Analisis Metode Pengiriman Covert Data pada Covert Channel TCP/IP

Edria Albert Varian W – NIM 13507031

Program Studi Teknik Informatika Institut Teknologi Bandung
Jalan Ganesha no 10 Bandung Indonesia 40132
e-mail: if17031@students.if.itb.ac.id

ABSTRAK

Steganografi merupakan proses menyembunyikan data rahasia ke dalam data lainnya. Steganografi merupakan salah satu teknik enkripsi yang dianggap paling aman saat ini, karena biasanya manusia kurang peka dengan pesan yang sekilas tidak memiliki arti lain. Steganografi ini dapat dilakukan pada media fisik maupun media digital (data). Data yang menjadi media merupakan data yang umum dikirimkan, bisa berupa teks, gambar, audio, maupun video.

Covert Channel adalah teknik untuk mengirim dan menerima informasi tersembunyi (covert data) dari data yang dikirim antara satu komputer ke komputer lain tanpa terdeteksi oleh keamanan sistem. Covert channel ini merupakan salah satu penerapan dari steganografi, yaitu steganografi pada paket data yang dikirimkan melalui jaringan.

Makalah ini membahas tentang beberapa metode yang digunakan dalam teknik covertchannel, seperti manipulasi pada header IP, manipulasi pada initial sequence number dan melakukan TCP acknowledge sequence nimber field "bounce".

Kata kunci: Steganografi, Covert Channel, Covert Data, Bounce

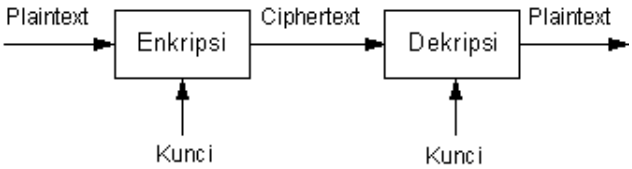
1. PENDAHULUAN

1.1 Kriptografi dan Steganografi

1.1.1 Kriptografi

Kriptografi secara umum adalah ilmu dan seni untuk menjaga kerahasiaan berita. Kriptografi memiliki empat tujuan dasar yaitu Kerahasiaan, Integritas data, Autentikasi, Non-repudiasi. Kerahasiaan menyangkut menjaga isi dari informasi dari siapapun keculai yang memiliki otoritas atau kunci rahasia untuk membuka informasi yang telah disandikan. Integritas data berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, system harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubstitusian data lain kedalam data yang sebenarnya. Autentikasi berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan system maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentifikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain. Non-repudiasi, atau nirpenyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman atau terciptanya suatu informasi oleh yang mengurimkan atau membuat.

Suatu pesan yang tidak disandikan disebut sebagai *plaintext* ataupun dapat disebut juga sebagai *cleartext*. Peroses yang digunakan untuk mengubah *plaintext* ke dalam *ciphertext* inilah yang dinamakan enkripsi. Sedangkan proses untuk mengubah *ciphertext* kembali menjadi *plaintext* disebut dekripsi. Secara sederhana proses kriptografi dapat digambarkan sebagai berikut.



Gambar 1. Proses enkripsi – dekripsi sederhana

Secara umum operasi enkripsi dan dekripsi dapat diterangkan secara matematis sebagai berikut:

E(P) = C [proses enkripsi]

D(C) = P [proses dekripsi]

Pada saat proses enkripsi kita menyandikan pesan P dengan suatu kunci K dan algoritma E lalu dihasilkan pesan C. Sedangkan pada proses dekripsi pesan C tersebut diuraikan dengan menggunakan kunci K sehingga dihasilkan pesan M yang sama seperti pesan sebelumnya.

1.1.2 Steganografi

Steganografi adalah ilmu dan seni menyembunyikan informasi dengan cara menyisipkan pesan di didalam pesan lain. Steganografi berasal dari bahasa Yunani, yaitu

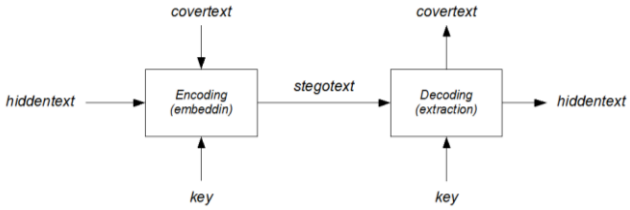
"steganos" yang artinya "tulisan tersembunyi (*covered writing*)". Steganografi membutuhkan dua property yaitu media penampung dan pesan rahasia. Media penampung yang umum digunakan adalah gambar, suara, video, atau teks. Pesan yang disembunyikan dapat berupa artikel, gambar, daftar barang, kode program atau pesan lain.

Penggunaan steganografi antara lain bertujuan untuk menyamarkan eksistensi (keberadaan) data rahasia sehingga sulit dideteksi dan melindungi hak cipta suatu produk. Steganografi dapat dipandang sebagai pengembangan dari kriptografi. Jika pada kriptografi, data yang telah disandikan tetap tersedia, maka dengan steganografi, cipherteks dapat disembunyikan sehingga pihak ketiga tidak mengetahui keberadaannya. Data rahasia yang disembunyikan dapat diekstraksi kembali seperti keadaan asliya.

Keuntungan steganografi dibandingkan dengan kriptografi adalah bahwa pesan yang dikirim tidak menarik perhatian sehingga media penampung yang membawa pesan tidak menimbukan kecurigaan bahwa pesan tersebut merupakan pesan rahasia.

Steganografi memiliki empat property umum yang ada dalam suatu system steganografi: *embedded message* (*hiddentext*) yaitu pesan yang disembunyikan, *cover-object (covertext)* yaitu pesan yang digunakan untuk menyembunyikan *embedded message*. *Stego-object (stegotext)* yaitu pesan yang sudah berisi pesan *embedded message* dan *stego-key* yaitu kunci yang digunakan untuk menyisipkan pesan dan mengekstraksi pesan dari stegotext.

Secara umum, proses steganografi dapat digambarkan secara sederhana sebagai berikut:



Gambar 2 Skema Steganografi

1.2 *Covert Channel* TCP/IP

Covert channel adalah mekanisme untuk mengirim dan menerima informasi tersembunyi dari data yang dikirim dari satu komputer ke komputer yang lain tanpa terdeteksi oleh keamanan system. *Cover channel* akan mengelabui IDS dengan membentuk paket data yang seolah-olah merupakan paket data biasa dan tidak mencurigakan, sementara terdapat informasi-informasi yang disisipkan pada paket data tersebut. *Covert channel* ini merupakan salah satu penerapan dari metode steganografi, yaitu steganografi pada paket data.

Pada kasus TCP/IP, terdapat beberapa metode yang dapat digunakan untuk membangun *covert channel* dan melewatkan data ke host lain. Metode ini dapat digunakan pada berbagai area seperti melakukan *bypass* pada filter paket, *sniffer* jaringan dengan mengenkapsulasi informasi yang sudah terenkripsi ataupun tidak dienkripsi pada paket informasi biasa untuk transmisi rahasia melalui jaringan yang melarang aktifitas untuk melakukan transmisi rahasia.

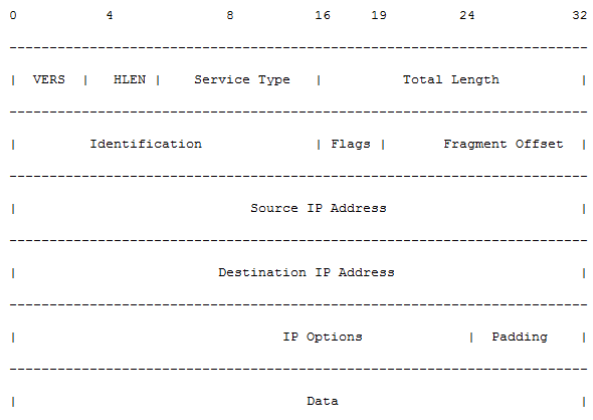
Untuk membuat suatu *covert channel* pada TPC/IP, kita akan melakukan manipulasi informasi pada *header* TCP/IP sedemikian rupa untuk melakukan *encode* pada nilai ASCII yang ditransmisikan. Sebagai awal dari langkah ini, perlu disadari bahwa TCP merupakan protocol yang "*connection oriented*" atau "*reliable*".

TCP memiliki beberapa fitur yang memastikan data akan diterima oleh client dalam keadaan lengkap. Dasar dari fitur ini ada pada saat inisialisasi koneksi TCP yaitu pada "*three way handshake*" yang dapat dideskripsikan dalam tiga langkah, yaitu:

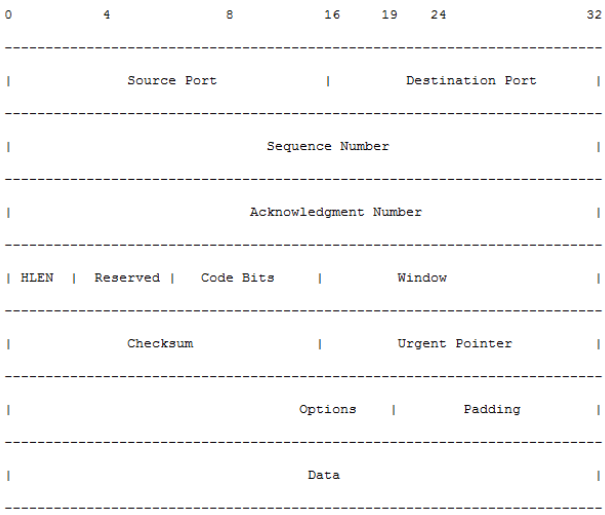
- Langkah pertama: mengirim sinkronisasi paket (SYN) dan *initial sequence number* (ISN).
- Langkah kedua yaitu menunggu host lain untuk merespon dengan mengirimkan *acknowledgement* (ACK).
- Langkah ketiga adalah melengkapi negosiasi dengan mengirim ACK final ke host lain tersebut.

Encode Information in a TCP/IP Header

Header TCP/IP berisi beberapa area dimana informasi dapat disimpan atau disisipkan dan dikirimkan ke host dengan metode *covert*.



Gambar 3 IP Header



Gambar 4 TCP Header

Dalam setiap header terdapat banyak area yang tidak digunakan dalam transmisi normal atau merupakan field pilihan yang dapat dipergunakan jika diperlukan oleh pengirim datagram. Dari hasil analisis terhadap beberapa area yang terdapat pada header IP yang umum, baik area yang tidak terpakai atau optional, terdapat banyak kemungkinan untuk menyisipkan data yang dapat di transmisikan. Untuk melakukan *covert channel* biasanya penyusupan enkapsulasi data dilakukan pada field yang masih cukup penting, bukan pada field-field yang tidak terpakai, karena terkadang field-field yang tidak terpakai akan diacuhkan dan dapat diubah atau dipotong oleh mekanisme filter paket. Maka dari itu dalam *covert channel* biasanya encode dan decode dilakukan pada:

- *IP packet identification field*
- *TCP initial sequence number field*
- *TCP acknowledge sequence number field*

Basis dari manipulasi informasi ini ada pada encode nilai ASCII yang berkisar 0-255 pada area-area diatas.
Jika menggunakan metode ini, dimungkinkan untuk melewatkan data antara host dengan paket yang terlihat sebagai paket awal untuk inisiasi koneksi, paket aliran data atau paket pada langkah-langkah lainnya. Paket ini dapat berisi data yang baru, benar-benar tidak ada data aslinya atau dapat berisi data yang didesain agar tidak tanpak mencurigakan. Paket ini juga dapat berupa *souce* palsu dan IP address tujuan yang palsu pula. Lebih lanjut, paket palsu ini juga dapat digunakan untuk menginisiasi *anonymous TCP/IP "bounced packet network"* dimana paket tersebut bisa dikirim tidak melalui host yang sebenarnya untuk menyulitkan pelacakan oleh system monitor jaringan.

2. METODE

2.1 *IP Identification Field Manipulation*

Field identifikasi dari protokol IP dibantu oleh *re-assembly* dari paket data oleh remote router dan system dari host. Tujuannya adalah untuk memberikan penanda nilai yang unik pada paket sehingga jika paket tersebut dipecah-pecah selama perjalanannya, paket tersebut dapat disusun ulang dengan akurat. Metode *encoding* pertama ini akan mengganti *IP identification field* dengan nilai ASCII dari karakter yang akan diencode. Metode ini dapat digunakan untuk transmisi ke *remote host* yang hanya membaca *IP identification field* dan mentranslasikan nilai dari ASCII yang telah diencode kedalam *buffer* data yang akan diolah. Contoh dibawah ini akan memperlihatkan tcpdump representation dari paket pada jaringan antara dua host "nemesis.psionic.com" dan "blast.psionic.com". Pesan yang telah diencode berisi kata "HELLO" dikirim antara kedua host dengan terlihat sebagai paket yang ditujukan untuk server WWW pada blast.psionic.com. Isi dari paket data yang sebenarnya, tidak akan diperlukan.

Field yang menjadi perhatian disini adalah area IP dari paket yang disebut "id" field, yang terletak di parenthesis. Field ID tersebut direpresentasikan dengan unsigned-integer selama proses pembuatan paket didalam program.

Packet One:

```
18:50:13.551117
nemesis.psionic.com.7180 >
blast.psionic.com.www: S
537657344:537657344(0) win 512 (ttl 64,
id 18432)

Decoding:...(ttl 64, id 18432/256)
[ASCII: 72(H)]
```

Packet Two:

```
18:50:14.551117
nemesis.psionic.com.51727 >
blast.psionic.com.www:
S1393295360:1393295360(0) win 512 (ttl
64, id 17664)

Decoding:...(ttl 64, id 17664/256)
[ASCII: 69(E)]
```

Packet Three:

```
18:50:15.551117
nemesis.psionic.com.9473 >
blast.psionic.com.www: S
```

```
3994419200:3994419200(0) win 512 (ttl
64, id 19456)

Decoding:...(ttl 64, id 19456/256)
[ASCII: 76(L)]
```

Packet Four:

```
18:50:16.551117
nemesis.psionic.com.56855 >
blast.psionic.com.www:
S3676635136:3676635136(0) win 512 (ttl
64, id 19456)

Decoding:...(ttl 64, id 19456/256)
[ASCII: 76(L)]
```

Packet Five:

```
18:50:17.551117
nemesis.psionic.com.1280 >
blast.psionic.com.www: S
774242304:774242304(0) win 512 (ttl 64,
id 20224)

Decoding:...(ttl 64, id 20224/256)
[ASCII: 79(O)]
```

Packet Six:

```
18:50:18.551117
nemesis.psionic.com.21004 >
blast.psionic.com.www:
S3843751936:3843751936(0) win 512 (ttl
64, id 2560)

Decoding:...(ttl 64, id
2560/256)
```
[ASCII: 10(Carriage Return)]

Metode ini digunakan dengan menyuruh host klien untuk membuat paket dengan tujuan host tertentu dan juga informasi host asal dan IP ID field yang telah di encode. Paket tersebut dikirim ke remote host yang sedang terkoneksi (*listening*) pada socket pasif yang mendecode data.

Metode ini relatif langsung dan mudah untuk diimplementasikan. Secara umum, metode ini tergantung pada manipulasi informasi header dari IP dan akan lebih rentan untuk paket *filtering* dan *network address translation* yang informasi headernya dapat ditulis ulang pada saat transit terutama jika terdapat dibelakang *firewall*. Jika ini terjadi, data yang telah diencode akan hilang.

2.2 *Initial Sequence Number Field*

Initial sequence number field (ISN) dari protokol TCP/IP memungkinkan klien untuk membuat suatu negosiasi protokol yg *reliable* dengan *remote server*. Dalam proses negosiasi untuk TCP/IP seperti telah dijelaskan pada metode sebelumnya, dilakukan *three way handshake*. Untuk melakukan *covert channel*, *sequence number field* ini menjadi media yang sempurna untuk mentransmisikan data tersembunyi karena ukurannya yang 32 bit. Dengan kondisi ini, ada banyak metode yang dapat digunakan. Yang paling mudah adalah membangkitkan serangkaian angka dari nilai ASCII karakter yang akan didecode. Ini adalah metode yang digunakan pada contoh dibawah. Pada metode ini kita juga tidak mengurutkan byte terlebih dahulu sebelum membangkitkan deretan angka. Cara ini menjadikan rangkaian angka yang kita bangkitkan lebih realistic. Pada contoh dibawah, rangkaian angka tersebut dikonversi kedalam ASCII dengan dibagi dengan 16777216 yang merupakan representasi dari 65536 * 256.

Packet One:

```
18:50:29.071117
nemesis.psionic.com.45321 >
blast.psionic.com.www: S
1207959552:1207959552(0) win 512 (ttl
64, id 49408)

Decoding:... S 1207959552/16777216
[ASCII: 72(H)]
```

Packet Two:

```
18:50:30.071117
nemesis.psionic.com.65292 >
blast.psionic.com.www: S
1157627904:1157627904(0) win 512 (ttl
64, id 47616)

Decoding:... S 1157627904/16777216
[ASCII: 69(E)]
```

Packet Three:

```
18:50:31.071117
nemesis.psionic.com.25120 >
blast.psionic.com.www: S
1275068416:1275068416(0) win 512 (ttl
64, id 41984)
```

```
Decoding:... S 1275068416/16777216
[ASCII: 76(L)]
```

Packet Four:

```
18:50:32.071117
nemesis.psionic.com.13603 >
blast.psionic.com.www: S
1275068416:1275068416(0) win 512 (ttl
64, id 7936)

Decoding:... S 1275068416/16777216
[ASCII: 76(L)]
```

Packet Five:

```
18:50:33.071117
nemesis.psionic.com.45830 >
blast.psionic.com.www: S
1325400064:1325400064(0) win 512 (ttl
64, id 3072)

Decoding:... S 1325400064/16777216
[ASCII: 79(O)]
```

Packet Six:

```
18:50:34.071117
nemesis.psionic.com.64535 >
blast.psionic.com.www: S
167772160:167772160(0) win 512 (ttl 64,
id 54528)

Decoding:... S 167772160/16777216
[ASCII: 10(Carriage Return)]
```

Dengan metode ini, paket dibentuk dengan data yang sesuai di SYN field dan dikirim ke host tujuan. Host tujuan tersebut yang diharapkan untuk menerima informasi cari klien, akan dengan mudah mengambil field SYN dari setiap paket yang masuk untuk menyusun kembali data yang telah diencode.

Karena ukuran dari informasinya yang dapat direpresentasikan dengan 32 bit *address space*, menjadikan sequence lokasi yang ideal untuk menyimpan data. Disamping contoh yang telah diberikan diatas, masih banyak teknik yang dapat digunakan untuk menyimpan informasi baik dengan mode byte maupun sebagai *bit of information* yang merepresentasikan *sequence number* yang telah dimanipulasi. Pada algoritma yang digunakan di contoh diatas, kita mengambil nilai ASCII dari data kita dan mengubahnya menjadi rangkaian angka yang berguna (biasa dilakukan oleh fungsi untuk menggenerate paket dan dikonversi kembali ke ASCII dengan metode yang simetris. Metode ini dan metode-metode lainnya pada prinsipnya mirip dengan metode "*substitution cipher*" jadi, paket yang berisi informasi yang sama akan menghasilkan rangkaian angka yang sama (seperti pada paket 3 dan 4).

2.3 *TCP Acknowledge Sequence Number Field "Bounce"*

Metode ini begantung pada *spoofing* alamat IP yang memungkinkan mesin yang mengirimnya untuk "*bounce*" paket ke *remote site* dan membuat *site* tersebut untuk meneruskan paket ke tujuan aslinya. Cara ini dapat menyembunyikan pengirim asli dari paket dan membuat terlihat seperti diikirim oleh "*bounce*" host. Metode ini dapat digunakan untuk membuat suatu jaringan komunikasi *anonymous* satu arah yang akan sulit untuk dideteksi apalagi jika *souce* servernya sangat sibuk.

Metode ini bergantung pada karakteristik dari TCP/IP dimana server tujuan merespon kepada inisial permintaan koneksi (paket SYN) dengan paket SYN/ACK yang berisi initial sequence number yang asli ditambah 1 (ISN + 1).

Pada metode ini, pengirim membuat paket yang berisi informasi:

- Forged SOURCE IP address.
- Forged SOURCE port.
- Forged DESTINATION IP address.
- Forged DESTINATION port.
- TCP SYN number dengan encoded data.

Port sumber dan tujuan yang dipilih tidak penting kecuali jika kita ingin menyembunyikan lalulintasnya seperti *service* yang sudah dikenal misalnya HTTP atau kita ingin membuat server melakukan *listening* data pada port yang belum ditentukan, yang juga berarti akan menyembunyikan port sumber.

Pada metode ini paket dikirim dari system komputer di klien dan diarahkan ke IP tujuan yang telah dipalsukan di header. *Bounce* server menerima paket dan mengirim baik SYN/ACK atau SYN/RST tergantung pada *state* dari port dari *bounce* server yang dituju oleh paket. Lalu paket kembalian dikirim kepada alamat sumber palsu dengan nomor ISN yang ditambah satu. *Listening destination* server mengambil paket yang datang tersebut dan mendecode informasi dengan mengubah *sequence number* yang telah dikurang satu menjadi ASCII yang berkesesuaian.

Pada metode *bounce* langkah yang ditempuh pertama adalah client A mengirim paket palsu dengan informasi yang diencode ke bounce server B. Paket berisi juga alamat dari server penerima C. Langkah kedua *bounce* server B menerima paket dan mengembalikan SYN/ACK atau SYN/RST yang cocok. Pada langkah ini *bounce* server B akan mengira bahwa paket datang dari server

penerima C, jadi paket akan dikirim kembali ke alamat C bersama *sequence number acknowledge*. Langkah ketiga, server C menerima paket dari *bounced* server B lalu mendecode data dan menuliskannya di disk. Metode ini sebenarnya mengelabui remote server untuk mengirim paket dan data yang telah dienkapsulasi kembali ke alamat IP sumber palsu dan pada penerima akhir, paket terlihat seperti berasal dari *bounced* server, walaupun sebenarnya memang demikian. Melakukan *bouncing* paket pada site internet yang sudah dikenal seperti (.mil, .gov, .com, dll) adalah salah satu teknik untuk menyembunyikan operasi pada *traffic* yang umum. Tetapi harus di pastikan juga bahwa *bounce* sitenya tidak menggunakan round-robin DNS atau jika iya pastikan jika site tersebut secara pasif melakukan *listening* pada port yang belum ditentukan untuk mendecode transmisi dari *multiple* site. Dengan teknik ini, klien pengirim bisa *bounce* paket melalui ratusan host internet saat server penerima menerima dan menulis semua data yang ditujukan ke port yang belum ditentukan tanpa menghiraukan alamat IP.

3. KESIMPULAN

Dari ketiga metode yang dibahas, setiap metode memiliki kelebihan dan kekurangannya masing-masing. Pada metode pertama yaitu dengan melakukan manipulasi pada header IP, kita dapat mengirimkan pesan dengan mudah tetapi sangat bergantung pada keamana *firewall* yang dipasang dan juga system paket *filtering* dan *network translation*. Metode kedua yaitu pada *initial sequence number field*, kelebihan pada metode ini yaitu terletak pada ukuran area yang bisa direpresentasikan dengan 32 bit sehingga mudah untuk mengubah data menjadi karakter, namun seperti pada metode manipulasi header IP, metode ini juga mudah dipecahkan karena prinsipnya mirip dengan *substitusion cipher* sehingga mudah dikenali, namun dapat lebih dikomplekskan dengan pembangkitan angka random. Metode ketida yaitu, menggunakan teknik *bounce* pada *sequence number field acknowledge* dari TCP. Pada metode ini kelebihannya pengirim paket asli menjadi sulit untuk dilacak apalagi jika dilewatkan pada site-site yang sudah umum diketahui orang tetapi metode ini dapat dicegah dengan melakukan pengaturan pada router. Jadi jika paket tersebut melewati router yang telah diproteksi maka metode ini akan gagal. Tetapi kebanyakan router memang tidak menerapkan proteksi ini sehingga cara ini masih efektif untuk dilakukan.

Implementasi dari teknik *covert channel* ini pun sangat beragam, baik untuk kegiatan yang bersifat proteksi maupun intrusi. Di satu sisi teknik ini dapat digunakan untuk memproteksi data penting yang dikirim agar tidak diketahui oleh orang yang berniat untuk mencuri data tesebut, tetapi disisi lain juga teknik ini dapat digunakan untuk melakukan penyusupan dan mengerusakan system karena dalam paket-paket data tersebut dapat disisipkan kode-kode berbahaya.

REFERENSI

[1] Tedi Heriyanti, *Pengenalan Kriptografi* http://tedi.heriyanto.net/papers/p_kripto.html Tanggal akses : 25 Maret 2010 pukul 16.20

[2] Ahsan, Kamran. *Covert Channel Analysis and Data Hiding in TCP/IP*. Department of Electrical and Computer Engineering. University of Toronto. 2002.

[3] Rowland, Craig. *Covert Channel in the TCP/IP Protocol Suite* http://131.193.153.231/www/issues/issue2_5/rowland/index.html Tanggal akses : 25 Maret 2010 pukul 19.00

[4] *Steganografi* http://www.ittelkom.ac.id/library/index.php?view=article&catid=20:informatika&id=595:steganografi&option=com_content&Itemid=15 Tanggal akses : 25 Maret 2010 pukul 17.00

[2] What is Covert Channel and What Are Some Examples? http://www.sans.org/security-resources/idfaq/covert_chan.php Tanggal Akses: 25 Maret 2010, Pukul 17.00