

Pendahuluan

Keamanan sistem jaringan komputer adalah bagian tak terpisahkan dari keamanan sistem komputer sebuah organisasi secara keseluruhan, terutama dengan semakin berkembangnya Internet. Semakin banyak aplikasi pengguna yang berbasis pada jaringan komputer. Jika sebuah jaringan komputer tidak aman, maka sistem komputer pada organisasi tersebut juga tidak aman.

Pada bab ini akan dijelaskan mengenai keamanan sistem jaringan komputer. Pembahasan akan dimulai dengan penjelasan mengenai arsitektur jaringan komputer, yang dilanjutkan dengan pembahasan mengenai mekanisme-mekanisme keamanan yang dimiliki arsitektur jaringan komputer. Setelah itu akan dijelaskan mengenai *tools-tools* yang bisa digunakan untuk melindungi jaringan komputer dari gangguan. Dijelaskan juga beberapa contoh gangguan yang mungkin dihadapi pengelola dalam melindungi sistem jaringan komputernya. Terakhir akan diberikan dua studi kasus keamanan jaringan komputer sebagai bahan pembelajaran.

I. Arsitektur Jaringan Komputer

Untuk dapat dengan jelas mengerti mengenai keamanan jaringan komputer, kita harus terlebih dahulu mengerti bagaimana jaringan komputer bekerja. Untuk mempermudah pemeliharaan serta meningkatkan komparabilitas antar berbagai pihak yang mungkin terlibat, jaringan komputer terbagi atas beberapa lapisan yang saling independen satu dengan yang lainnya. Menurut standard ISO/OSI, lapisan-lapisan dan tugas yang dimilikinya adalah :

- *Layer 1 - Physical*
Layer (lapisan) ini berhubungan dengan kabel dan media fisik lainnya yang menghubungkan satu peralatan jaringan komputer dengan peralatan jaringan komputer lainnya. Lapisan ini juga berhubungan dengan sinyal-sinyal listrik, sinar maupun gelombang radio yang digunakan untuk mengirimkan data. Pada lapisan ini juga dijelaskan mengenai jarak terjauh yang mungkin digunakan oleh sebuah media fisik. Pada lapisan ini juga diatur bagaimana cara melakukan *collision control*.
- *Layer 2 - Data Link*
Pada sisi pengirim, lapisan ini mengatur bagaimana data yang akan dikirimkan diubah menjadi deretan angka '1' dan '0' dan mengirimkannya ke media fisik. Sedangkan pada sisi penerima, lapisan ini akan merubah deretan angka '1' dan '0' yang diterima dari media fisik menjadi data yang lebih berarti. Pada lapisan ini juga diatur bagaimana kesalahan-kesalahan yang mungkin terjadi ketika transmisi data diperlakukan.
Lapisan ini terbagi atas dua bagian, yaitu *Media Access Control* (MAC) yang mengatur bagaimana sebuah peralatan dapat memiliki akses untuk mengirimkan data dan *Logical Link Control* (LLC) yang bertanggung jawab atas sinkronisasi *frame*, *flow control* dan pemeriksaan *error*. Pada MAC terdapat metode-metode yang digunakan untuk menentukan siapa yang berhak untuk melakukan pengiriman data. Pada dasarnya metode-metode itu dapat bersifat terdistribusi (contoh: CSMA/CD atau CSMA/CA) dan bersifat terpusat (contoh: token ring). Secara keseluruhan, lapisan *Data Link* bertanggung jawab terhadap koneksi dari satu node ke node berikutnya dalam komunikasi data.
- *Layer 3 - Network*
Lapisan *Network* bertanggung jawab terhadap koneksi dari pengirim sampai dengan penerima. Lapisan ini akan menterjemahkan alamat logik sebuah *host* menjadi sebuah alamat fisik. Lapisan ini juga bertanggung jawab untuk mengatur rute yang akan dilalui sebuah paket yang dikirim agar dapat sampai pada tujuan. Jika dibutuhkan penentuan jalur yang akan dilalui sebuah paket, maka sebuah *router* akan menentukan jalur 'terbaik' yang akan dilalui paket tersebut. Pemilihan jalur atau rute ini dapat ditentukan secara statik maupun secara dinamis.
- *Layer 4 - Transport*

Lapisan ini bertanggung jawab untuk menyediakan koneksi yang bebas dari gangguan. Ada dua jenis komunikasi data jaringan komputer, yaitu *Connection Oriented* dan *Connectionless*. Pada jenis komunikasi *Connection Oriented* data dipastikan sampai tanpa ada gangguan sedikitpun juga. Apabila ada gangguan, maka data akan dikirimkan kembali. Sedangkan jenis komunikasi *Connectionless*, tidak ada mekanisme untuk memastikan apabila data yang dikirim telah diterima dengan baik oleh penerima.

Biasanya lapisan ini mengubah layanan yang sangat sederhana dari lapisan *Network* menjadi sebuah layanan yang lebih lengkap bagi lapisan di atasnya. Misalnya, pada layer ini disediakan fungsi kontrol transmisi yang tidak dimiliki oleh lapisan di bawahnya.

- *Layer 5 - Session*

Lapisan ini bertanggung jawab untuk membangun, memelihara dan memutuskan koneksi antar aplikasi. Pada kenyataannya lapisan ini sering digabung dengan *Application Layer*.

- *Layer 6 - Presentation*

Agar berbagai aplikasi jaringan komputer yang ada di dunia dapat saling terhubung, seluruh aplikasi tersebut harus mempergunakan format data yang sama. Lapisan ini bertanggung jawab atas bentuk format data yang akan digunakan dalam melakukan komunikasi. Pada kenyataannya lapisan ini sering pula digabung dengan *Application Layer*.

- *Layer 7 - Application*

Lapisan ini adalah di mana interaksi dengan pengguna dilakukan. Pada lapisan inilah semua jenis program jaringan komputer seperti *browser* dan *email client* berjalan.

Pada implementasinya, lapisan jaringan komputer berdasarkan ISO/OSI tidak digunakan karena terlalu kompleks dan ada banyak duplikasi tugas dari setiap lapisan. Lapisan OSI/ISO digunakan hanya sebagai referensi. Lapisan jaringan komputer yang banyak digunakan adalah lapisan TCP/IP yang terdiri atas empat lapisan yaitu :

- Link (Lapisan OSI 1 dan 2)

Contoh dari lapisan ini adalah Ethernet, Wi-Fi dan MPLS. Implementasi untuk lapisan ini biasanya terletak pada *device driver* ataupun *chipset firmware*.

- Internetwork (Lapisan OSI 3)

Seperti halnya rancangan awal pada lapisan network (lapisan OSI 3), lapisan ini bertanggung-jawab atas sampainya sebuah paket ke tujuan melalui sebuah kelompok jaringan komputer. Lapisan Internetwork pada TCP/IP memiliki tugas tambahan yaitu mengatur bagaimana sebuah paket akan sampai tujuan melalui beberapa kelompok jaringan komputer apabila dibutuhkan.

- Transport (Lapisan OSI 4 dan 5)

Contoh dari lapisan ini adalah TCP, UDP dan RTP

- Applications (Lapisan OSI 5 sampai dengan 7)

Contoh dari lapisan ini adalah HTTP, FTP dan DNS.

Oleh sebab setiap lapisan memiliki tugas yang independen dari lapisan-lapisan lainnya, maka transparansi data akan terjamin. Sebagai contoh, semua jenis *browser* internet akan tetap digunakan, sekalipun media fisik yang digunakan berubah dari kabel tembaga menjadi sinyal radio misalnya. Untuk lebih mendalami mengenai hal ini, dapat dibaca pada referensi [CURTIN].

II. Tipe-tipe proteksi jaringan komputer

Dikarenakan perbedaan fungsi dalam setiap lapisan jaringan komputer, maka perlindungan yang dapat dilakukan juga berbeda-beda. Pada bagian ini akan dijelaskan mengenai perlindungan terhadap jaringan komputer yang bisa dilakukan pada setiap lapisan jaringan komputer, mulai dari lapisan terbawah sampai dengan lapisan teratas.

- Layer 2

Dalam usaha mengamankan sebuah gedung, tahap yang paling mendasar adalah dengan menjaga titik akses ke gedung tersebut. Begitu juga dengan pengamanan jaringan komputer, tahap paling mendasar adalah menjaga titik akses yang dapat digunakan seseorang untuk terhubung ke dalam jaringan. Pada umumnya, titik akses jaringan komputer adalah berupa hub atau switch. Dengan berkembangnya *wireless network*, maka peralatan *wireless access-point* juga termasuk dalam titik akses jaringan yang perlu untuk dilindungi.

Saat ini ada dua mekanisme umum yang biasa digunakan dalam mengamankan titik akses ke jaringan komputer, yaitu :

- Protokol 802.1x

Protokol 802.1x adalah sebuah protokol yang dapat melakukan otentikasi pengguna dari peralatan yang akan melakukan hubungan ke sebuah titik-akses. Dengan protokol ini, ketika sebuah komputer melakukan hubungan ke sebuah titik-akses (*hub* atau *switch*), maka pengguna komputer tersebut perlu melakukan otentikasi sebelum komputer tersebut terhubung ke jaringan komputer.

Protokol ini sangat berguna untuk melindungi jaringan komputer sekaligus mengakomodasi pengguna-pengguna yang memiliki peralatan atau komputer yang bersifat *mobile* seperti *notebook* atau PDA. Dengan digunakannya protokol ini, dapat dijamin bahwa peralatan komputer yang berusaha melakukan akses ke jaringan komputer sedang dipergunakan oleh pihak yang memang telah diizinkan untuk melakukan akses.

Tiga komponen yang terlibat dalam protokol ini adalah peralatan yang akan melakukan akses (*supplicant*), server yang akan melakukan otentikasi (server RADIUS) dan peralatan yang menjadi titik akses (otentikator). Secara umum, tahapan-tahapan dalam protokol ini adalah :

1. Secara *default* akses ke jaringan tertutup.
2. Sebuah *supplicant* melakukan akses dan meminta izin akses ke otentikator, yang kemudian meneruskannya ke server otentikasi.
3. Server otentikasi menjawab dengan memberikan 'tantangan' ke *supplicant* melalui otentikator.
4. Melalui otentikator, *supplicant* menjawab 'tantangan' yang diberikan.
5. Apabila jawaban yang diberikan *supplicant* benar, server otentikasi akan memberitahu ke otentikator yang kemudian akan memberikan akses jaringan ke *supplicant*.
6. Akses jaringan yang sudah terbuka, akan tetap terbuka sampai ketika terjadi perubahan status koneksi, misalnya koneksi diputus oleh pengguna atau alat yang terhubung berubah. Ketika terjadi perubahan status, akses akan kembali ditutup dan proses otentikasi akan berulang kembali.

Pada perkembangannya, protokol ini digunakan secara lebih mendalam, bukan hanya untuk melakukan otentikasi terhadap pengguna peralatan yang melakukan akses, melainkan juga akan digunakan untuk memeriksa apakah konfigurasi peralatan yang melakukan akses sudah sesuai dengan kebijakan yang berlaku. Misalkan akan dilakukan pemeriksaan apakah program antivirus yang berjalan pada sebuah *notebook* yang akan melakukan koneksi sudah mempergunakan versi yang terbaru, jika kondisi tersebut tidak terpenuhi maka akses jaringan tidak akan diberikan. Selain itu protokol ini juga dapat digunakan untuk menegakkan sebuah kebijakan pada peralatan-peralatan yang akan melakukan akses jaringan komputer.

Kelemahan dari protokol ini adalah, protokol ini harus diimplementasikan satu per satu pada semua *switch/hub* yang menjadi titik akses jaringan komputer.

Penjelasan mengenai protokol ini bisa dapat dibaca pada referensi [SNYDER]

- Mac Address

Mac Address Authentication adalah sebuah mekanisme di mana sebuah peralatan yang akan

melakukan akses pada sebuah titik-akses sudah terdaftar terlebih dahulu. Berbeda dengan protokol 802.1x yang memastikan bahwa alat yang melakukan koneksi dipergunakan oleh pihak yang berwenang, metode ini untuk memastikan apakah peralatan yang akan melakukan akses adalah peralatan yang berhak untuk akses tanpa mempedulikan siapa yang mempergunakannya.

Pada setiap peralatan jaringan komputer terdapat sebuah identitas yang unik. Berdasarkan identitas tersebutlah metode ini melakukan otentikasi. Pada setiap paket data yang dikirimkan sebuah peralatan akan mengandung informasi mengenai identitas peralatan tersebut, yang akan dibandingkan dengan daftar akses yang dimiliki setiap titik-akses, apabila ternyata identitas peralatan terdapat dalam daftar, paket yang dikirimkannya akan diteruskan apabila tidak, maka paket yang dikirimkannya tidak akan diteruskan.

Keuntungan metode ini jika dibandingkan dengan protokol 802.1x adalah metode ini sudah lebih banyak diimplementasikan pada *switch/hub* yang sering digunakan sebagai titik akses. Selain itu, untuk mempergunakan metode ini, tidak perlu semua *switch/hub* melakukan *filtering*, namun cukup *switch/hub* utama saja yang melakukannya.

Kelemahan utama dari metode ini adalah seseorang dapat dengan mudah memanipulasi identitas unik pada peralatan yang digunakannya, sehingga peralatan tersebut dapat melakukan akses ke sebuah jaringan komputer. Oleh karena itu sangat penting untuk menjaga integritas daftar identitas peralatan yang dapat melakukan akses ke jaringan.

Selain kedua protokol otentikasi yang telah disebutkan di atas, ada sebuah metode keamanan yang terletak pada lapisan *Data Link* tapi tidak berfungsi untuk melakukan otentikasi penggunaan titik-akses jaringan komputer, melainkan untuk melindungi data yang dikirimkan pada jaringan komputer tersebut. Metode tersebut adalah:

– *WEP dan WPA*

Perkembangan teknologi telah membuat transmisi data melalui media gelombang radio memiliki kualitas yang hampir sama dengan kualitas transmisi data melalui media kabel. Dengan menggunakan *wireless network*, koneksi ke sebuah jaringan komputer menjadi sangat mudah karena tidak lagi terhambat oleh penggunaan kabel. Asalkan sebuah peralatan jaringan komputer masih dalam jangkauan gelombang radio komputer penyedia jaringan, peralatan tersebut dapat terhubung ke dalam jaringan komputer.

Akan tetapi, penggunaan media gelombang radio untuk transmisi data memiliki berbagai permasalahan keamanan yang cukup serius. Sifat gelombang radio yang menyebar menyebabkan siapa saja yang berada pada jangkauan gelombang radio yang digunakan untuk komunikasi data dapat mencuri data yang dikirimkan oleh sebuah pihak ke pihak lain dengan mudah. Oleh karena itu dikembangkan metode yang disebut dengan *Wired Equivalent Privacy* (WEP).

Tujuan utama dari WEP adalah berusaha untuk memberikan tingkat privasi yang diberikan oleh penggunaan jaringan berbasis kabel. Dalam melakukan usaha itu, WEP akan melakukan enkripsi terhadap data-data yang dikirimkan antara dua peralatan jaringan komputer berbasis gelombang radio, sehingga data yang dikirimkan tidak dapat dicuri oleh pihak lain. Untuk ini, WEP mempergunakan algoritma *stream-cipher* RC4 untuk menjaga kerahasiaan data dan CRC-32 sebagai kontrol integritas data yang dikirimkan. Oleh karena ada peraturan pembatasan ekspor teknologi enkripsi oleh pemerintah Amerika Serikat, maka pada awalnya panjang kunci yang dipergunakan hanyalah sepanjang 40 bit. Setelah peraturan tersebut dicabut, maka kunci yang digunakan adalah sepanjang 104 bit.

Beberapa analis menemukan bahwa WEP tidak aman dan seseorang dapat dengan mudah menemukan kunci yang digunakan setelah melakukan analisa paket terenkripsi yang dia dapatkan. Oleh karena itu pada tahun 2003 dibuat standar baru yaitu *Wi-Fi Protected Access* (WPA). Perbedaan antara WEP dengan WPA adalah penggunaan protokol 802.1x untuk melakukan distribusi kunci yang digunakan dalam melakukan proses enkripsi dan dekripsi. Selain itu panjang kunci yang digunakan juga bertambah panjang menjadi 128 bit sehingga

menambah tingkat kesulitan dalam menebak kunci yang digunakan. Selain itu untuk meningkatkan keamanan, juga dibuat sebuah sistem yang disebut dengan *Temporal Key Integrity Control* yang akan melakukan perubahan kunci secara dinamis selama sistem sedang digunakan. Pada perkembangan selanjutnya, yaitu pada tahun 2004 dibuat standard WPA2, dimana algoritma RC4 digantikan oleh algoritma enkripsi baru yaitu *Advance Encryption System* (AES) dengan panjang kunci sepanjang 256 bit. Lebih lanjut mengenai kedua hal ini, dapat dilihat pada referensi [WEP]

- Layer 3

Pada lapisan ini, untuk membedakan sebuah peralatan jaringan komputer dengan peralatan jaringan komputer yang lainnya, digunakan alamat IP (*Internet Protocol*). Semua peralatan komputer aktif harus memiliki sebuah nomor IP unik yang akan menjadi identitasnya di jaringan komputer. Alamat IP yang saat ini banyak digunakan disebut dengan IPv4, yaitu sebuah deretan angka dengan format :

X.X.X.X

di mana x adalah angka antara 0 sampai dengan 255. Saat ini sedang dalam tahap pengembangan versi baru dari alamat IP yang disebut dengan IPv6. Selain alamat IP, pada lapisan ini juga dikenal istilah *Port*, yaitu sebuah pintu masuk ke dalam sebuah sistem komputer. Pada pintu inilah aplikasi jaringan komputer yang sedang berjalan dalam sebuah komputer menerima melakukan koneksi dengan pihak lain.

Pada lapisan ini, metode perlindungan jaringan komputer akan berdasarkan pada alamat IP dan *Port*. Pada setiap paket data yang dikirimkan oleh sebuah peralatan jaringan komputer ke peralatan lainnya akan mengandung alamat IP dan *Port* yang digunakan oleh pengirim serta alamat IP dan *Port* dari tujuan paket tersebut. Sebuah sistem pengamanan yang biasanya dikenal dengan nama *firewall* dapat melakukan *filtering* berdasarkan kedua hal tersebut. Pada umumnya *firewall* diletakkan pada gerbang masuk maupun keluar sebuah sistem jaringan komputer. Selain itu *firewall* juga dapat melakukan *filtering* berdasarkan protokol yang digunakan oleh sebuah paket data, misalnya sebuah *firewall* dapat dirancang untuk menolak paket jenis udp dan paket jenis icmp sementara mengizinkan paket jenis tcp.

Pada perkembangannya, *firewall* tidak hanya melakukan *filtering* berdasarkan alamat IP dan *Port*, tapi juga berdasarkan informasi lainnya yang tersedia dalam *header* sebuah paket IP. Sebagai contoh, sebuah *firewall* dapat melakukan *filtering* berdasarkan ukuran data sebuah paket data. Sebuah *firewall* juga bisa melakukan *filtering* berdasarkan status koneksi antara dua peralatan jaringan komputer, misalnya sebuah *firewall* dapat dirancang untuk menolak sebuah paket yang akan membuat sebuah koneksi baru dari sebuah alamat IP, tapi mengizinkan paket-paket lainnya dari alamat IP tersebut. Untuk menambah keamanan sistem jaringan komputer, saat ini sebagian besar *firewall* sudah bersifat *statefull* dan tidak lagi *stateless*. Pada *statefull firewall*, *firewall* akan membuat daftar sejarah status koneksi antara satu peralatan jaringan komputer dengan peralatan jaringan komputer lainnya. Hal ini untuk mencegah adanya penipuan status koneksi oleh sebuah peralatan jaringan komputer untuk dapat melewati proses *filtering* sebuah *firewall*.

Selain diimplementasikan pada gerbang masuk atau gerbang keluar dari sebuah sistem jaringan komputer, *firewall* juga dapat diimplementasikan pada sebuah *host*. Ini berguna untuk melindungi *host* tersebut dari serangan yang berasal dari *host* lain yang berada pada jaringan komputer yang sama.

Pada umumnya, implementasi *firewall* adalah metoda pengamanan sistem jaringan komputer yang pertama kali dilakukan. Walaupun cukup ampuh dan mudah untuk diimplementasikan, tanpa perencanaan yang baik, implementasi *firewall* dapat menyebabkan sebuah *firewall* tersusun atas peraturan-peraturan *filtering* yang sangat banyak. Hal ini dapat membuat *firewall* tersebut menjadi sulit untuk dikelola karena dengan banyaknya peraturan-peraturan *filtering* yang diimplementasikan akan lebih sulit untuk melakukan penelusuran proses penyaringan paket. Selain itu, banyaknya peraturan *filtering* yang terlalu banyak juga dapat mengganggu interaksi koneksi data jaringan komputer, karena semua paket yang lewat harus melalui proses penyaringan yang sangat banyak.

- Layer 4 /5

Pada lapisan ini, metode pengamanan lebih difokuskan dalam mengamankan data yang dikirimkan. Metode pengamanan yang banyak digunakan adalah :

- VPN

Pada banyak organisasi besar, organisasi tersebut memiliki kantor-kantor cabang yang tersebar di banyak tempat. Kantor cabang-kantor cabang tersebut tentu memiliki kebutuhan untuk saling berhubungan antara satu dengan yang lainnya. Pada masa-masa awal jaringan komputer, solusi yang biasa digunakan adalah dengan membangun jaringan privat yang menghubungkan seluruh kantor cabang yang ada atau yang biasa disebut dengan *Wide Area Network* (WAN). Dengan berkembangnya jaringan Internet, solusi dengan membangun WAN, menjadi solusi yang sangat mahal dan tidak fleksibel. Dengan berkembangnya *Virtual Private Network*, sebuah organisasi dapat membangun jaringan privat maya diatas jaringan publik untuk menghubungkan seluruh kantor cabang yang dimilikinya.

Kelebihan implementasi VPN dibandingkan dengan implementasi WAN adalah:

- Mempermudah perluasan konektivitas jaringan komputer secara geografis
Untuk menghubungkan beberapa lokasi yang terpisah secara geografis dapat mempergunakan jaringan publik (Internet) yang dimiliki oleh masing-masing lokasi. Koneksi Internet yang digunakan oleh sebuah lokasi bisa saja tidak menggunakan layanan dari *service provider* yang sama dengan koneksi Internet di lokasi lainnya.
- Peningkatan keamanan data
Data yang dikirimkan akan terlindungi sehingga tidak dapat dicuri oleh pihak lain karena data yang ditransmisikan melalui VPN melalui proses enkripsi.
- Mengurangi biaya operasional
Dengan menggunakan VPN, setiap lokasi hanya perlu memelihara satu buah koneksi Internet untuk seluruh kebutuhannya, baik kebutuhan koneksi Internet maupun kebutuhan koneksi internal organisasi.
- Menyederhanakan Topologi jaringan

Pada dasarnya, VPN adalah perkembangan dari *network tunneling*. Dengan *tunneling*, dua kelompok jaringan komputer yang terpisah oleh satu atau lebih kelompok jaringan komputer diantaranya dapat disatukan, sehingga seolah-olah kedua kelompok jaringan komputer tersebut tidak terpisah. Hal ini dapat dilakukan dengan melakukan enkapsulasi terhadap paket jaringan yang dikirimkan. *Tunneling* ini bersifat transparan bagi pengguna jaringan komputer di kedua sisi kelompok jaringan komputer. Hanya *router* di kedua sisi kelompok jaringan komputer yang melakukan proses enkapsulasi yang mengetahui adanya *tunnel* tersebut. Imbal baik dari proses *tunneling* adalah *Maximum Transfer Unit* (MTU) setiap paket yang dikirim menjadi lebih kecil, karena diperlukan ruang tambahan untuk menambahkan *header* IP hasil enkapsulasi paket yang dikirimkan. Berkurangnya MTU dapat menyebabkan berkurangnya kecepatan transfer data antara dua *host* yang sedang berkomunikasi. Salah satu implementasi dari *tunneling* adalah *mobile IP*. Dengan mempergunakan *mobile IP*, seorang pengguna dapat selalu mempergunakan alamat IP yang dia miliki dimanapun pengguna tersebut berada. Implementasi lainnya adalah dengan menambahkan proses kompresi data yang akan dikirimkan melalui *tunnel* yang sudah dibuat. Dengan cara ini, makan dengan ukuran *bandwidth* yang sama, besar data yang dikirimkan dapat lebih besar, sehingga meningkatkan kecepatan transfer data.

Seluruh sifat dasar dari *network tunneling* dimiliki oleh VPN, ditambah dengan proses enkripsi dan dekripsi. Dengan menggunakan VPN, seluruh data yang dikirimkan oleh sebuah pengguna jaringan komputer di sebuah kelompok jaringan komputer ke kelompok jaringan komputer lainnya yang terhubung dengan VPN akan melalui proses enkripsi, sehingga tidak dapat dibaca oleh pihak-pihak lain yang berada pada jalur pengiriman data.

Pada sisi penerima data, secara otomatis, data akan melalui proses dekripsi sebelum disampaikan ke pihak penerima. Sama dengan *tunneling*, proses enkripsi dan dekripsi data terjadi secara transparan tanpa diketahui oleh pengirim maupun penerima. VPN dapat mempergunakan berbagai macam algoritma enkripsi, baik itu yang bertipe *symmetric-key-encryption* maupun *public-key-encryption*. Kunci dari seluruh penggunaan VPN adalah pada proses enkripsi dan dekripsi data, dan oleh karena itu, pemilihan algoritma enkripsi menjadi sangat penting dalam implementasi VPN.

Selain untuk menghubungkan dua atau lebih lokasi kantor cabang, VPN juga banyak digunakan untuk mengakomodasi kebutuhan pekerja yang bekerja di luar kantor untuk melakukan akses ke sumber daya yang tersedia pada jaringan internal kantor. Hal ini dapat dilakukan dengan menganggap komputer yang digunakan oleh seorang pekerja yang berada di luar kantor sebagai kantor cabang lain yang sedang melakukan koneksi. Cara ini sangat mirip dengan konsep *mobile IP* yang sudah dijelaskan diatas, perbedaannya selain mempergunakan alamat IP yang dia miliki dimanapun dia berada, data yang dikirimkan akan selalu ter-enkripsi. Dengan cara ini, seorang pekerja yang sedang berada di luar kantor dapat dengan mudah dan aman mempergunakan fasilitas yang ada di jaringan komputer kantornya, asalkan yang bersangkutan dapat terhubung dengan Internet.

Kelemahan utama dari VPN adalah tidak adanya sebuah standard baku yang dapat diikuti oleh semua pihak yang berkepentingan. Akibatnya ada banyak implementasi VPN yang dapat digunakan, tapi antara satu implementasi dengan implementasi lainnya tidak dapat saling berhubungan. Oleh karena itu apabila sebuah organisasi memilih untuk mempergunakan sebuah implementasi VPN pada sebuah *router*, maka seluruh *router* yang dimiliki organisasi tersebut yang akan digunakan dalam jaringan VPN, harus mempergunakan implementasi VPN yang sama. Selain itu jika layanan VPN akan diberikan kepada para pengguna yang sering berpergian, maka pada setiap *host* yang digunakan oleh pengguna tersebut juga harus di-*install* aplikasi VPN yang sesuai. Selain itu, karena harus melalui proses enkripsi dan dekripsi, sehingga waktu yang dibutuhkan untuk melakukan transmisi bertambah, maka kemungkinan VPN tidak cocok untuk digunakan dalam mengirimkan data yang bersifat interaktif, seperti tranmisi suara ataupun transmisi video.

Untuk mempelajari lebih lanjut mengenai hal ini, dapat dibaca referensi [TYSON].

- Layer 7

Lapisan paling atas dari jaringan komputer adalah lapisan aplikasi. Oleh karena itu, keamanan sebuah sistem jaringan komputer tidak terlepas dari keamanan aplikasi yang menggunakan jaringan komputer tersebut, baik itu keamanan data yang dikirimkan dan diterima oleh sebuah aplikasi, maupun keamanan terhadap aplikasi jaringan komputer tersebut. Metode-metode yang digunakan dalam pengamanan aplikasi tersebut antara lain adalah:

- SSL

Secure Socket Layer (SSL) adalah sebuah protokol yang bekerja tepat di bawah sebuah aplikasi jaringan komputer. Protokol ini menjamin keamanan data yang dikirimkan satu *host* dengan *host* lainnya dan juga memberikan metode otentikasi, terutama untuk melakukan otentikasi terhadap *server* yang dihubungi. Untuk keamanan data, SSL menjamin bahwa data yang dikirimkan tidak dapat dicuri dan diubah oleh pihak lain. Selain itu, SSL juga melindungi pengguna dari pesan palsu yang mungkin dikirimkan oleh pihak lain.

Tahapan-tahapan yang harus dilalui dalam menggunakan SSL adalah :

1. Negosiasi algoritma yang akan digunakan kedua-belah pihak.
2. Otentikasi menggunakan *Public Key Encryption* atau Sertifikat elektronik.
3. Komunikasi data dengan menggunakan *Symmetric Key Encryption*.

Pada tahap negosiasi algoritma yang akan digunakan, pilihan-pilihan algoritma yang bisa digunakan adalah :

- *Public Key Encryption* : RSA, Diffie-Helman, DSA (Digital Signature Algorithm)

atau Fortezza

- *Symmetric Key Encryption* : RC2, RC4, IDEA (International Data Encryption Algorithm), DES (Data Encryption Standard), Triple DES atau AES
- Untuk fungsi *hash* 1 arah : MD5 (Message-Digest algorithm 5) atau SHA (Secure Hash Algorithm)

aplikasi yang banyak menggunakan SSL adalah aplikasi perbankan berbasis web.

Perkembangan lanjutan dari SSL adalah TLS, kepanjangan dari *Transport Layer Security*.

Kelebihan-kelebihan yang dimiliki oleh TLS adalah :

- Pemberian nomor pada semua data dan menggunakan nomor urut pada *Message Authentication Code* (MAC)
- *Message Digest* hanya dapat dipergunakan dengan kunci yang tepat.
- Perlindungan terhadap beberapa serangan yang sudah diketahui (seperti *Man in the Middle Attack*)
- Pihak yang menghentikan koneksi, mengirimkan resume dari seluruh data yang dipertukarkan oleh kedua belah pihak.
- Membagi data yang dikirimkan menjadi dua bagian, lalu menjalankan fungsi *hash* yang berbeda pada kedua bagian data.

Pada implementasinya banyak aplikasi di sisi server dapat memfasilitasi koneksi biasa ataupun koneksi dengan TLS, tergantung dengan kemampuan klien yang melakukan koneksi. Apabila klien dapat melakukan koneksi dengan TLS maka data yang dikirimkan akan melalui proses enkripsi. Sebaliknya, apabila klien tidak memiliki kemampuan TLS, maka data akan dikirimkan dalam format *plaintext*.

Lebih lanjut mengenai hal ini dapat dilihat di [SSL]

- Application Firewall

Selain permasalahan keamanan transaksi data, yang perlu diperhatikan pada lapisan ini adalah aplikasi itu sendiri. Sebuah aplikasi jaringan komputer yang terbuka untuk menerima koneksi dari pihak lain dapat memiliki kelemahan yang dapat dipergunakan oleh pihak yang tidak bertanggung jawab. Sebuah kelemahan pada sebuah aplikasi dapat mengancam keamanan *host* yang menjalankan aplikasi tersebut juga *host-host* lain yang berada pada sistem jaringan komputer yang sama.

Dengan semakin berkembangnya *virus* dan *worm* yang menyerang kelemahan-kelemahan yang ada pada aplikasi jaringan komputer, maka diperlukan keamanan lebih pada lapisan ini. Untuk melindungi aplikasi-aplikasi jaringan komputer yang ada, maka perlu dipastikan bahwa semua data yang diterima oleh aplikasi tersebut dari pihak lain adalah data yang valid dan tidak berbahaya.

Sebuah *Application Firewall* adalah sebuah sistem yang akan memeriksa seluruh data yang akan diterima oleh sebuah aplikasi jaringan komputer. Paket-paket data yang diterima dari pihak lain akan disatukan untuk kemudian diperiksa apakah data yang dikirimkan berbahaya atau tidak. Apabila ditemukan data yang berbahaya untuk sebuah aplikasi, maka data tersebut akan dibuang, sehingga tidak membahayakan sistem jaringan komputer secara keseluruhan.

Pada umumnya *Application Firewall* diletakkan pada setiap *host* untuk melindungi aplikasi jaringan komputer yang ada pada *host* tersebut. Kekurangan dari sistem ini adalah diperlukannya sumber daya komputasi yang sangat besar untuk menyatukan kemudian memeriksa seluruh paket yang diterima oleh sebuah *host*. Selain itu, dengan adanya sistem ini, maka waktu yang dibutuhkan agar sebuah data dapat sampai ke aplikasi yang dituju akan semakin lama, karena harus melalui pemeriksaan terlebih dahulu. Oleh karena itu, sistem ini tidak cocok untuk diimplementasikan pada sistem yang mengharuskan data dikirim dan diterima secara *real-time*.

Bentuk lain dari *Application Firewall* adalah *Network Proxy*. Tugas sebuah *proxy* adalah untuk mewakili klien-klien yang ada untuk melakukan hubungan dengan server-server

tujuan. Bagi klien yang akan melakukan koneksi ke sebuah server, *proxy* adalah server tersebut. Sedangkan bagi server yang dihubungi, *proxy* adalah klien-nya. Dengan menggunakan *proxy* akan lebih sulit bagi pihak luar untuk melakukan serangan ke jaringan komputer internal, karena pihak tersebut hanya dapat berhubungan dengan *proxy* tersebut, sehingga pihak luar tersebut tidak dapat mengetahui lokasi sebenarnya dari server yang dihubungnya. Selain itu sebuah *proxy* juga dapat memiliki sederetan *access-list* yang akan mengatur hak akses klien ke server. *Network Proxy* juga dapat difungsikan terbalik, menjadi sebuah *reverse proxy*. Dengan *reverse proxy* tujuan utamanya adalah untuk melindungi server-server di jaringan internal. Karena semua *request* dari klien eksternal akan diterima oleh *reverse proxy*, maka paket-paket *request* yang berbahaya bagi server akan tersaring dan tidak berbahaya bagi server internal organisasi.

Kelemahan dari *proxy* adalah antara klien dan server tidak memiliki hubungan langsung. Oleh karena itu, *proxy* tidak dapat digunakan pada protokol-protokol ataupun aplikasi yang membutuhkan interaksi langsung antara klien dan server.

Penjelasan lebih lanjut mengenai hal ini dapat dibaca pada referensi [GREENE]

III. Mekanisme pertahanan

Metode-metode yang dapat diterapkan untuk membuat jaringan komputer menjadi lebih aman, antara lain:

- IDS / IPS

Intrusion Detection System (IDS) dan *Intrusion Prevention System* (IPS) adalah sistem yang banyak digunakan untuk mendeteksi dan melindungi sebuah sistem keamanan dari serangan oleh pihak luar maupun dalam.

Sebuah IDS dapat berupa IDS berbasis jaringan komputer atau berbasis host. Pada IDS berbasis jaringan komputer, IDS akan menerima kopi paket yang ditujukan pada sebuah *host* untuk kemudian memeriksa paket-paket tersebut. Apabila ternyata ditemukan paket yang berbahaya, maka IDS akan memberikan peringatan pada pengelola sistem. Karena paket yang diperiksa hanyalah salinan dari paket yang asli, maka sekalipun ditemukan paket yang berbahaya, paket tersebut akan tetap mencapai *host* yang ditujunya.

Sebuah IPS bersifat lebih aktif daripada IDS. Bekerja sama dengan *firewall*, sebuah IPS dapat memberikan keputusan apakah sebuah paket dapat diterima atau tidak oleh sistem. Apabila IPS menemukan bahwa paket yang dikirimkan adalah paket yang berbahaya, maka IPS akan memberitahu *firewall* sistem untuk menolak paket data tersebut.

Dalam membuat keputusan apakah sebuah paket data berbahaya atau tidak, IDS dan IPS dapat menggunakan metode :

- *Signature-based Intrusion Detection System*. Pada metode ini, telah tersedia daftar *signature* yang dapat digunakan untuk menilai apakah paket yang dikirimkan berbahaya atau tidak. Sebuah paket data akan dibandingkan dengan daftar yang sudah ada. Metode ini akan melindungi sistem dari jenis-jenis serangan yang sudah diketahui sebelumnya. Oleh karena itu, untuk tetap menjaga keamanan sistem jaringan komputer, data *signature* yang ada harus tetap *ter-update*.
- *Anomaly-based Intrusion Detection System*. Pada metode ini, pengelola jaringan harus melakukan konfigurasi terhadap IDS dan IPS, sehingga IDS dan IPS dapat mengetahui pola paket seperti apa saja yang akan ada pada sebuah sistem jaringan komputer. Sebuah paket anomali adalah paket yang tidak sesuai dengan kebiasaan jaringan komputer tersebut. Apabila IDS dan IPS menemukan ada anomali pada paket yang diterima atau dikirimkan, maka IDS dan IPS akan memberikan peringatan pada pengelola jaringan (IDS) atau akan menolak paket tersebut untuk diteruskan (IPS). Untuk metode ini, pengelola jaringan harus terus-menerus memberi tahu IDS dan IPS bagaimana lalu lintas data yang normal pada sistem jaringan

komputer tersebut, untuk menghindari adanya salah penilaian oleh IDS atau IPS. Penggunaan IDS dan IPS pada sistem jaringan komputer dapat mempergunakan sumber daya komputasi yang cukup besar, dan khusus untuk IPS, dengan adanya IPS maka waktu yang dibutuhkan sebuah paket untuk dapat mencapai *host* tujuannya menjadi semakin lama, tidak cocok untuk aplikasi-aplikasi yang membutuhkan pengiriman data secara *real-time*. Selain itu IDS dan IPS masih membuka kesempatan untuk terjadinya *false-positive* dimana sebuah paket yang aman dinyatakan berbahaya dan *false-negative* dimana paket yang berbahaya dinyatakan aman. Untuk mengurangi tingkat *false-positive* dan *false-negative*, perlu dilakukan pembaharuan secara rutin terhadap sebuah IDS dan IPS. Dalam implementasinya, IDS adalah sebuah unit *host* yang terhubung pada sebuah *hub/switch* dan akan menerima salinan dari paket-paket yang diproses oleh *hub/switch* tersebut. Sedangkan untuk IPS biasanya diletakkan pada unit yang sama dengan *firewall* dan akan memproses paket-paket yang lewat melalui *firewall* tersebut.

Sedangkan pada IDS berbasis *host*, IDS akan memeriksa aktivitas *system call*, catatan kegiatan dan perubahan pada sistem berkas pada *host* tersebut untuk mencari anomali atau keanehan yang menandakan adanya usaha dari pihak luar untuk menyusup ke dalam sistem. IDS berbasis *host* akan membantu pengelola sistem untuk melakukan *audit trail* terhadap sistem apabila terjadi penyusupan dalam sistem.

Untuk mempelajari lebih jauh mengenai hal ini, dapat dilihat di referensi [IPS] dan [IDS]

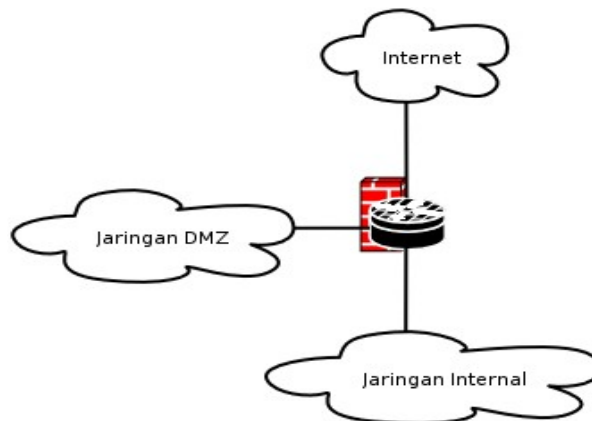
- Network Topology

Selain permasalahan aplikasi yang akan mempergunakan jaringan komputer, topologi jaringan komputer juga memiliki peranan yang sangat penting dalam keamanan jaringan komputer. Pembagian kelompok komputer sesuai dengan tugas yang akan diembannya adalah suatu hal yang perlu dilakukan. Dengan adanya pembagian kelompok-kelompok jaringan komputer, apabila terjadi gangguan keamanan pada sebuah kelompok jaringan komputer, tidak akan dengan mudah menyebar ke kelompok jaringan komputer lainnya. Selain itu metode keamanan yang diterapkan pada setiap kelompok jaringan komputer juga bisa berbeda-beda, sesuai dengan perannya masing-masing.

Secara mendasar, sebuah jaringan komputer dapat dibagi atas kelompok jaringan eksternal (Internet atau pihak luar), kelompok jaringan internal dan kelompok jaringan diantaranya atau yang biasa disebut sebagai *DeMilitarized Zone (DMZ)*. Komputer-komputer pada jaringan DMZ, adalah komputer-komputer yang perlu dihubungi secara langsung oleh pihak luar. Contohnya adalah *web-server*, *mail exchange server* dan *name server*. Komputer-komputer pada jaringan DMZ harus dipersiapkan secara khusus, karena mereka akan terbuka dari pihak luar. Aplikasi yang dipergunakan pada *host-host* pada DMZ harus merupakan aplikasi yang aman, terus menerus dipantau dan dilakukan *update* secara reguler. Aturan-aturan yang berlaku adalah sebagai berikut :

- Pihak luar hanya dapat berhubungan dengan *host-host* yang berada pada jaringan DMZ, sesuai dengan kebutuhan yang ada. Secara *default* pihak luar tidak bisa melakukan hubungan dengan *host-host* pada jaringan DMZ.
- *Host-host* pada jaringan DMZ secara *default* tidak dapat melakukan hubungan dengan *host-host* pada jaringan internal. Koneksi secara terbatas dapat dilakukan sesuai dengan kebutuhan.
- *Host-host* pada jaringan internal dapat melakukan koneksi secara bebas baik ke jaringan luar maupun ke jaringan DMZ. Pada beberapa implementasi, untuk meningkatkan keamanan, *host-host* pada jaringan internal tidak dapat melakukan koneksi ke jaringan luar, melainkan melalui perantara *host* pada jaringan DMZ, sehingga pihak luar tidak mengetahui keberadaan *host-host* pada jaringan komputer internal.

Bentuk topologi jaringan diatas, lebih jelasnya dapat dilihat pada gambar dibawah ini :

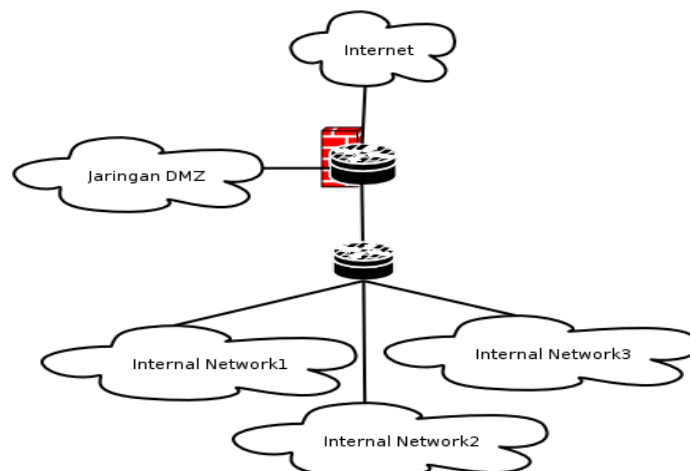


Gambar 1. Topologi Jaringan Sederhana

Selain meningkatkan keamanan, pembagian seperti ini juga menguntungkan karena penggunaan alamat IP yang lebih sedikit. Hanya *host-host* pada jaringan DMZ saja yang butuh untuk mempergunakan alamat IP publik internet, sedangkan untuk *host-host* jaringan internal bisa mempergunakan alamat IP privat. Hal ini terutama sangat menguntungkan bagi organisasi-organisasi yang hanya mendapatkan sedikit alokasi alamat IP yang dapat digunakan oleh organisasi tersebut dari *service provider* yang digunakan.

Kelemahan dari implementasi aturan-aturan yang ketat seperti ini adalah ada beberapa aplikasi yang tidak dapat digunakan. Sebagai contoh, untuk dapat melakukan *video-conference* ataupun *audio-conference* diperlukan koneksi langsung antara satu *host* dengan *host* lainnya. Dengan implementasi dimana pihak luar tidak dapat berhubungan dengan *host* pada jaringan internal, maka *host* pada jaringan internal tidak dapat melakukan *video-conference*.

Selain itu, untuk organisasi yang cukup besar, adanya pembagian lebih lanjut pada jaringan komputer internal akan lebih baik. Perlu dibuat sebuah panduan mengenai interaksi apa saja yang mungkin dilakukan dan dibutuhkan oleh satu bagian organisasi dengan bagian organisasi lainnya melalui jaringan komputer. Setelah panduan dibuat, maka interaksi-interaksi yang tidak diperlukan antar komputer pada jaringan yang berbeda dapat dibatasi. Aturan dasar yang saat ini banyak digunakan adalah untuk menutup semua pintu (*port*) yang ada dan buka hanya yang dibutuhkan dan aman saja.



Gambar 2. Topologi Jaringan Organisasi besar

Perlu diingat, semakin banyak pembagian kelompok jaringan komputer yang ada, maka akan semakin meningkatkan kompleksitas pemeliharaan jaringan komputer. Selain itu semakin banyak pembagian kelompok juga akan meningkatkan latensi koneksi antara satu *host* di sebuah kelompok jaringan dengan *host* lain di kelompok jaringan lainnya.

– Port Scanning

Metode *Port Scanning* biasanya digunakan oleh penyerang untuk mengetahui *port* apa saja yang terbuka dalam sebuah sistem jaringan komputer. Tetapi metode yang sama juga dapat digunakan oleh pengelola jaringan komputer untuk menjaga jaringan komputernya.

Sebuah *port* yang terbuka menandakan adanya aplikasi jaringan komputer yang siap menerima koneksi. Aplikasi ini dapat menjadi pintu masuk penyerang ke dalam sistem jaringan komputer sebuah organisasi. Oleh karena itu sangat penting bagi seorang pengelola jaringan komputer untuk tahu secara pasti, aplikasi jaringan komputer apa saja yang berjalan dan siap menerima koneksi pada sebuah *host*. Apabila ditemukan bahwa ada *port* yang terbuka dan tidak sesuai dengan perencanaan yang ada, maka aplikasi yang berjalan pada *port* tersebut harus segera dimatikan agar tidak menjadi lubang keamanan.

Cara kerja *port scanner* adalah dengan cara mengirimkan paket inisiasi koneksi ke setiap *port* yang sudah ditentukan sebelumnya. Apabila ternyata *port scanner* menerima jawaban dari sebuah *port*, maka ada aplikasi yang sedang bekerja dan siap menerima koneksi pada *port* tersebut.

Port Scanning sebagai bentuk serangan

Karena implementasinya yang cukup mudah dan informasinya yang cukup berguna, maka sering kali *port scanning* dilakukan sebagai tahap awal sebuah serangan. Untuk dapat melakukan penyerangan, seorang *cracker* perlu mengetahui aplikasi apa saja yang berjalan dan siap menerima koneksi dari lokasinya berada. *Port Scanner* dapat memberikan informasi ini.

Untuk dapat mendeteksi adanya usaha untuk melakukan *scanning* jaringan, seorang pengelola jaringan dapat melakukan *monitoring* dan mencari paket-paket IP yang berasal dari sumber yang sama dan berusaha melakukan akses ke sederetan *port*, baik yang terbuka maupun yang tertutup. Apabila ditemukan, pengelola jaringan dapat melakukan konfigurasi *firewall* untuk memblokir IP sumber serangan. Hal ini perlu dilakukan secara berhati-hati, karena apabila dilakukan tanpa ada toleransi, metode ini dapat mengakibatkan seluruh jaringan Internet terblokir oleh *firewall* organisasi. Oleh sebab itu, perlu ada keseimbangan antara keamanan dan performa dalam usaha mendeteksi kegiatan *port scanning* dalam sebuah jaringan komputer.

Lebih lanjut mengenai *Port Scanning* dapat dilihat pada referensi [BRADLEY]

– Packet Fingerprinting

Karena keunikan setiap vendor peralatan jaringan komputer dalam melakukan implementasi protokol TCP/IP, maka paket-paket data yang dikirimkan setiap peralatan menjadi unik peralatan tersebut. Dengan melakukan *Packet Fingerprinting*, kita dapat mengetahui peralatan apa saja yang ada dalam sebuah jaringan komputer. Hal ini sangat berguna terutama dalam sebuah organisasi besar dimana terdapat berbagai jenis peralatan jaringan komputer serta sistem operasi yang digunakan. Setiap peralatan dan sistem operasi memiliki karakteristik serta kelemahannya masing-masing, oleh karena itu, sangat penting bagi pengelola jaringan komputer untuk dapat mengetahui peralatan dan sistem operasi apa saja yang digunakan dalam organisasi tersebut. Dengan mengetahui peralatan jenis apa atau

sistem operasi apa saja yang ada pada sebuah organisasi, pengelola jaringan komputer dapat lebih siap dalam melakukan pengamanan jaringan komputer organisasi tersebut.

Untuk menentukan tipe peralatan atau sistem operasi ada, sebuah peralatan *fingerprinting* akan melihat bagaimana peralatan jaringan komputer atau sistem operasi yang bersangkutan memberikan nilai-nilai awal pada beberapa bagian di *header* IP. Bagian-bagian tersebut adalah:

- Time-to-Live – Setiap peralatan jaringan komputer mempergunakan nilai awal yang berbeda-beda dalam memberikan nilai ke bagian time-to-live pada *header* IP.
- Window-size - Setiap peralatan jaringan komputer, mempergunakan ukuran *TCP windows* yang berbeda-beda.
- bit DF pada paket – Apakah peralatan jaringan komputer yang mengirimkan paket tersebut mempergunakan bit DF (*dont' t fragment*), pada awal koneksi. Tidak terlalu berguna dalam membedakan satu peralatan dengan peralatan lainnya.
- bit *Type of Service* – Jenis layanan apa yang diberikan oleh sebuah peralatan jaringan komputer pada paket yang dikirimnya. Karena pada banyak implementasi, jenis layanan yang diinginkan, ditentukan oleh protokol atau aplikasi yang sedang berjalan dan bukan oleh sistem operasi atau peralatan yang digunakan, maka penggunaan bit *Type of Service* tidak terlalu berguna dalam membedakan satu peralatan dengan peralatan lainnya.

Setelah mendapatkan informasi-informasi di atas, peralatan *fingerprinting* akan melakukan perbandingan dengan data yang sudah dimiliki sebelumnya.

Fingerprinting dapat dilakukan secara aktif maupun secara pasif. Jika dilakukan secara aktif, analis akan mengirimkan sebuah paket *request* yang kemudian akan dibalas oleh *host* target. Paket balasan dari *host* target inilah yang kemudian dianalisa. Sedangkan jika dilakukan secara pasif, maka analis akan menunggu *host* target mengirimkan paket, kemudian paket tersebut akan dianalisa.

Selain dapat digunakan oleh pengelola jaringan komputer untuk mengamankan jaringan komputer organisasi, metode yang sama sering digunakan oleh pihak-pihak yang ingin mengganggu sebuah jaringan komputer.

Untuk lebih jauh mengenai hal ini, dapat dilihat pada referensi [HONEY]

– Security Information Management

Dalam usaha untuk meningkatkan keamanan jaringan komputer, sebuah organisasi mungkin akan meng-implementasikan beberapa teknologi keamanan jaringan komputer, seperti *firewall*, IDS dan IPS. Semua usaha tersebut dilakukan sehingga keamanan jaringan komputer organisasi tersebut menjadi lebih terjamin.

Namun, dengan semakin banyaknya peralatan jaringan komputer yang di-implementasikan, maka akan semakin banyak pula peralatan yang perlu dikelola. Pengelolaan akan dimulai dari konfigurasi peralatan agar sesuai dengan kebutuhan organisasi. Setelah itu setiap peralatan yang sudah terpasang perlu dipantau, perlu dianalisa apakah sudah berfungsi sesuai dengan rancangan awal. Salah satu bentuk pemantau yang perlu dilakukan adalah memantau *log* dan *alert* yang dihasilkan oleh setiap peralatan. Jumlah *log* dan *alert* yang dihasilkan oleh semua peralatan keamanan jaringan komputer yang terpasang dapat berukuran sangat besar. Akan membutuhkan banyak waktu pengelola jaringan komputer untuk menganalisa seluruh *log* dan *alert* yang ada, termasuk didalamnya adalah melakukan pencarian dimana *log* atau *alert* tersebut tersimpan.

Salah satu penyebab utama dari kegagalan sistem keamanan jaringan komputer adalah kesalahan pengelola dalam melakukan analisa informasi yang dihasilkan masing-masing perangkat keamanan jaringan komputer. Kesalahan analisa dapat menyebabkan pengelola lambat, salah atau tidak terarah dalam menghadapi serangan yang sedang berlangsung.

Oleh karena itu, salah satu alat bantu yang dapat digunakan oleh pengelola jaringan komputer adalah *Security Information Management* (SIM). SIM berfungsi untuk menyediakan seluruh informasi yang terkait dengan pengamanan jaringan komputer secara terpusat. Dengan menggunakan SIM, pengelola dapat dengan mudah mengetahui kondisi seluruh peralatan yang dimilikinya dan melakukan identifikasi serangan yang ada. Pada fungsi paling dasarnya, SIM akan mengumpulkan semua *log* dan *alert* yang dihasilkan oleh semua peralatan keamanan jaringan komputer yang ada ke dalam satu tempat, sehingga mempermudah pengelolaan. Pada perkembangannya SIM tidak hanya berfungsi untuk mengumpulkan data-data dari semua peralatan keamanan jaringan komputer tapi juga memiliki kemampuan untuk analisa data melalui teknik korelasi dan *query* data terbatas sehingga menghasilkan peringatan dan laporan yang lebih lengkap dari masing-masing serangan.

Dengan mempergunakan SIM, pengelola jaringan komputer dapat mengetahui secara lebih cepat bahwa sedang ada serangan dan dapat melakukan penanganan yang lebih terarah, sehingga keamanan jaringan komputer organisasi tersebut lebih terjamin.

Lebih lanjut mengenai topik ini dapat dilihat pada referensi [WUI]

IV. Jenis-jenis Ancaman

Berikut ini akan dijelaskan beberapa tipe-tipe serangan yang dapat dilancarkan oleh pihak-pihak tertentu terhadap sebuah jaringan komputer:

- DOS/DDOS

Denial of Services dan *Distributed Denial of Services* adalah sebuah metode serangan yang bertujuan untuk menghabiskan sumber daya sebuah peralatan jaringan komputer sehingga layanan jaringan komputer menjadi terganggu.

Salah satu bentuk serangan ini adalah 'SYN Flood Attack', yang mengandalkan kelemahan dalam sistem '*three-way-handshake*'. '*Three-way-handshake*' adalah proses awal dalam melakukan koneksi dengan protokol TCP. Proses ini dimulai dengan pihak klien mengirimkan paket dengan tanda SYN. Lalu kemudian pihak server akan menjawab dengan mengirimkan paket dengan tanda SYN dan ACK. Terakhir, pihak klien akan mengirimkan paket ACK. Setelah itu, koneksi akan dinyatakan terbuka, sampai salah satu pihak mengirimkan paket FIN atau paket RST atau terjadi *connection time-out*. Dalam proses '*three-way-handshake*', selain terjadi inisiasi koneksi, juga terjadi pertukaran data-data parameter yang dibutuhkan agar koneksi yang sedang dibuat dalam berjalan dengan baik.

Dalam serangan ini, sebuah *host* akan menerima paket inisiasi koneksi (Paket dengan flag SYN) dalam jumlah yang sangat banyak secara terus menerus. Akibatnya *host* yang sedang diserang akan melakukan alokasi memori yang akan digunakan untuk menerima koneksi tersebut dan karena paket inisiasi terus-menerus diterima maka ruang memori yang dapat digunakan untuk menerima koneksi akan habis. Karena semua ruang memori yang dapat digunakan untuk menerima koneksi sudah habis, maka ketika ada permintaan baru untuk melakukan inisiasi koneksi, *host* ini tidak dapat melakukan alokasi memori sehingga permintaan baru ini tidak dapat dilayani oleh *host* ini. Untuk menghindari pelacakan, biasanya paket serangan yang dikirimkan memiliki alamat IP sumber yang dipalsukan. Untuk menghadapi serangan seperti ini, sistem operasi – sistem operasi modern telah mengimplementasikan metode-metode penanganan, antara lain :

- *Micro-blocks*. Ketika ada sebuah *host* menerima paket inisiasi, maka *host* akan mengalokasikan ruang memori yang sangat kecil, sehingga *host* tersebut bisa menerima koneksi lebih banyak. Diharapkan ruang memori dapat menampung semua koneksi yang dikirimkan, sampai terjadi *connection-time-out*, dimana

koneksi-koneksi yang *stale*, yaitu koneksi yang tidak menyelesaikan proses *'three-way-handshake'* atau sudah lama tidak ada transaksi data, akan dihapuskan dari memori dan memberikan ruang bagi koneksi-koneksi baru. Metode ini tidak terlalu efektif karena bergantung pada kecepatan serangan dilakukan, apabila ternyata kecepatan paket serangan datang lebih cepat daripada lamanya waktu yang perlu ditunggu agar terjadi *connection-time-out* pada paket-paket yang *stale*, maka ruang memori yang dapat dialokasikan akan tetap habis.

- *SYN Cookies*. Ketika menerima paket inisiasi, *host* penerima akan mengirimkan paket tantangan yang harus dijawab pengirim, sebelum *host* penerima mengalokasikan memori yang dibutuhkan. Tantangan yang diberikan adalah berupa paket SYN-ACK dengan nomor urut khusus yang merupakan hasil dari fungsi *hash* dengan input alamat IP pengirim, nomor *port*, dll. Jawaban dari pengirim akan mengandung nomor urut tersebut. Tetapi untuk melakukan perhitungan *hash* membutuhkan sumber-daya komputasi yang cukup besar, sehingga banyak server-server yang aplikasinya membutuhkan kemampuan komputasi tinggi tidak mempergunakan metode ini. Metode ini merubah waktu peng-alokasian memori, yang tadinya pada awal dari proses *'three-way-handshake'*, menjadi diakhir dari proses tersebut. (notes: pada standard TCP/IP yang baru, ditentukan bahwa diperlukan cara yang lebih baik untuk menentukan urut paket, sehingga sulit untuk ditebak. Jadi kemungkinan secara *default*, metode ini akan digunakan pada seluruh peralatan jaringan komputer atau sistem operasi yang ada).
- *RST Cookies*. Mirip dengan *SYN Cookies*, hanya tantangan yang dikirimkan *host* penerima ke pengirim adalah sebuah paket yang salah. Apabila pengirim adalah pengirim yang valid, maka pengirim akan mengirimkan paket RST lalu mengulang kembali koneksi. Ketika penerima menerima paket RST, *host* tersebut tahu bahwa pengirim adalah valid dan akan menerima koneksi dari pengirim dengan normal. Karena ada masalah dengan implementasi lapisan TCP/IP, metode ini kemungkinan tidak kompatibel dengan beberapa sistem operasi. Metode ini merubah waktu peng-alokasian memori, yang tadinya pada awal dari proses *'three-way-handshake'*, menjadi diakhir dari proses tersebut.

Bentuk lain dari serangan DOS adalah *'Smurf Attack'* yang mempergunakan paket ping *request*. Dalam melakukan penyerangan, penyerang akan mengirimkan paket-paket ping *request* ke banyak *host* dengan merubah alamat IP sumber menjadi alamat *host* yang akan diserang. *Host-host* yang menerima paket ping *request* tersebut akan mengirimkan paket balasan ke alamat IP *host* korban serangan. Untuk serangan dapat mengganggu sistem korban, *host* yang menjawab paket ping *request* harus cukup banyak. Oleh karena itu, biasanya paket ping *request* akan dikirimkan ke alamat *broadcast* dari sebuah kelompok jaringan komputer, sehingga *host-host* pada kelompok jaringan komputer tersebut secara otomatis akan menjawab paket tersebut.

DOS juga dapat dilakukan dengan cara mengirimkan permintaan layanan yang diberikan oleh sebuah *host* secara berlebihan atau terus menerus. Tujuan dari serangan model ini adalah untuk membuat *host* menjadi terlalu sibuk atau kehabisan sumber daya komputasi sehingga tidak dapat melayani permintaan-permintaan lainnya.

Perkembangan lanjutan dari DOS adalah DDOS, dimana *host* yang terlibat dalam serangan lebih dari satu dan tersebar di banyak tempat. Banyaknya *host* yang terlibat dalam serangan akan meningkatkan efek serangan dan mempersulit pihak yang diserang untuk mempertahankan diri ataupun melakukan pelacakan asal serangan. Pada banyak kejadian, *host-host* yang terlibat dalam serangan, tidak semuanya sadar bahwa mereka terlibat dalam sebuah serangan DDOS. *Host-host* tersebut telah disusupi terlebih dahulu oleh penyerang,

sehingga penyerang dapat mempergunakan *host* tersebut untuk melakukan serangan. Penyusupan dapat dilakukan dengan cara mengirimkan *trojan* atau *worm* ke banyak *host*. Mengenai DOS dan DDOS dapat dilihat lebih lanjut pada referensi [MPRC] dan [LOOP]

– Packet Sniffing

Packet Sniffing adalah sebuah metode serangan dengan cara mendengarkan seluruh paket yang lewat pada sebuah media komunikasi, baik itu media kabel maupun radio. Setelah paket-paket yang lewat itu didapatkan, paket-paket tersebut kemudian disusun ulang sehingga data yang dikirimkan oleh sebuah pihak dapat dicuri oleh pihak yang tidak berwenang.

Hal ini dapat dilakukan karena pada dasarnya semua koneksi *ethernet* adalah koneksi yang bersifat *broadcast*, di mana semua *host* dalam sebuah kelompok jaringan akan menerima paket yang dikirimkan oleh sebuah *host*. Pada keadaan normal, hanya *host* yang menjadi tujuan paket yang akan memproses paket tersebut sedangkan *host* yang lainnya akan mengacuhkan paket-paket tersebut. Namun pada keadaan tertentu, sebuah *host* bisa merubah konfigurasi sehingga *host* tersebut akan memproses semua paket yang dikirimkan oleh *host* lainnya.

Cukup sulit untuk melindungi diri dari gangguan ini karena sifat dari *packet sniffing* yang merupakan metode pasif (pihak penyerang tidak perlu melakukan apapun, hanya perlu mendengar saja). Namun ada beberapa hal yang bisa dilakukan untuk mengatasi hal ini, yaitu:

- Secara rutin melakukan pemeriksaan apakah ada *host* di jaringan kita yang sedang dalam mode *promiscuous*, yaitu sebuah mode dimana *host* tersebut akan memproses semua paket yang diterima dari media fisik. Akan tetapi hal ini hanya akan melindungi diri kita terhadap *packet sniffer* yang berada pada satu kelompok jaringan dengan kita. Penyerang yang melakukan *sniffing* dari luar jaringan komputer kita tidak akan terdeteksi dengan menggunakan metode ini.
- Mempergunakan SSL atau TLS dalam melakukan pengiriman data. Ini tidak akan mencegah *packet sniffer* untuk mencuri paket yang dikirimkan, akan tetapi paket-paket yang dicuri tidak bisa dipergunakan karena dikirimkan dengan menggunakan format yang terenkripsi.
- Melakukan koneksi VPN, sehingga tetap bisa mempergunakan aplikasi yang tidak mendukung SSL atau TLS dengan aman.

***Packet Sniffing* sebagai tools pengelola jaringan**

Sebenarnya selain sebagai menjadi alat untuk melakukan kejahatan, *packet sniffer* juga bisa digunakan sebagai alat pertahanan. Dengan melakukan analisa paket-paket yang melalui sebuah media jaringan komputer, pengelola dapat mengetahui apabila ada sebuah *host* yang mengirimkan paket-paket yang tidak normal, misalnya karena terinfeksi virus. Sebuah IDS juga pada dasarnya adalah sebuah *packet sniffer* yang bertugas untuk mencari *host* yang mengirimkan paket-paket yang berbahaya bagi keamanan. Selain itu *packet sniffer* juga bisa menjadi alat untuk melakukan analisa permasalahan yang sedang dihadapi sebuah jaringan komputer. Misalkan ketika sebuah *host* tidak dapat berhubungan dengan *host* lainnya yang berada pada kelompok jaringan yang berbeda, maka dengan *packet sniffer*, pengelola jaringan komputer dapat melakukan penelusuran dimana permasalahan koneksi itu terletak.

– IP Spoofing

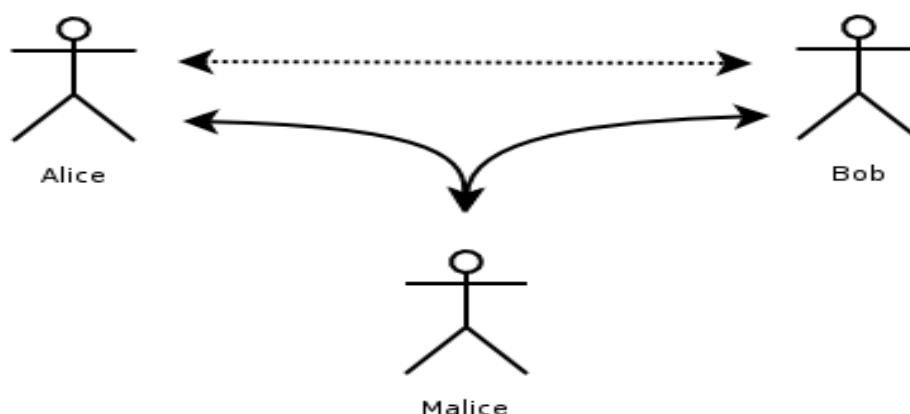
IP Spoofing adalah sebuah model serangan yang bertujuan untuk menipu seseorang. Serangan ini dilakukan dengan cara mengubah alamat asal sebuah paket, sehingga dapat melewati perlindungan *firewall* dan menipu *host* penerima data. Hal ini dapat dilakukan

karena pada dasarnya alamat IP asal sebuah paket dituliskan oleh sistem operasi *host* yang mengirimkan paket tersebut. Dengan melakukan *raw-socket-programming*, seseorang dapat menuliskan isi paket yang akan dikirimkan setiap bit-nya sehingga untuk melakukan pemalsuan data dapat dilakukan dengan mudah.

Salah satu bentuk serangan yang memanfaatkan metode *IP Spoofing* adalah '*man-in-the-middle-attack*'. Pada serangan ini, penyerang akan berperan sebagai orang ditengah antara dua pihak yang sedang berkomunikasi. Misalkan ada dua pihak yaitu pihak A dan pihak B lalu ada penyerang yaitu C. Setiap kali A mengirimkan data ke B, data tersebut akan dicegat oleh C, lalu C akan mengirimkan data buatannya sendiri ke B, dengan menyamar sebagai A. Paket balasan dari B ke A juga dicegat oleh C yang kemudian kembali mengirimkan data 'balasan' buatannya sendiri ke A. Dengan cara ini, C akan mendapatkan seluruh data yang dikirimkan antara A dan B, tanpa diketahui oleh A maupun C.

Untuk mengatasi serangan yang berdasarkan *IP Spoofing*, sebuah sistem operasi harus dapat memberikan nomor-urut yang acak ketika menjawab inisiasi koneksi dari sebuah *host*. Dengan nomor urut paket yang acak, akan sangat sulit bagi seorang penyerang untuk dapat melakukan pembajakan transmisi data.

Selain itu, untuk mengatasi model serangan '*man-in-the-middle-attack*', perlu ada sebuah metode untuk melakukan otentikasi *host* yang kita hubungi. Otentikasi dapat berupa *digital-certificate* yang eksklusif dimiliki oleh *host* tersebut.



Gambar 3. Man in the Middle Attack

Pada gambar 3, Alice dan Bob berpikir bahwa mereka saling berkomunikasi, dimana sebenarnya, mereka berkomunikasi dengan Malice.

Konfigurasi *firewall* yang tepat juga dapat meningkatkan kemampuan jaringan komputer dalam menghadapi *IP Spoofing*. *Firewall* harus dibuat agar dapat menolak paket-paket dengan alamat IP sumber jaringan internal yang masuk dari *interface* yang terhubung dengan jaringan eksternal.

Lebih lanjut mengenai topik ini dapat dilihat pada referensi [TANASE] dan [FOX]

- DNS Forgery

Salah satu cara yang dapat dilakukan oleh seseorang untuk mencuri data-data penting orang lain adalah dengan cara melakukan penipuan. Salah satu bentuk penipuan yang bisa dilakukan adalah penipuan data-data DNS. DNS adalah sebuah sistem yang akan

menterjemahkan nama sebuah situs atau *host* menjadi alamat IP situs atau *host* tersebut. Cara kerja DNS cukup sederhana, yaitu sebuah *host* mengirimkan paket (biasanya dengan tipe UDP) yang pada *header* paket tersebut berisikan alamat *host* penanya, alamat DNS *resolver*, pertanyaan yang diinginkan serta sebuah nomor identitas. DNS *resolver* akan mengirimkan paket jawaban yang sesuai ke penanya. Pada paket jawaban tersebut terdapat nomor identitas, yang dapat dicocokkan oleh penanya dengan nomor identitas yang dikirimnya. Oleh karena cara kerja yang sederhana dan tidak adanya metode otentikasi dalam sistem komunikasi dengan paket UDP, maka sangat memungkinkan seseorang untuk berpura-pura menjadi DNS *resolver* dan mengirimkan paket jawaban palsu dengan nomor identitas yang sesuai ke penanya sebelum paket jawaban dari DNS *resolver* resmi diterima oleh penanya. Dengan cara ini, seorang penyerang dapat dengan mudah mengarahkan seorang pengguna untuk melakukan akses ke sebuah layanan palsu tanpa diketahui pengguna tersebut. Sebagai contoh, seorang penyerang dapat mengarahkan seorang pengguna *Internet Banking* untuk melakukan akses ke situs *Internet Banking* palsu yang dibuatnya untuk mendapatkan data-data pribadi dan kartu kredit pengguna tersebut. Untuk dapat melakukan gangguan dengan memalsukan data DNS, seseorang membutuhkan informasi-informasi di bawah ini :

- Nomor identitas pertanyaan (16 bit)
- Port tujuan pertanyaan
- Alamat IP DNS *resolver*
- Informasi yang ditanyakan
- Waktu pertanyaan.

Pada beberapa implementasi sistem operasi, informasi diatas yang dibutuhkan seseorang untuk melakukan penipuan data DNS bisa didapatkan. Kunci dari serangan tipe ini adalah, jawaban yang diberikan DNS *resolver* palsu harus diterima oleh penanya sebelum jawaban yang sebenarnya diterima, kecuali penyerang dapat memastikan bahwa penanya tidak akan menerima jawaban yang sebenarnya dari DNS *resolver* yang resmi.

DNS Cache Poisoning

Bentuk lain serangan dengan menggunakan DNS adalah *DNS Cache Poisoning*. Serangan ini memanfaatkan cache dari setiap server DNS yang merupakan tempat penyimpanan sementara data-data domain yang bukan tanggung jawab server DNS tersebut. Sebagai contoh, sebuah organisasi 'X' memiliki server DNS (ns.x.org) yang menyimpan data mengenai domain 'x.org'. Setiap komputer pada organisasi 'X' akan bertanya pada server 'ns.x.org' setiap kali akan melakukan akses Internet. Setiap kali server ns.x.org menerima pertanyaan diluar domain 'x.org', server tersebut akan bertanya pada pihak otoritas domain. Setelah mendapatkan jawaban yang dibutuhkan, jawaban tersebut akan disimpan dalam *cache*, sehingga jika ada pertanyaan yang sama, server 'ns.x.org' dapat langsung memberikan jawaban yang benar. Dengan tahapan-tahapan tertentu, seorang penyerang dapat mengirimkan data-data palsu mengenai sebuah domain yang kemudian akan disimpan di *cache* sebuah server DNS, sehingga apabila server tersebut menerima pertanyaan mengenai domain tersebut, server akan memberikan jawaban yang salah. Patut dicatat, bahwa dalam serangan ini, data asli server DNS tidak mengalami perubahan sedikitpun. Perubahan data hanya terjadi pada *cache* server DNS tersebut.

Cara yang paling efektif dalam menghadapi serangan yang merubah DNS server adalah dengan melakukan otentikasi *host* yang akan kita hubungi. Model otentikasi yang banyak digunakan saat ini adalah dengan mempergunakan *digital certificate*. Dengan *digital certificate*, seseorang dapat dengan yakin bahwa *host* yang dia akses adalah *host* yang sebenarnya.

Mengenai Topik ini dapat dilihat pada referensi [SANF]

V. Studi Kasus

Berikut ini adalah dua studi kasus implementasi pengamanan jaringan komputer. Studi kasus pertama menggambarkan suatu kegagalan dalam usaha pengamanan jaringan komputer sementara studi kasus kedua merupakan sebuah kajian mengenai bagaimana jaringan komputer untuk suatu Usaha Kecil Menengah (UKM) dapat diimplementasikan.

- **Pengumuman Hasil Ujian Sistem Penerimaan Mahasiswa Baru (SPMB)**

Pada setiap awal tahun ajaran, Universitas Indonesia mendapatkan kehormatan untuk menjadi tempat *hosting* pengumuman penerimaan mahasiswa baru seluruh Indonesia. Sebagai sebuah *event* yang sangat penting, diperlukan persiapan yang sangat matang. Salah satu persiapan yang terpenting adalah persiapan jaringan komputer yang akan digunakan untuk menjamin tersedianya akses yang memadai bagi semua pihak yang berkepentingan dengan pengumuman tersebut, terutama pada saat-saat puncak, yaitu malam pertama pengumuman hasil SPMB.

Setelah beberapa tahun menjalani peranan tersebut, ada beberapa permasalahan yang dihadapi ketika acara sedang berlangsung, yaitu :

- Habisnya sumber daya pada *web server* yang digunakan, karena banyaknya *request* yang diterima, termasuk didalamnya usaha serangan DOS (*Denial of Service*), sedangkan kapasitas *web server* yang ada tidak sebanding dengan *request* yang masuk. Dalam beberapa kesempatan, *web server* yang digunakan kehabisan sumber daya memori, sehingga *web server* mengalami *crash*, yang selanjutnya dapat merusak komponen lain dari *web server* tersebut seperti media penyimpanan *hard-disk*. Jika hal ini terjadi, maka waktu untuk melakukan *recovery* akan cukup lama. Selama proses *recovery* tersebut, akses ke situs pengumuman hasil SPMB tidak dapat dilayani.
- *Bandwidth* koneksi Internet yang habis didominasi oleh beberapa pihak tertentu yang melakukan *mirroring* data. Walaupun hal ini tidak dilarang, tapi hal ini akan merugikan pengguna-pengguna lain yang tidak dapat melakukan akses karena *bandwidth* yang ada sudah habis digunakan. Seringkali dalam suatu kurun waktu, *bandwidth* yang ada habis untuk melayani satu klien saja.
- Akses data yang membutuhkan waktu cukup lama, karena terbatasnya sumber daya yang ada dan penggunaan struktur basis data penyimpanan data yang kompleks.

Berdasarkan pengalaman-pengalaman yang didapatkan dari tahun-tahun sebelumnya, maka persiapan-persiapan yang dilakukan pada tahun ini adalah :

- Penggunaan teknologi *virtual server* yang memungkinkan penggunaan beberapa *web server* dan melakukan pembagian beban di antaranya. Sebuah *virtual server* adalah sebuah *host* yang akan menerima koneksi dari klien-klien dan mengatur ke *web server* yang mana koneksi tersebut akan diarahkan. Pada *virtual server* tersebut juga akan dicatat jumlah koneksi yang sedang ditangani setiap *web server*, sehingga *virtual server* tersebut dapat membagi beban dengan sebaik mungkin. Ada beberapa algoritma yang dapat digunakan untuk membagi beban *web server*, antara lain membagi koneksi secara *round-robin* atau *least-connection*. Ada juga algoritma yang bisa memberikan bobot lebih pada server-server dengan kemampuan lebih dibandingkan dengan server lainnya. Algoritma lainnya adalah membagi koneksi berdasarkan lokasi klien yang melakukan akses. Untuk masalah konektivitas, *virtual server* juga memberikan beberapa pilihan, yaitu :
 - *Virtual server* bertindak sebagai *router*, dimana semua koneksi masuk dan keluar ke server akan melalui *host* ini. Hanya *host virtual server* yang

memakai alamat IP publik internet. Sedangkan untuk server-server tujuan akan mempergunakan alamat IP privat.

- *Virtual server* hanya bertindak sebagai penerus koneksi saja (network bridge), sedangkan server-server tujuan memiliki koneksi langsung sendiri. Pada skema ini, koneksi masuk akan melalui *virtual server* sedangkan koneksi keluar tidak. Baik *virtual server* maupun server-server tujuan akan mempergunakan alamat IP publik.
- *Virtual server* sebagai pengatur beban saja. Skema ini mirip dengan skema sebelumnya. Bedanya pada skema ini, server-server tujuan bisa terdapat pada kelompok jaringan komputer yang berbeda, bahkan bisa terdapat pada lokasi fisik yang berbeda.

Pada *host* tersebut juga diimplementasikan *traffic shaping* yang berguna untuk mengendalikan kecepatan *request* yang masuk ke setiap *web server*. Hal ini untuk mencegah sebuah *web server* menerima *request* yang terlalu banyak sehingga beban *web server* tersebut terlalu berat.

Dengan penggunaan *virtual server* dan *traffic shaping*, diharapkan sumber daya yang ada akan dapat menghadapi semua *request* yang masuk dan serangan DOS yang mungkin terjadi.

- Karena sifat data yang hanya perlu akses baca saja, maka dilakukan perubahan sistem penyimpanan data dari mempergunakan basis data menjadi *Lightweight Directory Access Protocol* (LDAP). Dengan LDAP akses baca data menjadi lebih cepat dan juga membutuhkan sumber daya memori maupun sumber daya komputasi yang lebih rendah dibandingkan dengan mempergunakan basis data. Hal ini dapat terjadi karena struktur data yang digunakan untuk menyimpan data dengan LDAP cukup sederhana. Dalam LDAP, data disimpan dalam struktur pohon, sehingga untuk melakukan pencarian data dapat dilakukan dengan cepat.

Selain itu untuk melakukan pengisian ulang data juga lebih cepat, sehingga apabila terjadi kerusakan pada data, proses perbaikan dapat dilakukan dengan cepat.

- Perbaikan *interface* aplikasi, yang mencegah agar tidak ada pengguna yang bisa melakukan *mirroring* data. Perbaikan mencakup penambahan kode rahasia yang perlu dimasukkan pengguna setiap kali pengguna ingin melakukan akses data. Kode rahasia ini ditampilkan dalam bentuk gambar di halaman situs yang diakses pengguna saat melakukan *request* dan kode ini akan berubah setiap kali pengguna melakukan akses.
- Penggunaan *Live-CD* linux, sehingga menghindari kemungkinan kerusakan media penyimpanan *hard-disk*. Dengan menggunakan hal ini juga mempermudah dalam melakukan duplikasi *web server*. Kemudahan ini akan sangat berguna ketika akan melakukan penambahan *web server* dalam waktu yang singkat.

Pada waktu pengumuman tiba, dideteksi bahwa tingkat koneksi *request* sangat rendah, sedangkan kecepatan untuk melakukan akses internet sangat lambat. Dugaan awal adalah adanya terjadi gangguan pada jaringan komunikasi data antara Universitas Indonesia ke jaringan Internet. Pemeriksaan pada peralatan jaringan komunikasi data tidak menunjukkan adanya gangguan. Setelah dilakukan analisa paket yang diterima oleh jaringan komunikasi data Internet Universitas Indonesia, ditemukan ada sebuah *host* di Internet yang mengirimkan paket dengan tipe UDP berukuran kecil tapi dengan jumlah yang sangat banyak dan dengan kecepatan pengiriman yang sangat tinggi. Karena konfigurasi *firewall* pada pintu gerbang Internet Universitas Indonesia, paket UDP tersebut tidak masuk ke dalam jaringan DMZ Universitas Indonesia. Akan tetapi, karena kecepatan pengiriman paket sangat tinggi, *wan router* yang berfungsi sebagai *router* akses Universitas Indonesia, kehabisan sumber daya komputasi dan tidak mampu melakukan proses *routing* paket-paket yang masuk. Akibatnya, sekalipun *bandwidth* yang digunakan tidak banyak, tidak ada paket lain yang dapat masuk ke dalam jaringan DMZ Universitas Indonesia ataupun ke luar ke

jaringan Internet. Oleh karena itu akses ke situs pengumuman tidak dapat dilakukan, walaupun ada paket *request* yang masuk, paket jawaban dari *web server* tidak dapat keluar dari Universitas Indonesia untuk mencapai komputer pengguna.

Sebagai langkah penanganan, *wan router* yang digunakan, diganti dengan yang memiliki sumber daya komputasi lebih tinggi. Selain itu, pihak Universitas Indonesia juga bekerja sama dengan pihak penyedia layanan Internet untuk melakukan pemblokiran terhadap paket-paket UDP yang mengganggu jaringan Internet Universitas Indonesia. Setelah proses penggantian *router* dan pemblokiran paket selesai, layanan situs pengumuman dapat berjalan dengan baik.

Pelajaran yang didapatkan dari studi kasus ini adalah, bahwa untuk untuk menjamin keamanan sebuah sistem, perlu persiapan dari seluruh komponen yang terlibat di dalamnya. Dalam kasus ini, penyelenggara hanya mempersiapkan dari sisi aplikasi yang akan digunakan saja dan tidak ada persiapan dari lingkungan lain yang berhubungan dengan aplikasi tersebut. Selain itu, untuk aplikasi-aplikasi penting, maka sebaiknya dibuat duplikasi layanan tersebut dan diletakkan di beberapa lokasi terpisah. Dengan cara ini, jika satu lokasi mengalami gangguan, layanan masih dapat berjalan dari beberapa lokasi lainnya. Persiapan lainnya adalah membina hubungan yang lebih dekat dengan penyedia layanan jaringan Internet. Hal ini diperlukan karena serangan-serangan seperti yang dihadapi layanan ini hanya dapat dihadapi dengan bantuan pihak-pihak eksternal.

- Usaha Kecil dan Menengah (UKM)

Usaha Kecil dan Menengah sebagai unit kerja yang kecil akan memiliki karakteristik utama :

- Sumber daya manusia yang terbatas
- Kompleksitas jaringan komputer yang rendah
- Dukungan finansial yang terbatas

Dengan karakteristik-karakteristik yang telah disebutkan di atas, maka diperlukan suatu pendekatan yang berbeda dalam mengelola keamanan jaringan komputer jika dibandingkan dengan perusahaan yang besar.

Dengan dukungan finansial yang terbatas, maka pemilihan teknologi menjadi area yang sangat penting. Diperlukan teknologi yang tidak mahal namun berkemampuan tinggi dan tidak membutuhkan biaya perawatan yang tinggi. Karena itu, penggunaan teknologi *open-source* menjadi pilihan yang tepat. Selain memerlukan biaya yang rendah untuk implementasinya, teknologi *open-source* cukup aman sehingga tidak memerlukan pengelolaan yang sulit. Namun pemilihan teknologi juga terkait erat dengan kemampuan sumber daya manusia yang ada. Oleh karena itu, perlu diperhatikan juga kapabilitas yang dimiliki oleh sumber daya manusia yang tersedia, jangan sampai dipilih teknologi yang dimana tidak ada sumber daya manusia yang mampu mengelolanya.

Dengan sumber daya manusia yang terbatas, maka topologi jaringan yang dibentuk harus cukup sederhana, sehingga tidak membutuhkan banyak personel untuk melakukan pengelolaan jaringan komputer. Topologi jaringan komputer yang terlalu kompleks akan membutuhkan banyak peralatan jaringan komputer, yang selain akan membutuhkan biaya lebih tinggi, juga akan membutuhkan lebih banyak upaya untuk mengelolanya. Topologi jaringan komputer sebuah UKM terdiri atas :

- Kelompok Jaringan DMZ
Terdiri atas *host* yang perlu berhubungan langsung dengan komputer. Dengan

kebutuhan UKM yang tidak banyak, maka *host-host* yang ada dalam kelompok jaringan ini terdiri atas : *proxy server*, *mail server* dan *web/ftp server*. Sebagai perlindungan awal dari serangan, dapat di-implementasikan *router* yang berfungsi sekaligus sebagai *firewall*.

- Kelompok jaringan komputer internal
Pada kelompok jaringan inilah komputer-komputer yang akan digunakan para staf UKM untuk bekerja.
- Backup server
Sebagai persiapan apabila terjadi gangguan yang merusak, maka perlu dilakukan proses *backup* secara rutin. Untuk itu perlu sebuah *host* yang fungsinya khusus menyimpan data-data yang di *backup*, sehingga apabila dibutuhkan dapat langsung digunakan.

Selain itu, dengan terbatasnya sumber daya yang dimiliki, maka faktor sumber daya manusia, baik itu pengelola jaringan komputer maupun pengguna memiliki peranan yang sangat penting. Baik pengelola maupun pengguna harus selalu waspada terhadap berbagai bentuk ancaman yang ada. Para pengguna harus dibiasakan untuk melakukan *update* sistem operasi dan perangkat lunak yang digunakannya (misalnya *update* antivirus). Pengelola harus selalu menjalankan fungsi pendidikan terhadap pengguna sehingga pengguna selalu tahu ancaman apa saja yang dihadapi saat ini. Hal ini menjadi sangat penting, mengingat keterbatasan kemampuan UKM untuk melakukan implementasi teknologi pengamanan jaringan komputer tingkat tinggi, maka kekurangan yang ada, perlu ditutupi dengan kedisiplinan sumber daya manusia yang ada untuk menjaga jaringan komputer organisasi.

VI. Kesimpulan

Keamanan jaringan komputer bagian yang tidak terpisahkan dari keamanan sistem informasi sebuah organisasi secara keseluruhan. Dengan semakin berkembangnya teknologi Internet, maka penggunaan Internet semakin luas dan begitu juga dengan usaha seseorang untuk melakukan gangguan dengan menggunakan teknologi tersebut.

Seperti halnya dengan di bidang lain, usaha untuk mengamankan sebuah jaringan komputer harus dipandang secara keseluruhan, tidak bisa secara *partial*. Setiap lapisan dalam jaringan komputer harus dapat melaksanakan fungsinya secara aman. Pemilihan teknologi yang tepat harus sesuai dengan kebutuhan yang ada. Pemilihan teknologi yang tidak tepat, selain akan mengeluarkan biaya terlalu besar, juga justru dapat mengurangi tingkat keamanan sebuah sistem. Selain itu yang perlu diingat, bahwa semakin banyak peralatan keamanan jaringan komputer yang kita implementasi, maka akan semakin banyak pula pekerjaan pengelola jaringan komputer. Akan semakin banyak *log* yang dihasilkan masing-masing peralatan, mulai dari yang paling penting sampai yang hanya berupa catatan saja. Kegagalan untuk mengelola informasi yang dihasilkan oleh setiap peralatan dapat membuat pengelola jaringan komputer lambat dalam mengantisipasi serangan yang sedang berjalan. Oleh karena itu, selain melakukan implementasi teknologi pengamanan jaringan komputer, perlu juga disediakan *tools* yang dapat digunakan pengelola dalam melakukan pengelolaan.

Daftar Pustaka

- [CURTIN] Matt Curtin, "introduction to network security" - <http://www.interhack.net/pubs/network-security/network-security.html#SECTION00022000000000000000> , 17 Desember 2005
- [SNYDER] Joel Snyder, "what is 802.1x" - <http://www.networkworld.com/research/2002/0506whatisit.html>, 17 Desember 2005
- [GREENE] Tim Greene, "the evolution of application layer firewall" <http://www.networkworld.com/news/2004/0202specialfocus.html>, 17 Desember 2005
- [CHDOI], Chris Chambers, Justin Dolske, Jayaraman Iyer, "TCP/IP Security" http://www.linuxsecurity.com/resource_files/documentation/tcpip-security.html, 17 Desember 2005
- [TYSON] Jeff Tyson, "How Virtual Private Network works" <http://computer.howstuffworks.com/vpn.htm>, 17 Desember 2005
- [WEP] "Wired Equivalent Privacy", http://en.wikipedia.org/wiki/Wired_Equivalent_Privacy, 17 Desember 2005
- [SSL] "Secure Socket Layer", http://en.wikipedia.org/wiki/Secure_Sockets_Layer, 17 Desember 2005
- [IDS] "Intrusion Detection System", http://en.wikipedia.org/wiki/Intrusion-detection_system, 17 Desember 2005
- [IPS] "Intrusion Prevention System", http://en.wikipedia.org/wiki/Intrusion_prevention_system, 17 Desember 2005
- [BRADLEY] Tony Bradley, "Introduction to port scanning", <http://netsecurity.about.com/cs/hackertools/a/aa121303.htm> , 17 Desember 2005
- [HONEY] HoneyNet Project, "Know Your Enemy, Passive Fingerprinting", <http://project.honeynet.org/papers/finger/>, 17 Desember 2005
- [MPRC] Alexander Murphy, Audrey Pender, Louise Reilly, Siobhan Connel, "Denial of Services and Countermeasures", <http://ntrg.cs.tcd.ie/undergrad/4ba2.05/group2/> , 17 Desember 2005
- [SYFL] "SYN Flood", http://www.iss.net/security_center/advice/Exploits/TCP/SYN_flood/default.htm , 17 Desember 2005
- [LOOP] John D Loop, "Three-way-handshake" <http://www.pccitizen.com/threewayhandshake.htm> ,17 Desember 2005
- [TANASE] Mathew Tanase, "IP Spoofing: An Introduction", 17 Desember 2005 <http://www.securityfocus.com/infocus/1674>, 17 Desember 2005
- [SANF] Salvatore Sanfilippo, "DNS Forgery" <http://wiki.hping.org/142>, 17 Desember 2005
- [FOX] Spacefox, "DNS Spoofing Technique", <http://www.securesphere.net/download/papers/dnsspoof.htm>, 17 Desember 2005
- [WUI] Chun Wui, "Mengidentifikasi dan Mitigasi secara Akurat", NOW – Buletin Pelanggan CISCO System Indonesia, kuartal keempat, 2005 vol 2

Daftar Isi

Pendahuluan.....	1
I. Arsitektur Jaringan Komputer.....	1
II. Tipe-tipe proteksi jaringan komputer.....	2
0Layer 2.....	3
0Layer 3.....	5
0Layer 4 /5.....	6
0Layer 7	7
III. Mekanisme pertahanan.....	9
0IDS / IPS.....	9
0Network Topology.....	10
0Port Scanning.....	12
0Packet Fingerprinting.....	12
0Security Information Management.....	13
IV. Jenis-jenis Ancaman	14
0DOS/DDOS.....	14
0Packet Sniffing.....	16
0IP Spoofing.....	16
0DNS Forgery.....	17
V. Studi Kasus.....	19
0Pengumuman Hasil Ujian Sistem Penerimaan Mahasiswa Baru (SPMB)	19
0Usaha Kecil dan Menengah (UKM).....	21
VI. Kesimpulan.....	22
Daftar Pustaka.....	23