JURNAL ILMIAH
ELEKTRO KRISNA

Vol. 1 No. 3 Juni 2013 ISSN : 2302-4712

Penerbit
Universitas Krisnadwipayana
(Dikelola Oleh Fakultas Teknik
Prodi Teknik Elektro)

SUSUNAN DEWAN REDAKSI

ALAMAT PENERBIT

Universitas Krisnadwipayana
Jl. Kampus UNKRIS Jatiwaringin, Jakarta 13077
Gedung G (Fakultas Teknik) Lantai 2 Ruang Sekretariat Jurusan Teknik Elektro
Telepon : 021-84998529
E-Mail : elektrounkrisna@yahoo.com

DAFTAR ISI

Pencegahan Dan Pendeteksian Serangan Penolakan Layanan (Denial of Service Attack) Dalam Jaringan Komunikasi

Sri Hartanto [1]

Abstrak - Aspek ketersediaan data atau informasi merupakan salah satu aspek keamanan yang penting dalam suatu jaringan komunikasi, karena sangat menentukan kualitas dan kehandalan suatu jaringan komunikasi. Tantangan atau ancaman yang berkaitan dengan jaminan ketersediaan data dapat berupa serangan penolakan layanan. Dengan demikian, diperlukan suatu metode untuk mencegah dan mendeteksi serangan penolakan layanan tersebut.
Kata Kunci - serangan penolakan layanan, ancaman, ketersediaan data

Abstract *– Data or information availability aspect is one of iimportant security aspects in a communication network, due to more determine quality and reliable of a communication network. Challenge or threat in relation with ensurement of data availability is denial of service attack. Therefore, it is required a method to prevent and detect that denial of service attack.*
Index Terms – denial of service attack, threats, availability

I. Pendahuluan

Pada jaringan komunikasi, terdapat enam aspek keamanan yang perlu diperhatikan, yaitu: aspek-aspek keamanan secara umum, yaitu berupa aspek *privacy/confidentially*, aspek *integrity*, aspek *authentication*, dan aspek *availabity*, serta aspek-aspek keamanan yang berkaitan secara khusus dengan perdagangan elektronik melalui internet (*e-commerce*), berupa aspek *access control* dan aspek *non repudiation*. Di antara keenam aspek tersebut, aspek *availability* merupakan aspek yang perlu diperhatikan secara khusus, karena aspek tersebut berkaitan dengan ketersediaan informasi atau data yang harus disajikan atas permintaan pengguna. Aspek ketersediaan data atau informasi tersebut merupakan salah satu aspek keamanan yang penting dalam suatu jaringan komunikasi, karena sangat menentukan kualitas dan kehandalan suatu jaringan komunikasi.

Ancaman atau serangan yang dihadapi dalam menjamin ketersediaan informasi, baik itu berasal dari luar sistem atau berupa kelemahan pada sistem jaringan itu sendiri dapat diidentifikasi berupa: serangan yang dapat menyebabkan terjadinya penolakan akses ke beberapa layanan atau sumber yang diberikan oleh suatu sistem, serangan yang memungkinkan seorang penyusup (*intruder)* dapat mengoperasikan suatu sistem dengan hak yang tidak sah (*unauthorized previlleges*), percobaan memasuki suatu sistem jaringan komunikasi untuk mencari kelemahan potensial yang terdapat pada sistem jaringan komunikasi,

serangan secara fisik yang perlu dihadapi oleh perangkat komunikasi atau komputer, serangan yang disebabkan oleh *worm* atau *virus*.[1]. Bentuk ancaman (*threats*) tersebut berupa serangan penolakan layanan atau *Denial of Service (DoS) attack.*

Oleh karena itu, diperlukan suatu metode untuk mendeteksi, mencegah dan menanggulangi serangan penolakan layanan tersebut; serta menentukan arsitektur jaringan komunikasi yang dapat dirancang untuk menghadapi serangan penolakan layanan.

II. Tinjauan Tentang Serangan Denial of Service (DoS)

Serangan penolakan layanan atau *Denial of Service (DoS) attack* merupakan suatu usaha untuk melumpuhkan sistem jaringan yang dijadikan target serangan sehingga sistem jaringan tidak dapat menyediakan layanan-layanannya. Serangan DoS merupakan serangan yang membanjiri (*flooding*) suatu *server* dengan paket-paket data yang tidak bermanfaat dalam jumlah yang tidak dapat dikendalikan [2]. Serangan DoS dapat menggunakan layanan yang tidak semestinya pada suatu layanan yang terhubung ke internet dengan tujuan untuk mengganggu pihak lain yang menggunakan layanan yang disediakan oleh suatu *server.*

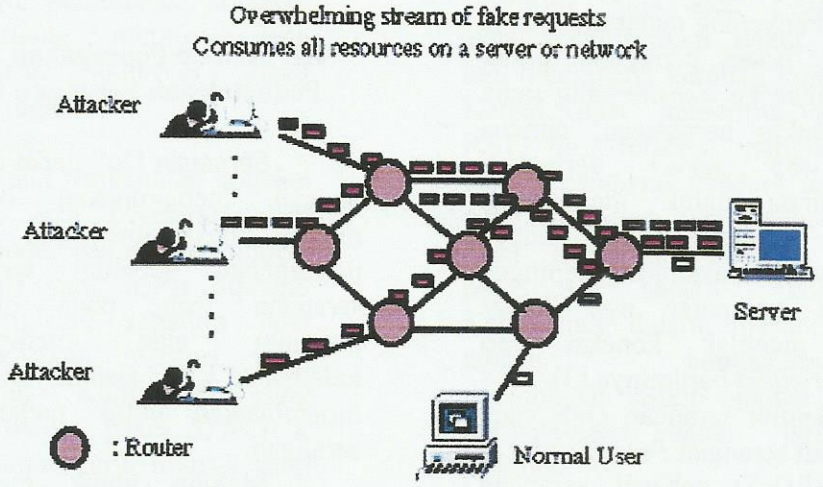
Salah satu bentuk serangan DoS, yaitu *TCP SYN Flood Attack,* dapat dilakukan dengan mengandalkan kelemahan atau efek samping metode peralihan data antara komputer *server* dengan komputer *client* melalui jaringan internet, yang dikenal dengan mekanisme *three way handshake.* Dalam serangan *TCP SYN Flood Attack*, suatu komputer *server* target dibanjiri dengan tanda SYN (*synchronization*), yang merupakan bagian pertama mekanisme *three way handshake*, dalam proses yang mengawali suatu koneksi menggunakan *Transmission Control Protocol (TCP).*

Selanjutnya, serangan DoS lainnya dapat berupa *UDP Flood Attack*, yang memanfaatkan dua layanan pada *User Datagram Protocol (UDP)*, yaitu *echo*, yang didefinisikan sebagai karakter-karakter yang telah diterima dan dikembalikan ke pengirim, serta *chargen*, yang didefinisikan sebagai karakter-karakter yang dihasilkan. Kedua layanan UDP tersebut digunakan di akhir pengetesan kondisi jaringan, yang akan digunakan untuk melewatkan data antara kedua perangkat jaringan, dimana hal ini dimungkinkan secara pasti pada hampir semua sistem jaringan. Namun, kedua layanan tersebut dapat dimanfaatkan untuk melancarkan serangan DoS dengan mengkoneksikan *port-port* yang digunakan untuk *chargen* dengan *port-port* yang digunakan untuk *echo* dalam suatu komputer atau perangkat komunikasi yang sama atau berbeda, sehingga menimbulkan sejumlah besar lalulintas data yang melewati jaringan.

Smurf Attack merupakan serangan DoS yang menggunakan permintaan *Packet Internet Gropher (PING Request)* ke semua alamat yang ada di dalam suatu jaringan komunikasi secara *broadcast.* Seluruh komputer (*device*) yang berada di dalam alamat yang di-*broadcast* tersebut akan menjawab permintaan PING tersebut. Jika

suatu sistem jaringan memiliki banyak komputer (*device*) dan PING yang di-*broadcast* ini dilakukan terus menerus, sistem jaringan dapat dipenuhi oleh tanggapan-tanggapan dari permintaan PING tersebut. Hal ini akan mengakibatkan *bandwith* yang dimiliki jaringan akan berkurang atau bahkan habis, sehingga jaringan menjadi lambat atau bahkan *hang*. *Smurf attack* biasanya dilakukan dengan menggunakan *IP Spoofing*, yaitu mengubah alamat IP pada perangkat komunikasi atau komputer penyerang dengan alamat IP yang dimiliki oleh perangkat komunikasi atau komputer target serangan (korban), sehingga seolah-olah perangkat komunikasi atau komputer target serangan (korban) merupakan asal (sumber) permintaan PING ke sistem jaringan. Dengan menggunakan IP *spoofing*, tanggapan dari PING tersebut kemudian dialamatkan ke komputer yang IPnya sudah di-*spoof* tersebut.. Akibatnya, perangkat komunikasi atau komputer target serangan (korban) akan menerima banyak paket data tanggapan permintaan IP. Dapat dibayangkan apabila komputer yang di-*spoof* tersebut memiliki koneksi berkecepatan rendah dan PING diarahkan ke sistem jaringan yang memiliki banyak *host*. Hal ini dapat menyebabkan terjadinya *hang* yang mengakibatkan terjadinya penolakan layanan.

Distributed Denial of Service (DDoS) merupakan serangan DoS dalam skala yang lebih besar, di mana serangan dilaksanakan secara terkoordinasi menggunakan ketersediaan layanan yang terdapat pada sistem jaringan korban [4], seperti terlihat pada Gambar 1 berikut :



Gambar 1 Distributed Denial of Service Attack [4]

Serangan DDoS dilancarkan dengan pengiriman suatu paket dengan *volume* yang sangat besar secara ekstrim ke mesin target melalui kerjasama secara simultan sejumlah *host* yang tersebar pada jaringan internet. Lalu lintas serangan yang begitu besar akan menghabiskan sumber daya *bandwidth* jaringan atau sumber

daya *computing* pada *host* target serangan, sehingga permintaan koneksi yang sebenarnya atau yang sah tidak dapat dilayani *(discarded)*. Akibat dari serangan ini dapat menyebabkan ketidaknyamanan pengunjung suatu *website* dalam skala yang kecil, hingga dalam skala yang lebih besar berupa kehilangan data keuangan yang dipercayakan pada layanan on-line melalui jaringan internet. Serangan DDoS muncul sebagai suatu cara yang lazim digunakan untuk mematikan aktifitas suatu organisasi di internet dan menghasilkan kerugian keuangan *(financial losses)* pada waktu yang bersamaan. Dalam serangan DDoS, koneksi salah satu elemen (perangkat) dalam jaringan komunikasi diputus dengan memutuskan sambungan *(link)* atau titik simpul *(node)* jaringan komunikasi [1]. Serangan DDoS merupakan serangan yang dikirimkan dari berbagai sistem sumber. Penyerang dapat mengelola sejumlah besar pengguna untuk berhubungan ke *website* yang sama dalam waktu bersamaan, dimana suatu *web server* seringkali dikonfigurasi untuk melewatkan koneksi komputer *client* hingga jumlah yang maksimum, sehingga membuat komputer *web server* korban menolak koneksi dari komputer *client* berikutnya.[3]

Bentuk serangan DoS yang lain adalah serangan *Pulse Denial of Srvice (PDoS)* sebagai serangan berlanjut atau berdenyut (*pulse*) pada aliran paket data yang menggunakan protokol TCP [2]. Serangan ini seringkali mengirimkan suatu urutan *(sequence)* serangan yang meningkatkan aktifitas *router* korban (target), dan penyilangan aliran TCP akan membuat paket-paket data hilang secara periodik. Dengan demikian, secara signifikan, akan terjadi penurunan tingkat keluaran. Serangan PDoS ini membatasi pengirim TCP untuk mengakses jaringan komunikasi sehingga terlempar keluar *(timeout state)* dari jaringan komunikasi. Hal ini dapat dilakukan oleh penyerang dengan mengirimkan serangan berdenyut *(pulse)* dengan pemilihan waktu secara instan. Selanjutnya, serangan dilakukan dengan mengaktifkan mekanisme pengelolaan antrian sehingga *router* korban memasuki kondisi *transient*. Akhirnya, serangan PDoS menggunakan serangan berdenyut (*pulses*) yang menyebabkan pengirim TCP korban atau target serangan mengalami kebuntuan (*congestion*) dalam mengakses jaringan komunikasi, sehingga memutuskan frekuensi sinyal yang dikirimkan.

III. Metode Pencegahan Dan Pendeteksian Serangan DoS

Serangan DoS dapat dicegah dengan menggunakan berbagai macam mekanisme atau metode penanganan sebelum terjadinya serangan yang pada dasarnya menutup atau memperbaiki kelemahan suatu sistem yang dapat dimanfaatkan untuk melancarkan serangan.

Metode untuk mencegah serangan *TCP SYN Flood* adalah sebagai berikut

1. *Micro Block*

   Ketika suatu *host* menerima paket pengenalan koneksi, maka *host* tersebut akan mengalokasikan ruang memori yang sangat kecil, sehingga dapat

menerima koneksi lebih banyak. Ruang memori diharapkan dapat menampung semua koneksi yang dikirimkan, sampai terjadi *connection time out*, yaitu suatu kondisi dimana koneksi tidak dapat menyelesaikan proses *three way handshake* secara keseluruhan, atau tidak dapat melakukan transaksi data dalam kurun waktu yang telah ditentukan, padahal sambungan koneksi sudah diberikan. Pada kondisi tersebut, koneksi akan diputus, dan paket dengan tanda *SYN* kemudian dihapuskan dari ruang memori, agar dapat memberikan ruang bagi permintaan koneksi yang baru. Metode ini tergantung pada kecepatan serangan yang dilakukan, karena apabila serangan paket pengenalan koneksi dikirimkan lebih cepat dari lamanya waktu untuk menunggu pemutusan koneksi, maka ruang memori yang dialokasikan akhirnya dapat diserang juga.

2. *SYN Cookies*
   Ketika paket pengenalan koneksi diterima, *host* penerima akan mengirimkan paket permintaan konfirmasi yang harus dijawab oleh pengirim, sebelum *host* mengalokasikan memori yang dibutuhkan. Konfirmasi yang diminta berupa paket *SYN-ACK* dengan nomor urut khusus yang merupakan hasil dari fungsi *hash* dengan masukan berupa alamat IP pengirim, nomor port dan lain-lain. Jawaban dari pengirim paket pengenalan koneksi harus berisikan data-data tersebut. Untuk melakukan perhitungan, *hash* memerlukan sumber daya komputasi yang cukup besar, sehingga banyak *server* yang aplikasinya membutuhkan kemampuan komputasi yang tinggi, tidak dapat mempergunakan metode ini. Metode ini mengubah waktu pengalokasian memori, yang seharusnya merupakan proses awal menjadi proses akhir dalam *three way handshake*. Diperlukan cara yang lebih baik untuk menentukan urutan paket tersebut, sehingga sulit untuk ditebak.
3. *RST Cookies*
   Sebagaimana halnya dengan metode *SYN Cookies*, *RST Cookies* juga mengirimkan paket permintaan konfirmasi yang harus dijawab oleh pengirim, sebelum *host* mengalokasikan memori yang dibutuhkan. Hanya saja, paket tersebut adalah paket yang salah. Apabila pengirim paket pengenalan koneksi adalah pengirim yang sah, pengirim akan mengirimkan paket RST lalu mengulang kembali koneksi. Ketika *host* menerima paket RST, *host* tersebut mengetahui bahwa pengirim tersebut adalah pengirm paket yang sah, dan akan menerima koneksi dari pengirim secara normal. Kelemahan dalam metode ini adalah ketidaksesuaian penggunaan sistem operasi yang berbeda-beda dalam jaringan internet.

Selain dengan metode pencegahan di atas, serangan *TCP SYN Flood Attack* dapat juga dicegah dengan menggunakan perangkat lunak *Ingress* atau *Egress Router Fiter* yang dipasang di *server* untuk menghindari *IP Spoofing* lokal.

Untuk mencegah serangan *UDP Flood Attack*, yang pertama

harus dilakukan adalah dengan menghilangkan (*disable*) kedua layanan UDP, berupa *echo* dan *chargen* yang digunakan untuk pengetesan akhir kondisi jaringan, dengan mengetikkan perintah ***/etc/inetd.conf*** di *Console* sistem operasi Linux dan ***no udp small services*** pada perangkat *Cisco IOS Router*. Kemudian, lalulintas paket data yang melalui protokol UDP disaring dengan tingkatan firewall tertentu. Hanya lalulintas yang sah saja yang diperbolehkan melewati port 53, yang memberikan layanan *Doman Name System (DNS)*, yaitu layanan penamaan suatu *web server* berdasarkan *IP Public* yang digunakan.

Untuk mencegah serangan *Smurf Attack*, *Router* yang ada perlu dikonfigurasi untuk menolak lalulintas data yang mem-*broadcast* (menyebarkan) alamat IP pada sistem jaringan komunikasi yang dikelola, dimana kemungkinan serangan berasal dari jaringan luar. Hampir pada semua kasus, fungsi IP yang di-*broadcast* tidak dibutuhkan. Kemudian, *host* atau komputer *server* perlu dikonfigurasi melalui *variable kernel* untuk tidak mengulangi (*not reply*) paket data ke pengirim paket yang menyebarkan alamat IP. Terakhir, adalah dengan mengkonfigurasi perangkat lunak *Ingress* atau *Engress Filter* pada *Router* yang dapat mengantisipasi adanya *IP Spoofing*.

Terdapat dua metode pendeteksian serangan DDoS, yaitu:

1. *Sequential Methode Detection (SMD)*
   Merupakan sistem pendeteksian dalam suatu lokal jaringan, dengan dua fase pendeteksian, yaitu fase pengetesan urutan atau *Sequential Test Method (STM)* dan fase untuk mengawasi adanya alamat IP yang baru. Dua fase tersbut akan memberitahukan pengelola jaringan apabila menemukan beberapa paket data yang melewati ambang batas kewajaran.
2. *Global Detector*
   Merupakan sistem pendeteksian pada jaringan secara menyeluruh.

Pada pendeteksian dengan metode SMD, fase pertama berupa pengetesan urutan, sejumlah permintaan dan sejumlah tanggapan dihitung, dengan deret waktu T_1, T_2, T_3,.....T_n, sehingga didapatkan jumlah paket yang ditandai dengan SYN dan paket yang ditandai dengan FIN (RST). Setiap periode pencuplikan (*sampling time*), dapat dihitung dengan jumlah rata-rata tanggapan R, dengan rumus sebagai berikut :

$$\Delta n = \frac{\sum_{i=1}^{n} X_i}{R}$$

dimana,

$\sum_{i=1}^{n} X_i$ = jumlah nomor permintaan untuk suatu periode pencuplikan

R = jumlah nomor tanggapan

Apabila Δn melebihi ambang batas nilai N, maka alarm akan dibangkitkan.

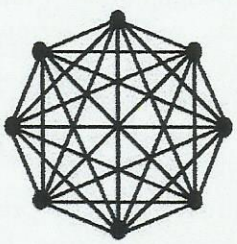
Sistem pendeteksian serangan *Pulse Denial of service (PDoS)* pada sisi penerima didasarkan pada kehadiran anomali lalulintas paket data yang dibuat oleh suatu serangan, dengan periode kedatangan paket TCP yang berubah-ubah (*fluctuation*) dan terdapat kecenderungan penurunan lalulintas paket TCP yang ditandai

dengan ACK (*acknowledgement*). Pada tahapan pertama, sistem mengawasi lalulintas paket data ACK yang dikirim dengan menggunakan d*iscrete wavelet transform*. Pada tahapan kedua, terdapat algoritma non parameter CUSUM untuk mendeteksi serangan PDoS dengan interval serangan yang konstan. Dengan perangkat *Vanguard*, serangan PDoS dapat dideteksi dari sisi penerima TCP dengan menganalisa lalulintas paket data TCP yang datang dan pergi menggunakan tanda ACK. *Vanguard* dirancang untuk mendeteksi serangan yang datang ke beberapa *host* yang ditempatkan di belakangnya (jaringan lokal). *Host-host* tersebut menjalankan aplikasi TCP pada *client* untuk menerima data dari jaringan luar.

IV Arsitektur Jaringan Untuk Pencegahan Serangan DoS

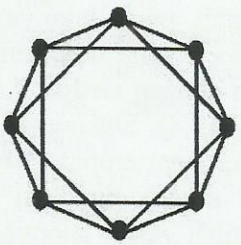
Arsitektur jaringan komunikasi yang dapat dibuat untuk mencegah serangan DoS dan DDoS adalah :

1. *Strongly connected network*
   Suatu graph G terhubung secara baik jika untuk semua x,y $\subset V$, terdapat pada tepian suatu jaringan. Sudut setiap puncak adalah d(v) = n-1, di mana n merupakan nomor titik simpul, seperti terlihat pada Gambar 2.



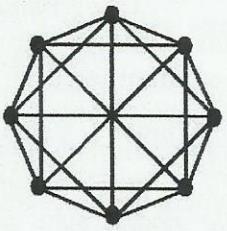
Gambar 2. Strongly Connected Network

2. m-*Conneced network*
   Nomor terkecil dari tepian suatu *m-connected graph* pada n puncak dapat berupa m, dengan asumsi bahwa m < n. Sudut tepian puncak adalah d(v) = m. Tergantung pada nilai m yang memiliki dua sub kasus, yaitu :
   a. m adalah genap
      Bila m= 2r, lalu G < n, 2r > dikonstruksikan mengikutinya. Hal ini akan memiliki nilai puncak 0,1,2,....,n-1 dan dua puncak i dan j yang digabung jika i-r = j = i + r (dimana tambahannya diambil dari *modulo* n), seperti terlihat pada Gambar 3.



Gambar 3. m-Conneced Network Genap

   b. m adalah ganjil
      Bila m = 2r+1, lalu G < n, 2r+1 > dikonstruksikan dengan penggambaran pertama G < n, 2r> dan lalu penambahan penggabungan tepian puncak i ke puncak 1 + (n/2) seperti terlihat pada Gambar 4.



Gambar 4 m-Conneced Network Ganjil

Pada ketiga gambar tersebut, rancangan arsitektur jaringan terlihat adanya kesamaan gambar, yaitu jaringan dengan topologi jala (*mesh*) yang sama-sama memiliki titik simpul atau *node (n)* yang sama, yaitu berjumlah delapan, dimana diasumsikan adanya kebutuhan delapan *Router* yang nantinya digunakan sebagai *node*, untuk menghubungkan bagian jaringan yang satu dengan bagian jaringan yang lain. Perbedaan dari ketiga gambar tersebut terletak pada seberapa banyak saluran koneksi (*link*) suatu *node* yang dapat dihubungkan ke *node* lainnya. Pada Gambar 2 terdapat tujuh *link* (n-1) yang dapat menghubungkan suatu *node* ke tujuh *node* lainnya, dari kedelapan *node* yang ada, sehingga disimbolkan dengan *Graph* 7 *link* 8 *node* atau G (7,8). Sedangkan pada Gambar 3 terdapat empat *link* (n/2) yang dapat menghubungkan suatu *node* ke empat *node* lainnya, dari kedelapan *node* yang ada, sehingga disimbolkan dengan *Graph* 4 *link* 8 *node* atau G (4,8). Terakhir, pada Gambar 4., terdapat lima *link* (1 + n/2) yang dapat menghubungkan suatu *node* ke lima *node* lainnya, dari kedelapan *node* yang ada, sehingga disimbolkan dengan *Graph* 5 *link* 8 *node* atau G (5,8)

Dalam menghadapi serangan DoS, dapat juga dirancang suatu bentuk jaringan yang dapat diprogram (*programmable networks*). Pada dasarnya, jaringan yang dapat diprogram tersusun dari sejumlah *Router* yang dapat diprogram, yang disebut dengan pemrograman titik simpul (*programmable node)* atau *active node* di dalam suatu jaringan *Internet Protocol (IP). Programmable node* mengidentifikasi paket khusus yang disebut dengan *active packet* dan memberikan suatu kode khusus untuk memprosesnya. *Active packet* berasal dari suatu sistem sumber akhir ke suatu sistem tujuan akhir, di mana *programmable node* berada di dalam suatu bagian antara sumber dan tujuan pemrosesan *active packet*,

Rancangan jaringan yang dapat diprogram meliputi jaringan dengan banyak layanan (*multiservice network*) dan jaringan dengan banyak penyedia layanan (*multidomain network*). Jaringan dengan banyak layanan membuat pengguna jaringan IP dapat menjalankan layanan pemrograman melalui *programmable node* yang berada di dalam arsitektur jaringan yang dirancang tersebut. Pemrograman titik simpul (*programmable node*) menjalankan kode untuk memproses paket data yang aktif (*active packet*), yang dapat membawa data pengguna dan mengendalikan informasi.

Pada jaringan yang dapat diprogram, bentuk jaringan atau topologinya dapat berubah, sehingga dapat menghasilkan perubahan rute jaringan dengan memanfaatkan titik simpul terprogram baru yang terlihat di dalam suatu jaringan atau pada saat *programmable node* tidak dapat berfungsi (*down*). Perubahaan topologi dapat menyebabkan perubahan pengambilan tempat secara tiba-tiba, sebagai permulaan *programmable node* yang baru dalam memproses *active packets* dari layanan terprogram atau pada saat *programmable node* lain menahan pemrosesan *active packet*. Arsitektur keamanan harus tahan

(immune) terhadap perubahan topologi ini.

Pada jaringan *multiservice,* proses *authentication* perlu diverifikasi pada saat seorang pengguna meminta suatu layanan ke *multiservice network*, memproses suatu kode *server* ke *programmable node,* dan pada saat suatu *programmable node* menerima suatu *active packet*. *Programmable node* dapat memodifikasi sebagian *active packet*, yang disebut dengan *dynamic packet*. Hal ini membutuhkan adanya mekanisme keamanan baru, di mana hanya modifikasi yang telah dibuat oleh *programmable node* yang sah *(authorized)* saja yang dapat dilewatkan. Mekanisme ini merupakan layanan terpadu yang dapat meyakinkan bahwa *dynamic part* dari *active packet* tidak dapat dimodifikasi oleh *programmable node* yang tidak sah. Hal ini tentu saja dilakukan untuk menghindari kemungkinan adanya serangan DOS, yang dapat meracuni proses *authentication*, dengan terlebih dahulu mencuri *active packet*.

Proses klarifikasi sistem keamanan pada rancangan jaringan komunikasi di atas dapat diuraikan sebagai berikut :

1. Pengguna meminta pengesahan (*authorization*) dari *Authorization Server*, dengan mengirimkan parameter C, SST, SET, D, S, U dan SP. Pengertian dari parameter- parameter tersebut adalah sebagai berikut :
   a. Parameter C atau *Code* adalah parameter yang mengidentifikasi kode yang dapat dijalankan (*executeable code*) yang harus memproses *active packet* untuk menawarkan layanan di dalam *programmable node*. Karena setiap layanan yang dapat diprogram tergabung dengan *executeable code* yang berbeda, C_i merupakan nilai yang mengidentifikasi layanan yang dapat diprogram, yang dibutuhkan oleh seorang pengguna.
   b. Parameter SST atau *Service Start Time* adalah parameter waktu di mana layanan terprogram dimulai. *Programmable node* tidak harus memproses *active packet* yang datang sebelum waktu diindikasi atau ditentukan oleh parameter SST ini.
   c. Parameter SET atau *Service End Time* adalah parameter waktu di mana layanan terprogram berakhir. *Programmable node* tidak harus memproses *active packet* yang datang setelah waktu diindikasi atau ditentukan oleh parameter SET ini.
   d. Parameter D atau *Destination* adalah parameter alamat IP tujuan dari *active packet*, dan merupakan akhir dari suatu sistem komunikasi yang dilayani oleh jaringan komunikasi. *Active packet* harus diverifikasi di antara parameter S dan D atau di antara awal dan akhir sistem komunikasi
   e. Parameter S atau *Source* adalah parameter yang menyatakan alamat IP sumber dari *active packet* yang dikirimkan
   f. Parameter U atau *User* adalah parameter yang mengidentifikasi pengguna

(*user*) yang meminta layanan terprogram, yang ditanggapi untuk kebutuhan penggunaan layanan yang terprogram.

g. Parameter SP atau *Spesific Parameter* adalah parameter khusus yang tergantung pada tanggapan dalam layanan terprogram. *Programmable node* harus mengaplikasikan aturan pengesahan untuk *active packet* yang datang. Untuk membuat mungkin, parameter pengesahan harus hadir pada setiap *programmable node*

2. *Authorization Server* sebagai pemberi pengesahan, menghasilkan *session key* dan mengirimkannya ke pengguna.
3. Pengguna menghasilkan suatu *active packet* dengan mengenalkan parameter pengesahan yang telah dikirimkan dan melindunginya dengan menggunakan *sessio key*. Akhirnya, pengguna akan mengirimkan *active packet* ke arah tujuan (D).
4. Pada saat *programmable node* menerima suatu *active packet*, yang mana tidak memiliki *execution code* yang diidentifikasi oleh paramter C, dan tergabung ke dalam kunci Kc_i, *programmable node* akan mengambilnya *(download)* dari *Code Server*. Lalu, *programmable node* menghasilkan *session key* dengan menggunakan kunci Kc_i dan mengesahkan parameter yang membawa *active packet*, serta memverifikasi *integrity* dan *authentication* dari *active packet* tersebut. *Programmable node* juga memverifikasi pengesahan untuk pemrosesan paket dengan menggunakan parameter-paramter pengesahan.
5. Sekali *active packet* diproses, jika kemudian dimodifikasi, *programmable node* akan melindungi *active packet* tersebut dengan menggunakan *session key*. Akhirnya, *programmable node* mengirimkan *active packet* ke arah tujuan.

Dalam skenario *multidomain network* terdapat sejumlah *domain*, di mana sejumlah *active packet* yang memasuki beberapa sesi akan dapat melewati *domain* tersebut. Pengguna harus bernegoisasi dengan *Server* yang berada dalam tingkatan di atas *domain-domain* atau penyedia layanan ini, untuk mendapatkan pengesahan (*authorization*). Terdapat peluang bagi semua *domain* untuk menentukan di mana pengguna yang sah dapat menerima permintaan layanan yang dapat diprogram. Sekali sudah ditentukan, yang mana *domain* yang akan mengambil layanan pemrograman, *domain* akan mengubah suatu kunci sesi. Kunci sesi ini digunakan oleh sistem pengguna terakhir dan suatu *programmable node* untuk melindungi *active packet*. Solusi keamanan di dalam suatu jaringan dengan banyak penyedia layanan yang dapat diprogram harus melalui beberapa fase sebagai berikut :

1. Proses pencarian di mana *domain* mengambil bagian di dalam suatu sesi pada layanan terprogram.
2. Proses negosiasi untuk semua sesi dengan *domain* yang dihadapi.
3. Proses perlindungan untuk *active packet*.

Solusi pada *multidomain network* harus memenuhi kebutuhan

topologi. Hal ini berarti bahwa pengguna dan *programmable node* tidak memerlukan pengetahuan topologi yang terdapat pada jaringan yang dapat diprogram. Solusi yang dirancang harus terukur, di mana pemrosesan pembawaan *active packet* oleh *programmable node* tidak akan meningkat pada saat *active packet* melewati beragam *domain*. Jaringan yang dapat diprogram harus berada pada tepi jaringan, sehingga program jaringan dapat selalu diambil oleh ISP yang memberikan layanan langsung ke pengguna. Oleh sebab itu, suatu sesi untuk *multidomain* akan menghasilkan dua *domain*, dan pada beberapa situasi yang ekstrim dapat meningkat jadi empat *domain*.

V. Kesimpulan

Dari pembahasan yang telah diuraikan di atas dapat diambil kesimpulan bahwa :

1. Serangan *DoS* merupakan suatu usaha untuk melumpuhkan sistem jaringan yang dijadikan target serangan sehingga sistem jaringan tidak dapat menyediakan layanan-layanannya, atau dapat saja tingkat kualitas layanan menjadi menurun dengan drastis dalam waktu yang singkat.
2. Penyerang tidak memiliki komputer-komputer yang digunakan untuk melakukan serangan DDoS, tetapi menggunakan komputer perantara dalam melancarkan suatu serangan DDoS. Penyerang umumnya menyebarkan *Trojan Horse* yang terdiri dari beberapa kode berbahaya (*malicious code*) yang dapat membuat seorang penyerang (*attacker*) dapat mengendalikan sistem yang mereka miliki.
3. Serangan DOS dapat dicegah dengan menggunakan berbagai macam mekanisme atau metode penanganan sebelum terjadinya serangan (*preventing of attack*) dengan menutup atau memperbaiki kelemahan suatu sistem yang kemungkinan dapat dimanfaatkan untuk melancarkan serangan.
4. Terdapat beberapa topologi atau bentuk jaringan yang menghubungkan antara beberapa *Router* yang terdapat di dalam suatu jaringan komunikasi. Macam macam topologi tersebut adalah topologi *ring, mesh, bus* dan *star*..
5. Dalam menghadapi serangan DoS, dapat juga dirancang suatu bentuk jaringan yang dapat diprogram (*programmable networks*). Jaringan yang dapat diprogram tersusun dari sejumlah *Router* yang dapat diprogram, yang disebut dengan pemrograman titik simpul (*programmable node)* atau *active node* di dalam suatu jaringan *Internet Protocol (IP). Programmable node* mengidentifikasi paket khusus yang disebut dengan *active packet* dan memberikan suatu kode khusus untuk memprosesnya.

Daftar Pustaka

[1] Behin Sam, S., Sujatha, S., Kannan, A., and Vivekanandan, P, Network

topology against distributed denial of service attacks, 2006, Information
Technology Journal 5 (3): 489-493
[2] Luo, Xiapu., W.W.Chan, Edmond., and K.C. Chang, Rocky, Detecting
pulsing denial-of-service attacks with nondeterministic attack intervals, 2009,
EURASIP Journal on Aadvances in Signal Processing, Volume 2009, Article
ID 256821, 13 pages, doi:10.1155/2009/256821: 1-13
[3] Nagesh, H.R., and Chandra Sekaran, K, Design and development of proactive
models for mitigating denial of service and distributed denial – of – service
attacks, 2009, IJCSNS International journal of Computer Science and
Network Security, Vol.7 No.7, July 2007 : 167-175
[4] Meenakshi, S and S.K Srivatsa, A distributed framework with less false
positive ratio against distributed denial of service attack. Information
Technology, 2007, Journal 6 (8): 1139-1145
[5] Alarco, Bernardo., Sedano, Marifeli., and Calderon, Maria, Multidomain
network based on programmable networks : security architecture, 2005, *ETRI Journal*, Volume 27, Number