Studi Perbandingan Metode *Stream Profiling, Active Warden* dan *Quantized Pumps* untuk Mendeteksi *Covert Channel*

Giovanni Sakti Nugraha [1)]
1) Program Studi Informatika ITB, Bandung 40132, email: if14096@students.if.itb.ac.id

Abstract – *Covert channel merupakan salah satu teknik steganografi yang digunakan untuk melakukan pengiriman pesan atau data melalui sebuah jaringan tanpa diketahui oleh selain pengirim dan penerima pesan. Covert channel memanfaatkan beberapa metode seperti menyelipkan bit pada pesan lain dalam protokol TCP/IP serta metode-metode lainnya yang bersifat menyembunyikan pesan atau data yang dikirim. Karena covert channel pada umumnya digunakan untuk hal-hal ilegal, seperti menyelundupkan data ilegal atau terlarang pada sebuah jaringan, maka ada perlunya untuk mencegah pemanfaatan teknik tersebut oleh orang-orang yang tidak bertanggung jawab. Untuk melakukan pencegahan, kita terlebih dahulu dapat menggunakan 3 teknik deteksi covert channel, seperti Stream Profiling, Active Warden dan Quantized Pumps yang masing-masing memiliki karakteristik sendiri. Makalah ini akan membahas mengenai perbandingan ketiga teknik deteksi tersebut dalam mendeteksi covert channel.*

Kata Kunci*: Stream Profiling, Active Warden, Quantized Pumps, Covert Channel, Steganography*

1. PENDAHULUAN

Salah satu bidang yang paling penting dalam dunia keamanan jaringan adalah *covert channel*. *Covert channel* adalah teknik steganografi untuk melakukan pengiriman data melalui sebuah saluran pada jaringan, tanpa diketahui oleh *administrator* jaringan maupun pemakai jaringan lainnya. Pada dasarnya *covert channel* merupakan salah satu cabang dalam steganografi, yaitu menyembunyikan pesan di dalam sebuah media yang dalam hal ini adalah sebuah saluran data atau *data stream* [8].

Terdapat 2 jenis covert channel yang umum digunakan pada pembahasan diskusi keamanan jaringan, yaitu *covert storage channels* dan *covert timing channels*. *Covert storage channels* merupakan teknik membuat saluran transmisi data dengan menggunakan bit penyimpanan pada media transmisi yang sebenarnya tidak ditujukan untuk transmisi data, sedangkan *covert timing channels* menggunakan manipulasi pada sumberdaya sistem yang berhubungan dengan waktu sehingga interval yang dihasilkan dapat digunakan untuk transmisi data.

C*overt channel* terkadang masih menjadi perdebatan dalam dunia keamanan jaringan, karena selain *covert channel* bisa membantu untuk meningkatkan keamanan dalam transmisi data namun selain itu juga dapat berdampak negatif dan menjadi hal yang merusak serta mengganggu jika digunakan oleh orang yang tidak bertanggung jawab. Karena penggunaannya pada sebuah jaringan dapat menyebabkan gangguan stabilitas dan penurunan performa pada jaringan tersebut. Efek lainnya yang dapat ditimbulkan adalah jika *covert channel* digunakan untuk hal-hal yang melanggar hukum maka ada kemungkinan orang yang bertanggung jawab atas jaringan yang bersangkutan akan dituduh sebagai tersangka, walaupun bukan ia yang melakukan kejahatan tersebut.

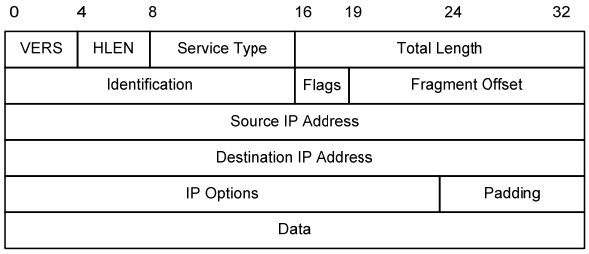
Penggunaan covert channel hingga saat ini masih didominasi oleh kepentingan ilegal dan digunakan oleh orang-orang yang tidak bertanggung jawab. Oleh karena itu, banyak diskusi yang dilakukan mengenai cara mendeteksi penggunaan *covert channel* dalam sebuah jaringan. Beberapa teknik yang cukup populer dan paling visibel untuk diterapkan adalah *stream profiling, activer warden* dan *quantized pumps*.

2. MEDIA PADA JARINGAN YANG DAPAT DIEKSPLOITASI UNTUK *COVERT CHANNEL*

Sebelum membahas lebih lanjut mengenai metode yang dapat dilakukan untuk mendeteksi *covert channel*, terlebih dahulu kita akan membahas mengenai media-media pada jaringan yang dapat dieksploitasi untuk *covert channel* sehingga meningkatkan pemahaman kita mengenai cara kerja *covert channel* tersebut.

2.1. *Internet Protocol* (IP)

IP merupakan protokol *connection-less* yang bekerja di dalam *network layer*. IP sangat umum digunakan pada internet karena efisiensinya yang cukup tinggi dan kepraktisannya. Sebagai contoh, IP menyediakan layanan bagi *layer* di atas *network layer* untuk mengirimkan data tanpa memerlukan koneksi dan pengirim data tidak perlu mengawasi perjalanan data tersebut karena telah diatur oleh IP.

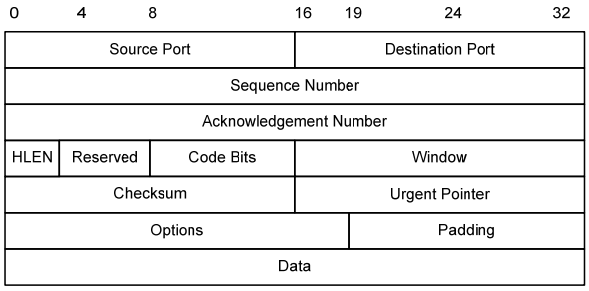


Gambar 1. IP *header*

Sebagaimana dapat dilihat pada gambar di atas, sebuah IP *header* memiliki beberapa *field* yang digunakan dalam berkomunikasi. Namun pada keadaan yang sebenarnya, banyak *field* yang opsional maupun yang tidak digunakan ketika berkomunikasi. Oleh karena itu, *field* yang ada tersebut dapat dimanfaatkan untuk diisi dengan data tersembunyi untuk kemudian dikirimkan ke penerima data tersembunyi tersebut [1]. Contoh *field* yang umum digunakan untuk *covert channel* adalah *field* Identification yang berukuran 16 bit serta *field flags* yang berukuran 3 bit.

2.2. *Transport Control Protocol* (TCP)

TCP adalah protokol *connection-oriented* yang digunakan jika pertukaran data dalam jaringan membutuhkan reliabilitas yang lebih tinggi jika dibandingkan dengan protokol *connection-less* seperti IP. Karena TCP memiliki peningkatan *error-correction* dan *reliability* yang lebih baik dari IP, TCP juga tentunya memiliki lebih banyak bit kendali yang saling *overhead*. Bit-bit kendali tersebut dapat dimanfaatkan sebagai saluran *covert channel* [1].



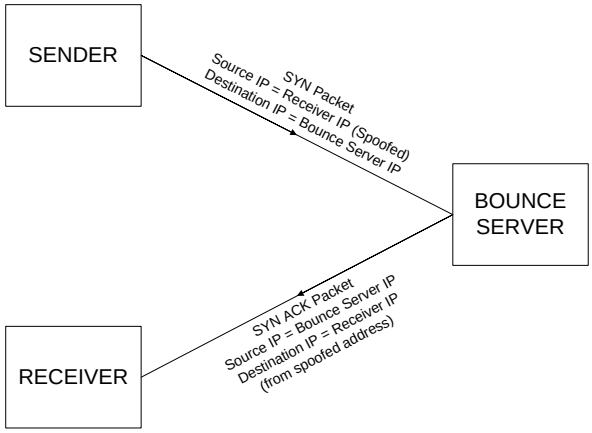
Gambar 2. TCP *header*

2.2.1. Metode ISN

Pada umumnya *Initial Sequence Number* atau ISN merupakan *field* yang digunakan untuk membentuk koneksi pada TCP, *field* ini biasanya digunakan pada 3-*way handshake* yang merupakan 3 transmisi pertama dalam membangung koneksi. Hal ini mengakibatkan ISN dapat dimanfaatkan untuk membangun *covert channel* dua arah yang independen [6].

2.2.2. Metode ACK *bounce*

Pada TCP juga dapat digunakan metode lain untuk membentuk saluran *covert channel*. Metode ini dinamakan ACK *bounce*, metode yang dapat memberikan anonimitas yang sangat tinggi bagi penggunanya, namun koneksi yang dapat terbentuk hanya satu arah. Metode ini melakukan manipulasi terhadap alamat pengirim, sehingga ketika penerima ingin mengirim kembali ACK ke pengirim, pesannya akan dikirim ke penerima yang diinginkan oleh pengirim [9].



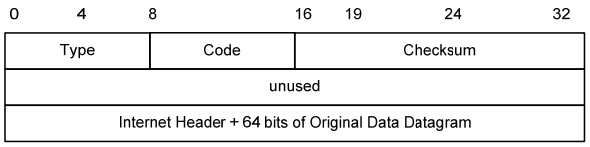
Gambar 3. Skema metode ACK bounce untuk *covert channel*

2.2.3. Metode ACK *tunneling*

Metode ini diciptakan untuk melewati *firewall* yang biasanya melakukan *blocking* terhadap paket dari IP *address* yang tidak dikenal [9]. Pada umumnya *firewall* tersebut akan mengizinkan ACK karena menganggap ACK tersebut datang dari server yang telah dikontak oleh *host* dari *firewall* tersebut. ACK tersebut kemudian dapat dimanfaatkan untuk diisi dengan data yang ingin dikirim kepada *host* yang berada di dalam *firewall*.

2.3. *Internet Control Message Protocol* (ICMP)

ICMP digunakan oleh *host* dan *server* pada sebuah jaringan untuk saling mengirimkan pesan kesalahan jika terjadi kerusakan atau anomali pada jaringan tersebut.



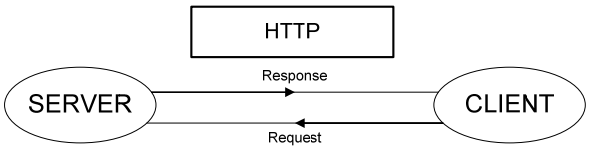
Gambar 4. ICMP *header*

ICMP memiliki beberapa tipe sesuai dengan kode identifikasi yang dituliskan pada *field* tipe. ada dua tipe yang cukup menarik untuk dijadikan *covert channel*, yaitu ICMP *echo request* dan ICMP *echo reply*. Kedua tipe ICMP di atas merupakan paket yang digunakan untuk melakukan verifikasi apakah suatu

host ada dan aktif dalam sebuah jaringan atau tidak. Karena terkadang dirasa cukup mengganggu dan dapat dimanfaatkan untuk serangan *denial-of-service* (DOS) maka mayoritas *firewall* akan melarang pake ICMP *echo request* untuk masuk. Namun, mayoritas *firewall* ternyata masih memperbolehkan pake ICMP *echo replies* untuk masuk sehingga dapat dimanfaatkan untuk *covert channel*.

2.4. *Hyper Text Transfer Protocol* (HTTP)

HTTP merupakan protokol yang digunakan di dalam dunia *World Wide Web*. HTTP juga merupakan salah satu protokol yang paling populer di Internet sehingga hampir seluruh jaringan memperbolehkan HTTP untuk melewati jaringan mereka tersebut. Sayangnya, walaupun HTTP sangat populer dan digunakan oleh mayortias pemakai internet, HTTP memiliki banyak kesalahan desain yang dapat dimanfaatkan untuk *covert channel* [7]. Hal ini juga dipertegas dengan kenyataan bahwa HTTP berada pada *application layer*, tidak seperti mediator lainnya yang berada di *network layer*, penggunaan HTTP sebagai *covert channel* jauh lebih unggul karena kapasitas yang dibawa lebih besar dan tidak memiliki limitasi *bandwidth.*



Gambar 5. Skema HTTP *request-response*

HTTP merupakan protokol yang berjenis *request-response*, sehingga pemanfaatan *covert channel* juga dapat dilakukan dengan *request-response*. Proses yang terjadi dalam melakukan komunikasi adalah, *Server* akan membuka port 80 dan menunggu *request* dari *client*, *client* kemudian akan melakukan koneksi dan komunikasi akan dilakukan seperti halnya koneksi HTTP biasa. Untuk menyembunyikan komunikasi tersebut, dapat dilakukan dengan beberapa teknik seperti menggunakan *proxy* khusus, enkripsi, menggunakan banyak HTTP *header* sekaligus untuk mengecoh dan trik-trik lainnya dapat ditambahkan untuk mengurangi kemungkinan deteksi oleh *admin*.

2.5. IPv6

IPv6 merupakan generasi baru dari IP. IPv6 memiliki beberapa kemampuan tambahan jika dibandingkan dengan IP seperti reliabilitas yang meningkat, *space* alamat yang lebih luas dan keamanan yang ditingkatkan jika dibandingkan dengan IP.

IPv6 memiliki *field options* yang dapat dimanfaatkan untuk melakukan komunikasi dengan *covert channel*. Pertama-tama kita dapat mengisi *field option type* dengan nilai yang tidak sesuai, kemudian kita dapat mengisi 2 bit pertama *field options* dengan 00. Hal ini untuk memberitahu kepada penerima bahwa paket harus tetap diproses walaupun tipenya tidak dikenali. Setelah itu bit berikutnya dari *field options* dapat kita manfaatkan untuk pengiriman data tersembunyi.

2.6. *Domain Name Service* (DNS)

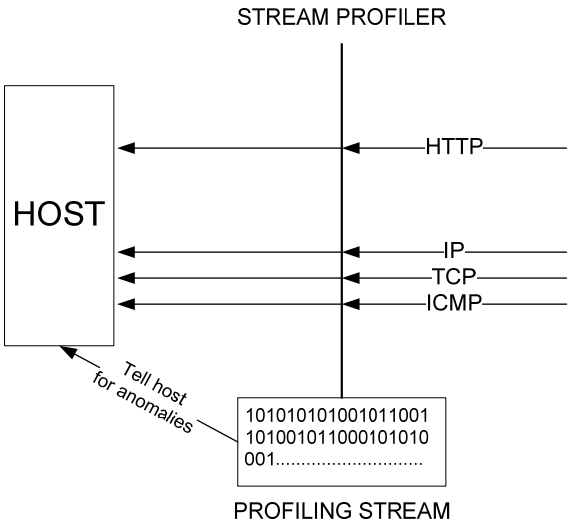
Protokol DNS yang menjadi tulang punggung bagi sistem penamaan domain internet juga dapat digunakan sebagai media *covert channel* [7].

3. DESKRIPSI KETIGA METODE DETEKSI

Bagian ini akan mendeskripsikan masing-masing metode deteksi yang menjadi topik utama dari makalah.

3.1. *Stream Profiling*

Stream profiling merupakan salah satu metode yang paling awal muncul untuk melakukan deteksi *covert channel*. Dalam melakukan deteksi, *stream profiling* pertama-tama akan membangun karakteristik dari jaringan yang diawasi. Kemudian metode ini akan melakukan analisa terhadap aliran data yang mengalir di dalam jaringan dan melakukan perbandingan dengan karakteristik jaringan yang telah disimpan. Metode ini akan mengirimkan peringatan kepada *admin* jika terdapat keanehan atau perbedaan dalam melakukan analisa dan *admin* kemudian akan dapat mengambil tindakan yang sesuai.



Gambar 6. Skema *Stream Profiling*

Metode ini memiliki kemiripan dengan *Anomaly Detection System* (ADS) dan *Instrusion Detection System* (IDS) yang digunakan untuk mendeteksi serangan terhadap *host* seperti *denial of service* (DOS). *Stream profiling* juga merupakan metode yang paling banyak digunakan oleh aplikasi komersial untuk melakukan deteksi *covert channel*, bahkan banyak aplikasi yang menggabungkan *stream profiling* dengan IDS sebagai kumpulan *tools* untuk

menanggulangi penyimpangan pada jaringan.

3.2. *Active Warden*

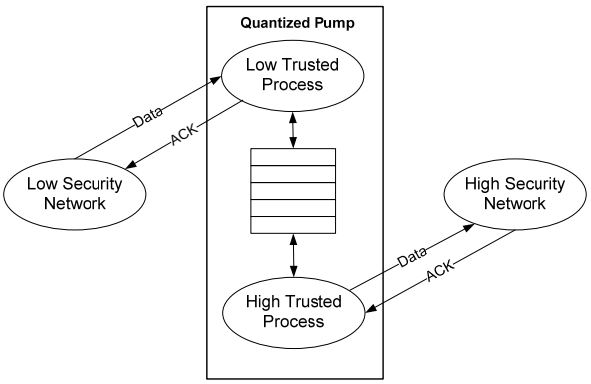
Metode deteksi ini juga telah lama cukup dibahas dalam diskusi untuk mendeteksi *covert channel*. Pada awalnya terdapat cerita yang berjudul "*Prisoner's problem*" yang mengisahkan tentang Alica dan Bob yang sedang dipenjara dan ingin berkomunikasi untuk membicarakan rencana kabur. Namun mereka diawasi oleh pengawas penjara yang bernama Willy, jika Willy menemukan bukti apapun bahwa mereka berkomunikasi secara rahasia, maka Willy akan mencabut izin mereka untuk saling berkomunikasi. Willy tentunya sadar bahwa mereka akan berkomunikasi secara rahasia, maka ia harus mampu memeriksa semua pesan yang lewat tanpa mengubah pesan ataupun mengakibatkan pesan menjadi lama sampai agar Alice dan Bob tidak sadar kalau mereka sedang diawasi. Dalam kasus ini, Willy adalah *active warden* yang bertugas mengawasi dan mendeteksi adanya *covert channel* tanpa membuat pemakai saluran tersebut menjadi curiga, walaupun dalam praktekya nanti terdapat teknik *active warden* yang melakukan normalisasi, dimana pesan akan dirubah untuk menghindari pemanfaatan bit opsional pada protokol yang ada [2].

Active warden merupakan kumpulan dari beberapa teknik-teknik yang dapat digunakan untuk mendeteksi, sekaligus membatasi atau menghambat *covert channel*. Contoh yang paling sederhana, adalah teknik normalisasi yang melakukan perubahan terhadap paket-paket protokol yang lewat. Sebagaimana yang telah kita ketahui pada bagian-bagian sebelumnya, banyak teknik *covert channel* yang memanfaatkan *field-field* opsional pada paket protokol yang seharusnya bernilai 0 jika tidak digunakan. Normalisasi akan melakukan pengisian pada seluruh *field* opsional sehingga *field* tersebut menjadi tidak bermanfaat untuk *covert channel*. Namun, kelemahannya adalah apabila *covert channel* memanfaatkan *field* lain yang bukan merupakan *field* opsional, cara ini menjadi tidak dapat digunakan. Tersedia juga pengembangan dari Normalisasi yang disebut *Minimum Requisite Fidelity* atau MRF [2]. Sebagaimana kita ketahui gambar dan suara yang kita nikmati dapat langsung kita rasakan dengan penglihatan atau pendengaran. Namun, sebenarnya dalam file gambar dan suara terdapat *bit-bit* yang tidak banyak berpengaruh terhadap apa yang kita lihat atau dengar sehingga dapat dimanfaatkan untuk steganografi. Hal yang sama juga berlaku pada pengiriman data melalui protokol, terdapat *bit-bit* yang tidak berpengaruh terhadap data yang dikirim namun dapat dimanfaatkan untuk menjadi *covert channel*. MRF menyempurnakan normalisasi dengan mendeteksi dan mengubah atau mengacak *bit-bit* tersebut agar tidak dapat dimanfaatkan untuk *covert channel* [2]. Jika MRF dikonjungsikan dengan normalisasi maka akan didapatkan kombinasi yang hampir sempurna untuk mengeliminasi *covert channel* pada sebuah jaringan, khususnya untuk *network layer*.

Konsep *active warden* sendiri sebenarnya belum banyak yang direalisasikan, hingga saat ini baru *firewall-firewall* tertentu yang mengimplementasikan *active warden* secara sederhana. Namun pengembangan terus dilakukan untuk metode ini yang merupakan salah satu metode yang paling populer untuk melakukan deteksi dan menghambat *covert channel*.

3.3. *Quantized Pumps*

Metode ini merupakan metode yang dapat dimanfaatkan untuk membatasi *covert channel* tanpa mengurangi *bandwidth* dari jaringan dan merupakan pengembangan dari teknologi pendahulunya, protokol *store and forward* dan *pump and upwards channel* [3]. *Quantized pumps* dapat secara tepat mengontrol kecepatan transfer data *covert channel* hingga seminimal mungkin tanpa mengurangi *bandwidth* dari jaringan sehingga *covert channel* dengan metode *timing* akan terganggu dan tidak dapat berjalan dengan baik [4]. *Quantized pumps* juga melakukan generasi ACK secara independen dan tidak memanfaatkan ACK dalam paket protokol yang dikirim, sehingga pemanfaatan ACK sebagai mediator *covert channel* dapat dikurangi.



Gambar 7. *Quantized pumps*

Secara umum, langkah-langkah yang dilakukan quantized pumps adalah sebagai berikut :

1. *Low trusted process* menerima data dari *low security network*, ACK dari *quantized pumps* dikirim kembali dengan *delay* tertentu.
2. Data dimasukkan ke dalam *buffer* dan ditambahkan *delay* dan *noise* sebelum dikirimkan kepada *High Trusted Process*
3. *High trusted process* menerima dan mengirimkan data ke *high security network*, ACK yang dikirim oleh *high security network* kemudian diterima oleh *high trusted process*.

4. ANALISIS DAN PERBANDINGAN KETIGA METODE DETEKSI *COVERT CHANNEL*

Perbandingan ketiga metode deteksi akan dilakukan dengan analisis dan studi kasus.

4.1. Studi Kasus deteksi *Covert Channel* pada *Network Layer*

Network layer merupakan lapisan yang menampung seluruh protokol primitif seperti TCP, IP dan ICMP. Kita telah mengetahui mengenai masing-masing metode untuk mengeksploitasi protokol tersebut, marilah kita analisis apa yang akan dilakukan masing-masing metode deteksi jika menemukan kasus-kasus yang bersesuaian.

4.1.1. *Covert channel* dengan protokol yang memanfaatkan *field* opsional

Pertama-tama jika kita tinjau *covert channel* ini dengan menggunakan *stream profiling* dalam bentuk aplikasi syncd. syncd akan melakukan pembangunan profil seiring dengan penggunaannya berdasarkan karakteristik jaringan yang diawasi. Setelah berjalan cukup lama, syncd akan memiliki cukup pengetahuan untuk melakukan seleksi terhadap aliran data yang masuk, tentunya jika pada kebiasaan umumnya pada jaringan tersebut *field opsional* jarang digunakan maka syncd tentunya akan segera mendeteksi paket-paket yang menggunakan *field opsional* dan memberikan informasi tersebut kepada pengawas sistem. Namun bagaimana jika ketika pembangunan profil, telah banyak aliran data yang menggunakan *field opsional* atau bahkan *covert channel* telah umum terjadi pada jaringan tersebut. Hal ini tentunya mengakibatkan syncd gagal untuk mendeteksi *covert channel* pada kesempatan berikutnya. Dapat disimpulkan bahwa penggunaan *stream profiling* untuk kasus ini sangat bergantung kepada pembangunan profil, sehingga kita harus yakin terlebih dahulu bahwa profil yang dibentuk adalah profil yang benar. Kemungkinan lainnya adalah jika ditambahkan fungsi heuristik yang didapat berdasarkan *database covert channel signature* kepada metode *stream profiling* sehingga metode tersebut memiliki 2 alternatif solusi, yaitu profil dan heuristik. Masing-masing solusi kemudian diberi pembobotan atau tingkat *confidence* tertentu untuk memudahkan pemilihan solusi.

Metode *active warden* dengan normalisasi dan MRF akan melakukan perubahan terhadap tiap *field* opsional dengan *bit* 0, sehingga kemungkinannya sangat kecil saluran *covert channel* bisa melewati jaringan dengan penerapan metode tersebut. Namun perlu diperhatikan bahwa metode ini mungkin cukup sulit diterapkan bagi jaringan yang umum menggunakan *field* opsional tersebut.

Metode *quantized pumps* akan membatasi pengiriman data dari dalam jaringan ke luar sehingga *covert channel* jenis ini akan sangat terbatasi menjadi hubungan satu arah saja. Jika di dalam buffer *quantized pumps* kita tambahkan *active warden* maka kemungkinan besar komunikasi *covert channel* ini menjadi benar-benar tidak dapat dilakukan.

4.1. 2. *Covert channel* dengan Protokol TCP yang memanfaatkan *field* ACK

Tentunya metode *stream profiling* akan mengalami kesulitan untuk mengatasi *covert channel* ini. Jika metode yang digunakan merupakan ACK *tunneling*, *stream profiling* dapat melakukan deteksi dengan syarat profil yang tersedia sudah cukup lengkap. Namun lain halnya dengan ACK *bounce*, *stream profiling* tentunya akan kesulitan, karena tidak mengetahui mengenai ip *address* pengirim ACK. Hal ini kemungkinan dapat diatasi dengan menggunakan daftar jaringan yang dipercaya dan yang tidak, jika ada SYN yang berasal dari jaringan yang tidak dipercaya, paket tersebut dapat dilewati.

Metode *active warden* akan melakukan normalisasi dan MRF terhadap *field-field* dari ACK yang diterima dan ACK yang akan digunakan untuk dikirim. Hal ini mengakibatkan komunikasi *covert channel* tidak dapat terjadi.

Quantized pumps akan membentuk ACK sendiri untuk dikirimkan kepada pengirim paket dan penerima paket sehingga *covert channel* yang memanfaatkan ACK akan kesulitan untuk dibentuk.

4.2. Studi Kasus deteksi *Covert Channel* pada *Application Layer*

Sayangnya ketiga metode ternyata belum didesain untuk secara khusus menangani *covert channel* pada *application layer*. Bahkan sampai saat ini sangat sulit ditemukan metode yang dapat mendeteksi *covert channel* pada *application layer*. Hal ini disebabkan karena protokol-protokol pada *application layer* memiliki standar yang lebih bebas dan formatnya banyak yang tidak baku sehingga cukup rumit untuk dibuat profilnya atau dinormalisasi [5]. Aplikasi yang paling mendekati dalam mendeteksi dan menghindari *covert channel* pada *application layer* adalah aplikasi yang memanfaatkan teknik mirip dengan *active warden*. Aplikasi ini melakukan normalisasi berdasarkan format HTTP dengan menghapus seluruh spasi dan baris yang berlebihan dan tidak seharusnya ada pada HTTP *header*. Dengan demikian *covert channel* yang memanfaatkan spasi dan baris tersebut dapat dihilangkan.

5. KESIMPULAN

Masing-masing metode deteksi ternyata memiliki kelebihan dan kelemahannya masing-masing pada lingkungan yang berbeda-beda. Kesimpulan yang

dapat diambil adalah, dalam melakukan deteksi *covert channel* kita sebaiknya terlebih dahulu memahami tentang metode-metode *covert channel* yang dapat digunakan dengan membaca jurnal atau referensi yang bersesuaian. Setelah itu kita harus benar-benar memahami jaringan yang ingin kita awasi, tipe-tipe pemakai jaringan, data yang diizinkan lewat dan *policy* yang ada pada institusi pemilik jaringan.

Setelah itu kita dapat memanfaatkan teknik-teknik deteksi yang telah dijelaskan. Sebagai contohnya, jika kita ingin mengawasi sebuah jaringan sederhana yang tidak terlalu tinggi resiko keamanannya, kita dapat mencoba *stream profiling* yang biasanya terintegrasi dengan banyak *tools* dan aplikasi IDS seperti snort dan syncd. Namun perlu diingat, *stream profiling* bukan berarti tidak dapat menangani jaringan dengan resiko keamanan tinggi, aplikasi yang memanfaatkan metode tersebut hanya butuh waktu hingga mendapatkan akurasi yang tinggi dan sesuai dengan spesifikasi jaringan masing-masing. Sedangkan jika kita ingin memanfaatkan *active warden*, kita dapat mencoba *firewall-firewall* komersial yang umumnya memiliki *active warden* sederhana untuk membatasi *covert channel*. Terakhir, jika kita ingin melakukan pengawasan terhadap jaringan dengan tingkat keamanan yang sangat tinggi, maka kita dapat mencoba *quantized pumps*. umumnya metode ini diaplikasikan dalam bentuk *hardware* seperti *physical firewall*. Seluruh koneksi ke luar dan ke dalam jaringan harus melewati *hardware* tersebut terlebih dahulu yang berfungsi sebagai *quantized pumps*. Hal yang perlu dipikirkan juga adalah, sebaiknya *quantized pumps* ini dikonjungsikan juga dengan *active warden* atau *stream profiling* untuk meningkatkan keamanan.

DAFTAR REFERENSI

[1] Allix, Pierre., “Covert channel analysis in TCP/IP networks”, University of Paris-Sud XI, 2007.

[2] Gina Fisk, Mike Fisk, Christos Papadopoulos., “Eliminating Steganography in Internet Traffic with Active Wardens”, University of Southern California, 2002.

[3] Myong H. Kang, Ira S. Moskowitz, Stanley Chincheck., “The Pump: A Decade of Covert Fun”, Center for High Assurance Computer System, Naval Research Laboratory, 2005.

[4] Nick Ogurstov, Hilarie Orman, Richard Schroeppel, Sean O`Malley, Oliver Spatscheck., “Covert Channel Elimination Protocols”, Department of Computer Science, University of Arizona, 1996.

[5] Norka B. Lucena, James Pease, Payman Yadollahpour, Steve J. Chapin., “Syntax and Semantics-Preserving Application-Layer Protocol Steganography”, Syracuse University, 2003.

[6] Owens, Mark., “A Discussion of Covert Channels and Steganography”, SANS, 2002.

[7] Singh, Pukhraj., “Network Based Covert Channels”, Gray-World.

[8] Ir. Rinaldi Munir, M.T., “Kriptografi”, Institut Teknologi Bandung, 2006.

[9] Rowland, Craig H., “Covert Channels in the TCP/IP Protocol Suite”, First Monday, 2007.