PENDETEKSIAN DINI SERANGAN UDP FLOOD BERDASARKAN ANOMALI PERUBAHAN TRAFFIC JARINGAN BERBASIS CUSUM ALGORITHM

Kafi Ramadhani [1)] , Muhammad Yusuf [2)] , Henni Endah Wahanani [3)]
Jurusan Teknik Informatika, Fakultas Teknologi Industri, UPN "Veteran" Jatim
Jl. Rungkut Madya, Surabaya
email : kafiramadhani@gmail.com [1)] , thea.uzty@gmail.com [2)] , henni_endah@yahoo.com [3)] ,

Abstrak. Serangan DOS (*Denial Of Service*) merupakan sebuah serangan yang sedang ramai dibicarakan di dunia penelitian saat ini. Di zaman yang maju ini, Serangan DOS sudah berkembang menjadi serangan yang terdistribusi yang biasa disebut DDOS (*Distributed Denial Of Services*). Para penyerang (*hacker*) dapat melakukan serangan DOS lebih banyak lagi dengan *zombie host* (komputer yang sudah di injeksi dengan *script* pengontrol jarak jauh / *botnet*) pada target secara terdistribusi dan bersamaan sehingga efek dari serangan ini adalah dapat melumpuhkan target dengan cepat. Didasari dari beberapa penelitian yang ada algoritma CUSUM diakui memiliki titik akurasi yang cukup handal dalam mendeteksi serangan DDOS yang sering terjadi saat ini. Serangan *UDP Flood* juga mendominasi beberapa serangan besar di dunia. Didasari masalah kenyataan dimana *UDP flood* mendominasi serangan yang ada saat ini, penulis ingin membuat IDS (*Intrusion Detection System*) menggunakan algoritma CUSUM. Diharapkan dengan adanya penerapan algoritma CUSUM pada sistem IDS mampu mendeteksi serangan *UDP Flood* dengan mendekati keakuratan yang tinggi dan waktu pendeteksian yang cepat.

Kata kunci: DDOS Attack, UDP Flood, Algoritma CUSUM, IDS

1. PENDAHULUAN

1.1 Latar Belakang

Serangan DOS (Denial Of Service) merupakan sebuah serangan yang sedang ramai dibicarakan di dunia penelitian saat ini. Dan pada umumnya penelitian pada serangan ini difokuskan pada 2 hal yaitu deteksi atau pencegahan [4],[5]. Serangan DOS bisa terjadi pada tipe jaringan apapun, sehingga perlu penelitian-penelitian untuk menemukan bagaimana mendeteksi serangan tersebut. Dampak dari serangan tersebut tidak bisa diremehkan lagi, karena dapat membuat sebuah jaringan dari skala jaringan kecil hingga besar berhenti bekerja (*Down*).

Di zaman yang maju ini, Serangan DOS sudah berkembang menjadi serangan yang terdistribusi yang biasa disebut DDOS (Distributed Denial Of Services). Para penyerang (hacker) dapat melakukan serangan DOS lebih banyak lagi dengan *zombie host* (komputer yang sudah di injeksi dengan *script* pengontrol jarak jauh / botnet) pada target secara terdistribusi dan bersamaan sehingga efek dari serangan ini adalah dapat melumpuhkan target dengan cepat. [6] Serangan DDoS memiliki tiga karakteristik utama: (1) jumlah sumber serangan adalah raksasa tapi lalu lintas menyerang individu sedikit, (2) trafik penyerang sering menyerupai lalu lintas yang sah dan (3) pola serangan akan dicampur untuk menyalakan serangan yang nyata. Adapun macam-macam serangan *DDOS attack* yang sering digunakan para *hacker* untuk menyerang yaitu *SYN-Flooding, SMURF Attack, UDP-Flooding, ICMP-Flooding, DNS-Flooding*. Berujuk pada survey dari lembaga Arbor'S pada tahun 2008, serangan *SYN-Flooding, DNS-Flooding*, dan *SMURF attack* diklasifikasikan sebagai serangan terbesar pada tahun tersebut yang menyerang situs-situs pemerintahan dan 76% diantaranya *SYN-Flooding*. IRC (*Internet Relay Chat*) [11] menjadi sarana tempat yang terpopuler untuk menjadi Master-Bot yang digunakan mengontrol *zombie-Bot* yang menyerang target secara bersamaan dan terdistribusi.

Pada penelitian-penelitian sebelumnya, serangan *SYN-Flooding* menjadi riset utama dari sisi pendeteksian dan pencegahan. Dan algoritma CUSUM [4], [5], [6], [8] menjadi algoritma yang sering digunakan untuk mendeteksi pola serangan ini secara *real-time*. [4] Cusum menjadi algoritma penting dalam penentuan pendeteksian serangan pada penelitian ini. Algoritma Cusum ini digunakan untuk menetapkan parameter

serangan *SYN flooding* dan mendeteksi pola-pola serangan persebut berdasarkan perubahan nilai rata-rata selama proses [5] serangan berlangsung. Dari hasil pendeteksian menggunakan algoritma CUSUM [4] ini, pola-pola pendeteksian serangan lebih akurat dengan mendefiniskan nilai-nilai dan parameter serangan *SYN-flooding*. Algoritma ini mendeteksi setiap source-end serangan dan juga dapat diterapkan pada serangan lainnya yaitu *UDP-Flooding* dan *ICMP-Flooding* [4] karena sifatnya yang universal dengan mengklasifikasikan paket dari jenis serangan, serta mengambil metode yang sesuai untuk menyaring dan mendeteksi paket yang ada.

Pada riset selanjutnya [6], penulis menggunakan algoritma CUSUM untuk mendeteksi pola-pola serangan berdasarkan analisa trafik abnormal per-IP address dalam sebuah jaringan secara real-time. Pada penelitian ini, pendeteksian serangan DDOS menggunakan Algoritma CUSUM ini dibagi menjadi dua tahap yaitu Pencocokan data (*recognition*) dan penentuan serangannya (*Decision*). Setiap IP-Record dikelola dan dicocokkan dengan data yang ada untuk meneliti pola normal atau abnormal (yang melakukan serangan) setiap pengguna jaringan yang ada. Setelah proses pencocokkan (*recognition*) dan penelitian data IP-record, Proses *Decision* mengambil keputusan apakah pola tersebut termasuk serangan atau bukan dan parameternya diambil dari hasil proses recognition. Kesimpulan yang bisa diambil dari penelitian ini adalah berhasilnya mendeteksi dan mencegah serangan secara *real time* melalui pendekatan perilaku traffik setiap IP dalam sebuah jaringan. Sistem ini [6] memiliki tiga keunggulan yaitu : (1) berdasarkan analisis perilaku lalu lintas trafic per-IP, lebih mudah membedakan mana penyerang dari pengguna yang normal, (2) pendekatan ini membutuhkan konsumsi memori yng cukup kecil karena perhitungan yang cukup mudah, (3) Dengan menerapkan algoritma CUSUM non-parameter dan algoritma keputusan, sistem ini dapat mendeteksi serangan secara akurat. Serta sistem ini dapat menyaring lalu lintas serangan dan meneruskan trafik yang normal secara bersamaan dengan menggunakan teknologi identifikasi yang cepat. Hasil menunjukkan bahwa sistem ini memiliki akurasi deteksi DDOS yang tinggi dan waktu deteksi yang pendek. Didasari dari beberapa penelitian yang ada [4], [5],[6],[8] algoritma CUSUM diakui memiliki titik akurasi yang cukup handal dalam mendeteksi serangan DDOS yang sering terjadi saat ini.

Beberapa penelitian yang telah dibahas sudah mampu diterapkan dan di implementasikan pada jaringan saat ini. Penelitian yang ada [4], [5],[6], hanya membahas serangan yang berbasis protocol TCP [1], namun saat ini serangan menggunakan *UDP Flood* telah menjadi topic yang menonjol berkat sejumlah serangan pada *website* perusahaan besar dan pemerintahan yang diluncurkan kelompok oleh *hacker Anonymous* [2], [3]. Trend serangan saat ini, *UDP-Flood* banyak dijadikan senjata utama untuk melakukan serangan *DDOS attack*. Banyak penelitian-penelitian yang telah dilakukan untuk mendeteksi serangan *UDP-Flood* ini [1],[8],[9],[10]. Salah satunya pada penelitan dalam jaringan VANET [8]. Penulis mendeteksi serangan *spoofing* berbasis UDP menggunakan algoritma CUSUM dan *A Bloom filter based IPCHOCKREFERENCE* (BFICR) untuk mendeteksi paket data UDP tersebut termasuk serangan atau bukan. Percobaan simulasi ini menunjukkan bahwa metode yang diusulkan menghasilkan deteksi yang sangat akurat dan hasil klasifikasi namun dengan biaya komputasi rendah. Dari beberapa penelitian yang ada, pendeteksian *UDP-Flooding* ini hanya sebatas simulasi. Sehingga hasil pendeteksian belum bisa dikatakan sempurna, dikarenakan *problem* yang terjadi antara simulasi dan implementasi real-word sangatlah berbeda.

Berdasarkan masalah yang terjadi [2],[3] dan kebutuhan dunia akan pendeteksian serangan *UDP-Flooding* secara *real-world* maka dibutuhkan penelitian lebih lanjut. Penulis ingin mengajukan sebuah penelitian guna mencari titik akurat pendeteksian serangan *UDP Flooding* dan mencari waktu tercepat serta komputasi yang kecil untuk mendeteksi serangan tersebut. Untuk mewujudkan tujuan tersebut, Penelitian ini akan diimplementasikan di dunia nyata. Metode Analisis anomali perubahan trafik dan Algoritma CUSUM dipilih dan digunakan dalam penelitian ini karena menurut beberapa penelitian yang telah dilakukan [4],[5],[6],[8],[9],[10], Algoritma CUSUM merupakan algoritma yang memiliki keuntungan: mampu melakukan deteksi "real-time" dengan memonitor variabel acak yang masuk tanpa jeda [5].

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam pembuatan penelitian ini dijelaskan sebagai berikut:

1. Bagaimana cara menganalisa paket UDP yang bersifat serangan atau bukan serangan berdasarkan analisa trafik jaringan

2. Bagaimana cara menganalisa dan menentukan parameter dalam mendeteksi serangan *UDP-Flooding* menggunakan algoritma CUSUM.
3. Bagaimana cara mengambil dan menganalisis data uji dan data set untuk menganalisa pola serangan *UDP Flood*.
4. Bagaimana cara mengalihkan paket serangan *UDP Flood* setelah dideteksi sehingga tidak mengganggu sistem kerja dari *server*.

1.3 Tujuan Penelitian

Tujuan yang akan dicapai dalam penelitian ini adalah sebagai berikut:

1. Menganlisa paket UDP yang bersifat serangan atau bukan
2. Mendeteksi serangan *UDP-Flooding* menggunakan algoritma CUSUM
3. Mengalihkan paket serangan *UDP-Flooding* ke *server* bayangan sehingga tidak mengganggu kinerja jaringan dan sistem *server* yang dijadikan target serangan

1.4 Manfaat Penelitian

Adapun manfaat penelitian yang diusulkan sebagai berikut:

1. Mendapatkan hasil yang akurat untuk pendeteksian serangan *UDP Flood*
2. Mendapatkan kecepatan waktu mendeteksi serangan *UDP Flood*
3. Mengurangi *false* negatif dalam mendeteksi dengan menggunakan algoritma CUSUM

1.5 Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Penelitian ini menggunakan hanya 2 Data uji untuk dibandingkan yaitu KDD-Cup dan data uji yang dibuat dan diskenariokan dari serangan *UDP Flood* sendiri.
2. Menguji keakuratan dan kecepatan waktu untuk mendeteksi serangan *UDP-Flood*
3. Ruang Lingkup jaringan area kampus
4. Skenario serangan menggunakan 4 botnet dan master bot dilakukan dari IRC Chat.

2. Kajian Pustaka

2.1 DOS (Denial Of Service)

Serangan DOS adalah jenis serangan terhadap sebuah komputer atau *server* di dalam jaringan internet dengan cara menghabiskan sumber (*resource*) yang dimiliki oleh komputer tersebut sampai komputer tersebut tidak dapat menjalankan fungsinya dengan benar sehingga secara tidak langsung mencegah pengguna lain untuk memperoleh akses layanan dari komputer yang diserang tersebut [12].

Dalam sebuah serangan DOS, si penyerang akan mencoba untuk mencegah akses seorang pengguna terhadap sistem atau jaringan dengan menggunakan beberapa cara, yakni sebagai berikut:

1. Membanjiri lalu lintas jaringan dengan banyak data sehingga lalu lintas jaringan yang datang dari pengguna yang terdaftar menjadi tidak dapat masuk ke dalam sistem jaringan. Teknik ini disebut sebagai *traffic flooding*.
2. Membanjiri jaringan dengan banyak *request* terhadap sebuah layanan jaringan yang disediakan oleh sebuah *host* sehingga *request* yang datang dari pengguna terdaftar tidak dapat dilayani oleh layanan tersebut. Teknik ini disebut sebagai *request flooding*.
3. Mengganggu komunikasi antara sebuah *host* dan kliennya yang terdaftar dengan menggunakan banyak cara, termasuk dengan mengubah informasi konfigurasi sistem atau bahkan perusakan fisik terhadap komponen dan *server*. [13] DOS memiliki beberapa jenis serangan, diantaranya adalah :
   - *Ping of Death*
   - *Teardrop*
   - *SYN Attack*
   - *Land Attack*
   - *Smurf Attack*
   - *UDP Flood*

Selain itu, agar komputer yang diserang lumpuh total karena kehabisan *resource* dan pada akhirnya komputer akan menjadi *hang*, maka dibutuhkan *resource* yang cukup besar untuk seorang penyerang dalam melakukan aksi penyerangannya terhadap sasaran. Berikut ini merupakan beberapa *resource* yang dihabiskan :

1. *SwapSpace*. *Swapspace* biasanya digunakan untuk mem-*forked child* proses.
2. *Bandwidth*. Dalam serangan DOS, bukan hal yang aneh bila *bandwith* yang dipakai oleh korban akan dimakan habis.
3. *Kernel Tables*.Serangan pada *kernel tables*, bisa berakibat sangat buruk pada sistem. Alokasi memori kepada kernel juga merupakan target serangan yang sensitif. Kernel memiliki kernelmap limit, jika sistem mencapai posisi ini, maka sistem tidak bisa lagi mengalokasikan memory untuk kernel dan sistem harus di reboot.
4. RAM. Serangan DOS banyak menghabiskan RAM sehingga sistem mau tidak mau harus di *reboot*.
5. *Disk*. Serangan klasik banyak dilakukan dengan memenuhi *Disk*. Data diatas merupakan

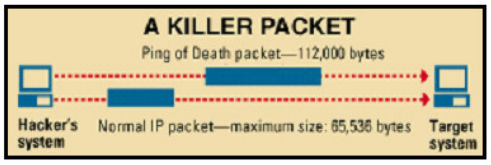
beberapa bagian dari *resource* yang dihabiskan oleh serangan DOS.

Ada beberapa hal yang harus di perhatikan sebelum melakukan penyerangan DOS:

- Serangan membutuhkan Shell Linux (Unix/Comp)
- Mendapatkan exploits di http://packetstormsecurity.nl (gunakan fungsi *search* agar lebih mudah)
- Menggunakan/membutuhkan GCC (*Gnu C Compiler*)

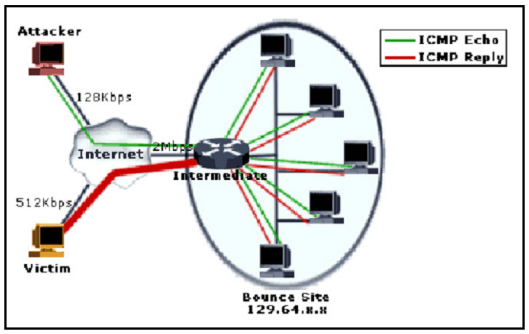
2.1.1 Ping Of Death

[13] *Ping of Death* merupakan jenis serangan yang sudah tidak baru lagi, semua vendor sistem operasi sudah memperbaiki sistemnya. Jenis serangan ini menggunakan *utility ping* yang ada pada sistem operasi komputer. *Ping* ini digunakan untuk mengecek waktu yang akan diperlukan untuk mengirim data tertentu dari satu komputer ke komputerlainnya. Panjang maksimum data menurut TCP protocol IP adalah 65,536 byte.



Gambar 2.1 Mekanisme Ping Of Death [12]

Selain itu, paket serangan *Ping of Death* dapat dengan mudah di *spoof* atau direkayasa sehingga tidak bisa diketahui asal sesungguhnya dari mana, dan penyerang hanya perlu mengetahui alamat IP dari komputer yang ingin diserangnya.

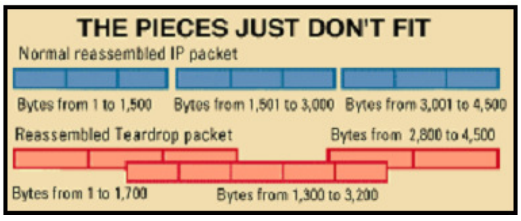


Gambar 2.2 Model Serangan Ping Of Death [12]

Penyerang dapat mengirimkan berbagai aket ICMP (digunakan untuk melakukan ping) yang ter fragmentasi sehingga waktu paket-paket tersebut disatukan kembali, maka ukuran paket seluruhnya melebihi batas 65536 *byte*. Contoh yang sederhana adalah sebagai berikut: C:\windows>ping -l 65540 Perintah MSDOS di atas melakukan ping atau pengiriman paket ICMP berukuran 65540 *byte* ke suatu *host*/*server*. Pada jenis serangan ini, data yang akan dikirim melebihi panjang maksimum yang disediakan. Jika sistem tidak siap pada saat penerimaan data, maka sistem akan *hang*, *crash* atau *reboot*.

2.1.2 Tear Drop

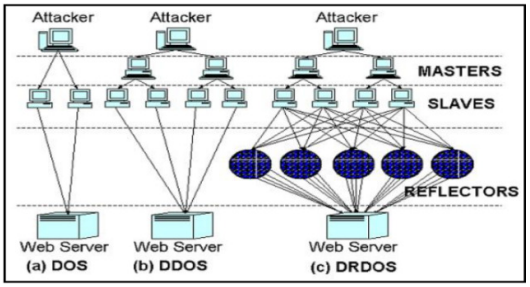
[12] *Tear drop attack* adalah suatu serangan bertipe DOS terhadap suatu *server*/komputer yang terhubung dalam suatu jaringan. *Teardrop attack* ini memanfaatkan fitur yang ada di TCP/IP yaitu packet *fragmentation* atau pemecahan paket, dan kelemahan yang ada di TCP/IP pada waktu paket-paket yang ter fragmentasi tersebut disatukan kembali. Jenis serangan ini. dikembangkan dengan cara mengeksplotasi proses *disassembly-reassembly* paket data. Dalam jaringan Internet, seringkali data harus di potong kecil-kecil untuk menjamin reliabilitas & proses *multiple* akses jaringan. Potongan paket data ini, kadang harus dipotong ulang menjadi lebih kecil lagi pada saat di salurkan melalui saluran *Wide Area Network* (WAN) agar pada saat melalui saluran WAN yang tidak *reliable* proses pengiriman data menjadi lebih *reliable*.



Gambar 2.3 Mekanisme Serangan TearDrop [13]

Pada proses pemotongan data paket yang normal setiap potongan di berikan informasi *offset* data yang kira-kira berbunyi "potongan paket ini merupakan potongan 600 *byte* dari total 800 *byte* paket yang dikirim". Program *teardrop* akan memanipulasi *offset* potongan data sehingga akhirnya terjadi *overlapping* antara paket yang diterima di bagian penerima setelah potongan-potongan paket ini di *reassembly*. Misalnya ada data sebesar 4000 *byte* yang ingin dikirim dari komputer A ke komputer B. Maka, data tersebut akan dipecah menjadi 3 paket demikian:

Di komputer B, ketiga paket tersebut diurutkan dan disatukan sesua dengan OFFSET yang ada di TCP *header* dari masing-masing paket. Terlihat di atas bahwa ketiga paket dapat diurutkan dan disatukan kembali menjadi data yang berukuran 4000 *byte* tanpa masalah.

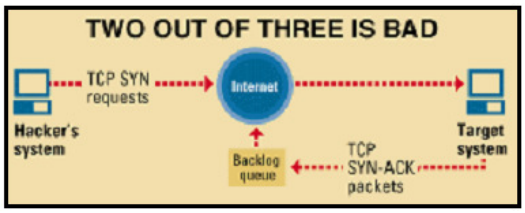


Gambar 2.4 Model serangan TearDrop [12]

gap dan *overlap* pada waktu paket-paket tersebut disatukan kembali. *Byte* 1501 sampai 1600 tidak ada, dan ada *overlap* di *byte* 2501 sampai 3100. Adapun akibat dari serangan ini adalah pada waktu *server* yang tidak terproteksi menerima paket-paket seringkali, *overlapping* ini menimbulkan system yang *crash*, *hang* & *reboot* di ujung sebelah sana.

2.1.3 Syn Flooding

[13] *SYN Flooding* merupakan network DOS yang memanfaatkan '*loophole*' pada saat koneksi TCP/IP terbentuk. Kernel Linux terbaru (2.0.30 dan yang lebih baru) telah mempunyai *option* konfigurasi untuk mencegah DOS dengan mencegah/menolak *cracker* untuk mengakses sistem.

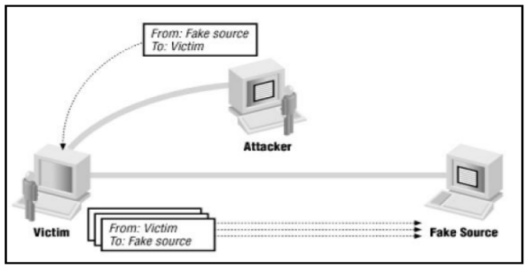


Gambar 2.5 Mekanisme Serangan Syn Flooding [12]

Pada kondisi normal, *client* akan mengirimkan paket data berupa SYN (*synchronization*) untuk men-sinkron kan pada *server*. Lalu *server* akan menerima *request* dari *client* dan akan memberikan jawaban ke client berupa ACK (Acknowledgement). Sebagai tanda bahwa transaksi sudah dimulai (pengiriman & penerimaan data), maka *client* akan mengirimkan kembali sebuah paket yang berupa SYN lagi. Jenis serangan ini akan membajiri *server* dengan banyak paket SYN. Karena setiap pengiriman paket SYN oleh *client*, *server* pasti akan membalasnya dengan mengirim paket SYN ACK ke *client*.
Server akan terus mencatat dan membuat antrian *backlog* untuk menunggu respon ACK dari *client* yang sudah mengirim paket SYN tadi. Biasanya memori yang disediakan untuk *backlog* sangat kecil. Pada saat antrian *backlog* ini penuh, sistem tidak akan merespond paket TCP SYN lain yang masuk dalam bahasa sederhananya sistem tampak *hang*. Sialnya paket TCP SYN ACK yang masuk antrian *backlog* hanya akan dibuang dari *backlog* pada saat terjadi *time out* dari *timer* TCP yang menandakan tidak ada respon dari pengirim.

2.1.4 Land Attack

[13] *Land attack* merupakan salah satu enis serangan SYN, karena menggunakan paket SYN (*synchronization*) pada saat melakukan 3-*way Handshake* untuk membentuk suatu hubungan TCP/IP antara *client* dengan *server*. Namun jenis serangan ini sudah tidak efektif lagi karena hampir pada setiap sistem sudah di proteksi melalui paket *filtering* ataupun *firewall*.



Gambar 2.6 Model Serangan Land Attack [12]

Berikut ini merupakan langkah – langkah yang akan dilakukan dalam melancarkan serangan *Land Attack*:

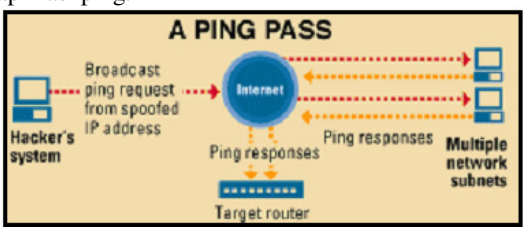
- pertama-tama *client* akan mengirimkan sebuah paket pada *server/host*. Paket yang dikirim yaitu berupa paket SYN.
- Setelah itu *server/host* akan menjawab permintaan dari *client* tersebut dengan cara mengirim paket SYN/ACK (*Synchronization/Acknowledgement*)
- Setelah *server* mengirimkan balasan atas permintaan dari *client*, *client* pun akan kembali menjawab dengan cara mengirimkan sebuah paket ACK kembali pada *server*. Dengan demikian hubungan antara *client* dengan *server* sudah terjalin, sehingga transfer data bisa dimulai.
- *Client* yang bertindak sebagai penyerang akan mengirimkan sebuah paket SYN ke *server* yang sudah di *Dispoof* (direkayasa). Paket data yang sudah direkayasa tersebut akan berisikan alamat asal (*source address*) dan *port number* asal (alamat dan *port number* dari *server*). Dimana akan sama persis dengan alamat tujuan (*destination source*) dan nomor *port* tujuan (*destination port number*). Pada saat

server/host mengirimkan SYN/ACKK kembali ke pada *client*, maka akan terjadi suatu *infinite loop*. Hal ini sebenarnya si *server* bukan mengirimkan paket tersebut ke client melainkan pada dirinya sendiri.

Adapun akibat dari serangan *land attack* ini yaitu seandainya *server/host* tersebut belum terproteksi terhadap jenis serangan ini, *server* akan *crash/ hang*.

2.1.5 Smurf Attack

[13] *Smurf attack* adalah serangan secara paksa pada fitur spesifikasi IP yang kita kenal sebagai *direct broadcast addressing*. Seorang *Smurf hacker* biasanya membanjiri *router* kita dengan paket permintaan *echo Internet Control Message Protocol* (ICMP) yang kita kenal sebagai aplikasi ping.



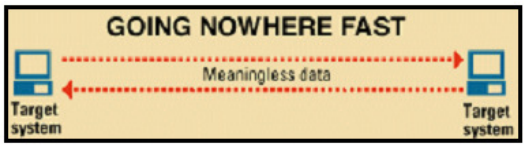
Gambar 2.7 Mekanisme Serangan Smurf Attack [12]

Alamat IP tujuan pada paket yang dikirim adalah alamat *broadcast* dari jaringan, maka *router* akan mengirimkan permintaan *ICMP echo* ini ke semua mesin yang ada di jaringan. Kalau ada banyak *host* di jaringan, maka akan terhadi trafik *ICMP echo* respons & permintaan dalam jumlah yang sangat besar.

Akibat serangan *Smurf attack* ini adalah jika *hacker* ini memilih untuk men-*spoof* alamat IP sumber permintaan ICMP tersebut, akibatnya ICMP trafik tidak hanya akan membuat macet jaringan komputer perantara saja, tapi jaringan yang alamat IP-nya di *spoof* jaringan ini di kenal sebagai jaringan korban (*victim*). Untuk menjaga agar jaringan kita tidak menjadi perantara bagi serangan Smurf ini, maka *broadcast addressing* harus di matikan di *router* kecuali jika kita sangat membutuhkannya untuk keperluan *multicast*, yang saat ini belum 100% didefinisikan. Alternatif lain, dengan cara mem-*filter* permohonan *ICMP echo* pada *firewall*.

2.1.6 UDP Flood

[13] *UDP flood* merupakan serangan yang bersifat *connectionless*, yaitu tidak memperhatikan apakah paket yang dikirim diterima atau tidak. *flood attack* akan menempel pada servis UDP chargen di salah satu mesin, yang untuk keperluan "percobaan" akan mengirimkan sekelompok karakter ke mesin lain, yang di program untuk meng-*echo* setiap kiriman karakter yang di terima melalui servis chargen. Hal ini karena paket UDP tersebut di *spoofing* antara ke dua mesin tersebut, maka yang terjadi adalah banjir tanpa henti kiriman karakter yang tidak berguna antara ke dua mesin tersebut.



Gambar 2.8 Mekanisme serangan UDP Flooding [12]

Adapun karakteristik *UDP flooding* yang dapat disebutkan sebagai berikut:

1. Menggunakan beberapa *IP-Spoof* untuk melakukan serangan secara bersamaan.
2. Mengirimkan paket UDP dengan jumlah yang besar pada interval waktu yang telah ditentukan.
3. Serangan dikirim ke *random port* pada target.

2.2 DDOS (Distributed Denial Of Services)

DDOS [2], [3], [15] merupakan serangan yang saat ini sangat ditakuti di dunia internet. Salah satu jenis serangan DOS [12] yang menggunakan banyak *host* penyerang (baik itu menggunakan komputer yang didedikasikan untuk melakukan penyerangan atau komputer yang "dipaksa" menjadi *zombie*) untuk menyerang satu buah *host* target dalam sebuah jaringan.

Serangan DOS klasik bersifat "satu lawan satu", sehingga dibutuhkan sebuah *host* yang kuat (baik itu dari kekuatan pemrosesan atau sistem operasinya) demi membanjiri lalu lintas *host* target sehingga mencegah *client* yang valid untuk mengakses layanan jaringan pada *server* yang dijadikan target serangan. Serangan DDOS ini menggunakan teknik yang lebih canggih dibandingkan dengan serangan DOS yang klasik, yakni dengan meningkatkan serangan beberapa kali dengan menggunakan beberapa buah komputer sekaligus, sehingga dapat mengakibatkan *server* atau keseluruhan segmen jaringan dapat menjadi "tidak berguna sama sekali" bagi klien.

Serangan DDOS pertama kali muncul pada tahun 1999, tiga tahun setelah serangan DOS yang klasik muncul, dengan menggunakan serangan *SYN Flooding*, yang mengakibatkan beberapa *server* web di Internet mengalami "*downtime*".

Pada awal Februari 2000, sebuah serangan yang besar dilakukan sehingga beberapa situs web terkenal seperti Amazon, CNN, eBay, dan Yahoo! mengalami "*downtime*" selama beberapa jam. Serangan yang lebih baru lagi pernah dilancarkan pada bulan Oktober 2002 ketika 9 dari 13 root DNS *Server* diserang dengan menggunakan DDOS yang sangat besar yang disebut dengan "Ping Flood". Pada puncak serangan, beberapa *server* tersebut pada tiap detiknya mendapatkan lebih dari 150.000 *request* paket *Internet Control Message Protocol* (ICMP). Untungnya, karena serangan hanya dilakukan selama setengah jam saja, lalu lintas Internet pun tidak terlalu terpengaruh dengan serangan tersebut (setidaknya tidak semuanya mengalami kerusakan).

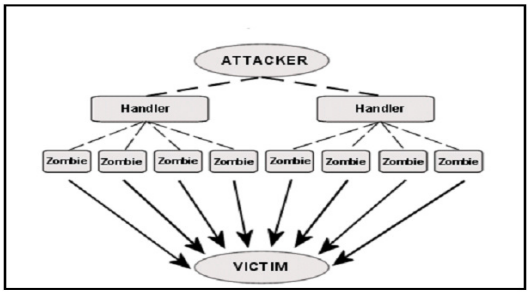
Tidak seperti akibatnya yang menjadi suatu kerumitan yang sangat tinggi (bagi para administrator jaringan dan *server* yang melakukan perbaikan *server* akibat dari serangan), teori dan praktik untuk melakukan serangan DDOS justru sederhana, yakni sebagai berikut:

1. Menjalankan *tool* (biasanya berupa program (perangkat lunak) kecil) yang secara otomatis akan memindai jaringan untuk menemukan *host-host* yang rentan (*vulnerable*) yang terkoneksi ke Internet. Setelah *host* yang rentan ditemukan, *tool* tersebut dapat menginstalasikan salah satu jenis dari Trojan Horse yang disebut sebagai DDOS Trojan, yang akan mengakibatkan *host* tersebut menjadi *zombie* yang dapat dikontrol secara jarak jauh (bahasa Inggris: remote) oleh sebuah komputer master yang digunakan oleh si penyerang asli untuk melancarkan serangan. Beberapa *tool* (*software)* yang digunakan untuk melakukan serangan serperti ini adalah TFN, TFN2K, Trinoo, dan Stacheldraht, yang dapat diunduh secara bebas di Internet.
2. Ketika si penyerang merasa telah mendapatkan jumlah host yang cukup (sebagai *zombie*) untuk melakukan penyerangan, penyerang akan menggunakan komputer master untuk memberikan sinyal penyerangan terhadap jaringan target atau *host* target. Serangan ini umumnya dilakukan dengan menggunakan beberapa bentuk *SYN Flood*, *UDP-Flood*, *Smurf Attack* atau skema serangan DOS yang sederhana, tapi karena dilakukan oleh banyak *host zombie*, maka jumlah lalu lintas jaringan yang diciptakan oleh mereka adalah sangat besar, sehingga "memakan habis" semua sumber daya *Transmission Control Protocol* yang terdapat di dalam komputer atau jaringan target dan dapat mengakibatkan *host* atau jaringan tersebut mengalami "*downtime*".

Karakteristik DDOS attack [6] dapat disebutkan sebagai berikut:

1. Jumlah sumber serangan adalah *gigantic* (raksasa) akan tetapi lalu lintas trafik serangan individu kecil.
2. Penyerang sering disamakan dengan lalu lintas yang sah.
3. Pola serangan akan bercampur untuk memicu serangan yang nyata.

Adapun langkah-langkah utama DDOS attack adalah sebagai berikut (1) Menanamkan Botnet, (2) Menginstruksikan botnet untuk menyerang secara bersamaan seperti pada gambar 2.9. [6] Oleh karena serangan tersebut dilakukan secara bersamaan atau terdistribusi maka efek yang ditimbulkan adalah Memory mengalami peningkatan yang tinggi sehingga menyebabkan korban tidak bisa beraktifitas.



Gambar 2.9 Mekanisme serangan DDOS

2.3 UDP (User Datagram Protocol)

UDP adalah sebuah transport-layer protocol yang memiliki servis transport yang minimal. Protocol ini tidak banyak diajukan dikarenakan dia memiliki sifat connection less atau paket yang dikirim tidak akan dimintai report pengiriman oleh pengirim atau penerima[1]. Data hanya selalu dikirim ke tujuan tanpa harus tau data paket tersebut sampai atau tidak. *packet header* UDP memiliki 4 *fields* [12] seperti yang dtampilkan pada gambar 2.10 : source port, destination port, length and checksum. Hanya 2 yang sangat dibutuhkan dalam packet header tersebut yaitu (length dan destination port); dan 2 fields lainnya (source port dan checksum) adalah optional.

Gambar 2.10 Header UDP [12]

Gambar 2.11 RFC 768 untuk UDP [12]

2.4 Algoritma CUSUM

Algoritma CUSUM adalah algoritma umum dalam statistik proses kontrol, dapat mendeteksi perubahan nilai rata-rata selama proses. CUSUM didasari pada Fakta ini: Jika perubahan terjadi, distribusi probabilitas urutan acak juga akan berubah. Sayangnya, Internet adalah sebuah entitas yang sangat kompleks dan dinamis, dan struktur teori model bisnis internet adalah masalah yang cukup rumit. Dengan demikian, perhatian utama adalah bagaimana untuk mensimulasikan urutan acak. Jadi algoritma kita gunakan di sini lebih cocok untuk menganalisis internet. [4] Dalam algoritma ini, penulis mengumpulkan nilai dalam urutan acak yang tampak lebih besar daripada rata-rata. Salah satu keuntungan dari CUSUM adalah mampu melakukan deteksi secara "real-time" dengan memonitor variabel acak yang masuk tanpa jeda. Metode analisis anomali perubahan trafik dan Algoritma CUSUM dipilih dan digunakan dalam penelitian ini karena menurut beberapa penelitian yang telah dilakukan [4],[5],[6],[8],[9],[10].

2.5 IDS (Intrusion Detection System)

IDS merupakan sebuah sistem yang digunakan untuk mendeteksi adanya serangan dalam sebuah jaringan. IDS ini memiliki sebuah keuntungan yaitu: (1) memonitoring sebuah resource jaringan untuk mendeteksi adanya intrusi atau gangguan dan serangan yang tidak bisa difilter oleh firewall yang sudah ada. (2) menyediakan beberapa opsi untuk me-manage resiko dari sebuah vulnerabillities dan threats. Didalam IDS memiliki beberapa kriteria kondisi yang harus dimengerti dan dipahami yaitu :

- IDS harus menganalisis dan mengidentifikasi intrusi dan serangan
  1. True Positif
  2. True Negatives
- False Negatives adalah kondisi dimana IDS tidak bisa mendeteksi adanya proses serangan
- False Positive adalah kondisi aktifitas biasa yang dianggap sebagai serangan

Type dari IDS dibagi menjadi 2 yaitu **:**

- NIDS (*network Based*)
  Dimana IDS ini memonitoring traffic jaringan dan Memberitahukan dengan cepat ketika ada serangan
- HIDS (host based)
  Dimana IDS ini memonitoring aktifitas sebuah host dan memberitahukan dengan cepat ketika ada serangan.

3. METODOLOGI PENELITIAN

3.1 Perancangan Algoritma CUSUM

Berdasarkan karakteristik serangan UDP Flood dan algoritma CUSUM yang telah dijabarkan pada bab 2, penulis mencoba mendefinisikan parameter-parameter untuk mendapatkan nilai cumulative dari serangan ini. Didasari dari IP-spoof (s) , packet data UDP (Pudp), port (p), time (t). Nilai UDP yang tercatat dalam TCP dump akan dimasukkan ke data dan dicatat semua namun untuk di identifikasi jika memiliki kriteria sebagai berikut Pudp > Np-udp dan nilai Np-udp adalah 150 Kb paket udp. Jika Nudp = {Pudp1,Pudp2,Pudp3, / Pnudp}. Jumlah IP-Spoof yang mengirimkan paket UDP didefinisikan sebagai $\Delta s = \{s1,s2,s3,sn\}$, Nilai s1dst didapat dari jumlah ip yang sama yang mengirimkan data udp. Dan dianggap salah satu kriteria UDP Flood jika ip source penyerang lebih dari 1 Δs. Dapat didefinisikan jika $\Delta s > 1$. Lalu dilihat dari jumlah paket data UDP didefinisikan sebagai Δudp = {Nudp+Nudp2+Nudp3+ .. / Nupdn}, dan paket data UDP-Flood dikirim ke beberapa port secara random, jumlah port didefinisikan $\Delta p = \{p1,p2,p3\}$. Interval waktu serangan didefinisikan sebagai Δt = {Nt1, Nt2, Nt3, Ntn} jika waktu serangan lebih dari Nts yang sudah dtentukan yaitu 60 detik. Dari klasifikasi diatas dapat ditulis rumus untuk pendeteksian cusum sebagai berikut:

Dianggap serangan $\Delta udp > Npudp$ yang dimana $\Delta p > 3$, $\Delta t > 60$ s, $\Delta s > 1$ bukan serangan jika: $Dcsm0 = \sum (\Delta s = (Audp, \Delta t, \Delta p, \Delta s) \leq \Delta p > 3 , \Delta t > 60$ s, $\Delta s > 1$ Dianggap serangan jika: $Dcsm1 = \sum (\Delta s = (Nudp, \Delta t, \Delta p, \Delta s) > \Delta p > 3 , \Delta t > 60$ s, $\Delta s > 1$

3.1.1 Implementasi algoritma CUSUM

Setelah merancang rumus CUSUM dari sub-bab diatas, maka algoritma ini akan diterapkan pada Router yang dilewati data UDP. Cusum digunakan untuk menyaring dan memfilter paket data UDP sesuai parameter yang ditentukan. Cusum

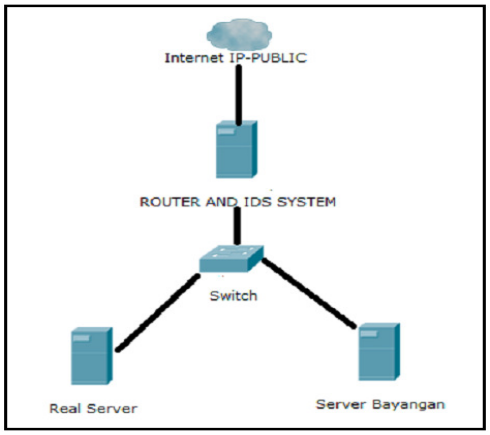
akan ditulis dan di implementasikan menggunakan Bahasa pemograman Python. Dan akan dinamai dengan Cusum-IDS.py. akan disiapkan database untuk menampung data yang masuk dan yang akan diujikan sehingga menemukan data yang valid untuk mendeteksi serangan ini.

Beberapa Table dalam database yang dibuat yaitu IP-Source, Packet Data, Port, Time. Di isi oleh data yang sudah difilter melalui rumus diatas dan data uji. Data uji disini memakai DD-up dan data uji yang dibuat sendiri (yang telah diskenariokan).

3.2 Pengujian dan Evaluasi

3.2.1 Rancangan Jaringan

Adapun rancangan jaringan yang akan dilakukan dalam penelitian ini:



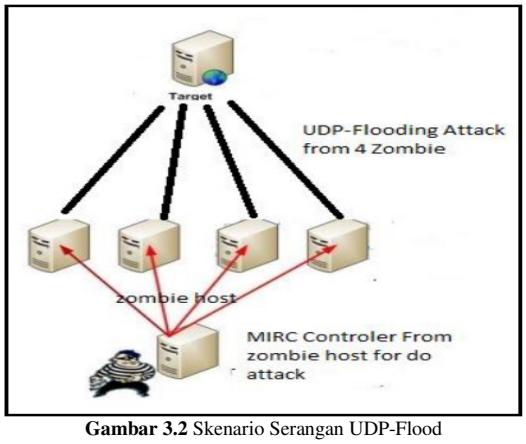
Gambar 3.1 Rancangan Jaringan untuk penelitian

Keterangan:

1. *Real Server*: merupakan *Server* berbasis web-based yang menjadi sasaran target serangan UDP-Flood
2. Router dan IDS *server*: merupakan *server* yang mendeteksi paket data yang masuk.
3. *Server* Bayangan: merupakan *server* yang dijadikan lemparan paket UDP yang sudah dideteksi serangan. Sehingga tidak mengganggu kinerja *Real server*.

3.2.2 Rancangan Serangan UDP-Flooding

Adapun rancangan Serangan UDP-flooding yang akan dilakukan dalam penelitian ini:



Gambar 3.2 Skenario Serangan UDP-Flood

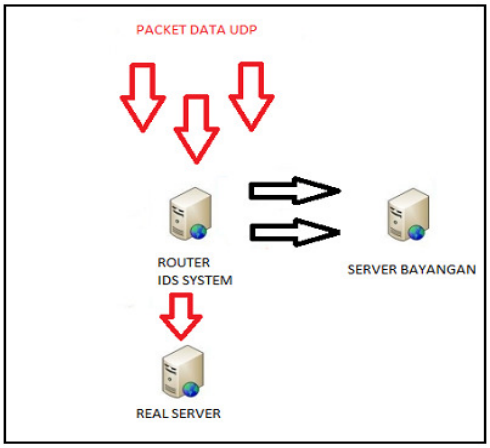
Skenario serangan UDP Flood guna mendapatkan data Set dan menguji sistem IDS:

1. Penyerang menggunakan 4 *zombie* dari website yang sudah di injeksi oleh botnet.
2. Penyerang menggunakan media MIRC sebagai sarana mengontrol *Zombie* untuk melakukan serangan DDOS attack (UDP-Flood) secara bersamaan dan terdistribusi
3. Serangan diluncurkan dengan targer *server* yang telah disiapkan
4. Skenario Serangan akan dilakukan 3 kali dalam sehari, untuk mencari data set.

Serangan dilakukan selama 5 menit. Serangan ke 1 = 1.000 paket/s dengan 1 MB/packetnya. Serangan ke 2 = 10.000 paket/s dengan 2 MB/paketnya. Serangan ke 3 = 100.000 paket/s dengan 1 MB/paketnya.

3.2.3 Rancangan Pengujian Sistem Deteksi

Adapun rancangan pengujian sistem IDS yang akan dilakukan dalam penelitian ini:



Gambar 3.3 Pengujian System IDS

Skenario Pengujian System:

1. Paket data UDP akan dikirim ke *real server*.
2. Sebelum paket UDP dilempar ke real *server* oleh Router, paket UDP akan diteliti oleh system IDS guna menganalisa dan menentukan data packet ini merupakan serangan atau bukan.
3. Jika Packet UDP tersebut bukan serangan, maka paket tersebut akan diteruskan ke *real server* tempat tujuannya
4. Jika Packet UDP tersebut dideteksi serangan, maka paket tersebut akan di alihkan ke *Server* Bayangan.
5. Pengujian System IDS menggunakan data uji KDD-Cup dan Data Uji sendiri.
6. Pengujian sistem IDS ini membandingkan data trafik UDP yang normal dan serangan.

4. Daftar Pustaka

[1] Bardas A.G Zomlot.L Sundaramurthy.C.S. "Classification of UDP trafic for DDOS detection" ieeexplore conference .2011

[2] Hackers Step Up Attacks After Megaupload Shutdown. http://bits.blogs.nytimes.com/2012/01/24/, 2012

[3] National Cyber Alert System - Anonymous DDoS Activity. http://www.uscert.gov/cas/techalerts/TA12-024A.html, 2012

[4] shu.phi, "CHOOSING PARAMETERS FOR DETECTING DDOS ATTACK" Wavelet Active Media Technology and Information Processing (ICWAMTIP), 2012 International Conference on Digital Object Identifier. pp.239-242. 2012

[5] Pinghei. Wang, Qinghua. Zheng, Guolin. Niu, Xiaohong. Guan, and Zhongming, Cai, "Port scan detection algorithms based on statistical traffic features", Journal on Communications, vol 28, no. 12, pp. 14-18, Dec. 2007

[6] zhang.yi liu Q, zhao.G "A Real-Time DDoS Attack Detection and Prevention System Based on per-IP Traffic Behavioral Analysis" ieee-explore conference. 2010

[7] liu L, Jin X, Xu Li. "Real-Time Diagnosis of network Anomaly based on Statisticl Traffic Analysis" ieee-xplore conference, 2012.

[8] verma K, Hasbullah H, Kumar.A "An Efficient Defense Method agains UDP-Spoofed Flooding Traffic of Denial Of Service (DOS) attacks in VANET" IACC,2013.

[9] master of thesis, Daan van der sanden " Detecting UDP attack in High speed network using packet symmetry with only flow data ", University of twente, juli 2008.

[10] rui Xu, Wen-li Ma, wen-ling Z."Defending Against UDP Flooding By negative selection based on Eigenvalui sets" Fifth international conference on information Assurance and Security. 2009

[11] zilong W, Jinsong W, Wenyi H, Chengyi X "The detecton Of IRC botnet Based on abnormal behavior" Second international conference on Multimedia and information technology. 2010

[12] anonim, "DOS Attack" id.wikipedia.com. 2013

[13] khasanah nur. "Metode pencegahan serangan Denial of Services" Tugas Akhir Universitas Sriwijaya. 2008

[14] Subramani rao Sridhar rao ."Denial of Service attacks and mitigation techniques: Real time implementation with detailed analysis" Final Project University Essex. 2010 Parallel Processing Workshops (ICPPW'05).

[15] Arbor Networks. Worldwide Infrastructure Security Report, http://www.arbometworks.com/report. Sept 2008.