JET.LOG dan JET*.LOG, dan System.MDB.

BOOT, sebagai file konfigurasi utama, mendeklarasikan file-file yang digunakan untuk menginisialisasikan server DNS dengan menggunakan tiga directive yaitu *directory*, *primary* dan *cache*. Bentuk dari file sebagai contoh:

```
;directory entry not required. Files in default directory
;directory \windows\system32\drivers
primary     depok.edu                   depok.dom
primary     154.76.200.in-addr.arpa     200.76.154.rev
primary     0.0127.in-addr.arpa         127.0.0.rev

cache       .                           cache.dom
```

File 127.0.0.REV berisi data *lookup* kebalikan untuk alamat IP pada jaringan 127 *(loopback)* seperti halnya localhost. Domain.DOM, di setiap domain yang dikelola server DNS terdapat file *lookup reverse* yang diperlukan untuk menentukan pemetaan nama ke alamat.

CACHE file merupakan database untuk setiap *host* yang memberikan konektivitas dasar kepada DNS. Dengan kata lain pemetaan dari pemetaan yang terdapat pada jaringan dari *root* yang menjadi server sampai dengan tujuan. Untuk mengetahui server nama *root* yang resmi dapat dicari dengan menggunakan FTP, Gopher dan e-mail (Dhiraj Bhagchandka, 2003). Sedangkan Netid.REV pada setiap jaringan netid yang dikelola server DNS terdapat file *lookup reverse* yang diperlukan untuk menentukan pemetaan alamat ke nama.

Server DNS sekunder memperoleh data-data dari server nama lain melalui proses yang dikenal sebagai *zone transfer* sehingga memerlukan file BOOT, CACHE.DOM dan 127.0.0.REV karena server DNS yang kedua ini juga merupakan *back-up* agar kegagalan satu server nama tidak menyebabkan gangguan resolusi nama bagi domain sendiri.

Setelah syarat untuk satu *scope* terpenuhi, klien dapat meminta sebuah *lease* yang mengijinkan alamat IP tertentu untuk digunakan oleh klien. *Scope* merupakan range alamat IP berikut kumpulan *option* konfigurasi yang dapat digunakan oleh klien yang menerima alamat IP dari *scope*. Durasi penggunaan dari *lease* ditentukan oleh kebutuhan dari jaringan.

Bila jumlah kumpulan alamat yang tersedia lebih besar dari jumlah *host* yang memerlukan alamat maka durasi *lease* dapat dibuat cukup lama, tetapi untuk jangka waktu yang tak terbatas tidak dianjurkan karena setiap jaringan pasti mengalami perubahan dalam bentuk tertentu dan untuk menentukan durasi dari *lease* yang sudah tersimpan lama pasti akan dilepaskan.

Untuk konfigurasi jaringan yang sering berubah dan jumlah pemakai TCP/IP yang mendekati jumlah alamat yang tersedia, durasi dari *lease* akan singkat. DHCP yang berbentuk database ini mempunyai file-file yang penting yaitu DHCP.MDB, DHCP.TMP, JET.LOG dan JET*.LOG, dan System.MDB.

DHCP.MDB adalah file database DHCP, sedangkan DHCP.TMP merupakan file yang digunakan oleh DHCP untuk menyimpan *working data* untuk sementara. JET.LOG dan JET*.LOG merupakan file yang merekam seluruh transaksi yang terjadi pada database. Data yang terdapat pada file-file ini digunakan untuk memulihkan sistem DHCP apabila database mengalami kerusakan. Sistem.MDB menyimpan informasi tetang struktur database DHCP.

Cara kerja sistem *packet filtering*

Sistem *packet filtering* mengawasi secara individual dengan melihat melalui *router* (perangkat keras yang dapat berfungsi sebagai sebuah server karena harus membuat keputusan untuk me-*rout* seluruh paket yang diterima). *Router* juga harus menentukan seperti apakah pengiriman paket yang telah didapat itu kepada tujuan yang sebenarnya. *Router* saling berkomunikasi dengan protokol-protokol untuk me-*rout*. Protokol yang dimaksud adalah *Routing Information Protocol* (RIP) atau *Open Shortest Path First* (OSPF) yang menghasilkan sebuah *table routing*.

Tabel routing menunjukkan tujuan dari paket yang diterima. *Router* yang menjadi filter pada *packet filtering* dapat menyediakan sebuah *choke point* (*channel* sempit yang sering digunakan untuk dipakai oleh penyerang sistem dan tentu saja dapat dipantau dan dikontrol oleh *user*) untuk semua pengguna yang memasuki dan meninggalkan network. Sistem ini beroperasi di tingkat *Network Layer* dan *Transport Layer* dari tingkatan protokol pada tingkatan TCP/IP. Bagian kepala dari network dan transport mengawasi informasi-informasi berikut :

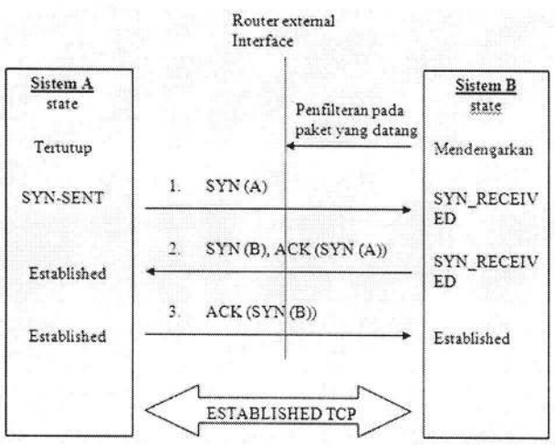
- Protokol (IP header, pada network layer): di dalamnya byte 9 mengidentifikasikan protokol dari paket.
- *Source address* (IP header, pada

network layer): alamat IP 32 bit dari host yang menciptakan oleh paket.

- *Destination address* (Ipheader, pada network layer): alamat tujuan yang berukuran 32 bit dari host yang menjadi tujuan dari paket.
- *Source port* (TCP atau UDP header, pada transport layer) pada setiap akhir dari koneksi TCP atau UDP tersambung dengan sebuah port walaupun port-port TCP terpisah dan cukup jauh dari port-port UDP. Port-port yang mempunyai nomor di bawah 1024 diterbalikan karena nomor-nomor ini telah didefinisikan secara khusus, sedangkan untuk port-port yang bernomor di atas 1024 (inklusif) lebih dikenal dengan port ephermal. Konfigurasi dari nomor pengalamatan ini diberikan sesuai dengan pilihan dari vendor.
- *Destination port* (TCP atau UDP header, transport layer): nomor port dari tujuan mengindikasikan port yang dikirimi paket. Servis yang akan diberikan pada sebuah host dengan mendengarkan port. Adapun port yang difilter adalah 20/TCP dan 21/TCP untuk koneksi ftp atau data, 23/TCP untuk telnet, 80/TCP untuk http dan 53/TCP untuk zona transfer DNS.
- *Connection status* (TCP atau UDP header, transport layer): status dari koneksi memberitahukan apakah paket yang dikirim merupakan paket pertama dari sesi di network. Jika merupakan paket pertama maka pada TCP header diberlakukan '*false*' atau 0 dan untuk mencegah sebuah *host* untuk mengadakan koneksi dengan menolak atau membuang paket yang mempunyai bit set '*false*' atau 0.

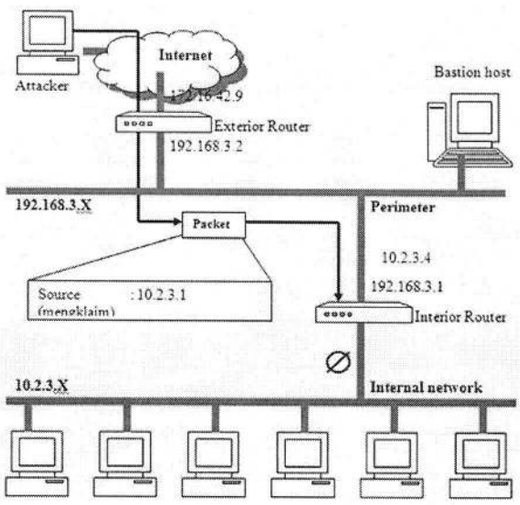
Komunikasi pada TCP

Pada jaringan komunikasi antar-host yang terjadi merupakan standar dari apa yang dikatakan sebagai membuka koneksi. Koneksi yang terjadi karena ada dua jaringan berkomunikasi disebut *three-way handshake*, di mana dari *host* yang saling berkomunikasi ada bit-bit yang digunakan untuk membangun suatu koneksi seperti diperlihatkan pada Gambar 3



Gambar 3. Bagan Komunikasi Pada TCP

Beberapa sistem pengamanan dapat disediakan hanya oleh *router filtering* jika program untuk sistem itu dijalankan pada lokasi tertentu di network. Contoh, suatu ide yang sangat bagus apabila menolak semua paket eksternal yang mempunyai alamat sumber internal atau dengan kata lain paket tersebut mengklaim bahwa paket tersebut berasal dari dalam mesin tetapi paket itu sebenarnya berasal dari luar, karena paket biasanya sebuah bagian dari penyerangan dari *address-spoofing*. Untuk itu *router* yang berfungsi sebagai *filtering* harus memiliki kemampuan untuk memberikan keputusan apakah paket berasal dari dalam atau luar kegiatan yang sering disebut dengan *source address forgery* seperti dapat dilihat pada Gambar 4.



Gambar 4. Implementasi Source Address Forgery

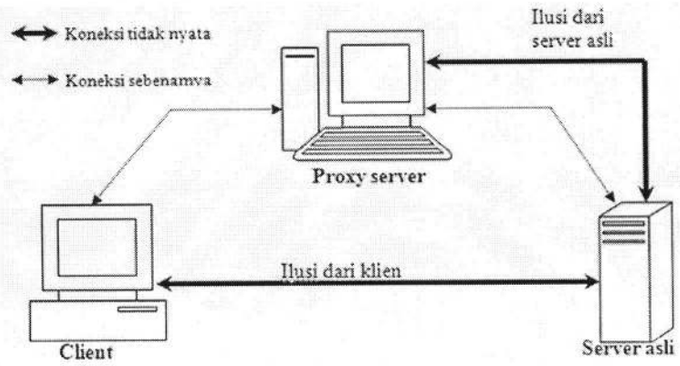
Cara kerja proxy

Proxy memberikan akses internet untuk satu *host* atau *host* yang dalam jumlah kecil dengan terlihat seperti menyediakan akses untuk seluruh host. Sebuah *proxy server* untuk protokol tertentu atau sebuah set dari protokol dapat dijalankan pada sebuah *dual-homed host* atau pada *bastion host*. *Proxy* sangat mendukung arsitektur dari *client/server*. *Client/server* membentuk sebuah sistem di mana komponen-komponen dari software saling berinteraksi.

Para klien dapat meminta seluruh kebutuhan dan pelayanan yang dinginkan, dan server menyediakannya. Sistem *proxy* harus mendukung seluruh pelayanan yang diminta dan diperlukan oleh klien. Server harus mempunyai file server yang sangat besar dan selalu aktif. File-file yang terdapat pada server akan digunakan oleh setiap komputer yang terhubung baik dalam *Local Area Network* (LAN) ataupun *Wide Area Network* (WAN). Pada file server selain dari list yang cukup panjang sebagai database yang dapat digunakan oleh setiap klien yang menggunakan alamat IP yang legal, terdapat juga file-file untuk aplikasi yang bekerja pada server utama.

Proxy merupakan sistem pengamanan yang memerlukan alamat IP yang jelas dan valid, karena server yang utama terdapat di internet. Pada *proxy* terdapat empat pendekatan yang akan dilakukan pada sisi klien yang sangat berperan penting. Pendekatan-pendekatan tersebut yaitu :

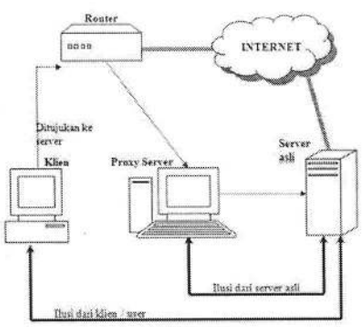
1. *Proxy-aware application software.* Dengan pendekatan ini software harus mengetahui bagaimana membuat kontak dengan *proxy server* daripada dengan server yang sebenarnya ketika user membuat suatu permintaan dan bagaimana memberitahukan proxy server, server asli yang mana yang harus dibuatkan koneksi seperti dapat dilihat pada Gambar 5. Mekanisasi dari ini sangat bergantung pada *runtime linking* yang dinamis (kemampuan untuk memberikan *library* ketika program dijalankan). Mekanisme ini tidak selalu berjalan dengan mulus dan dapat gagal yang tidak wajar untuk user.



Gambar 5. Proxy Aware Application Software

2. *Proxy-aware user procedures*. Pada pendekatan ini pengguna menggunakan *software client* yang tidak mengerti bagaimana me-*proxy*, di mana untuk berbicara (berkomunikasi) ke *server proxy* dan memberitahukan *proxy server* untuk melakukan hubungan kepada server yang sebenarnya daripada memberitahukan software klien untuk berkomunikasi secara langsung ke server yang sebenarnya.

3. *Proxy-aware router*. Pada pendekatan ini software yang digunakan klien tidak dimodifikasikan tetapi sebuah *router* mengantisipasi koneksi dan melangsungkan ke *proxy server* atau *proxy* yang diminta. Mekanisme ini membutuhkan sebuah *router* yang pintar di samping *software proxy* (meskipun me-*proxy* dan me-*rout* tidak bisa tampil pada mesin yang sama). Mekanisme pada *proxy* ini diperlihatkan pada Gambar 6.



Gambar 6. Proxy-Aware Router Pada Sistem Proxy

HASIL DAN PEMBAHASAN

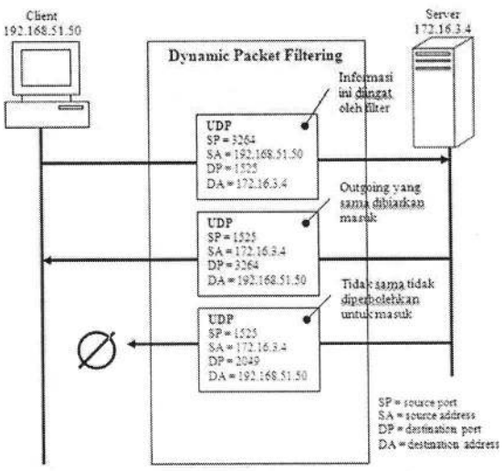
Paket filtering yang paling simpel seperti *Ipchains* mempunyai sebuah daftar aturan-aturan yang dapat diperbandingkan untuk seluruh lalulintas data yang masuk maupun yang keluar. Sebagai contoh:

```
Input-I eth1 -s 0.0.0.0/0 1024:65535 -d
0.0.0.0/0 80 -p tcp -j ACCEPT
Input-I eth0 -s 0.0.0.0/0 80 -d
0.0.0.0/0 1024:65535 -p tcp -! -SYN -j
ACCEPT
Input-j REJECT -|
```

Melihat IP *chain* di atas maka terdapat dua sistem paket filtering yang lebih *advanced*, yakni *Stateful Packet Filtering* dan *Dynamic Packet Filtering*.

Stateful packet filtering tidak memerlukan peraturan dalam memberikan keleluasaan dalam lalulintas yang hilir mudik. Pada saat permintaan *outgoing* dibuat dari sisi terpercaya dari *network*, firewall akan mulai mendengar untuk memberikan balasan. Bila sampai batas waktu tertentu tidak ada balasan, firewall segera membuang aturan-aturan yang tercipta secara dinamis dan tidak ada lalulintas data yang diperbolehkan untuk lewat.

Supaya balasan benar-benar datang, pintu untuk keluar diubah atau dihilangkan secara halus atau kasar sehingga tidak ada lubang yang nampak dari luar sistem. Sebuah *dynamic packet filtering* karena perilaku dari sistem yang diakibatkan dari perubahan-perubahan yang terjadi dilihat dari lalulintas data pada saat itu seperti dapat dilihat pada Gambar 7.



Gambar 7. Dynamic Packet Filtering Sebagai Firewall

Perbandingan

Pada internet terdapat fasilitas untuk mengirim data dan informasi melalui sebuah protokol yang dikenal dengan 'File Transfer Protocol (FTP)'. Protokol ini digunakan untuk mentransfer segala macam data dari mesin yang satu ke mesin yang lain. Penggunaan FTP tidak dibatasi oleh jenis data, baik binary yang dapat dijalankan, gambar grafik, teks ASCII, postscript, file bunyi dan file video.

Dilihat dari cara mengakses, ada dua tipe FTP yaitu *User FTP* atau *authenticated FTP* dan *Anonymous FTP*. User FTP membutuhkan sebuah account yang ada pada server (umumnya FTP ini ditujukan kepada pengguna yang telah mempunyai account di dalam mesin dan membiarkan pengguna ini untuk mengakses file apa saja yang dapat diakses setelah mereka melakukan *log in*). *Anonymous FTP* untuk orang-orang yang belum mempunyai account dan FTP ini digunakan untuk menyediakan akses untuk file-file spesifik pada dunia yang luas.

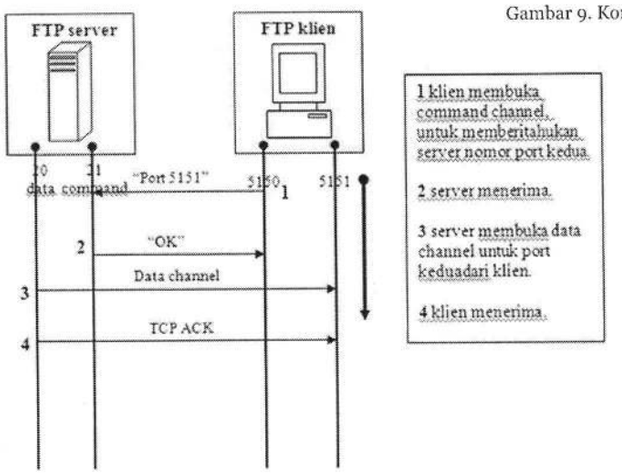
Adapun terdapat karakteristik tersendiri atas sistem keamanan yang terdapat pada firewall. FTP menggunakan dua koneksi TCP yang terpisah: yang satu untuk membawa perintah dan hasil dari *client-server* (sering disebut *command channel*), dan yang lainnya membawa file yang sudah teraktual dan daftar-daftar direktori yang tertransfer (dikenal dengan *data channel*).

Command channel menggunakan port 21 untuk server dan port di atas 1023 untuk klien. Pada FTP terdapat dua cara untuk merancang *data channel* yaitu mode normal dan mode pasif. Pada mode normal server menggunakan port bernomor 20 untuk *data channel*, sedangkan pada mode pasif server menggunakan port bernomor di atas 1023, dan klien selalu menggunakan port bernomor di atas 1023 untuk *data channel*.

Untuk memulai sesi FTP pada mode normal, seorang klien mula-mula harus mengalokasikan dua port TCP untuk dirinya sendiri, masing-masing dari port menggunakan nomor di atas 1023. Itu dilakukan pertama kali untuk membuka koneksi *command channel* untuk server dan kemudian mengisukan port perintah

pada FTP untuk memberitahukan server nomor pada port kedua, yang ingin digunakan klien untuk *data channel*.

Server kemudian membuka koneksi *data channel*. Koneksi *data channel* berjalan ke belakang, berbeda dengan kebanyakan protokol, di mana koneksi yang dibuka dari klien ke server. Pembukaan yang berjalan mundur ini mempersulit situs-situs yang akan mengadakan koneksi pada paket filtering untuk memastikan bahwa seluruh koneksi FTP telah dirintis dari dalam karena server FTP luar akan mencoba untuk memulai koneksi data untuk klien dalam sebagai respon dari koneksi perintah yang dibuka dari klien dalam tersebut. Selanjutnya koneksi ini berjalan ke port-port yang diketahui tidak dalam lingkup yang aman seperti terlihat pada Gambar 8.



Gambar 9. Koneksi TCP Dalam Mode Pasif

Gambar 8. Koneksi FTP Pada Mode Normal

Pada koneksi mode pasif, klien FTP membagi dua port TCP untuk digunakan sendiri dan digunakan untuk port pertama yang akan menghubungi server FTP, sama seperti saat menggunakan mode normal. Tanpa menggunakan port perintah untuk memberitahukan kepada server port kedua dari klien, klien mengisukan dengan perintah PASV.

Ini mengakibatkan server harus membagi port keduanya untuk *data channel* (untuk alasan arsitektur server menggunakan port-port bernomor di atas 1023 secara acak untuk ini, bukan port 20 seperti pada mode normal; karena tidak mungkin mempunyai dua server pada mesin yang sama secara simulasi mendengarkan PASV-mode yang datang dari port 20) dan memberitahukan klien nomor pada port tersebut. Klien kemudian membuka koneksi data dari port tersebut ke port data dari server yang telah disebutkan seperti terlihat pada Gambar 9.

KESIMPULAN DAN SARAN

Sistem keamanan yang dibutuhkan tergantung pada data apa yang akan diamankan. Pada *packet filtering* pengawasan atas file-file yang masuk maupun keluar dilakukan secara ketat karena *packet filtering* ditujukan untuk pengguna yang ingin membuat jaringan lokal yang terdiri dari banyak komputer yang mengakses satu tempat sekaligus. Pada *proxy* lebih ditekankan untuk pengguna secara individual dari pada jaringan lokal yang cukup besar. Dengan kelebihan dan kekurangannya, *proxy services* lebih ditujukan kepada komputer pribadi atau HOME PC, sedangkan *packet filtering* merupakan pilihan yang paling tepat untuk jaringan yang dibangun pada dasar jaringan lokal.

Untuk sistem keamanan yang baik, khususnya jaringan harus mempunyai perangkat keras yang dapat menjalankan *packet filtering* sampai pada tingkat alamat IP dan sumber daya manusia yang siap untuk menerima tanggung jawab untuk mengawasi sistem secara teratur.

DAFTAR PUSTAKA

Bhagchandka, Dhiraj. 2003. *Classification of Firewalls and Proxies*. Department of Computer Sciences, The University of Texas at Austin Computer Science Research and Writing.

Grennan, Mark. 2000. *Firewall and Proxy Server*. HOWTO, vo.80, Feb. 26.

Haas, J, 2003. *Squid-ProxyServer*. , Access Date: 15 Juli 2011.
, Karanjit B., Parker, , and Parker, 2002. *TCP/IP Unleashed*. Third Edition, Sams; 3 Sub edition, February 20.

Zwicky, Elizabeth D., Cooper, Simon and Chapman, D. Brent. 2000. *Building Internet Firewalls*. Second Edition.