
TUGASKULIAH

KEAMANAN SISTEM LANJUT (EC7010)

Tinjauan ICMP TRACEBACK Sebagai Teknik Mencegah Serangan Distributed Denial of Service (DDoS) pada Jaringan Komputer

Muhammad Nasrun

232 02 017

Departemen Teknik Elektro
Fakultas Teknik Industri
Institut Teknologi Bandung



DAFTAR ISI

DAFTAR ISI	2
ABSTRAK	3
DEFINISI ISTILAH	4
I PENDAHULUAN	5
II TINJAUAN UMUM DoS dan DDoS	7
2.1 Klasifikasi Umum Serangan (attack)	7
2.2 Defenisi DoS dan DDoS	7
2.3 Klasifikasi Serangan DoS	9
2.4 Uraian Umum Serangan DoS	10
2.4.1 IP Spoofing	10
2.5 Uraian spesifik Serangan DoS	11
2.5.1 Smurf : ICMP flood	11
2.5.2 Trinoo: UDP flood	12
2.5.3 SYN Flooding	13
2.5.4 Stealth Bomb	14
2.6 Karakteristik Serangan DDoS	14
2.7 Attack Tools	16
2.8 Beberapa Teknik Bertahan dari Serangan DDoS	18
III ICMP TRACEBACK	20
3.1 Format Message	20
3.1.1 Elemen Forward dan Backward Link	22
3.1.1.1 Back Link (TAG=0x01)	23
3.1.1.2 Forward Link (TAG=0x02)	23
3.1.2 Interface Identifier (TAG=0x03)	23
3.1.3 IPv4 Address Pair	24
3.1.4 IPv6 Address Pair	24
3.1.5 Mac Addresss Pair (TAG=0x06)	24
3.1.6 Operator -defined Link Identifier (TAG=0x07)	24
3.1.7 Timestamp (TAG=0x08)	25
3.1.8 Traced Pcked Packet (TAG=0x09)	25
3.1.9 Probability (TAG=0x0A)	25
3.1.10 RouterId (Tag=0x0B)	25
3.1.11 Authentication Data	26
3.1.11.1 HMAC Authentication Data (TAG=0x0C)	26
3.1.12 Key Disclosure List (TAG=0x0D)	27
3.1.13 Key Disclosure (TAG=0x0E)	27
3.1.14 Public-key Information (TAG=0x0F)	28
3.2 Prosedur Kerja	29
3.2.1 Membangun ICMP Traceback	29
3.2.1.1 Kebutuhan Implementasi dalam Membangun Message	29
3.2.1.2 Kebutuhan Implementasi dalam Penerimaan Message.. ..	30
3.2.2 Konfigurasi	30
3.2.3 Proses Penerimaan Message	30
IV ANALISA ICMP TRACEBACK	31
V KESIMPULAN	33
DAFTAR PUSTAKA	34

DEFENISI ISTILAH

1. Elemen : Komponen dari message yang secara eksplisit diidentifikasi oleh tag, diencoded menggunakan dalam format Tag-Length-Value (TLV). Beberapa elemen terdiri atas elemen lain.
2. Field : Komponen message yang diidentifikasi lewat posisinya dalam header atau dalam elemen tertentu
3. Generator : Router yang menghasilkan ICMP Traceback Message atau yang mewakili membangun message oleh entity lain
4. Link : Logika koneksi antara Generator dan entity lain selama traced packet dilewatkan.
5. Peer : Entity lain di akhir link, dimana dapat mengirimkan traced packet ke Generator atau menerinya dari Generator. (entity ini masih router, atau edge router trace packet yang diterima dari atau mengirim ke host)
6. Traced packed : Paket yang dibahas dalam ICMP Traceback message.

ABSTRAK

Teknik untuk mencegah serangan yang bersifat Distributen Denial of Service (DDoS) sangat dibutuhkan sekarang. Karena serangan ini telah menghancurkan beberapa website terkenal seperti Ebay, Amazon, Buy.com, CNN.com, dan Yahoo.com, sehingga mengakibatkan kerugian yang cukup besar. Teknik yang ada diperlukan suatu pembahasan, agar dapat diketahui kemampuan dalam mencegah serangan DDoS. Pada bahasan ini kita membahas ICMP Traceback dalam mengatasi serangan DDoS pada jaringan. Diharapkan dengan mengetahui prinsip kerja ICMP Traceback dapat membuka wawasan baru tentang cara mencegah serangan yang bersifat DDos.

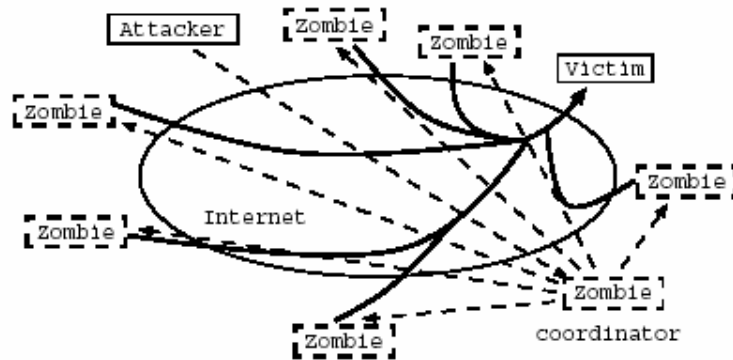
Keyword : Denial of Service (DoS), Distributed Denial of Service (DDoS)

BAB 1

PENDAHULUAN

Perkembangan teknologi jaringan internet di era sekarang ini semakin pesat. Feature-feature *service* (layanan) yang disediakan dalam jaringan internet juga begitu banyak ragamnya. Mulai dari web server, *File Transfer Protocol (ftp)*, layanan E-mail, sampai feature-feature yang berhubungan dengan layanan transaksi yang semakin marak di dalam jaringan internet. Layanan tersebut seperti *Electronic Commerce (E-Commerce)*, *Electronic Banking (E-Banking)*, *Electronic Government (E-Gov)* dan sebagainya. Karena internet yang begitu banyak memberikan manfaat dan bersifat publik, maka dibutuhkan suatu sistem keamanan dalam menjaga informasi yang ada di internet supaya tidak dirusak oleh pihak-pihak yang potensial melakukan pengrusakan seperti *hacker* dan *cracker*.

Salah satu teknik pengrusakan yang dilakukan adalah dengan meniadakan ketersediaan informasi ketika dibutuhkan yang sering disebut *Denial of Service attack (DoS)*. Jenis serangan ini dilakukan dengan cara server dikirim permintaan (biasanya palsu) atau permintaan yang diluar perkiraan sehingga tidak dapat permintaan lain atau bahkan *down, hang, dan crash*. Apabila serangan DoS tersebut dilakukan secara terdistribusi maka jenis serangan tersebut disebut *Distributed Denial of Service attack (DDoS)*. Yakni *attacker* (penyerang) dapat men-*crack* beberapa mesin menjadi *zombie*, kemudian mesin yang menjadi *zombie* mengendalikan beberapa mesin lagi menjadi *zombie-zombie*, akhirnya *attacker* akan mengendalikan *zombie-zombie* tersebut secara terdistribusi menyerang korban (*victim*) untuk meniadakan ketersediaan informasi dari korban [1]. Gambaran tentang jenis serangan ini dapat dilihat pada gambar dibawah ini :



Gambar 1.1 Skema serangan DDoS

Beberapa website yang diserang dengan menggunakan prinsip DDoS adalah Ebay, Amazon, Buy.com [2], CNN.com, dan Yahoo.com.

Karena begitu hebatnya efek dari serangan DDoS ini bagi website-website tersebut sampai website-website tersebut mengalami kerugian yang cukup banyak untuk menanggulangi serangan ini. Sehingga dibutuhkan suatu teknik pencegahan untuk mendeteksi dan mencegah serangan DDoS ini. Pada bahasan ini akan dipaparkan suatu tinjauan teknik untuk mencegah dari serangan DDoS yakni ICMP TRACEBACK. Dalam bahasan ini tidak membahas bagaimana *Intrusion Detection System (IDS)* melakukan deteksi pada Serangan DDoS

BAB II

TINJAUAN UMUM DoS dan DDoS

2.1 Klasifikasi Umum Serangan (attack)

Sebelum membahas secara detail mekanisme menanggulangi DDoS attack, lebih dahulu membahas uraian tentang klasifikasi ancaman keamanan komputer secara umum.

Serangan pada *IT (Information Technologi)* menurut tujuan dari *cracker* dapat diklasifikasikan ke dalam [Bellavin, S.M., W.R. Cheswick. *Firewalls and Internet Security*. Addison Wesley Longmen, 1994] [3]:

- *Denial of service (DoS)* : sasaran utama serangan adalah merusak layanan yang disediakan, sehingga layanan menjadi tidak tersedia.
- *Intrusion* : pihak yang tidak berwenang berusaha memperoleh akses sistem, tujuan seperti ini adalah gambaran klasik dari seorang *hacker*. Mereka biasanya berusaha mencapai tujuannya keluar dari sistem dengan melakukan beberapa pengrusakan pada sistem, kemudian melaporkan ke *administrator* bahwa ada “*bug*” ditemukan dalam sistem.
- *Information Theft* [Pencurian informasi] : sasaran utama dari jenis serangan ini adalah berusaha mengakses informasi yang dibatasi pengaksesannya (*restricted*), informasi yang sensitif.
- *Modification* : disini *attacker* dengan aktif berusaha mengubah informasi yang ada dalam sistem. Jenis serangan macam ini semakin hari jumlahnya bertambah seperti mengubah isi website.[4]

2.2 Defenisi DoS dan DDoS

Ada beberapa definisi dari DoS, FAQ dari WWW. Security [5] mendefinisikan DoS sebagai berikut :

... suatu serangan yang dilakukan untuk membuat komputer atau jaringan komputer tidak dapat menyediakan layanan secara normal. Pada umumnya serangan DoS menargetkan serangan pada bandwidth jaringan komputer atau koneksi jaringan (*connectivity*). *Bandwidth attack* membanjiri jaringan dengan volume traffic yang tinggi, sehingga semua *resources* (sumber daya) yang ada, tidak dapat melayani *request* (permintaan) dari *legitimate user* (user yang sah). *Connectivity attack* membanjiri komputer dengan volume request koneksi yang

tinggi, sehingga semua *resources* sistem operasi komputer yang ada tidak dapat memproses lebih lama *request* dari *legitimate user*.

J.D. Howard mendefinisikan DoS sebagai [6]:

Apabila hardware, software, dan data komputer tidak dapat terjaga ketersediaannya, maka produktivitas operasional jadi turun, walaupun tidak ada kerusakan yang terjadi. Denial of Service dapat mencakup kedua keadaan tersebut yang secara disengaja maupun tidak disengaja melakukan serangan kepada ketersediaan sistem (*system availability*). Perspektif yang muncul tanpa melihat sebab yang terjadi adalah apabila layanan diibaratkan tersedia, padahal tidak ada, sehingga mengakibatkan layanan *denied* (tidak ada).

Suatu serangan, bagaimanapun, adalah suatu tindakan disengaja. Denial of Service attack diyakini berlangsung ketika mengakses ke komputer atau resource jaringan dengan sengaja di blocked atau hak aksesnya diturunkan dari user lain. Serangan ini tidak perlu merusak data secara langsung, atau permanen (walaupun mereka dapat melakukannya), tetapi mereka dengan sengaja berkompromi (mengganggu) ketersediaan dari *resource*.

Macam serangan DoS attack umumnya melalui jaringan, dimana target utama dari serangan adalah *website* yang populer seperti contoh yang disebutkan pada bab pendahuluan. Umumnya site-site tersebut mempunyai banyak hardware yang mereka gunakan, sehingga *attacker* akan bekerja keras untuk menyerang. Website normalnya terdiri dari beberapa web-server dengan sistem load balancing dan memiliki koneksi jaringan multi megabit.

Sebagai konsekuensi attacker harus menemukan jalur baru untuk menaklukkan sistem. Attacker tidak menggunakan satu host dalam penyerangan mereka, tetapi menggunakan beberapa ratus bahkan ribuan komputer untuk melakukan serangan yang terkoordinir. Jenis serangan seperti ini disebut *Distributed Denial of Service attack (DDoS attack)*. FAQ dari WWW. Security [5] pada *Distributed Denial of Service (DDoS) attack* mendefinisikan jenis serangan ini sebagai berikut :

Distributed Denial of Service (DDoS) attack menggunakan banyak komputer untuk mengkoordinir DoS attack ke satu atau lebih target korban. Penggunaan teknologi *client/server attacker* dapat mengaktifkan Denial of Service dengan memanfaatkan *resource* dari bagian komputer yang tanpa disadari bertindak sebagai *platform* serangan. Sehingga program master *DDoS* diinstall pada salah satu komputer dengan menggunakan *account* yang telah dicuri. Program master pada saat waktu yang ditentukan akan melakukan komunikasi ke sejumlah program "agent" yang akhirnya menginstall program tersebut dikomputer-komputer yang terhubung dengan internet. Ketika agent menerima command memulai serangan, penggunaan teknologi client/server program master dapat mengendalikan ratusan bahkan ribuan komputer yang memiliki program agent untuk memulai serangan dalam beberapa detik.

Aggregat bandwidth dari sejumlah besar program agent mungkin lebih besar dari kapasitas *uplink* dari website tersebut. Sehingga efek dari serangan ini lebih utama ke *IP Router* atau tidak dapat akses internet [7]

2.3 Klasifikasi Serangan DoS.

DoS attacks dapat dibagi menjadi tiga jenis [8] :

1. Pemanfaatan implementasi Bugs dari system.

DoS attacks dengan pemanfaatan implementasi bugs pada prinsipnya lebih mudah ditanggulangi dengan menginstall *patch* dari sistem

2. Serangan pada resource server.

Serangan pada resource server (memory, space disk, dan lainnya) lebih sulit ditanggulangi sebab attacker sering memanfaatkan kelemahan bagian-bagian "*legitimate protocol*" dibanding bug yang sederhana. Seperti serangan pada feature-feature aplikasi atau protokol yang berada diatas layer transport. Contoh serangan ini adalah nama server mail palsu yang membuat mail berulang-ulang secara eksponential atau mencreate sejumlah account samaran dalam waktu singkat [9]. Jenis serangan DoS ini biasanya ditanggulangi dengan memperbaiki aplikasi yang bermasalah, karena tidak mungkin lapisan bawah jaringan atau lapisan sistem dapat mendeteksi masalah ini.

3. Serangan pada bandwidth server.

Serangan pada bandwidth server dilakukan dengan membebani jaringan korban dengan traffic yang sia-sia. Bugs pada router jaringan korban dapat menyebabkan router crash, karena menemukan banyak masalah. Beberapa serangan dengan mudah dapat diidentifikasi, dengan memfilter atau membatasi paket karena dalam operasi normal paket-paket tersebut tidak pernah volumenya besar[10]. Kesulitan mengatasi *bandwidth attack* disebabkan oleh traffic yang kelihatannya normal pada volume besar [11]. Biasanya *bandwidth attack* membutuhkan sebuah group attacker untuk bekerja sama menghasilkan traffic yang cukup.

2.4 Uraian Umum Serangan DoS

2.4.1 IP Spoofing

IP spoofing adalah menggunakan IP address orang lain untuk melakukan penyerangan. Ketika menulis ke raw socket, program dapat mengisi header field dari suatu paket IP apapun yang diinginkan. Dalam system Linux, user yang melakukan proses ini memerlukan ijin dari root. Karena routing hanya berdasarkan *IP destination address* (tujuan), maka *IP source address* (alamat IP sumber) dapat diganti dengan alamat apa saja. Dalam beberapa kasus, *attacker* menggunakan satu *IP source address* yang spesifik pada semua paket IP yang keluar untuk membuat semua pengembalian paket IP dan *ICMP message* ke pemilik address tersebut. *Attacker* juga menggunakan *IP spoofing* untuk menyembunyikan lokasi mereka pada jaringan. Pada bahasan Smurf : ICMP Flood memperlihatkan serangan dengan menggunakan *IP spoofing* untuk membanjiri korban dengan ECHO REQUEST.

Filterisasi Ingress/Egress [12][13] yang ditempatkan pada *router* dapat mengeliminasi secara efektif *IP spoofing*. *Router* mencocokkan *IP source address* dari masing-masing paket keluar terhadap *IP address* yang ditetapkan. Jika *IP source address* ditemukan tidak cocok paket di *dropped* (dihilangkan). Sebagai contoh *Router* di MIT, rute paket keluar hanya dari *IP source address* dari subnet 18.0.0.0/8. Walaupun *IP spoofing* adalah suatu senjata bagi *attacker* untuk melakukan penyerangan, dalam banyak kasus penggunaan *IP spoofing* ini oleh *attacker* hanya sebatas dalam pemakaian sementara *IP address* tersebut. Banyak *attacker* dilibatkan dalam suatu serangan, masing-masing operasi dari jaringan yang berbeda, sehingga kebutuhan *IP spoofing* menjadi berkurang. Filterisasi Ingress dan Egress membuat seorang *attacker* lebih sulit melakukan serangan, walaupun cara ini belum menjamin mengatasi *IP spoofing*.

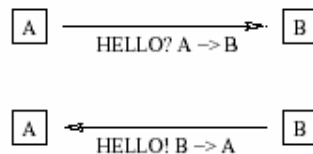
Stefan Savage dan teman-temannya [14] membuat suatu cara lain untuk mengatasi *IP Spoofing* yakni disebut *IP Traceback* [14] yang membantu dalam menemukan *attacker*. Teknik mereka membutuhkan *router* untuk menandai paket yang diterima kemudian merekonstruksi rute baru untuk diikuti paket tersebut, untuk membuat paket-paket dikirim. Teknik ini sangat menjanjikan, tetapi dengan teknik ini hanya membantu dalam menemukan *attacker*, belum mencegah *bandwidth attack*. *IP Traceback* pada dasarnya hanya efektif sebagai penghalang.

2.5 Uraian spesifik Serangan DoS

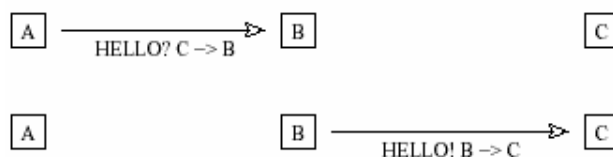
Uraian spesifik DoS attack dapat dijelaskan pada Smurf : ICMP Flood, Trinoo : UDP Flood, SYN Flooding, dan Stealth Bomb, dan uraian attack tools.

2.5.1 Smurf : ICMP flood

Salah satu mekanisme jenis serangan yang baru-baru ini mulai marak digunakan adalah menggunakan “ping” ke alamat *broadcast*, ini sering disebut smurf. Smurf attack [11] dilakukan dengan menggunakan ICMP (Internet Control Message Protocol). ICMP adalah protocol yang digunakan untuk mengontrol message yang dikirim. ECHO REQUEST dan ECHO REPLY adalah dua bagian dari message. Program “ping” sebagai contoh yang menggunakan ICMP untuk mengukur delay round-trip antara dua mesin. Gambar 2.1 dapat menerangkan proses ini. A mengirim message ECHO REQUEST ke B. B menjawab dengan message ECHO REPLY. Notasi “A → B” mengartikan IP source address dan IP destination address dalam paket IP yang membawa message ICMP. B mengetahui kemana mengirim ECHO REPLY dari melihat IP source address dalam paket IP yang tiba.



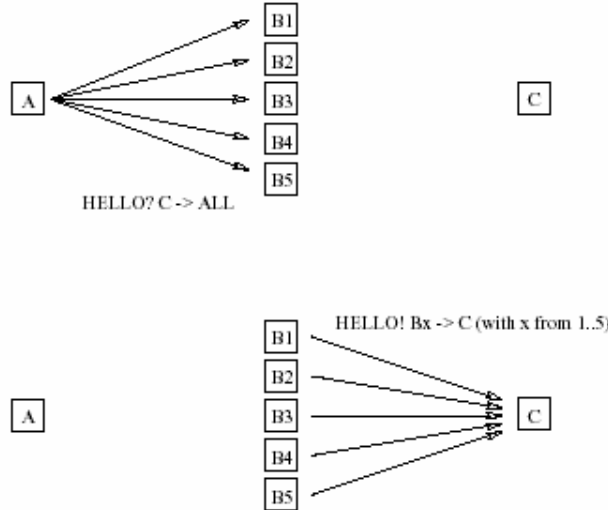
Gambar 2.1 ICMP tanpa IP Spoofing



Gambar 2.2. ICMP dengan IP Spoofing

Jika A spoof IP source address, situasi digambarkan pada gambar 2.2. A mengirim sebuah message ECHO REQUEST ke B, dengan spoof IP source address dengan C, bukan A. Sehingga C menerima ICMP ECHO REPLY dari B, nampaknya tidak diduga oleh C. Ini belum menjadi suatu ancaman ke C, sebab C dapat dengan mudah membuang message tersebut. Situasi ini mejadi berbahaya ketika A mengirim ECHO REQUEST ke *broadcast address*. Masing-masing mesin pada jaringan yang menerima mendapatkan ECHO REQUEST dan masing-masing mesin meresponnya.

Jika A spoof ke IP source address, suatu mesin yang tidak tahu akan menerima semua ECHO REPLY (seperti pada gambar 2.3 dibawah ini). konsekuensinya, link router ke C mungkin dapat tersumbat oleh semua traffic tersebut.



Gambar 2.3 “Smurf” attack

“Smurf” attack bekerja disebabkan bug pada banyak implementasi ICMP. Sebuah host seharusnya tidak boleh mengirim ECHO REPLY ke broadcast untuk merespon ECHO REQUEST. Disayangkan banyak implementasi ICMP gagal memeriksa IP destination address dari kedatangan ECHO REQUEST. IP spoofing adalah suatu kenyataan mutlak berhasilnya suatu serangan. Jika attacker gagal untuk spoof IP source address, banjir ECHO REPLY akan balik ke dirinya sendiri dengan kata lain melumpuhkan (menyerang) diri sendiri.

2.5.2 Trinoo: UDP flood

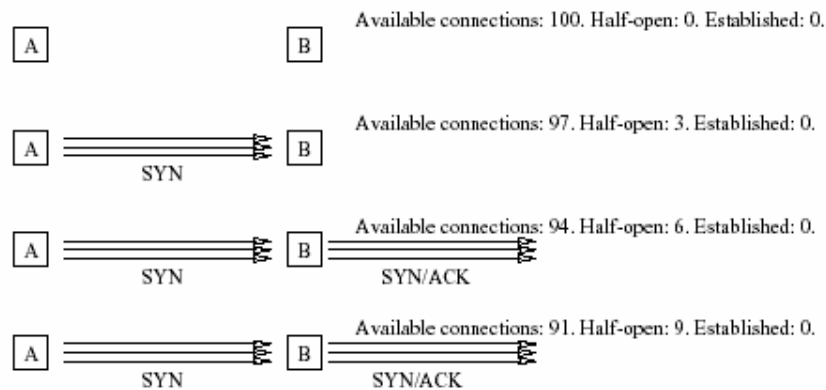
Trinoo attack adalah lebih canggih dari “Smurf” attack. Setelah mesin berkompromi –pembahasan tentang ini [15]– sebuah daemon kecil diinstall seterusnya menunggu perintah dari master : mesin yang berkompromi diubah menjadi zombie. Komunikasi antara master dan zombie-zombie sering di encrypt supaya mempersulit deteksi dari *network intrusion detector*. Suatu saat, master dapat menginstruksikan semua zombie untuk memulai mengirimkam paket UDP (User Datagram Protocol) ke satu tujuan.

2.5.3 SYN Flooding

Koneksi normal TCP memerlukan tiga paket yang akan dikirim antara client dan server, yakni :

1. Client mengirim paket SYN.
2. Server mengalokasikan TCP control block dan mengirim balik paket SYN/ACK.
3. Server menunggu ACK datang dari client.

Cara ini disebut 3-way handshake. Sepanjang ACK ada dari langkah 3 tidak dari server, koneksi adalah state half-open. Ketika tiada ada ACK datang dari client, koneksi dalam keadaan half-open sampai TCP time out setelah beberapa menit. Ketika TCP time out, alokasi control block menjadi tersedia lagi.



Gambar 2.4 SYN flood

Serangan sederhana (gambar 2.4) adalah A terus mengirim paket SYN dalam spoof paket IP. Paket SYN/ACK akan pergi ke pihak ketiga yang tidak tahu apa-apa (yang akan mendrop paket tersebut), yang membutuhkan paket ACK yang tidak dikirim oleh siapa pun. Kasus ini menyebabkan B kehabisan TCP control block dengan cepat dengan semua koneksi yang tersedia dalam state half-open. Suatu server tanpa tersedia TCP control block tidak akan mampu untuk menerima koneksi TCP. Beberapa solusi disarankan untuk mengatasi SYN flood dengan cara menurunkan waktu timeout TCP, meningkatkan jumlah TCP control block, SYN cookies dapat mengeliminasi kebutuhan penyimpanan informasi pada koneksi half-open, dan firewall khusus sebagai buffer paket SYN.

2.5.4 Stealth Bomb

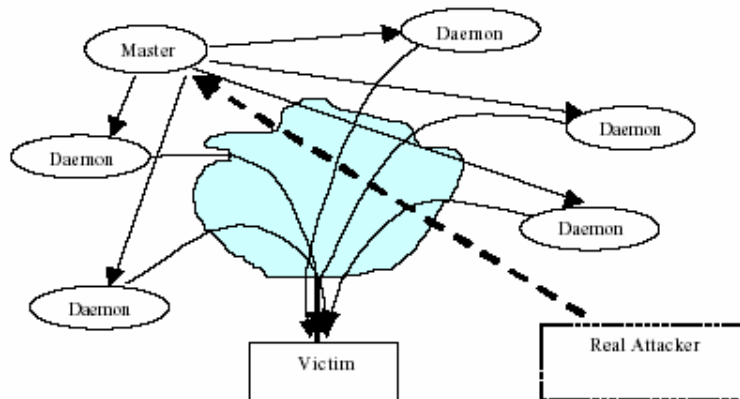
Generasi berikutnya bandwidth attack mungkin menghasilkan sejumlah besar traffic TCP normal. JavaScript yang running dalam browser yang memunculkan beberapa lusin window pada saat halaman web diambil dari server, maka berarti kematian bagi server jika beberapa ribu orang akan me-run script ini di browser mereka secara serempak [11]. Script seperti ini dapat dengan mudah tersebar lewat replikasi sendiri melalui virus e-mail. Kita dapat membedakan antara Stealth bom dan flashcrowd. *Flashcrowd* adalah kelompok client yang overload pada web server atau link-nya tertutup ke web server disebabkan terlalu banyak traffic. *Flashcrowd* adalah seperti SYN flood yang dapat menghabiskan semua TCP control block yang tersedia pada server. Perbedaannya dengan SYN flood, 3-way handshake disini diperlukan untuk kemajuan koneksi secara normal.

2.6. Karakteristik Serangan DDoS

DoS attack ditandai oleh usaha attacker untuk mencegah legitimate user dari penggunaan resource yang diinginkan. Cara DoS attack:

- a. Mencoba untuk membanjiri (flood) network, dengan demikian mencegah lalu lintas yang legitimate pada network.
- b. Mencoba mengganggu koneksi antara dua mesin, dengan demikian mencegah suatu akses layanan.
- c. Mencoba untuk mencegah individu tertentu dari mengakses layanan.
- d. Mencoba untuk mengganggu layanan sistem yang spesifik atau layanan itu sendiri.

Format terdistribusi membuat dmensi menjadi “many to one”, dimana jenis serangan ini lebih sulit untuk dicegah. DDoS adalah terdiri dari 4 elemen seperti gambar dibawah ini .



Gambar 2.5 Empat elemen DDoS attack.

Empat elemen tersebut adalah:

1. Korban (victim) yakni host yang dipilih untuk diserang.
2. Attack Daemon Agents, yakni program agen yang benar-benar melakukan serangan pada target korban. Serangan daemon biasanya menyebar ke computer-komputer host. Daemon ini mempengaruhi target dan komputer-komputer host. Manfaat serangan daem on ini dipergunakan attacker untuk untuk memperoleh akses dan menyusup ke komputer-komputer host.
3. Kendali Program Master, Yakni Tugasnya adalah untuk mengkoordinir serangan.
4. Attacker (penyerang), yakni penyerang riil, dalang di belakang serangan.

Dengan penggunaan kendali master program, penyerang riil dapat berdiri dibelakang layer dari serangan. Langkah-Langkah yang berikut berlangsung pada saat serangan terdistribusi:[16]

1. Penyerang riil mengirimkan suatu “execute” pesan kepada kendali master program.
2. Kendali master program menerima “execute” pesan kemudian menyebarkan perintah ini kepada attack daemons dibawah kendalinya.
3. Ketika menerima perintah penyerangan, attack daemons mulai menyerang korban (victim).

Walaupun tampaknya penyerang riil hanya melakukan sedikit pekerjaan disini, tetapi dengan meakukan pengiriman “execute” command, dia sebenarnya telah merencanakan pelaksanaan DDoS attacks. Attacker harus menyusup ke semua komputer host dan network dimana daemon attacker dapat disebar. Attacker harus mempelajari topologi jaringan target, kemudian melakukan pencarian bottlenecks dan kelemahan jaringan untuk dimanfaatkan selama serangan. Oleh karena penggunaan attack daemon dan kendali master program, penyerang real tidak secara langsung dilibatkan sepanjang serangan, dimana keadaan ini membuat dia sulit dilacak sebagai pembuat serangan.

2.7 Attack Tools

Beberapa attack tools yang digunakan dalam DDoS Attack.

1. Trinoo : menggunakan TCP untuk komunikasi antara attacker dan kendali master program. Master program berkomunikasi dengan attack daemon menggunakan paket UDP. Trinoo attack daemon menerapkan UDP Flood untuk menyerang korban.[17]
2. TFN (Tribe Flood Network): menggunakan suatu garis perintah untuk berkomunikasi antara attacker dan kendali master program. Komunikasi antara kendali master program dan attack daemon dilakukan lewat ICMP echo reply packet. Telnet atau SSH digunakan untuk mengirim kendali message ke kendali master program, dan paket ICMP ECHO REPLY digunakan untuk berkomunikasi antara kendali master program dan attack daemon, karena message ini tidak diblocked oleh firewall. TFN dapat menerapkan smurf attack, SYN Flood attack, UDP flooding attack, TCP dan ICMP flood attack.[17]
3. Stacheldraht (dalam istilah bahasa jerman “kawat-berduri”):teknik ini didasarkan serangan TFN. Tetapi tidak sama dengan TFN, stacheldraht menggunakan koneksi encrypsi TCP untuk komunikasi antara attacker dan kendali master program. Komunikasi antara kendali master program dan attack daemon dilakukan dengan menggunakan TCP dan ICMP, kemudian secara otomatis dapat mengupdate teknik attack daemon (serangan daemon). Attack daemon dalam stacheldraht dapat menerapkan Smurf, SYN Flood, UDP Flood, dan serangan ICMP Flood.[17]

4. TFN2K : menggunakan TCP, UDP, ICMP, atau ketiganya secara acak untuk berkomunikasi antara kendali master program dengan attack daemon. Komunikasi antara penyerang riil dan kendali master dienkripsi menggunakan algoritma key-based CAST-256. Sebagai tambahan TFN2K melakukan latihan rahasia untuk menyembunyikan dirinya sendiri dari deteksi sistem DS (Intrusion Detection System), sehingga sulit dideteksi oleh sistem IDS tersebut. TFN2K dapat menerapkan serangan Smurf, SYN, UDP, dan serangan ICMP Flood. [18]
5. Shaft : salah satu model setelah Trinoo. Komunikasi antara kendali master program dengan attack daemon dicapai menggunakan paket UDP. Kendali master program dan attacker berkomunikasi lewat koneksi telnet TCP. Yang membedakan antara Trinoo, Shaft mampu mengendalikan switch master server dan port dalam real time, sehingga membuat pendeteksian sistem IDS sulit. [19]
6. Mstream: kependekan dari “multiple stream”, tools ini dapat membanjiri dengan sangat efisien point to point stream TCP ACK. Tools ini memiliki kendali yang terbatas, dan melakukan sistem random terhadap semua 32 bit dari sumber alamat IP dalam penyerangannya. Tools muncul pertama kali pada musim semi 2000.[20]
7. Omega : pertama kali muncul awal musim summer 2000, tools ini dapat melakukan TCP ACK flooding, UDP packet flooding, ICMP flooding, IGMP packet flooding, dan campuran keempatnya. Tools ini mirip dengan Shaft, yang menyediakan statistic pembanjiran yang dihasilkan. Omega melakukan sistem random semua 32 bit sumber alamat IP dalam penyerangan, dan memiliki fungsi chat untuk berkomunikasi antara attacker. [21]
8. Trinity : berhubungan erat dengan mutasi Entitee, yang melakukan pendekatan baru ke arah model DDoS. Prinsip kerjanya tidak bersandar pada sistem jaringan umum, tetapi mengambil keuntungan dari jaringan IRC (Internet Relay Chat) untuk handler-to-agent communication, kemudian membuat channel pada IRC sebagai “handler”. Disamping UDP, TCP SYN, TCP ACK, TCP NUL packet flood, tools ini juga menyerang dengan cara

TCP fragment Flood, TCP RST packet flood, TCP random flag packet flood, dan TCP established flood. Ketika penyerangan melakuka sistem random pada semua 32 bit sumber alamat IP.

9. myServer : berbeda jauh dengan Trinity yang sama-sama di buat pada musim summer 2000. myServer adalah suatu tools DDoS sederhana. Tools ini hanya bersandar pada program eksternal yang menyediakan prinsip denial of service (DoS).
10. Plague :tools ketiga dari generasi yang sama dari Trinity dan myServer. Plague ini (artinya : wabah) didesain oleh attacker dengan melakukan tinjauan ulang kepada tools yang sudah ada dan melakukan peningkatan kinerja dari beberapa tools yang ada. Tools ini dapat melakukan TCP ACK dan TCP SYN Flooding, dan mengklaim telah memperbaharui atau memperbaiki kinerja tools TCP ACK flood yang ada sebelumnya.
11. Analisis tentang DDoS tools ada pada [22] dan [23]

2.8. Beberapa Teknik Bertahan dari Serangan DDoS

Pendeteksian DDoS attack dapat diklasifikasikan *Signature-based detection* atau kejadian *anomaly*. *Signature-based detection* suatu metode mendeteksi DDoS attack lewat proses pencarian pola (signature) dari jaringan yang diamati, pencocokan signature serangan dilakukan lewat database. *Anomaly detection* adalah metode mendeteksi DDoS attack lewat membandingkan parameter (matematis) dari traffic jaringan yang diamati dengan traffic normal jaringan.[24]

Sebagian besar jurnal membahas teknik bertahan dari serangan DDoS attack sebagai berikut [25]:

1. ICMP traceback.
2. Probabilistic IP Traceback.
3. IP Traceback menggunakan pendekatan aljabar.
4. Phusback
5. Tunneling – IP Overlay
6. Mengontrol flooding
7. Ingress Filtering dan Egress Filtering
8. Honey pots

Beberapa ukuran yakni 1, 2, 3, 4,5, dan 6 tujuan utamanya adalah untuk merekonstruksi alur serangan lewat penelusuran sumber DDoS attack. Dengan cara tersebut diharapkan dapat menghentikan serangan DDOS.

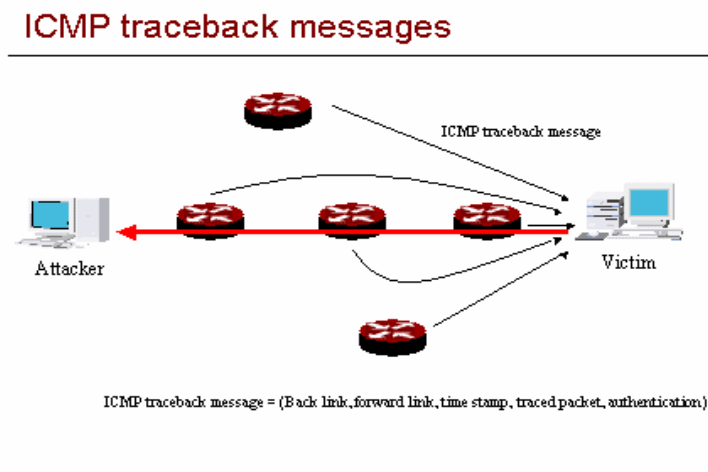
Dalam bab berikutnya akan membahas ICMP Traceback, sebagai suatu teknik mencegah serangan DDoS.

BAB III

ICMP TRACEBACK

ICMP Trace-Back Message sering disebut juga “*ITrace*”. Metode ini sangat berguna untuk mempelajari path dari paket yang melalui internet. Khususnya yang berhubungan dengan denial of servis attack, yang menggunakan IP Spoofing. Disamping itu dapat digunakan untuk mengetahui karakterisasi path dan deteksi asymmetric route. Ada beberapa tools yang dapat melakukan hal tersebut, seperti “tracert” tapi tools ini hanya menginformasikan forward path, tidak sebaliknya.

Permasalahan seperti itu dapat diselesaikan dengan ICMP Traceback message. Ketika menforward paket, router-router (dengan probabilitas rendah) dapat membuat Traceback message yang dikirim ke tujuan (destination). Dengan cukup Traceback message dari router-router sepanjang path, traffic source dan path dapat ditentukan. Gambaran dari ICMP Traceback dapat dilihat pada gambar 3.1 dibawah ini.

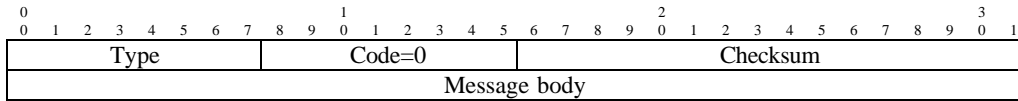


Gambar 3.1 ICMP Traceback

3.1 Format Message [26]

Format message adalah message yang dibawa paket ICMP, dengan ICMP TYPE dari TRACEBACK. Numeric value untuk setiap field diusulkan oleh IANA (Internet Assigned Number Authority). Untuk IPv6, TRACEBACK harus diklasifikasikan lewat

informasi tersebut. Field CODE harus selalu diset 0 (tidak ada code) dan harus selalu diabaikan (ignored) oleh receiver.



Body dari ICMP TRACEBACK message terdiri dari rangkaian elemen individu yang terdiri dari identifikasi masing-masing, dengan menggunakan TAG-LENGTH-VALUE dengan skema berikut:



Struktur ini adalah rekursif dimana setiap tipe elemen field VALUE akan berisi satu atau lebih komponen yang berada pada format TAG-LENGTH-VALUE (TLV). Top-level elemen boleh muncul dalam beberapa order dan receiver harus bisa memprosesnya dalam beberapa order. Elemen dalam field VALUE suatu parent element yang boleh muncul dalam beberapa order dalam field dan menyajikan elemen yang sama dibutuhkan oleh receiver. Elemen-elemen ditempatkan dengan teratur di dalam message body tanpa melihat lapisan, karena elemen-elemen secara umum tidak dibariskan dalam pembatas kata (word boundaries).

Field Tag adalah Single octet, dengan value berikut ;

Tag	Nama Elemen	Keterangan
0x01	Back Link	
0x02	Forward Link	
0x03	Interface Name	1
0x04	Ipv4 Address Pair	1,2
0x05	Ipv6 Address Pair	1,2
0x06	MAC Address Pair	1,3
0x07	Operator -defined Link Identifier	1,3
0x08	Timestamp	
0x09	Traced Packet Contents	
0x0A	Probability	
0x0B	RouterId	
0x0C	HMAC Authentication	
0x0D	Key Disclosure List	4
0x0E	Key Disclosure	4
0x0F	Public-Key Information	4

Keterangan :

1. Item ini adalah sub elemen dari elemen Back Link atau forward link.
2. Sekurang-kurangnya elemen ini harus disediakan oleh elemen Link.
3. Elemen MAC Address Pair atau Operator-defined Link Identifier tapi keduanya tidak harus disediakan oleh elemen Link.
4. Key Disclosure List harus berisi satu atau lebih elemen Key Disclosure dan pasti satu elemen Public-Key Information

3.1.1 Elemen Forward dan Backward Link.

Sebuah ICMP Trace back message harus berisi satu element Forward Link atau Satu elemen Back Link, karena kemungkinanakan terjadi suatu kejadian dari salah satu elemen tersebut. Sebuah elemen Link dispesifikasikan link sepanjang penelusuran paket (Traced Packet) yang datang ke korban atau dari pembuatnya (Generator). Tujuan dari elemen Forward dan Back Link adalah untuk membuat lebih mudah suatu kontruksi dari rantai Traceback message. Kedua elemen tersebut didesain untuk pengujian personil operasi jaringan, yang berisi informasi penting seperti *interface name*.

Value field dari elemen link terdiri dari tiga komponen :

- a. Interface name hanya ada di Generator. (asumsi ini bahwa Generator tidak mengetahui interface name tetangga).
- b. Alamat IP sumber dan tujuan dari Generator dan peer, di encode dalam elemen IPv4 address Pair atau IPv6 address Pair.
- c. Level link asosiasi string

Asosiasi string adalah suatu alat untuk menyatukan Traceback message dari pancaran router didekatnya. Dengan begitu semua elemen link mengacu pada link yang sama dan harus menggunakan value sama untuk asosiasi string, dengan mengabaikan entity yang membuat elemen-elemen link tersebut.

Pada LAN, asosiasi string dibangun oleh jalinan MAC address sumber dan tujuan dari dua interface yang melakukan link, semua proses ini diencode dalam elemen MAC address Pair. Jika tidak ada alamat seperti itu (pada point-to-point link) sebuah string yang pantas harus disediakan oleh kedua router, ini diencode pada elemen Operator Defined Link Identifier.

Field dari elemen Address Pair selalu diatur oleh “forward order” dari nilai traced packet. Oleh karena itu field “destination” selalu menunjukkan kedekatan alamat penerima paket traceback. Kemudian didalam elemen Back Link, alamat generator ditempatkan pada field destination dari sub elemen IP dan MAC Address Pair, didalam elemen Forward Link alamat generator ditempatkan dalam field sumber.

3.1.1.1 Back Link (TAG=0x01)

Elemen Back Link menyediakan identifikasi informasi, dari perpektif Generator, yakni link traced packet yang tiba dari. VALUE field dari elemen terdiri dari tiga Sub elemen TLV, masing-masing Interface Identifier, IP address Pair, dan asosiasi string. Panjang elemen yang ditunjukkan termasuk panjang tag dan field.

TAG=0x01	LENGTH (variable)
INTERFACE IDENTIFIER (variable length)	
IPv4 or IPv6 ADDRESS PAIR (11 or 35 octets)	
MAC ADDRESS (15 octets) or OPERATOR-DEFINED LINK IDENTIFIER (variable length)	
...	

3.1.1.2 Forward Link (TAG=0x02)

Elemen Forward Link menyediakan identifikasi informasi, dari perpektif Generator, yakni link traced packet yang telah diforward dari. Strukturnya sama dengan elemen Back Link.

TAG=0x02	LENGTH (variable)
INTERFACE IDENTIFIER (variable length)	
IPv4 or IPv6 ADDRESS PAIR (11 or 35 octets)	
MAC ADDRESS (15 octets) or OPERATOR-DEFINED LINK IDENTIFIER (variable length)	
...	

3.1.2 Interface Identifier (TAG=0x03)

Elemen ini berisi nama dari interface yang link ke router generator. Panjangnya diukur engan variable. Tipikal VALUE field berisi berisi string karakter yang dapat dibaca manusia.

TAG=0x03	LENGTH (variable)
INTERFACE IDENTIFIER (variable length)	
...	

3.1.3 IPv4 Address Pair

Elemen ini berisi dua 4 octet address IPv4 dari akhir link yang sesuai, karena LENGTH field-nya selalu 0x0008. Sebagai catatan, address harus selalu dimunculkan dalam permintaan traversal mereka oleh traced packet.

TAG=0x04	LENGTH=0x0008
UPSTREAM ADDRESS (4 octets)	
DOWNSTREAM ADDRESS (4 octets)	

3.1.4 IPv6 Address Pair

Elemen ini berisi dua 16 octet address IPv6 dari akhir link yang sesuai, karena LENGTH field-nya selalu 0x0020. Sebagai catatan, address harus selalu dimunculkan dalam permintaan traversal mereka oleh traced packet.

TAG=0x05	LENGTH=0x0020
UPSTREAM ADDRESS (16 octets)	
...	
DOWNSTREAM ADDRESS (16 octets)	
...	

3.1.5 Mac Address Pair (TAG=0x06)

Elemen ini berisi dua 6 octet IEEE MAC address dari akhir link yang sesuai, karena LENGTH field-nya selalu 0x0020. Sebagai catatan, address harus selalu dimunculkan dalam permintaan traversal mereka oleh traced packet.

TAG=0x06	LENGTH=0x0020
UPSTREAM ADDRESS (6 octets)	
DOWNSTREAM ADDRESS (6 octets)	

3.1.6 Operator-defined Link Identifier (TAG=0x07)

Value dari elemen ini adalah suatu field yang bermacam-macam length-nya. Jika peer juga memancarkan ICMP TRACEBACK message dari link yang sama, maka harus menggunakan value yang sama.

TAG=0x07	LENGTH (variable)
LINK IDENTIFIER (variable length)	
...	

3.1.7 Timestamp (TAG=0x08)

Elemen ini berisi waktu, dalam NTP (Network Time Protocol) format timestamp[27], dimana traces packed tiba dari Generator. Elemen ini harus dimunculkan pada level tertinggi pada TRACEBACK message.

TAG=0x08	LENGTH (variable)
INTEGER PART (4 octets)	
FRACTION PART (4 octets)	

3.1.8 Traced PckedPacket (TAG=0x09)

Elemen ini berisi traced packet, sedapat mungkin menunjuk ke link dan batasan sumber daya pada router. Elemen ini harus dimunculkan pada level tertinggi pada TRACEBACK message, harus berisi sekurang-kurangnya IP header dan 64 bit pertama dari body traced packet.

TAG=0x09	LENGTH (variable)
Complete Packet Header (>=24 octets)	
Packet body (>= 8 octets)	
...	

3.1.9 Probability (TAG=0x0A)

Elemen ini berisi invers probabilitas dari yang digunakan untuk memilih traced packet. Elemen nampak sebagai unsigned integer, dari satu, dua, atau empat octet. Elemen ini sebaiknya dimunculkan pada level tertinggi dari TRACEBACK message.

TAG=0x0A	LENGTH =0x0001/2/4
VALUE (1,2, or 4 octets)	

3.1.10 RouterId (Tag=0x0B)

Elemen ini berisi alat identifikasi informasi, berguna bagi organisasi yang mengoperasikan router pemancar ITRACE message. Elemen ini harus dimunculkan pada level tertinggi pada TRACEBACK message.

TAG=0x0B	LENGTH (variable)
ROUTER IDENTIFIER (variable length)	
...	

3.1.11 Authentication Data

Attacker mungkin mencoba membangun Trceback message palsu, terutama untuk menghilangkan sumber trafik serangan sebenarnya, tetapi ini juga bisa sebagai suatu cara bentuk serangan lain. Makanya kita membutuhkan suatu teknik autentikasi yang baik (robust) dan dapat memverifikasi.

Format autentikasi yang ideal adalah digital signature. Tapi itu tidak mungkin, walaupun router dapat melakukan signature pada semua paket Traceback. Yang dibuat sebagai penggantinya adalah kode hash (HMAC Authentication Data Element), yang mendukung pengungkapan bukti keabsahan yang banyak digunakan akhir-akhir ini (elemen Key Disclosure dan Public Key Information). Kunci terbaru tidak termasuk pada disclosure.

3.1.11.1 HMAC Authentication Data (TAG=0x0C)

Elemen ini harus ada, yang berisi empat sub field:

- a. Algoritma, satu octet : HMAC-MD5-128, HMAC-MD5-96, HMAC-SHA1-160, HMAC-SHA1-96, dan sebagainya. Code point harus diselidiki.. satu kandidat adalah terdiri dari kumpulan 16 bit Authentication Algorithm codepoints yang ditentukan oleh IANA dengan sekumpulan ISAKMP Codepoint. Lebih lanjut dapat dilihat pada [28]
- b. Keyid : delapan octet key identifier.
- c. Waktu Timestamp, dimana hash diambil, format NTP (8 octet).
- d. MAC data : Variabel.

Field MAC data meliputi seluruh IP datagram, termasuk header information. Dimana header information adalah dapat bermutasi selama pengangkutan, seperti informasi diset nol (0x00) untuk tujuan menghitung HMAC. Field ini baik sesuai dengan pemberian algoritma MAC.

TAG=0x0C	LENGTH (variable)
HMAC ALG (1?)	
TIMESTAMP (8 octets)	
MAC DATA (algorithm dependent)	

3.1.12 Key Disclosure List (TAG=0x0D)

Sebuah paket sebaiknya berisi sebuah list recently-used keys untuk algoritma hash. Ini disediakan oleh elemen Key Disclosure List. Elemen ini harus berisi sedikitnya satu sub elemen Key Disclosure, juga harus berisi sub elemen Public Key Information untuk menunjuk ke key yang digunakan untuk tanda Key Disclosure. Menurut aturan yang umum untuk membangun Traceback message, sub elemen mungkin dimunculkan di beberapa order, receiver harus dapat memprosesnya dengan mengabaikan dimana mereka dihadirkan.

TAG=0x0D	LENGTH (variable)
KEY DISCLOSURE (s) and PUBLIC KEY INFORMATION	
...	

3.1.13 Key Disclosure (TAG=0x0E)

Isi utama dari elemen Key Disclosure berisi key yang digunakan untuk mengautentikasi previous TRACEBACK message, dari starting, dan ending antara kunci yang digunakan. Algoritma diasumsikan sama seperti yang digunakan untuk mengautentikasi message baru (elemen yang ditunjukkan oleh HMAC Authentication). Elemen harus berisi digital signature mencakup elemen Key Disclosure.

Struktur elemen Key Disclosure sebagai berikut :

- a. Keyid untuk key yang di disclosed : delapan octet
- b. Validity : dua NTP timestamps memberikan periode validitas (start, dan end).
- c. Key length : satu octet
- d. Key material : variabel [key length] octet
Key material untuk memilih Fungsi HMAC, harus disesuaikan untuk kebutuhan kunci. (lebih lanjut dapat dilihat pada [29])
- e. Public key signature algorithm identifier, satu octet : PKCS1-RSA-MD5, PKCS1-RSA-SHA1, DSS-SHA1, X9.62-ECDSA-SHA1 ...(lebih lanjut dapat dilihat pada RFC2459 bagian 7, dan [PKALGS] : L. Bassham, R. Housley, W. Polk, "Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and CRL Profile", Internet Engineering Task Force, work in progress, untuk informasi algoritma signature).
- f. Signature length : dua octet. Sejumlah unsigned integer octet dari signature.

g. Signature : variable (siglengt) octet.

Signature meliputi seluruh elemen key disclosure, dikurangi signature fieldnya sendiri.

TAG=0x0E	LENGTH (variable)
KEY IDENTIFIER (8 octets)	
START TIME (8 octets)	
END TIME (8 octets)	
KEY LEN (1)	
KEY MATERIAL (KEY LEN octets)	
...	
SIG ALG (1?)	
SIGNATURE LENGTH (2 octets)	
SIGNATURE (SIG LEN octets)	
...	

3.1.14 Public-key Information (TAG=0x0F)

Digital signature adalah sia-sia (tidak berguna) tanpa beberapa cara autentikasi public key dari signer. Bentuk ideal dari autentikasi berdasarkan skema root sertifikasi (certificate-based scheme rooted) dalam address registries.. registries adalah sumber informasi yang mempunyai authority (wewenang) dari address yang memilikinya, karena mereka yang dapat melakukan sertifikasi.

Public-key Information (PKI) bisa ada, kita anjurkan masing-masing ISP (Internet Service Provider) memiliki root public key sendiri. Database registry-based baru dapat digunakan untuk memverifikasi owner (pemilik) dari address block, informasi ini dapat kembali digunakan untuk menempatkan root key.

Elemen Public-key Information dapat digunakan untuk menemukan public key yang sesuai, dan informasi lain yang berhubungan. Elemen ini brisi URL, menunjuk halaman XML yang berisi public key yang digunakan untuk menandai elemen key disclosure.

TAG=0x0ZF	LENGTH (variable)
URL (variable)	

3.2 Prosedur Kerja

3.2.1 Membangun ICMP Traceback

Implementasi router yang diangun untuk skema ini yakni probabilitas paket ICMP Traceback yang dipancarkan kira-kira $1/20000$, penerapan untuk link local disarankan menggunakan nilai ini dengan mengatur penggunaan metric link local.

Beberapa yang dibutuhkan dibebankan kepada IP Header dari Traceback message. Khususnya source address sebaiknya dihubungkan dengan interface dimana paket tiba. Jika interface memiliki multiple address, address harus dipilih, sehingga diharapkan satu diantaranya router ini mengetahui previous hop. Jika interface tidak memiliki IP address, “primary” IP address dihubungkan dengan router yang digunakan. (“primary “ IP address dibahas dibawah ini).

Inisial field TTL harus diset 255. Jika paket Traceback mengikuti path yang sama dengan paket data, ini dapat mengindikasikan dengan baik jarak dari router dengan tujuan. Yang lebih penting lagi, membandingkan jarak tersebut dengan elemen-elemen link, sehingga suatu chain (rantai) dapat dibentuk, dan secara parsial dapat diverifikasi tanpa menguji field autentikasi.

3.2.1.1 Kebutuhan Implementasi dalam Membangun Message (iTrace)

Probabilitas dari Traceback yang dibangun harus dapat diatur (disetel) oleh operator router. Nilai defaultnya disarankan $1/20000$. Jika rata-rata maksimum diameter dari internet 20 hops, ini menyebabkan trafik akan bertambah ke tujuan kira-kira 0.1%, sehingga probabilitas tidak boleh lebih besar dari $1/1000$.

Pemilihan Paket harus berdasarkan pseudo-random number (nomor acak) dibanding counter sederhana. Cara ini memungkinkan membantu memblokir serangan time burst. Serangan seperti ini tidak akan muncul jika secara cryptography menggunakan pseudo-random number.

Disarankan melakukan pengujian low-order bit dari linear congruential pseudo-random number generator (LCPRNG). Jika semuanya diset 1, paket dapat dipancarkan. Hal ini memungkinkan melakukan pemilihan probabilitas yang digunakan seperti $1/8191$, $1/16383$, dan sebagainya. Selama periode generator maksimal, semua value, termasuk semua nilai 1 dalam low order bit, akan memunculkan probabilitas yang sesuai.

Walaupun pembahasan ini menggambarkan implementasi Traceback message berdasarkan router. Kebanyakan fungsi ini dapat diimplementasikan lewat outboard device (device luar).. sebagai contoh, komputer laptop dapat digunakan untuk memonitor LAN, memancarkan traceback message yang sesuai mewakili semua router pada LAN.

3.2.1.2 Kebutuhan Implementasi dalam Penerimaan Message (iTrace)

Host-host harus didesain sehingga operator dapat melakukan “enable” dan “disable” pengumpulan dan penyimpanan ICMP Traceback yang dibutuhkan. Ini dimaksudkan jika hanya ada serangan sistem kerja ini dapat dilakukan, sehingga jika tidak ada serangan kerja router dapat difungsikan secara normal umumnya.

3.2.2. Konfigurasi

Asosiasi string digunakan pada elemen Forward dan Back link dapat dibangun dari MAC Address dari link endpoint. Jika tidak ada address seperti itu (yang digunakan untuk point-to-point link), string yang sesuai harus ditetapkan di kedua router, yang digunakan nantinya pada Operator-Defined Link Identifier.

3.2.3 Proses Penerimaan Message

Serangan yang rumit menggunakan pancaran ICMP Traceback message pada, message seperti ini harus divalidasi dulu sebelum digunakan. Beberapa cara validasi dilakukan sebelum HMAC keying Information di disclosed. Khususnya, ketika message muncul yang dihubungkan kepada segmen rantai (chain) yang telah diidentifikasi, penerima harus menggunakan field TTL yang berbeda yang dihubungkan dengan elemen-elemen link, untuk memverifikasi rantai (chain).

Karena HMAC key disclosure selesai hanya ketika akhir dari validasi untuk key. Autentikasi dari sekumpulan Traceback message membutuhkan pengumpulan message lebih lanjut, kemudian diuji di luar dari periode yang ada sampai key yang dibutuhkan muncul. Proses entity harus dilakukan untuk memverifikasi signature dari key sebelum mengambil key itu sendiri untuk memvalidasi message.

BAB IV

ANALISA ICMP TRACEBACK

Dengan rendahnya probabilitas ($1/20K$, tidak lebih dari $1/1K$), router secara acak menghasilkan message ICMP baru yang berisi message Traceback, kemudian mengirim message ke tujuan selama paket di forward. Jika message traceback cukup dikumpulkan korban, traffic asal dapat diidentifikasi oleh konstruksi suatu rantai (chain) message Traceback

Tetapi ICMP Traceback ini belum dapat mengatasi semua serangan denial of service (DoS). Karena ada pembatasan, dimana trafik yang datang harus relative kecil dari sejumlah sumber. Malah kadang sumber berasal dari jaringan sendiri sehingga menimbulkan kerusakan pada jaringan sendiri. Prinsip serangan adalah berdasarkan pada mempengaruhi host lain (“daemon” atau “zombie”) yang tidak dirusak untuk mengirim trafik ke korban, sehingga yang sering dapat ditelusuri hanya sampai pada daemon., tidak sampai pada penyerang riil.

Dengan rendahnya probabilitas ($1/20K$, tidak lebih dari $1/1K$) dari ICMP Traceback. Ketika serangan DDoS terjadi setiap daemon (“Zombie”) akan secara relative memproduksi sejumlah kecil trafik, sehingga membutuhkan waktu yang panjang untuk router terdekat untuk membangun paket ICMP Traceback. Secara statistik, router terdekat dengan korban akan membangun ICMP Traceback message yang menuju korban lebih cepat dibanding router terdekat dengan daemon, sehingga susah menelusuri penyerang sebenarnya (riil), yang didapat malah daemon-nya saja.

Sebagai contoh kita memiliki satu korban serangan DDoS, yakni X., dan R adalah salah satu router yang mem-forward trafik DDoS ke arah X. Ketika R membangun ICMP Traceback message, probabilitas ICMP traceback adalah $1/20000$ untuk semua paket yang dikirim ke X. Jika rate paket untuk X adalah $1/10000$ paket per detik, secara statistik satu paket ICMP Traceback akan dibangun kira-kira 2 detik. Jika rate paket adalah 100 paket per detik, maka waktu membangun ICMP Traceback akan menjadi 200 detik. Jika daemon (menyerang X) adalah dalam jaringan sendiri, maka router terdekat ke X secara statistik akan membangun ICMP Traceback message yang menuju ke X setelah serangan berlangsung. Malah mungkin akan memakan waktu beberapa menit, sebelum korban melihat ICMP Traceback message pertama dari border

router dari jaringannya sendiri. Masalah diatas dalam membangun ICMP Traceback jadi mungkin tidak berguna. Karena saat itu mungkin tidak menjadi masalah besar untuk non-victim menerima ICMP Traceback message, resources (seperti siklus CPU). Semua ICMP Traceback message menjadi sampah, dan aktivitas penelusuran ke arah korban jadi tertunda.

Kelemahan lain dari ICMP Traceback adalah pembatasan masalah. Hal tersebut memungkinkan sulit memperoleh Traceback message yang berguna untuk menanggulangi serangan DDoS.

Beberapa pembatasan masalah tersebut adalah:

- Extra traffic yang dibangun ketika sinyal out of band
- Traffic ICMP mungkin subjek yang difilter atau dibatasi rate -nya dari traffic normal.
- Walaupun PKI diharapkan dapat mencegah Attacker (penyerang) dari membangun ICMP Traceback palsu, tetapi tidaklah semua router menggunakan certificate-based scheme.

Untuk mengatasi kelemahan dari ICMP Traceback tersebut dalam mengatasi serangan DDoS, dilakukan beberapa perubahan pada prinsip ICMP Traceback message. Perubahan tersebut seperti :

1. Dengan probabilitas yang dispesifikasikan, resource untuk membangun ICMP Traceback message dibuat tersendiri, seperti paket yang mengarah ke korban DDoS. Dengan kata lain, probabilitas lebih kecil dari 1/20000.
2. Total jumlah dari ICMP Traceback yang dibangun oleh masing-masing router tetap sama. (secara statistic tetap 1/20000).

Dengan cara tersebut diharapkan dapat mengatasi serangan DDOS. Lebih lanjut tentang perbaikan prinsip dapat dilihat dalam evaluasi ICMP Traceback pada [30].

Keuntungan dari menggunakan ICMP Traceback adalah sebagai berikut:

- ICMP Traceback tidak membutuhkan peta upstream pada router untuk merekonstruksi alur serangan ketika IP address dari router dikodekan message ICMP Traceback.

BAB V

KESIMPULAN

Dari bahasan ini kita dapat mengambil, kesimpulan sebagai berikut.

- Dengan ICMP Traceback memungkinkan kita dapat menelusuri serangan yang bersifat denial of service (DoS), yakni mengetahui alur path dari serangan, sehingga dapat difilter seranga tersebut.
- ICMP Traceback mengalami kesulitan dalam mencegah jenis serangan Distributed Denial of Service (DDoS), karena adanya pemabtaan masalah.
- Untuk memperbaharui ICMP Traceback untuk bisa mengatasi serangan DDoS, diperlukan perubahan terhadap pemakaian probabilitas.

DAFTAR PUSTAKA

1. Bellavin, S.M., "Distributed Denial of Service Attacks," <http://www.research.att.com/~smb>, 2000.
2. M. Williams. Ebay, amazon, buy.com hit by attacks, 02/09/00. IDG News Service, 02/09/00, <http://www.nwfusion.com/news/2000/0209attack.html>
3. Bellavin, S.M., W.R. Cheswick. Firewalls and Internet Security. Addison Wesley Longmen, 1994
4. Attrition mirrored sites. <http://Attrition.org/mirror/attrition/>
5. L. Stein. The world wide web security faq, version 2.0.1. <http://www.w3.org/Security/Faq/>
6. J.D. Howard. An analysis of security incidents on the internet 1989 - 1995. Carnegie Mellon University, Carnegie Institute of Technology, <http://www.cert.org/research/JHThesis/>
7. Results of the Distributed-Systems Intruder Tools Workshop Pittsburgh, Pennsylvania USA, November 24 1999, CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, http://www.cert.org/reports/dsit_workshop.pdf
8. Thomer M. Gil, "MULTOPS: a data structure for denial-of-service attack detection", thesis at Mathematics and Computer Science, VRIJE UNIVERSITEIT, 2000
9. David Mazières and M. Frans Kaashoek. The design, implementation and operation of an email pseudonym server. *Proceedings of the 5th ACM Conference on Computer and Communications Security*, 1998
10. CERT Coordination Center. CERT Advisory CA-98.01 "smurf" IP Denial-of-Service Attacks, 1998. Available at <http://www.cert.org/advisories/CA-98.01.smurf.html>.
11. The electrohippies collective. Client-side Distributed Denial-of-Service, 2000. Available at <http://www.gn.apc.org/pmhp/ehippies/files/op1.pdf>
12. P. Ferguson, and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," RFC 2827, 2000
13. SANS Institute. Egress filtering v 0.2, 2000. <http://www.sans.org/y2k/egress.htm>
14. Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical Network Support for IP Traceback. Technical report, Department of Computer Science and Engineering, University of Washington, 2000.
15. Lance Spitzner. The Tools and Methodologies of the Script Kiddie. Know Your Enemy, 2000. <http://www.enteract.com/~lspitz/enemy.html>
16. CERT® Coordination Center, "Results of the distributed systems intruder tools workshop," Nov. 1999, http://www.cert.org/reports/dsit_workshop.pdf
17. D. Dittrich, "The DoS project's 'Trinoo' distributed denial of service attack tool," Oct. 1999; "The 'Stacheldraht' distributed denial of service attack tool," Dec. 1999; "The 'Tribe Flood Network' distributed denial of service attack tool," Oct. 1999, <http://www.washington.edu/People/dad>.
18. J. Barlow and W. Thrower, "TFN2K – an analysis," Feb. 2000, http://packetstorm.securify.com/distributed/TFN2k_Analysis.htm.

19. D. Dittrich, S. Dietrich, and N. Long, "An analysis of the 'Shaft' distributed denial of device tool," Mar. 2000, http://netsec.gsfc.nasa.gov/~spock/shaft_analysis
20. Dittrich, David, George Weaver, Sven Dietrich, and Neil Long, *The "mstream" distributed denial of service attack tool*, <http://staff.washington.edu/dittrich/misc/mstream.analysis.txt>
21. Dietrich, Sven; Dittrich, David; long, Neil, Analyzing distributed Denial of Service Tools : The Shaft Case, ----
22. Dave Dittrich, "Distributed Denial of Service (DDoS) attacks/tools resource page," <http://staff.washington.edu/dittrich/misc/ddos/> Sven Dietrich, Neil Long,
23. David Dittrich, "Analyzing Distributed Denial of Service Tools: The Shaft Case," 14th Systems Administration Conference, LISA 2000
24. Rudolf B. Blazek, Hongjoong Kim, Boris Rozovskii, and Alexander Tartakovsky, "A novel approach to detection of denial-of-service" attacks via adaptive sequential and batch-sequential change-point detection methods," IEEE Systems, Man, and cybernetics Information Assurance Workshop, June 2001
25. Ho Chung, "An Evaluation on Defensive Measures against Denial-of-Service Attack", Departemen of Computer Science, University of Southern California, Los Angeles, CA, 2002
26. S.M. Bellovin, ICMP Traceback Messages. <http://www.ietf.org/proceedings/01dec/I-D/draft-ietf-itrace-01.txt>.
27. [RFC1305]: David L. Mills, "Network Time Protocol (Version 3): Specification, Implementation and Analysis", RFC 1305, Internet Engineering Task Force, March 1992.
28. <http://www.iana.org/assignments/isakmp-registry>.
29. [RFC2104]: H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, Internet Engineering Task Force, February 1997
30. Allison Mankin, Dan Massey, Chie Lung Wu, S. Felix Wu, Lixia Zhang, "On Design and Evaluation of "Intention-Driven ICMP Traceback," 10th International Conference on Computer Communications and Networks (IC3N'2001), Arizona, October 2001