SISTEM KEAMANAN JARINGAN DALAM MENCEGAH *FLOODING DATA* DENGAN METODE BLOKING IP DAN *PORT*

Budi Triandi

Teknik Informatika, Fakultas Teknik dan Ilmukomputer Universitas Potensi Utama
Jl. KL Yos Sudarso Km. 6,5 No. 3-A Tanjung Mulia-Medan
Email : buditriandi@gmail.com

Abstrak

Serangan yang terjadi pada server dapat terjadi kapan saja. Dengan demikian dibutuhkan sistem pertahanan didalam server yang dapat menganalisa langsung apakah setiap paket yang masuk adalah data yang diharapkan atau data yang tidak diharapkan. Jika data yang masuk merupakan data yang tidak diharapkan maka semestinya komputer harus dapat mengambil tindakan yaitu dengan memblok IP asal paket tersebut. Pada makalah ini akan membahas bagaimana membangun sebuah sistem yang dapat mencegah terjadinya flooding data dengan cara pemblokiran IP dan Port dengan cara membuat suatu firewall yang aktif yang bisa mendefenisikan setiap data yang masuk kedalam server untuk mendeteksi apakah data tersebut merupakan sebuah data flood atau data yang diperlukan oleh user, pendeteksian data dilakukan melaui paket TCP, UDP dan ICMP, dari hasil pengujian Flooding ICMP dan Flooding UDP Paket Besar dan Paket Kecil dengan pengujian besar paket data dibawah 100 byte untuk paket kecil dan diatas 100 byte untuk paket besar, sistem dapat memblok flooding data berdasarkan ketentuan, Pengujian flooding TCP dengan Port yang diperbolehkan 8080, 3128, 80 apabila paket data TCP SYN yang datang melebihi ketentuan maka sistem dapat melakukan pemblokan IP dengan priode TCP SYN 1 - 2000

Kata kunci: *Blokir IP, Flooding data, paket TCP.*

1. Pendahuluan

Saat ini banyak perusahaan yang telah memanfaatkan teknologi internet sebagai sarana komunikasi data untuk melaksanakan rutinitas harian perusahaan dalam oprasional perusahaan. Dalam hal ini tidak hanya perusahaan yang bergerak di bidang telekomunikasi saja yang menggunakan internet, tetapi juga perusahaan lain yang tidak bergerak di bidang tersebut. Kecenderungan penggunaan internet ini disebabkan oleh dengan adanya internet akan didapatkan kemudahan dalam hal komunikasi dan transfer data. Kenyataan ini bisa di lihat pada bidang perbankan sistem komunikasi data sangat berguna membantu perusahaan tersebut untuk melayani para nasabahnya, juga dalam bidang marketing suatu barang hasil industri suatu perusahaan. Kemudahan dan kepraktisan merupakan kunci dari mengapa dipilihnya internet. Tetapi disamping keuntungan yang banyak tersebut, internet juga menyimpan banyak kekurangan yang sangat mengkhawatirkan bagi para penggunanya. Salah satu yang sangat menjadi kendala adalah dalam bidang keamanan. Banyak kasus yang membuktikan bahwa perusahaan yang tersambung dengan internet sering kali mendapatkan gangguan baik data yang dimiliki maupun peralatannya. Kerugian yang diderita akan hal ini bisa dibilang tidak kecil. Dalam faktor keamanan ini biasanya perusahaan menempatkan administrator untuk menjaga. Dari hasil penelitian sebelumnya network administrator dituntut dapat menjaga sistem yang dibangun dari serangan *hacker* atau seorang *client* yang ingin merusak sistem. Otentikasi *user* merupakan menu yang harus ada untuk memberikan hak akses pada client [1]. Permasalahan yang terjadi yaitu keterbatasan waktu administrator, saat jam kerja. Meskipun di jam kerja pun kadang kala karena terlalu banyaknya aliran data tentunya administrator akan kesulitan menganalisa apakah data yang diterima oleh server adalah data yang diharapkan atau data yang tidak diharapkan. Sedangkan suatu serangan ke sistem keamanan bisa terjadi kapan saja. Baik pada saat administrator sedang kerja atupun tengah malam dimana tidak ada yang menjaga *server* tersebut. Dengan demikian dibutuhkan sistem pertahanan didalam server itu sendiri yang bisa menganalisa langsung apakah setiap paket yang masuk tersebut adalah data yang diharapkan ataupun data yang tidak diharapkan. Kalau paket tersebut merupakan data yang tidak diharapkan, diusahakan agar komputer bisa mengambil tindakan untuk mengantisipasi agar serangan yang terjadi tidak menimbulkan kerugian yang besar. Akan lebih baik kalau server bisa mengantisipasinya langsung, sehingga kerugian bisa mendekati nol atau tidak ada sama sekali. Makalah ini bertujuan membahas perancangan sistem untuk mencegah terjadinya *flooding* data, merancang sebuah sistem yang dapat memisahkan data sehingga dapat dideteksi sejak dini dan merancang sistem keamanan yang dapat memdeteksi serangan paket *flood* data yang besar.

1.1 Flood Data

Traffic data yang ada dalam suatu jaringan akan mengalami turun naik selama pemakaiannya. Pada jam-jam sibuk traffic suatu data akan sangat padat, sehingga traffic data tersebut akan terganggu. Baik data yang akan

dikirim maupun data yang akan datang akan mengalami antrian data yang mengakibatkan kelambatan dalam pengiriman dan penerimaan data.
Macam-macam Flood attack :

1. *Ping of death*
   Pengiriman paket *echo request* ICMP ke dalam suatu jaringan secara berlebihan. Pengiriman paket ini dapat mengakibatkan sistem crash, hang ataupun *reboot*
2. *Smurf Attack*
   Hampir sama dengan *Ping of death* tetapi untuk *smurf attack* paket ICMP tidak dikirim secara langsung ke korban, melainkan melalui perantara. Pada awalnya dikirim sebuah paket ICMP *echo request* ke sebuah *host* lain, paket ini bertujuan agar host tersebut mengirimkan paket ICMP PING secara terus menerus ke korban terakhirnya.
3. *Syn Flooding*
   Dalam proses pengiriman data yang melalui TCP, proses data yang terjadi adalah sebagai berikut :

Gambar 1. *Proses data TCP*

Pada gambar 1 Proses data TCP menunjukan Hubungan TCP dimulai dengan mengirimkan paket SYN-TCP ke host yang dituju, pengiriman paket SYN adalah merupakan pembuka untuk membuka jalur koneksi antara dua *host* melalui protokol TCP. Apabila hubungan tersebut disetujui host tujuan akan mengirimkan paket SYN-ACK sebagai tanda bahwa jalur sudah terbentuk. Dan bagian terakhir adalah pengiriman paket ACK dari *host* awal ke host tujuan sebagai konfirmasi.

Sedangkan *flood SYN* terjadi bila suatu host hanya mengirimkan paket SYN TCP saja secara kontinyu tanpa mengirimkan paket ACK sebagai konfirmasinya. Hal ini akan menyebabkan host tujuan akan terus menunggu paket tersebut dengan menyimpannya kedalam *backlog*. Meskipun besar paket kecil, tetapi apabila pengiriman SYN tersebut terus menerus akan memperbesar *backlog*. Hal yang terjadi apabila backlog sudah besar akan mengakibatkan host tujuan akan otomatis menolak semua paket SYN yang datang, sehingga host tersebut tidak bisa dikoneksi oleh host-host yang lain .

4. *UDP flood*
   Pengiriman data UDP secara berlebihan kedalam suatu jaringan, pengiriman UDP *flood* ini akan membentuk suatu jalur hubungan dengan suatu *servis* UDP dari *host* tujuan. Flood UDP ini akan mengirimkan karakter-karakter yang akan mengetes jaringan korban. Sehingga terjadi aliran data yang tidak perlu dalam jaringan korban tersebut[2].

1.2 Metode Pengambilan data

Agar bisa mengidentifikasikan suatu data apakah data tersebut termasuk data yang diperlukan oleh user dari *server* tersebut ataukah data yang termasuk data yang tidak dibutuhkan dengan kata lain data serangan[2]. Maka terlebih dahulu harus bisa mendapatkan keterangan-keterangan dari semua data yang masuk. Kemana tujuan dari data itu, darimana datangnya data itu ataupun berapa jumlah byte yang dibawa oleh data. Keterangan-keterangan yang di butuhkan tersebut dapat di peroleh langsung dari data itu.
Yang jadi permasalahan adalah "Bagaimana cara mendapatkan data yang terdapat pada paket tersebut dari jaringan yang ada ?". Semua komunikasi data yang terjadi di jaringan akan melewati router sebagai perantaranya. Baik data dari jaringan lokal yang akan keluar ke internet, maupun data yang datang dari luar jaringan lokal yang menuju ke lokal. Bisa di katakan router merupakan satu-satunya tempat yang memungkinkan untuk mendapatkan semua data paket yang ada. Pengambilan data paket yang akan berfungsi sebagai masukan untuk mengidentifikasikan paket bisa juga dilakukan di router tersebut.
Tetapi fungsi router sebagai pengatur bisa terganggu kalau harus dilakukan pengecekan yang rutin terhadap sejumlah database yang besar. Untuk itu di buat suatu komputer yang di tujukan untuk mengatur lalulintas keluar masuknya data di tempat lain.
Untuk pengambilan data tersebut dilakukan oleh *sniffer*, *Sniffer* disini adalah suatu program yang dapat mengambil setiap paket yang masuk dan keluar didalam suatu jaringan. Sehingga setiap ada data yang melewati jaringan tersebut bisa terdeteksi dan bisa dilihat isi dari data tersebut, dengan cara mengidentifikasikannya sesuai dengan aturan yang ada disetiap protokol pembawanya [3] [4].
Tetapi proses sniffing ini mempunyai kelemahan bila ditempatkan di luar router. Kelemahannya ini bergantung dari penghubung yang ada didalam jaringan tersebut. Penghubung dalam suatu jaringan bisa dalam bentuk HUB atau SWITCH HUB, yang mempunyai karakteristik masing-masing dalam menyalurkan data nya Berikut dijelaskan karakteristik masing-masing penghubung.

1.3 Pemblokiran IP

Pemblokiran IP tersebut disesuaikan dengan *operating system* yang ada di *router*, apakah Linux, Windows 2000, ataukah FreeBSD

1. WINDOWS 2000
   Di dalam *windows 2000 server* telah dilengkapi dengan cara untuk mengatur IP baik itu mengeblok IP maupun melewatkan suatu IP. Program tersebut adalah IPSECPOL. Utility ini hampir sama kegunaannya pada iptables dan ipchains dalam program LINUX. Hanya saja untuk utility ini hanya bekerja pada *windows 2000 server* [5].
   Pengaturan IPSECPOL pada tampilan windows dapat dijumpai pada "IP *Security Policies on Local Machine*" yang berada pada "*Computer Configuration*

Security Settings" di MMC *(Microsoft Management Console)*. Yang pada defaultnya terdapat 3 ketentuan yang telah ditetapkan dapat dilihat pada gambar 2 :

a. Client (Respond Only)

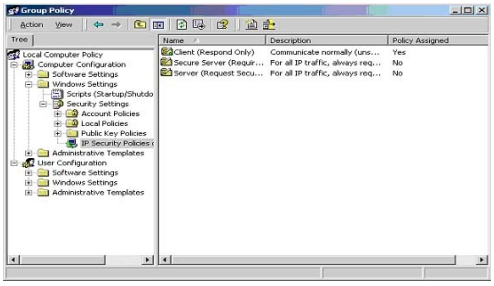
Digunakan oleh client untuk memberikan respon kepada *windows 2000 server* pada saat ada permintaan menggunakan servis yang ada didalamnya.

b. Secure Server (Require Security)

Ketentuan ini digunakan pada *windows 2000 server* dan *windows 2000 host* yang menghasilkan *network-based services* untuk meyakin kan bawa tidak ada *non-authentication* dan *non-encryption traffic* yang di abaikan.

c. Server (Request Security)

Ketentuan ini hampir sama dengan ketentuan yang ada pada *Secure Server*, yang menjadi perbedaan adalah pada ketentuan ini terdapat ketentuan untuk mengadakan hubungan enkripsi pada tingkat lebih tinggi di *user*[6][7].



Gambar 2. *IPSECPOL Pada Tampilan Windows*

2. Linux

Untuk sistem pemblokiran dengan menggunakan operating system ini dengan menggunakan aplikasi yang sudah tersedia yaitu dengan menggunakan IPTABLES atau IPCHAINS tergantung versi yang digunakan. Pada aplikasi ini tersedia berbagai fungsi tentang *routing* baik *forwarding*, *accepting* ataupun *bloking* [8][9].

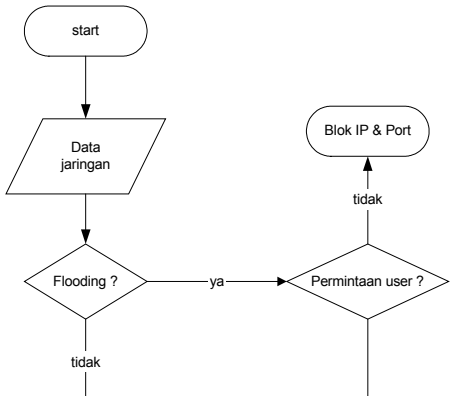
3. FreeBSD

FreeBSD juga mempunyai aplikasi untuk pengaturan routing yang fungsinya mirip dengan IPTABLES pada linux ataupun IPSECPOL pada windows, hanya saja pada sistem FreeBSD untuk pengaturannya menggunakan perintah IPFW [9].

2. Pembahasan

2.1 Rancangan Sistem Secara Umum

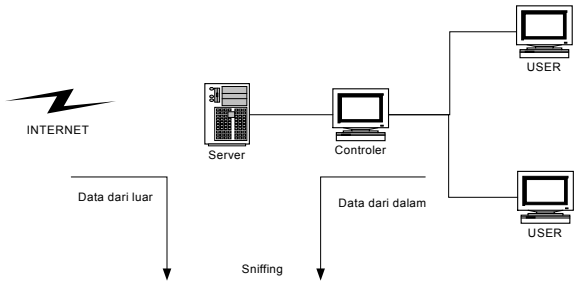
Secara umum sistem yang akan dibangun adalah sebagai berikut :



Gambar 3 . *Desain Umum Program Blokir Otomatis Pada Flood*

2.2 Rancangan Pengambilan Data

Apabila menggunakan Windows 2000 sebagai router tentunya hal tersebut akan tidak bisa dilakukan secara langsung. Karena harus mengambil data dari paket tersebut secara detail, walaupun yang di ambil hanya sebatas header dari data [2]. Bukan keseluruhan dari paket tersebut untuk menjaga privasi user dari server. Dengan demikian perlu menempatkan *sniffer* untuk memperoleh *header* dari data itu. Seperti yang tergambar sebagai berikut



Gambar 4. *Proses Pengambilan Data*

Dari gambar 4 Proses Pengambilan Data tersebut bisa dijelaskan data yang akan masuk ataupun akan keluar di belokkan terlebih dahulu untuk diambil datanya sebelum dilanjutkan ketujuan sebenarnya. Didalam pembelokan ini tidak berarti bahwa data paket ditahan dulu untuk di teliti melainkan data hanya yang datang maupun keluar di-capture headernya[10].

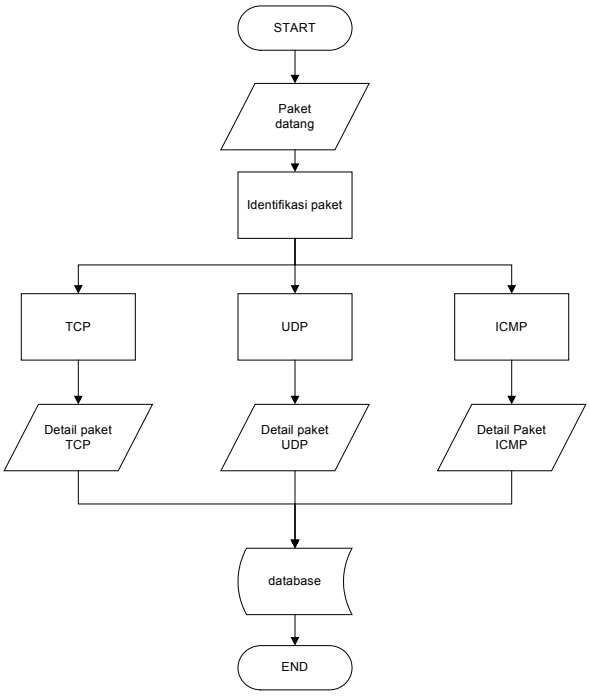
2.3 Rancangan Pengidentikasian Data

Menurut standar protokol yang ada sekarang ini hampir semuanya menggunakan struktur data Ethernet II sebagai struktur untuk mengirimkan atau menerima data. Struktur data dalam Ethernet II itu sendiri disebut frame. Format frame Ethernet II adalah sebagai berikut

Preamble (8 octets)	Destination Address (6	Source Address (6 octets)	Type (2 octets)	Data (46-**1500 octets**)	FCS (3 octets)

Gambar 5. *Header IP*

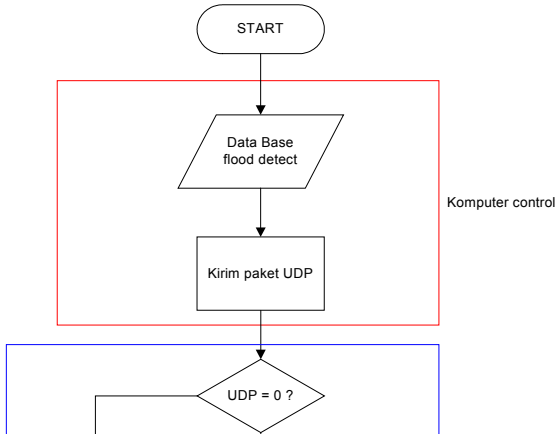
Pada gambar 5 pembukaannya mengandung serangkaian 8 bit dengan pola tertentu yang memberitahu node penerimaan bahwa setiap bahwa suatu frame dimulai. Untuk alamat tujuan dan sumber masing-masing 48 bit (6 oktet). Field type menunjukkan jenis data dari field data, dengan besar 16 bit (2 oktet).
Dengan mengetahui header dari setiap paket yang masuk dapat kita peroleh data-data dari paket, yang kemudian bisa mengklasifikasikan setiap data yang datang apakah itu paket TCP, UDP atau juga ICMP pada gambar 6 Pengidentifikasian Data Paket menunjukan proses klasifikasi paket data Beserta semua keterangan dari mana paket itu berasal, kemana tujuannya, juga besar dari paket tersebut.



Gambar 6. *Pengidentifikasian Data Paket*

2.4 Rancangan Pemblokiran IP

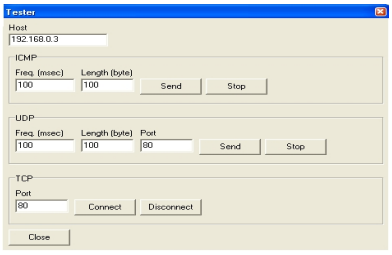
Dapat dilihat pada gambar 7 Desain blokir IP, yaitu setelah data terbukti melakukan flooding pada jaringan maka sistem akan mengirimpkan paket UDP ke server untuk mengirimkan perintah blooking kepada IP yang besangkutan. Sebelumnya program daemon sudah diletakkan didalam server terlebih dahulu dan dijalankannya,untuk program daemon akan ditanyakan pakah paket UDP sama dengan nol, jika sama maka data akan ditujukan ketujuanya,sebaliknya jika tidak maka data akan diblok.



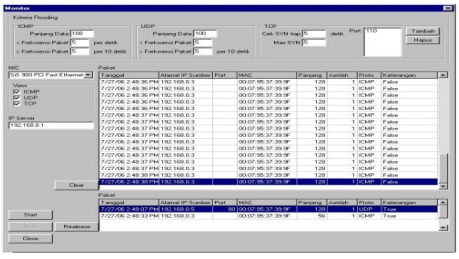
Gambar 7. *Desain blokir IP*

2.5 Pengujian Sistem

Sistem yang dibangun mempunyai kemampuan melihat semua paket yang datang dalam bentuk apapun. Meskipun demikian sistem hanya mengambil paket-paket dari tiga protokol utama yang biasa digunakan untuk mentransfer data. Protokol itu adalah TCP, UDP dan ICMP. Hal ini disebabkan karena *flood* yang biasa terjadi dalam jaringan dilakukan melalui tiga protokol tersebut. Sedangkan protokol lain hampir tidak pernah mengalami *data flooding proses* pengiriman data terlihat pada gambar 8 dan proses pengambilan data terlihat pada gambar 9.



Gambar 8. *Pengiriman Data*



Gambar 9. *Pengambilan Data*

2.6 Kemampuan Sistem Dalam Mengolah Data

Pengolahan data dari setiap paket datang yang masuk ditujukan untuk mengoptimasikan kerja komputer agar tidak terjadi komputer mengalami crash atau hang. Karena sistem mengolah data-data yang tidak perlu.
Pengolahan data tersebut meliputi:
a. Pemisahan paket data

Pada pengolahan ini sistem akan memisahkan data-data mana yang perlu ditampilkan pada tabel dan data-data mana yang tidak perlu ditampilkan pada tabel. Untuk paket UDP dan ICMP data-data yang tidak masuk kategori flood dihapus langsung. Untuk data yang menggunakan TCP sebagai protokol, data yang di tampilkan hanya data yang merupakan paket SYN saja, data acknowledge tidak ditampilkan.

b. Pengidentifikasian *data flood*
Sistem mampu mengidentifikasikan apakah data yang datang itu *flood* atau tidak. Secara visual data yang dikategorikan *flood* akan masuk lansung dalam tabel blacklist untuk di blok IP pengirimnya.

c. Pengiriman data UDP secara otomatis untuk blokir IP
Apabila setelah teridentifikasi *flood* maka control akan mengirimkan satu paket data melalui protokol UDP. Pengiriman paket UDP ini digunakan untuk memerintahkan *server* melakukan pengeblokan IP *sever* yang melakukan *flooding* terhadap jariangan.

2.7 Pengujian Ketahanan Sistem pada Flooding Data

a. Pengujian dengan protokol ICMP
Kondisi awal :
1. Panjang data maksimal = 100 byte
2. Frekuensi paket besar (lebih besar dari 100 byte) maksimal = 5 / 10 s
3. Frekuensi paket kecil (lebih kecil dari 100 byte) maksimal = 5 / 1 s

Tabel 1 dan 2 Pengujian *Flooding ICMP* Paket Besar dan Paket Kecil

Tabel 1. Paket Besar

Periode (ms)	Besar paket (byte)	Pemblokiran
1000	2500	Ya
500	2500	Ya
100	2500	Ya
50	2500	Ya
10	2500	Ya
1	2500	Ya

Tabel 2. Paket Kecil

Periode (ms)	Besar paket (byte)	Pemblokiran
1000	80	Tidak
500	80	Tidak
100	80	Ya
50	80	Ya
10	80	Ya
1	80	Ya

Dengan melihat data yang ada diatas maka semua data-data yang melewati batas yang telah ditentukan atau melewati batas ketentuan flooding akan dilakukan blooking pada IP-nya. Sedang data paket yang tidak melewati ketentuan akan diteruskan, seperti yang terlihat pada pengujian di data paket kecil yang mempunyai periode 1000 dan 500 ms

2.8 Pengujian dengan protokol UDP

Kondisi awal :
1. Panjang data maksimal = 100 byte
2. Frekuensi paket besar (lebih besar dari 100 byte) maksimal = 5 / 10 s
3. Frekuensi paket kecil (lebih kecil dari 100 byte) maksimal = 5 / 1 s

Tabel 3 dan4 Pengujian *Flooding UDP* Untuk Paket Besar dan Paket Kecil

Tabel 3. Paket Besar

Periode (ms)	Besar paket (byte)	Pemblokiran
1000	2500	Ya
500	2500	Ya
100	2500	Ya
50	2500	Ya
10	2500	Ya
1	2500	Ya

Tabel 4. Paket Kecil

Periode (ms)	Besar paket (byte)	Pemblokiran
1000	80	Tidak
500	80	Tidak
100	80	Ya
50	80	Ya
10	80	Ya
1	80	Ya

Data-data UDP yang datang apabila melewati batas ketentuan akan di blok sedang yang tidak melwati akan diteruskan

2.9 Pengujian dengan protokol TCP

Kondisi awal :
1. Pengecekan setiap = 10 s
2. Banyak TCP SYN maksimal (dalam satu kali pengecekan) = 5 s

Port yang diperbolehkan : 8080, 3128, 80
Tabel 5 dan 6 Pengujian *flooding TCP*

Tabel 5. Pengiriman Dalam Port yang Diperbolehkan

Port	PeriodeTCP SYN	Pemblokiran
8080	2000	Tidak
8080	1000	Ya
8080	500	Ya
8080	100	Ya
8080	10	Ya
8080	1	Ya

Tabel 6. Pengiriman Dalam Port yang Tidak Diperbolehkan

Port	PeriodeTCP SYN	Pemblokiran
5000	2000	Ya
5000	1000	Ya
5000	500	Ya
5000	100	Ya
5000	10	Ya
5000	1	Ya

Pengiriman data TCP melalui port yang tidak di perbolehkan langsung di blok sedangkan apabila melalui port yang diperbolehkan maka dilakukan pemerikasaan

apakah banyak TCP SYN yang datang melebihi ketentuan atau tidak. Jika ternyata banyak data paket datang melebihi ketentuan akan dilakukan pengeblokan IP.

3. Kesimpulan

Dari hasil penelitian, analisis, perancangan, pembuatan hingga pengujian, dapat disimpulkan sebagai berikut

1. Sistem dapat mendeteksi flooding data yang keluar masuk sehingga semua data bisa dilihat apakah data itu merupakan flooding atau bukan.
2. Sistem dapat melakukan pemisahan data sehingga data tersebut dapat diketahui bersifat *flooding* atau tidak.
3. Sistem dapat bekerja meskipun data yang dikirimkan memiliki *flood* yang besar karena pembatasan paket data yang masuk merupakan variabel yang bisa diubah besar kecilnya maka berapapun besar flood yang masuk dapat di deteksi dan diatasi.
4. Selain itu pengolahan data bukan semua data yang ada melainkan data-data yang sudah sangat terseleksi.
5. Sistem dapat menjamin keamanan data karena konsep sistem melakukan pencegahan flooding sejak dini.

Daftar Pustaka

[1] S.N.M.P. Simamora, Y. S. Pratiwi, A. Sularsa, "Implementasi Perangkat Lunak Simulator Penanganan Man-In-The-Middle-Attack menggunakan Interlock Protocol", The Conference on Information Technology and Electrical Engineering (CITEE) 2011, Department of Electrical Engineering and Information Technology, Gajah Mada University, in Proc. ISSN: 2085-6350, pp.240-252 2011.

[2] Sopandi. Dede, Instalasi dan Konfigurasi Jaringan Komputer ,Bandung : Informatika, pp. 54-67, 2010.

[3] Sutomo. Erwin,.Jaringan Komputer dan Pengamanannya, Surabaya : STIKOM Surabaya, pp. 105-111, 2010.

[4] Wahana Komputer,. Penanganan Jaringan Komputer. Yogyakarta, pp. 21-32, 2001.

[5] Harianto. Bambang,. Sistem Operasi revisi keempat, Bandung : Informatika, pp. 67-69, 2006.

[6] Riadi. Imam, Optimalisasi Keamanan Jaringan Menggunakan Pemfilteran Aplikasi Berbasis Mikrotik, JUSI Vol.1, Program Studi Sistem Informasi, Yogyakarta : Universitas Ahmad Dahlan, pp. 201-211, 2011.

[7] Sukmaaji. Anjik. Rianto. Jaringan Komputer :Konsep Dasar Pengembangan Jaringan Dan Keamanan Jaringan, Yogyakarta : Andi, pp. 90-98, 2008

[8] Mancill. T, (). Linux Routers : A Primer for Network Administrator : Prentice Hall, 2 ed., pp. 244-246, 2002.

[9] Purbo. O. W., Linux Untuk Warung Internet, Jakarta: Elex Media Komputindo, pp. 25-31, 2000.

[10] Setiawan. Dkk, Modul Pratikum Jaringan Komputer II,Bandung, pp. 20-24, 2011

Biodata Penulis

Budi Triandi, M.Kom **,**memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK Potensi Utama, lulus tahun 2007. Memperoleh gelar Magister Komputer (M.Kom) Program Pasca Sarjana Magister Teknik Informatika Universitas Putra Indonesia Padang, lulus tahun 2009 .Saat ini menjadi Dosen di Universitas Potensi Utama.