

## KAJIAN PENGGUNAAN *PACKET FILTERING FIREWALL* MENGUNAKAN *CISCO IP ACCESS CONTROL LIST*

Imam Sutoyo<sup>1</sup>, Mochamad Wahyudi<sup>2</sup>

<sup>1</sup> Program Studi Teknik Komputer AMIK Bina Sarana Informatika  
Jl. Kramat Raya No. 18 Jakarta Pusat, Indonesia  
<sup>2</sup> Program Pascasarjana Magister Ilmu Komputer STMIK Nusa Mandiri  
Jl. Salemba Raya No. 5 (10250) Jakarta Pusat, Indonesia  
imam@bsi.ac.id  
wahyudi@nusamandiri.ac.id

### **Abstract**

*A computer network has become a necessity for any organization implementing a computer-based information system. Hence, keeping the security aspect is important to maintain the network performance so as to provide optimum service to its users and to be up against any attacks especially when it is connected to the Internet. This paper is intended to give input to computer network administrators who implement IP Access Control List (ACL) network security system as firewall. It discusses the strengths and vulnerabilities of packet filtering firewall using Cisco IP Access Control List (ACL). The findings of this study will give computer network administrators better understanding on implementing Packet filtering firewall with Cisco IP ACL and comprehending the potential security holes due to its vulnerabilities.*

*Keywords : Vulnerabilities, Packet Filtering Firewall, Cisco IP Access Control List (ACL), Cracker*

### **I. PENDAHULUAN**

Untuk mengamankan suatu jaringan komputer, ada berbagai macam cara yang dapat dilakukan. *Firewall* istilah yang sangat dikenal dalam dunia keamanan jaringan, merupakan teknologi yang telah banyak diterapkan untuk melaksanakan fungsi keamanan jaringan.

Ada berbagai macam jenis *firewall*, namun *packet filtering* merupakan metode mendasar yang merupakan pondasi sebuah sistem *firewall*. Cisco System Inc. menyediakan fungsi *packet filtering* pada jajaran produk router mereka dengan nama *IP Access Control List (ACL)*. ACL dapat digunakan sebagai *packet filtering firewall* yang bertugas melaksanakan fungsi penyaringan paket data yang akan masuk ke dalam jaringan internal (*inbound*) maupun yang akan keluar dari jaringan internal (*outbound*).

ACL merupakan suatu fasilitas keamanan yang terdapat pada Cisco *Internetwork*

*Operating System (IOS)* yang terdapat pada suatu perangkat *router* keluaran Cisco. Jadi apabila kita akan menggunakan suatu perangkat *router* keluaran Cisco untuk membangun sebuah *internetwork*, kita tidak memerlukan kembali perangkat keamanan tambahan lain untuk membuat sebuah *firewall* sederhana.

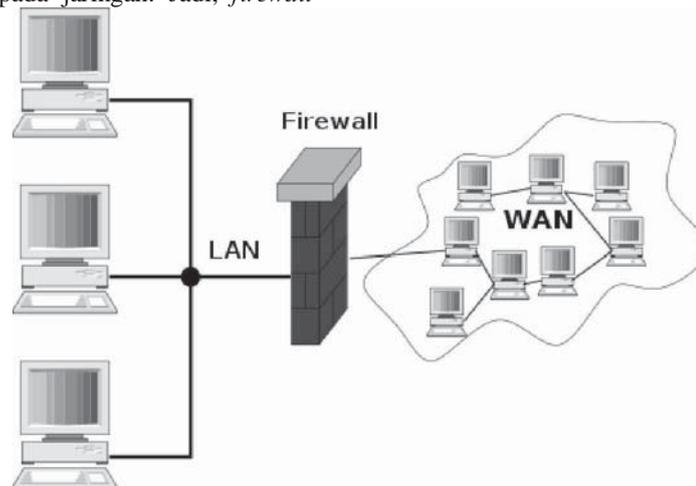
ACL tentu memiliki banyak keterbatasan. Bagi para pengguna ACL, mengetahui keterbatasan atau kekurangan (*vulnerabilities*) ACL sangat penting. Hal ini dimaksudkan agar kita dapat diambil tindakan yang diperlukan untuk menutupi keterbatasan atau kekurangan (*vulnerabilities*) tersebut. Keterbatasan ACL, sebagaimana juga umumnya keterbatasan perangkat keamanan lainnya, tentu menimbulkan konsekuensi berupa *vulnerabilities* yang dapat berpotensi untuk dieksploitasi oleh *cracker*. Salah satu metode untuk menganalisa *vulnerabilitas* tersebut adalah dengan menggunakan metode *vulnerabilities taxonomi*.

## II. PEMBAHASAN

### 2.1. Firewall

*Firewall* merupakan teknologi yang telah sangat dikenal dalam dunia keamanan jaringan. Menurut Brenton (2003) *firewall* adalah sebuah sistem atau kelompok sistem yang menerapkan sebuah kebijakan *access control* terhadap lalu lintas data yang melewati titik-titik akses pada jaringan. Jadi, *firewall*

berfungsi untuk menyaring (*filtering*) lalu lintas data yang melewati titik-titik akses pada jaringan atau pintu-pintu keluar masuk lalu lintas data, baik yang ingin masuk ke dalam jaringan internal (*inbound*) maupun yang ingin keluar ke jaringan eksternal (*outbound*).



Gambar 2.1. Firewall

Ada berbagai macam jenis *firewall*. Menurut Beny (2004), klasifikasi *firewall* menurut fungsi atau cara kerjanya adalah :

#### 1. Packet Filtering Firewall

Sesuai dengan namanya, prinsip kerja *firewall* jenis ini adalah melaksanakan penyaringan terhadap setiap paket data. Dari hasil penyaringan tersebut selanjutnya dapat diputuskan, apakah paket data tersebut dapat diproses lebih lanjut atau ditolak.

*Firewall* jenis ini umumnya diimplementasikan pada perangkat *router*. Router adalah perangkat jaringan yang bekerja pada lapisan jaringan (*network layer*) pada model *Open System Interconnection* (OSI). Dengan demikian, penyaringan paket data oleh *packet filtering firewall* setidaknya didasarkan pada informasi yang diolah pada lapisan tersebut, yaitu Alamat IP.

Kelebihan *firewall* jenis ini, antara lain : sifatnya independen, mudah disesuaikan dengan kebutuhan sistem, memiliki transparansi yang tinggi, dan unjuk kerjanya pun tinggi. Sebaliknya, kelemahan *firewall* jenis ini, antara lain pengamanan yang

dilaksanakan masih sangat rendah dibandingkan dengan banyaknya ancaman yang mengintai jaringan komputer yang dijaga, sangat rentan terhadap *IP Spoofing*, tidak memiliki metode untuk memeriksa aktivitas yang dilakukan oleh koneksi-koneksi yang aktif (*stateless*), tidak memiliki metode otentikasi, dan kemampuannya sangat terbatas karena hanya bekerja pada lapisan jaringan (*network layer*).

#### 2. Application Level Gateway

*Firewall* jenis ini biasa dikenal sebagai *proxy*. Prinsip kerjanya adalah sebagai perantara antara *host* pada jaringan internal dengan sumber daya eksternal yang diakses oleh *host* tersebut. Nama yang umum dikenal untuk *firewall* jenis ini adalah *proxy server*.

Dengan menggunakan *proxy server*, *host* pada jaringan internal tidak pernah berhubungan langsung dengan sumber daya jaringan di luar jaringan lokal tempat ia berada. Setiap ada permintaan koneksi untuk mengakses sumber daya jaringan di luar jaringan lokal harus selalu diarahkan ke *proxy server* terlebih dahulu. *Proxy server* inilah yang nantinya akan memutuskan, apakah

koneksi boleh dilaksanakan atau tidak.

Kelebihan *firewall* jenis ini, antara lain pengamanan yang dilaksanakan lebih bagus dibandingkan *packet filter*, memiliki metode otentikasi, memiliki metode kendali akses (*access control*), memiliki fasilitas *logging*, dan memiliki fasilitas *caching* untuk membantu menghemat *bandwidth*. Sebaliknya, kelemahan *firewall* jenis ini, antara lain : kurangnya transparansi terhadap pengguna dimana aplikasi pengguna harus dikonfigurasi untuk mendukung fungsi *proxy*, aplikasi yang digunakan harus mendukung fasilitas *proxy*, dan unjuk kerja yang lebih rendah dibandingkan *packet filter*.

### 3. Circuit Level Gateway

*Firewall* jenis ini merupakan pengembangan dari *Application Level Gateway*. Prinsip kerja *Circuit Level Gateway* serupa dengan *Application Level Gateway*, yakni sebagai perantara antara *host* pada jaringan internal dengan sumber daya eksternal yang diakses oleh *host* tersebut. Perbedaannya adalah pada tingkatan atau lokasi pelaksanaan fungsi perantara (*proxy*) tersebut dilaksanakan.

Kelebihan *firewall* jenis ini kurang lebih sama dengan *Application Level Gateway*, namun ia menutupi kelemahan kurangnya transparansi dari *Application Level Gateway*, dimana setiap aplikasi tidak perlu dikonfigurasi untuk mendukung fungsi *proxy*, bahkan aplikasi yang tidak memiliki dukungan terhadap fungsi *proxy* pun masih dapat berjalan. Sebaliknya, kelemahan *Firewall* jenis ini pun kurang lebih sama, bedanya adalah aplikasi harus kompatibel dengan *platform* yang digunakan, misalnya harus kompatibel dengan *Application Programming Interface* (API).

### 4. Statefull Inspection

*Firewall* jenis ini adalah *firewall* yang paling canggih dibandingkan dengan tiga jenis *firewall* sebelumnya. Prinsip kerja dari *Statefull Inspection* adalah selalu aktif mengawasi setiap koneksi yang terjadi, sehingga selalu dapat diketahui status dari koneksi-koneksi tersebut (*statefull*) dan dapat dilaksanakan tindakan yang semestinya jika ditemukan adanya penyimpangan-penyimpangan dari koneksi yang ada.

Kelebihan *firewall* jenis ini, antara lain tingkat pengamanannya paling tinggi, pengamanannya paling lengkap karena mendukung dan dapat melaksanakan pengawasan pada seluruh lapisan OSI, memiliki unjuk kerja yang tinggi, memiliki *skalabilitas* yang bagus, dan memiliki transparansi yang tinggi. Sebaliknya, kelemahan *firewall* jenis ini adalah diperlukannya sumber daya yang sangat besar untuk menjalankannya, apalagi saat jumlah koneksi makin bertambah banyak.

## 2.2. Cisco IP Access Control List (ACL)

Pengendalian akses (*access control*) merupakan mekanisme pengamanan yang umum diimplementasikan dalam skenario pengamanan sebuah sistem informasi. *Access Control* dapat diterapkan melalui ketiga komponen sistem informasi, yaitu *hardware*, *software*, dan *brainware*.

Salah satu contoh penerapan *Access Control* melalui *hardware* secara *embedded*, lebih spesifik lagi pada sebuah perangkat *router* adalah seperti yang diterapkan oleh Cisco Systems Inc. Pada jajaran produk *router* mereka, yaitu *Cisco IP Access Control List* (ACL). ACL merupakan sebuah fasilitas keamanan yang dimiliki oleh perangkat *router* Cisco untuk menyaring paket data yang masuk maupun yang keluar dari *router*.

Menurut Cisco (2008), "*IP Access Control List* adalah sebuah daftar berurutan yang paling sedikit terdiri dari satu pernyataan *permit* dan mungkin satu atau lebih pernyataan *deny*". Mekanisme penyaringan paket data pada ACL didasarkan pada dua pernyataan tersebut, yakni sebuah paket data akan diteruskan jika memenuhi kriteria pada pernyataan *permit* dan tidak memenuhi kriteria pada pernyataan *deny* dan sebaliknya paket data tidak akan diteruskan jika tidak memenuhi kriteria pada pernyataan *permit* atau memenuhi kriteria pernyataan *deny*. Terurut maknanya adalah, daftar pernyataan pada ACL diproses secara terurut atau *sekuensial* dari baris pertama hingga baris terakhir.

### 2.2.1. Klasifikasi ACL

Ada dua jenis ACL, yaitu *Standard ACL* dan *Extended ACL*. Sesuai dengan nama yang diberikan oleh pembuatnya, *Extended ACL* memiliki fasilitas *filtering* yang lebih lengkap dibandingkan *Standard ACL*, namun dalam

penggunaannya keduanya saling melengkapi.

Cisco memberikan arahan berkaitan dengan penempatan ACL, bahwa *Standard ACL* efektif jika diletakkan dilokasi tujuan, sedangkan *Extended ACL* seharusnya diletakkan pada lokasi asal dari paket data.

1. *Standard ACL*

*Standard ACL* hanya menyaring paket data berdasarkan alamat IP pengirim paket. Saat sebuah paket data ingin melewati sebuah *interface* dari *router*, alamat pengirim akan dibandingkan dengan alamat IP yang didefinisikan pada baris-baris pernyataan yang terdapat pada ACL yang dipasang pada *interface* tersebut. Paket data akan dilewatkan atau tidak berdasarkan hasil perbandingan.

*Standard ACL* menggunakan penomoran dari 1 sampai 99. Berikut ini sintaks dari *Standard ACL*.

```
access-list access-list-number {deny | permit}
source-ip-address [wildcard-mask]
```

Dimana :

- a. *Access-list*  
Keyword yang digunakan untuk membuat ACL.
- b. *access-list-number*  
Nomor yang menjadi identitas dari ACL, jangkauannya adalah 1-99.
- c. *{permit | deny}*  
Tindakan terhadap paket data. Paket data dapat dilewatkan atau ditolak.
- d. *source-ip-address*  
Alamat IP pengirim.
- e. *[wildcard-mask]*  
Wildcard yang digunakan.

Contoh :

```
access-list 1 permit host 202.101.51.3 0.0.0.0
```

*Standard ACL* tersebut hanya akan meneruskan paket dari alamat IP 202.101.51.3.

2. *Extended ACL*

*Extended ACL* memiliki fasilitas *filtering* yang lebih lengkap dibandingkan *Standard ACL*. Selain penyaringan paket data berdasarkan alamat IP pengirim, *Extended ACL* dapat digunakan untuk menyaring paket data berdasarkan alamat IP tujuan, nomor *port*, dan jenis protokol yang digunakan.

*Extended ACL* menggunakan penomoran dari 100 sampai 199. Berikut ini sintaks dari *Extended ACL*.

```
access-list access-list-number {deny | permit}
protocol source-ip-address [wildcard-mask]
destination-ip-address [wildcard-mask]
operator
```

Dimana :

- a. *access-list*  
Keyword yang digunakan untuk membuat ACL.
- b. *access-list-number*  
Nomor yang menjadi identitas dari ACL, jangkauannya adalah 100-199.
- c. *{permit | deny}*  
Tindakan terhadap paket data. Paket data dapat dilewatkan atau ditolak.
- d. *protocol*  
Nama atau nomor protokol.
- e. *source-ip-address*  
Alamat IP pengirim.
- f. *destination-ip-address*  
Alamat IP tujuan.
- g. *[wildcard-mask]*  
wildcard yang digunakan.
- h. *operator*  
Operator yang digunakan.

Tabel 2.1 Operator untuk *Extended ACL*

Operator	Penjelasan
eq ( <i>equal</i> )	Menentukan satu nomor <i>port</i>
neq ( <i>not equal</i> )	Bentuk negasi atau kebalikan dari operator eq
gt ( <i>greater than</i> )	Digunakan untuk menentukan jangkauan nomor <i>port</i> yang lebih besar dari nomor <i>port</i> yang diberikan
lt ( <i>less than</i> )	Digunakan untuk menentukan jangkauan nomor <i>port</i> yang lebih kecil dari nomor <i>port</i> yang diberikan

Contoh :

```
access-list 100 permit tcp 202.101.51.3 0.0.0.0  
host 172.16.1.1 0.0.0.0 eq 80
```

*Extended ACL* tersebut hanya akan meneruskan paket dari alamat IP 202.101.51.3 ke alamat IP 172.16.1.1 untuk protokol TCP dengan nomor port 80.

### 2.2.2. Memasang ACL pada Interface

Membuat aturan penyangkangan paket data melalui baris-baris aturan pada ACL merupakan langkah awal dalam menggunakan ACL. Langkah selanjutnya, aturan penyangkangan yang telah kita buat tersebut harus dipasang ke sebuah *interface* pada *router*.

Penyangkangan terhadap paket data pada sebuah *interface* dapat dilaksanakan dalam dua arah, yaitu diterapkan untuk paket data yang masuk maupun yang keluar dari *router* melalui *interface* tersebut. ACL yang dipasang untuk menyaring paket data yang masuk melalui sebuah *interface* disebut *ACL inbound*. Sedangkan, ACL yang dipasang untuk menyaring paket data yang akan keluar melalui sebuah *interface* disebut *ACL outbound*.

*ACL inbound* akan menyaring paket data yang masuk melalui *interface router*. Saat sebuah paket data masuk, paket data tersebut akan diperiksa, yaitu dicocokkan dengan aturan-aturan yang ada pada ACL yang dipasang pada *interface* tempat paket data tersebut ingin masuk.

Jika paket data tersebut cocok dengan satu baris pernyataan *permit*, maka akan langsung diproses lebih lanjut, yaitu di-*routing* ke jaringan tujuan dari paket data. Jadi, paket data tersebut tidak akan dicocokkan lagi dengan baris-baris berikutnya dari ACL. Sebaliknya, jika paket data tersebut cocok dengan satu baris pernyataan *deny*, akan langsung ditolak, dan *router* akan mengirimkan paket ICMP *destination unreachable* kepada pengirim paket data. Jadi, paket data tersebut sudah tidak akan dicocokkan lagi dengan baris-baris berikutnya.

*ACL outbound* akan menyaring paket data yang akan keluar melalui *interface Router*. Saat sebuah paket data akan keluar, yakni minta di-*routing* keluar dari jaringan internal, paket data tersebut akan dicocokkan dengan aturan-aturan yang ada pada ACL yang dipasang pada *interface* tempat paket data tersebut ingin keluar.

Jika paket data tersebut cocok dengan satu baris pernyataan *permit*, ia akan langsung diproses lebih lanjut, yakni di-*routing* ke jaringan tujuan

dari paket. Jadi, paket data tersebut tidak akan dicocokkan lagi dengan baris-baris berikutnya dari ACL. Sebaliknya, jika paket data tersebut cocok dengan satu baris pernyataan *deny*, maka akan langsung ditolak, dan *router* akan mengirimkan paket ICMP *destination unreachable* kepada pengirim paket. Jadi, paket data tersebut sudah tidak akan dicocokkan lagi dengan baris-baris berikutnya.

Pembuatan ACL dan pemasangannya pada *interface* memiliki beberapa aturan penting yang harus diperhatikan, antara lain:

1. Hanya satu ACL untuk satu protokol untuk satu arah penyangkangan.
2. *Standard ACL* seharusnya dipasang sedekat mungkin dengan lokasi tujuan paket data.
3. *Extended ACL* seharusnya dipasang sedekat mungkin dengan lokasi asal paket data.
4. Sudut pandang arah penyangkangan dilihat dari dalam Router, jadi *interface* Router dipandang sebagai sebuah pintu masuk.
5. Setiap pernyataan diproses secara berurutan, mulai dari pernyataan pada baris pertama sampai baris terakhir.
6. Pada akhir baris dari setiap ACL ada baris *implicit deny*, berupa pernyataan *deny all* yang berfungsi untuk menolak setiap paket yang tidak memenuhi satupun kriteria pernyataan *permit* pada baris-baris di atasnya.
7. Isi ACL seharusnya melakukan penyangkangan paket data dari khusus ke umum, misalnya sebuah *host* harus ditentukan dahulu aturannya baru aturan untuk sekelompok *host*.
8. Proses pemeriksaan atau pencocokan sesuai dengan kriteria yang ditentukan dilaksanakan terlebih dahulu sebelum keputusan *permit* atau *deny* diterapkan.
9. Jangan memanipulasi sebuah ACL yang sedang aktif pada sebuah *interface*.
10. Menghapus ACL dari sebuah *interface* yang aktif harus dilakukan dengan hati-hati, sebaiknya *interface* tersebut dinonaktifkan terlebih dahulu.
11. Jika paket data yang datang ke sebuah *interface* ditolak, ACL akan mengirimkan pesan ICMP *Destination Unreachable* kepada pengirim paket tersebut.

### 2.3. Metode Vulnerabilities Taxonomi

Menurut Krsul (1998), "*taxonomi* merupakan kajian teoritis mengenai klasifikasi, termasuk landasan dasar, prinsip, prosedur, dan aturan-aturan yang berkaitan dengan klasifikasi

tersebut”. Metode klasifikasi itu sendiri dapat digunakan untuk mempermudah analisa terhadap objek atau permasalahan yang akan diteliti.

Umumnya, orang mengenal istilah *taxonomi* dalam disiplin Ilmu Biologi, namun dalam disiplin Ilmu Komputer, khususnya bidang kajian Keamanan Sistem Informasi, wacana yang berkaitan dengan metode *taxonomi* telah dikenal sejak awal perkembangan ilmu ini. Menurut Wright (2007), “kebutuhan akan dibuatnya sebuah *taxonomi* yang terstruktur (sistem penamaan) untuk istilah-istilah atau layanan-layanan dalam dunia keamanan sistem informasi bukanlah hal baru. Semua layanan tersebut telah tersedia sejak dunia bisnis dan pemerintahan mulai menggunakan komputer, yakni sekitar tahun 70-an”.

Salah satu contoh metode *taxonomi* dalam bidang kajian Keamanan Sistem Informasi adalah metode *taxonomi vulnerabilities*, yaitu metode sistematis yang digunakan untuk menjelaskan kelemahan-kelemahan sebuah sistem yang dapat dieksploitasi oleh orang-orang yang tidak berwenang terhadap sistem tersebut, serta metode penanggulangannya.

Menurut Krsul (1998), “Fungsi dari *taxonomi* adalah agar pemisahan atau pengurutan spesies dapat dilaksanakan sehingga dapat dibuat sebuah generalisasi yang telah mencakup seluruh spesies tersebut. Jadi, dapat kita katakan bahwa *taxonomi* memiliki nilai penjelasan. Taksonomi dapat juga digunakan untuk membuat prediksi adanya spesies-spesies lain yang belum dikenal dengan mempelajari pola dari spesies-spesies yang telah dikenal”. Jadi, dapat kita katakan bahwa *taxonomi* memiliki nilai prediksi”.

Dengan demikian, dapat disimpulkan bahwa metode *taxonomi vulnerabilities* sangat bermanfaat untuk menganalisa dan mengklasifikasikan kelemahan-kelemahan sebuah sistem, atau sebuah perangkat keamanan sistem, misalnya *firewall*. Dengan menggunakan metode *taxonomi vulnerabilities*, kita tidak hanya dapat mendefinisikan penyebab-penyebab kelemahan, akibat buruk dari kelemahan tersebut, dan teknik penanggulangannya, tetapi kita juga dapat mengelompokkan beragam kelemahan tersebut sehingga memudahkan analisa yang tidak hanya dapat digunakan untuk menyelesaikan permasalahan yang ada saat ini, namun juga menjadi landasan dalam memecahkan permasalahan yang datang dikemudian hari.

Kamara, dkk (2003) mengklasifikasikan penyebab *vulnerabilities* yang berkaitan dengan

*firewall* menjadi tujuh, yaitu:

1. Kesalahan Validasi (*Validation Error*). *Validation Error* terjadi saat program, perangkat, atau sistem berinteraksi dengan lingkungannya, yakni dalam rangka mengolah data-data yang datang dari lingkungannya, tanpa memeriksa terlebih dahulu keabsahan dari data-data tersebut.  
Ada tiga jenis data yang memerlukan validasi, yaitu *input*, *origin*, dan *target*. Validasi *Input*, artinya memeriksa bahwa data masukan tidak menyimpang, yakni benar-benar sesuai dengan yang diharapkan atau yang seharusnya, baik dalam hal nomor urutnya, jenis datanya, maupun formatnya. Validasi *origin*, artinya memeriksa bahwa data yang diolah benar-benar asli sesuai dengan apa yang dinyatakan oleh data tersebut. Validasi *target*, artinya memeriksa bahwa data hasil pengolahan diberikan kepada penerima yang berhak atas data tersebut. Tidak hanya itu, validasi *target* juga harus dapat meyakinkan bahwa data tersebut tidak diberikan kepada pihak yang tidak berhak.
2. Kesalahan Autorisasi (*Authorization Error*). *Authorization Error* disebut juga kesalahan otentikasi. Kesalahan ini terjadi saat pihak yang tidak berhak atau tidak berwenang diijinkan untuk melaksanakan operasi terhadap program, perangkat, atau pun sistem.
3. Kesalahan Serialisasi/Aliasing (*Serialization/Aliasing Error*). *Serialization Error* terjadi saat muncul eksploitasi terhadap sistem akibat perilaku *asinkron* dari dua sistem yang berbeda diijinkan untuk dioperasikan dalam waktu yang bersamaan. Sedangkan, *Aliasing Error* terjadi saat ada dua nama untuk sebuah objek yang sama dapat mengakibatkan perubahan pada isi objek tersebut secara tidak terduga, sehingga konsekuensinya adalah dapat mengakibatkan perubahan validasi yang sebelumnya telah diaplikasikan kepada objek tersebut.
4. Kesalahan Pengecekan Batasan Sistem (*Boundary Checking Error*). *Boundary Checking Error* muncul akibat kegagalan dalam memeriksa batasan-batasan yang diperbolehkan, yakni pelanggaran terhadap batasan-batasan yang telah ditetapkan. Akibatnya, terjadi *Buffer Overflow*.
5. Kesalahan Domain (*Domain Error*). *Domain*

*Error* terjadi saat muncul celah keamanan pada sebuah *Domain*, yakni batasan *Domain* tersebut dilanggar, sehingga mengakibatkan adanya informasi yang seharusnya hanya boleh diakses oleh pengguna pada *Domain* tersebut bocor ke pihak luar yang tidak berhak.

6. Rancangan yang lemah atau kesalahan Rancangan (*Weak/Design Error*). *Weak/Design Error* terjadi saat tahapan proses perancangan sistem. Contoh dari kesalahan rancangan misalnya lemahnya algoritma enkripsi dimana hasil enkripsinya, yakni *cipher text*-nya mudah dipecahkan atau dilaksanakan *kriptanalisis* terhadapnya.
7. Kesalahan-kesalahan lainnya. Kesalahan-kesalahan lain yang tidak masuk ke dalam enam kategori kesalahan sebelumnya masuk ke dalam kategori ini.

#### 2.4. Keterbatasan ACL

Cisco System Inc. membuat ACL sebagai perangkat pengamanan dasar pada jajaran produk Router mereka. ACL dibuat untuk memberikan fungsi penyaringan paket data yang merupakan fungsi mendasar dari sebuah *firewall*. ACL dirancang untuk memberikan fungsi pengamanan yang optimal, namun tetap sederhana dalam pembuatan dan pengimplementasiannya.

Sebuah sistem *firewall* yang kokoh karena dibangun dari beragam fungsi pengamanan sekalipun, tetap bukan merupakan solusi tunggal dalam mengamankan sebuah sistem. Apalagi ACL yang hanya melaksanakan penyaringan paket data saja dalam melaksanakan fungsi pengamanan. Sebelum menggunakan ACL, penting untuk diketahui keterbatasan-keterbatasan dari ACL, sehingga dapat dilaksanakan tindakan yang diperlukan agar keamanan sistem tetap terjaga secara optimal. Berikut ini keterbatasan-keterbatasan dari ACL dan analisa berikut solusinya sekaligus klasifikasinya berdasarkan metode *taxonomi vulnerabilitas*.

1. Pembuatan ACL harus dilaksanakan secara berurutan atau *sekuensial*. Saat pernyataan-pernyataan pada ACL jumlahnya makin banyak, hal ini tidak hanya akan sangat merepotkan dalam pembuatannya, namun juga sangat sulit untuk melaksanakan audit dan merawatnya.

Kesalahan yang mungkin muncul pada ACL yang memiliki banyak baris aturan adalah permasalahan logik, yakni sulit untuk tetap menjaga konsistensi logik dari seluruh baris ACL tersebut. Ada beberapa kesalahan logik yang dapat muncul, misalnya baris aturan yang terduplikasi (*redundant*), baris aturan yang saling bersilangan (*intersection*), dan baris aturan yang tidak konsisten.

Baris aturan yang terduplikasi (*redundant*), artinya ada dua atau lebih baris aturan yang memiliki makna atau aturan yang sama, yakni sebuah baris aturan ternyata merupakan bagian atau *subset* dari baris aturan lain yang lebih lengkap.

Baris aturan yang saling bersilangan (*intersection*), artinya ada dua atau lebih baris aturan yang memiliki makna atau aturan yang saling bersilangan, yakni baris aturan yang satu memiliki aturan yang telah diatur oleh baris aturan lain atau aturan penyaringan mereka saling beririsan.

Baris aturan yang tidak konsisten, artinya ada dua atau lebih baris aturan yang memiliki makna atau aturan yang saling berlawanan atau bertolak belakang, yakni adanya sebuah baris aturan yang menyatakan *permit* untuk sebuah paket data sementara ada baris aturan lain yang menyatakan *deny* untuk paket data yang sama atau sebaliknya.

Kesalahan-kesalahan logik tersebut muncul umumnya karena makin berkembangnya ACL, yakni makin bertambahnya baris aturan pada ACL tersebut yang disebabkan oleh makin berkembangnya jaringan dan adanya perubahan atau penyesuaian aturan kebijakan keamanan mengakibatkan makin sulitnya menyesuaikan baris-baris aturan yang telah ada dengan aturan-aturan baru yang harus ditambahkan. Diperlukan pemahaman yang baik mengenai aturan pembuatan ACL dan ketelitian yang tinggi serta ketekunan untuk membuat ACL yang akurat dan efisien.

Sebenarnya, Cisco telah menyediakan alat bantu yang dapat digunakan untuk manajemen ACL, yaitu *Cisco Works*. *Cisco Works* merupakan aplikasi berbasis *Graphical User Interface* (GUI) yang dapat digunakan sebagai alat bantu untuk melakukan pemantauan dan mengatur perangkat-perangkat jaringan milik Cisco.

Untuk manajemen ACL, Cisco *Works* menyediakan fasilitas *ACL Manager* yang dapat digunakan untuk membuat (*create*), mengubah (*edit*), dan mengatur urutan baris-baris aturan pada ACL serta fungsi-fungsi yang berkaitan dengan manajemen ACL lainnya.

Berdasarkan *taxonomi vulnerabilitas*, keterbatasan ACL ini masuk ke dalam kesalahan atau kelemahan desain dimana akibat dari keterbatasan ini adalah menyulitkan para penggunanya. Meskipun demikian, secara objektif, dimana Cisco merancang ACL agar sederhana untuk menjamin kinerja yang tinggi maka administrator jaringan yang diharapkan dapat memahami dengan baik ACL agar efek dari keterbatasan ACL ini dapat diminimalkan dan keuntungan dari kesederhanaan dan unjuk kerja yang tinggi dari ACL didapatkan.

2. ACL merupakan *packet filtering firewall* yang bersifat *stateless*, artinya ACL tidak dapat memeriksa hakikat dari setiap koneksi yang sedang terjalin. Selama paket-paket data yang meminta lewat memenuhi persyaratan yang tercantum pada ACL, paket data tersebut akan dilewatkan. Jadi, bisa saja sebuah paket data yang memiliki alamat IP yang sah yang diperbolehkan lewat oleh ACL ternyata merupakan paket data dari penyerang yang sedang berusaha untuk mengeksploitasi sistem, yakni si penyerang tadi telah melakukan *IP Spoofing*.

Berdasarkan *taxonomi vulnerabilitas*, keterbatasan ACL ini masuk ke dalam *validation error* yang berkaitan dengan validasi *origin* dimana ACL tidak mampu atau salah dalam memeriksa kebenaran alamat IP dari paket data yang lewat akibat *IP Spoofing* yang dilakukan oleh penyerang, sehingga penyerang dapat masuk untuk mengeksploitasi jaringan dengan menyamarkan alamat IP yang dipergunakannya.

3. ACL hanya akan melaksanakan penyaringan terhadap paket-paket data yang ingin melewati sebuah *interface* dari *router*, yaitu paket data yang ingin masuk dari jaringan eksternal ke jaringan internal maupun sebaliknya. ACL tidak dapat menyaring paket data yang berasal dari *router* itu sendiri. Misalnya, paket-paket data *routing*

*protocol*, seperti *OSPF hello*, paket *routing update* dan sejenisnya yang digunakan oleh *router* untuk saling berbagi informasi *routing* akan dilewatkan begitu saja oleh ACL tanpa perlu diperiksa.

Penyerang dapat membuat paket *route advertisement* yang berisi rute-rute yang dapat melewati paket-paket data ke jaringan miliknya. Jika *router* yang kita gunakan menggunakan protokol *routing* yang dinamis, dan umumnya memang protokol *routing* dinamis yang digunakan karena kemudahan administrasinya dibandingkan harus mengisi tabel *routing* secara manual (*static routing*), maka ACL tidak mampu menyaring mana paket *route advertisement* yang sah dengan yang tidak.

*Static routing* memang lebih aman dibandingkan *Dynamic Routing*. Menurut Brenton (2003), "Meskipun *routing statis* memerlukan *maintenance* atau pemeliharaan yang cukup banyak, namun *routing statis* adalah cara yang paling aman untuk membangun *routing table* Anda. *Routing* dinamis memungkinkan *routing table* untuk diperbaharui (*update*) secara dinamis oleh alat-alat di *network*. Seorang penyerang bisa mengeksploitasi fasilitas ini untuk memberikan informasi *routing* yang tidak benar kepada *router-router* kita yang bisa menghalangi *network* kita untuk bekerja dengan baik".

Solusi dari masalah ini adalah penggunaan protokol *routing* yang memiliki fitur otentikasi dan enkripsi, seperti *Open Shortest Path First* (OSPF). Protokol OSPF mensyaratkan *router-router* yang berpartisipasi dalam pertukaran *routing table* untuk memberikan kata sandi agar informasi rute mereka dapat diterima. Informasi kata sandi tersebut berikut informasi *routing table* yang ingin diberikan dan kunci kriptografi yang digunakan dienkripsi dan disertakan dalam paket *update routing table*.

Jadi, pengamanan pertukaran informasi rute diserahkan pada protokol *routing*. Tentu saja, protokol *routing* yang digunakan harus telah memiliki fitur keamanan seperti OSPF. Tanpa adanya fasilitas keamanan, *router* yang kita miliki rentan terhadap masuknya informasi-informasi rute yang ilegal. Namun, seorang pengelola jaringan komputer tentu saja selalu dapat dengan

mudah memeriksa secara manual *routing table* dari *router-router* yang berada dalam pengelolaannya menggunakan perintah-perintah yang disediakan IOS melalui *console* menggunakan sebuah *terminal*.

Berdasarkan *taxonomi vulnerabilitas*, keterbatasan ACL ini masuk ke dalam kesalahan atau kelemahan desain dimana akibat dari keterbatasan ini menyebabkan rentannya Router terhadap masuknya informasi-informasi rute yang tidak benar hasil dari paket *routing update* dari pihak-pihak yang tidak terpercaya.

4. Saat sebuah paket telah memenuhi persyaratan dari sebuah baris pernyataan pada ACL, baik pernyataan *permit* maupun *deny*, tindakan yang sesuai akan segera dilaksanakan. Jadi, paket tersebut tidak akan dicocokkan lagi dengan baris-baris berikutnya. Artinya, ACL harus dibuat dengan sangat teliti agar jangan sampai salah dalam mengambil tindakan terhadap sebuah paket data yang dapat mengakibatkan paket data yang seharusnya ditolak menjadi diteruskan dan sebaliknya.

Meskipun tidak mutlak sebagai *design error* berdasarkan *taxonomi vulnerabilitas*, hal ini merupakan keterbatasan rancangan dari ACL, dimana ACL dirancang agar sederhana dan cepat sehingga operasinya tidak terlalu mengganggu kinerja dari *router*. Jadi, begitu sebuah paket telah cocok dengan satu baris aturan dari ACL, baris-baris berikutnya tidak perlu diperiksa lagi agar tidak menghabiskan sumber daya pemrosesan dari *router*.

5. ACL tidak dirancang untuk mendeteksi penyerang dari dalam, yaitu pengguna yang sah dari jaringan internal yang memanfaatkan sumber daya jaringan untuk melaksanakan tindakan jahatnya. Jadi, seorang pengguna jaringan yang sah dapat melakukan kegiatan apapun yang diinginkannya dalam jaringan tanpa terdeteksi oleh ACL.

Untuk dapat menganalisa paket-paket data yang hilir mudik dalam jaringan internal diperlukan sebuah program *network analyzer* atau *traffic analyzer*. Program *analyzer* tersebut dapat menangkap setiap paket data yang hilir mudik dalam jaringan dan menganalisanya, yaitu dengan membandingkannya dengan *database* modus operandi serangan yang dimilikinya

sehingga dapat ditentukan aktivitas apa yang sedang dilaksanakan oleh pemilik paket data tersebut.

Jika ternyata terbukti bahwa paket data tersebut mencurigakan, yaitu cocok dengan modus operandi yang terdaftar dalam *database*, maka dapat segera diambil tindakan yang semestinya, misalnya pemutusan koneksi secara langsung atau dikirimkannya laporan kepada pengelola jaringan agar pengelola jaringan tersebut dapat segera melaksanakan tindakan yang diperlukan. Inilah yang dilaksanakan oleh perangkat keamanan yang dikenal dengan nama *Intrusion Detection System (IDS)*.

IDS tidak lagi memperdulikan apakah paket data tersebut menggunakan alamat IP yang sah atau tidak. Selama aktivitas koneksi yang sedang dibangun oleh paket data tersebut cocok dengan modus operandi atau pola serangan yang terdaftar dalam *database* IDS, koneksi yang dibangun oleh paket data tersebut adalah koneksi yang ilegal dan setiap paket data yang berkaitan dengannya harus ditahan.

Berdasarkan *taxonomi vulnerabilitas*, keterbatasan ACL ini masuk ke dalam kesalahan atau kelemahan desain, yaitu ACL adalah *packet filtering* yang bersifat *stateless*.

6. ACL tidak dapat mengenali *malware (malicious ware)*, seperti : *virus, worm, trojan horse*, dan sejenisnya. *Malware* merupakan alat yang sangat ampuh yang sering menjadi senjata pamungkas oleh penyerang untuk mengeksploitasi bahkan melumpuhkan sebuah sistem. ACL dapat saja melewatkan sebuah paket yang merupakan sebuah *malware*, selama paket tersebut memenuhi persyaratan *permit* pada ACL.

ACL hanya mampu menerapkan *access control* melalui mekanisme *filtering*. ACL tidak mampu mengenali apalagi menangani *malware*. Menurut Brenton (2003), "*Access Control* tidak akan menghilangkan atau mendeteksi keberadaan sebuah program kosmetik. *Access control* hanya sebuah metode untuk membantu sistem kita menahan infeksi *virus*". Maksudnya, metode *access control* yang kita terapkan, baik melalui mekanisme *filtering*, otentikasi, dan sejenisnya hanya akan menghalangi

*malware* untuk masuk. Ia tidak akan mampu mendeteksi *malware* apalagi memperbaiki sumber daya yang telah terinfeksi oleh *malware*. Kita membutuhkan aplikasi lain untuk melakukannya, yaitu aplikasi *antimalware*.

ACL hanya dapat menghalangi *malware* untuk masuk dengan berasumsi bahwa *malware* hanya dibawa oleh paket data yang tidak diijinkan untuk lewat, yaitu : asumsi bahwa aturan pada ACL memang telah dirancang untuk hanya melewatkan paket data yang bersih dan terpercaya. Jadi, jika seorang penyerang ingin menyebarkan *malware* maka ia dapat melakukannya dengan menyamarkan paket data yang berisi *malware* sehingga seolah-olah merupakan paket data yang bersih dan terpercaya, misalnya paket data tersebut memiliki alamat IP yang terpercaya, yaitu penyerang melakukan *IP Spoofing*.

Berdasarkan *taxonomi vulnerabilitas*, secara prinsip keterbatasan ACL ini masuk ke dalam kesalahan lain-lain. Namun, berkaitan dengan *IP Spoofing* sebagai modus operasinya, keterbatasan ini dapat masuk ke dalam kesalahan validasi.

7. *Standard* dan *Extended* ACL tidak memiliki metode otentikasi. Metode otentikasi yang diterapkan bagi pengguna digunakan untuk menguji keabsahan seorang pengguna yang ingin mengakses sumber daya jaringan.

Umumnya, metode otentikasi diterapkan menggunakan pasangan *user name* dan *password*. Jadi, pengguna yang ingin mengakses sumber daya sistem terlebih dahulu harus memberikan *user name* dan *password* yang dimilikinya. Mekanisme otentikasi yang diterapkan pada sistem tersebut kemudian melaksanakan *query* ke database yang sesuai untuk mencari *account* yang cocok. Jika ditemukan, pengguna tersebut baru boleh mengakses sumber daya sistem, sebaliknya jika tidak ada yang cocok, maka pengguna tersebut tidak diijinkan untuk mengaksesnya.

Karena *Standard* dan *Extended* ACL tidak memiliki metode otentikasi, maka tidak ada mekanisme pengujian terhadap pengguna yang ingin mengakses sumber daya jaringan. Siapapun boleh mengakses sumber daya jaringan selama ybs memenuhi persyaratan pada aturan yang terdapat pada ACL,

misalnya alamat IP yang sah untuk digunakan pada jaringan tersebut. Dengan demikian, seorang penyerang yang telah berhasil mendapatkan akses ke jaringan sudah tidak ada bedanya lagi dengan pengguna jaringan yang sah, karena pada dasarnya pengguna jaringan yang sah juga tidak memiliki *user account*, yakni pada jaringan tersebut memang tidak dikenal adanya *user account* untuk membedakan antara pengguna jaringan yang sah dengan yang tidak. Kesimpulannya, tanpa adanya metode otentikasi, maka tidak ada mekanisme untuk membedakan antara pengguna jaringan yang sah dengan pengguna ilegal, konsekuensinya adalah tidak dapat diterapkannya pembatasan hak akses atau *access control* berdasarkan *user account*.

Berdasarkan *taxonomi vulnerabilitas*, keterbatasan ACL ini masuk ke dalam *authorization error* dimana ACL tidak mampu untuk melaksanakan otentikasi terhadap permintaan koneksi sehingga tidak dapat dibedakan antara pengguna yang sah dan pengguna ilegal.

### III. PENUTUP

#### 3.1. Kesimpulan

Berdasarkan analisa yang telah dijabarkan pada bagian sebelumnya, dapat diambil beberapa kesimpulan, antara lain:

1. Salah satu teknik untuk mengamankan sistem adalah dengan konsep *Access Control* atau pengendalian akses.
2. Salah satu contoh penerapan *Access Control* pada Router adalah Cisco *IP Access Control List (ACL)* yang diterapkan oleh Cisco Systems Inc. pada jajaran produk Router mereka.
3. *ACL* adalah sebuah daftar berurut yang paling sedikit terdiri dari satu pernyataan *permit* dan mungkin satu atau lebih pernyataan *deny*.
4. Sebuah paket data akan diteruskan jika memenuhi kriteria pada pernyataan *permit* dan tidak memenuhi kriteria pada pernyataan *deny* dan sebaliknya paket data tidak akan diteruskan jika tidak memenuhi kriteria pada pernyataan *permit* atau memenuhi kriteria pernyataan *deny*.
5. Daftar pernyataan pada *ACL* diproses secara

- terurut atau *sekuensial* dari baris pertama hingga baris terakhir, dimana jika sebuah paket data telah cocok dengan sebuah pernyataan, baik *permit* ataupun *deny*, tindakan yang sesuai akan segera dilaksanakan tanpa perlu membandingkan paket data tersebut dengan baris-baris berikutnya.
6. Ada dua jenis *IP Access Control List*, yaitu *Standard ACL* dan *Extended ACL*.
  7. *Standard ACL* hanya menyaring paket data berdasarkan alamat IP pengirim paket.
  8. *Extended ACL* dapat menyaring paket data berdasarkan alamat IP pengirim, alamat IP tujuan, nomor *port*, dan jenis protokol yang digunakan.
  9. ACL yang telah dibuat harus dipasang pada sebuah *interface* dari Router sekaligus ditentukan arah penyaringan yang dilaksanakan oleh ACL tersebut.
  10. ACL yang dipasang untuk menyaring paket yang masuk ke sebuah *interface* disebut ACL *inbound*. Sedangkan, ACL yang dipasang untuk menyaring paket yang akan keluar melalui sebuah *interface* disebut ACL *outbound*.
  11. ACL dapat dipasang di setiap *interface* dari Router dengan syarat satu ACL untuk satu protokol dan satu arah penyaringan.
  12. *Standard ACL* seharusnya dipasang sedekat mungkin dengan tujuan paket data untuk mencegah pemblokiran paket data yang sah.
  13. *Extended ACL* seharusnya dipasang sedekat mungkin dengan asal paket data untuk menghemat *bandwith* jaringan, yakni paket data tersebut tidak perlu masuk ke dalam jaringan jika memang harus ditolak.
  14. Pada akhir baris dari setiap ACL ada baris *implicit deny*, berupa pernyataan *deny all* yang berfungsi untuk menolak setiap paket yang tidak memenuhi satupun kriteria pernyataan *permit* pada baris-baris di atasnya.
  15. Jika sebuah ACL tidak memiliki satupun pernyataan, baik pernyataan *permit* maupun *deny*, maka ACL tersebut tidak akan melewatkan paket data apapun.
  16. Metode *taxonomi* vulnerabilitas dapat digunakan untuk menganalisa dan mengklasifikasikan kelemahan-kelemahan sebuah sistem secara sistematis, sehingga tidak hanya dapat digunakan untuk menyelesaikan permasalahan yang ada saat ini, namun juga menjadi landasan dalam memecahkan permasalahan yang datang di kemudian hari.
  17. ACL dapat digunakan untuk membuat *Packet Filtering Firewall* yang akan melaksanakan penyaringan terhadap setiap paket data.
  18. ACL bukan merupakan solusi total untuk membangun sebuah sistem *Firewall*, apalagi sebagai pengaman tunggal sebuah sistem.
  19. Makin bertambahnya baris pernyataan pada sebuah ACL, makin sulit menjaga konsistensi aturan ACL tersebut sehingga kesalahan makin mungkin terjadi.
  20. Proses penyaringan paket data oleh ACL memerlukan sumber daya pemrosesan milik Router sehingga akan menurunkan kinerja Router secara keseluruhan.
  21. ACL bersifat *stateless*, sehingga tidak dapat memeriksa status dari setiap koneksi yang sedang terjalin pada jaringan.
  22. ACL tidak dapat mengenali apalagi menanggulangi serangan *malware*.
  23. ACL sangat rentan terhadap *IP Spoofing* atau pemalsuan alamat IP.
  24. ACL tidak memiliki metode otentikasi sehingga tidak dapat membedakan antara pengguna yang sah dengan yang tidak.

### 3.2. Saran

Berikut ini saran-saran yang dapat dikemukakan berkaitan dengan penggunaan ACL sebagai perangkat pengamanan.

1. Jika aturan penyaringan paket data masih sederhana, gunakanlah *Standard ACL* agar pembuatan, perawatan, dan auditnya lebih mudah.
2. Saat aturan penyaringan paket data makin rumit, gunakanlah *Extended ACL* agar aturan tersebut dapat terakomodasi dengan baik.
3. Sebelum membuat ACL, rancang terlebih dahulu aturan penyaringan paket data yang diinginkan dengan teliti.
4. Jangan menerapkan ACL pada Router kecuali Anda telah yakin bahwa baris-baris aturan pada ACL tersebut telah sesuai dengan aturan yang ingin Anda terapkan.
5. Untuk menutupi kekurangan ACL dalam hal ketidakmampuannya untuk mendeteksi paket-paket data yang berbahaya yang berhasil menyusup masuk ke jaringan internal, gunakanlah IDS (*Intrusion Detection System*).
6. Gunakanlah NAT (*Network Address Translation*) dan PAT (*port Address Translation*) yang juga merupakan fitur

- keamanan yang terdapat pada Router Cisco untuk mendukung fungsi pengamanan dari ACL, yakni dengan menyembunyikan alamat-alamat IP yang digunakan pada jaringan internal.
7. Untuk menutupi kekurangan ACL dalam hal ketidakmampuannya untuk melaksanakan otentikasi, Anda dapat menggunakan RADIUS (*Remote Authentication Dial-In User Service*) atau TACACS+ (*Terminal Access Controller Access Control System*) dimana dukungan terhadap keduanya telah disediakan dengan baik oleh Cisco.
  8. Gunakanlah juga perangkat keamanan lain untuk mendukung ACL, seperti *Proxy Server*, aplikasi *antimalware*, *bastion host*, dan sebagainya.
  9. Untuk pengamanan yang lebih optimal dan jika tersedia sumber daya manusia yang cakap gunakanlah *honeypot* untuk mengalihkan setiap usaha serangan yang masuk.
  10. Lakukanlah *update* terhadap Cisco IOS dan kunjungi situs resmi mereka untuk mendapatkan *white paper* dan berbagai laporan penting lainnya yang berkaitan dengan peralatan yang Anda gunakan.
  11. Kunjungilah situs-situs yang memberikan informasi aktual berkaitan dengan keamanan jaringan komputer untuk mengantisipasi jenis serangan, dan celah keamanan serta *bug* yang berkaitan dengan perangkat-perangkat yang Anda gunakan.
  12. Gunakanlah aplikasi pembantu untuk manajemen ACL, seperti Cisco *Works* agar manajemen ACL lebih mudah.
2009. *ACL Analysis Tool*. King Saud University. Arab Saudi.
- Benardi, Beny. 2004. *Membangun Firewall dengan Cisco Router*. Penerbit PT Elex Media Komputindo. Jakarta.
- Brenton, Chris dan Hunt, Cameron. 2005. *Network Security*. Penerbit PT Elex Media Komputindo. Jakarta.
- Cisco System, Inc. Cisco IOS Security Configuration Guide Release 12.2SX [[http://www.cisco.com/en/US/docs/ios/security/configuration/guide/12\\_2sx/sec\\_12\\_2sx\\_book.html](http://www.cisco.com/en/US/docs/ios/security/configuration/guide/12_2sx/sec_12_2sx_book.html)] (Accessed November 17, 2008)
- Cisco System, Inc. TACACS+ and RADIUS Comparison [[http://www.cisco.com/en/US/tech/tk59/technologies\\_tech\\_note09186a0080094e99.shtml](http://www.cisco.com/en/US/tech/tk59/technologies_tech_note09186a0080094e99.shtml)] (Accessed January 14, 2008)
- Habtamu, Abie. 2000. *An Overview of Firewall Technologies*. Norwegian Computing Center. Norwegia.
- Ivan Victor, Krsul. 1998. *Software Vulnerability Analysis*. Purdue University. Amerika Serikat.
- Kamara, Seny dkk. 2003. *Analysis of Vulnerabilities in Internet Firewalls*. Purdue University. Amerika Serikat.
- Wright, Craig S.. 2007. *A Taxonomy of Information Systems Audits, Assessments and Reviews*. SANS Institute. Amerika Serikat
- DAFTAR PUSTAKA**
- Al-Wabel Abdulelah A. dan Al-Shayea Hamid I.