Studi Mengenai SILC Sebagai Chat Security Protocol

Haryus Aminul Akbar 13507016
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
If17016@students.if.itb.ac.id

***Abstraksi*—SILC (Secure Internet Live Conferencing) merupakan protokol untuk aktivitas online chat yang berupa open source dan dirilis sejak tahun 2000. SILC bertujuan menjadi protokol standar yang mampu mengamankan pertukaran informasi melalui media chatting. Namun sedikitnya pihak yang ikut serta dalam pengembangan serta beberapa kelemahan dalam kegiatan proses pengamanannya membuat protokol ini membutuhkan banyak penelitian lebih lanjut agar menjadikannya protokol yang dapat diandalkan dalam menjamin keamanan data saat melakukan kegiatan chatting.**

***Kata Kunci*—SILC, security protocol, chat.**

I. PENDAHULUAN

Sejak berkembangnya internet sebagai media komunikasi sekunder, *chat protocol* merupakan hal yang amat populer di kalangan pengguna internet. Sejak kemunculan chat protocol pertama, IRC (Internet Relay Chat), pengguna internet diberi kemudahan untuk melakukan komunikasi secara langsung dengan biaya yang amat murah. Kesuksesan IRC ini diikuti dengan banyak munculnya produk chat lain berupa protokol Instant Messaging (IM) seperti Skype, Yahoo Messenger, ICQ, dll.

Pertukaran informasi yang kerap dilakukan saat melakukan aktivitas chatting tanpa disadari sering dilakukan. Jika informasi yang dikirim merupakan informasi biasa maka tidak akan ada masalah. Lain halnya jika informasi yang dikirim merupakan informasi penting atau topik yang diperbincangkan dalam ruang chat merupakan topik yang tidak bisa disebarluaskan. Apalagi perbincangan online melalui chatting ternyata tidak seaman yang diduga.

Serangan dan penyadapan terhadap informasi yang dikirim melalui chatting dapat dilakukan dengan mudah terhadap kebanyakan chat protocol. Apalagi terhadap chat protocol yang agak kuno seperti IRC, aspek sekuriti dalam chat protocol-chat protocol tersebut tidak diperhasatikan.

Serangan yang paling sering dilakukan terhadap aktivitas chatting adalah *packet sniffing* (penyadapan) terhadap data yang dikirim oleh satu client ke client lain. Dengan menggunakan perangkat gratis seperti Wireshark, tcpdump, atau Cain and Abel, paket yang dikirim dapat dimonitor dengan mudah dan dapat diketahui isinya.

Protokol Secure Internet Live Conferencing (SILC) merupakan chat protocol generasi baru yang menyediakan layanan untuk melakukan konferensi layaknya chat protocol lainnya dengan tambahan dari aspek keamanan. SILC memberi enkripsi dan autentikasi pada pesan-pesan yang dikirim melalui jaringannya. Semua pesan yang dikirim melalui SILC akan dienkripsi, tanpa kemungkinan opsi untuk mengirim pesan tanpa enkripsi. Hal ini dimaksudkan agar tidak terjadi pengirim yang tanpa sengaja mengirim pesan tanpa terenkripsi, seperti pada protokol lain yang menggunakan enkripsi sebagai *plugin* atau *add-in*. Topologi jaringan SILC juga diklaim lebih kuat dan *scalable* daripada jaringan IRC.

II. SILC

Protokol SILC dikembangkan oleh Pekka Riikonen antara tahun 1996 dan 1999 dan pertama kali diluncurkan ke publik pada tahun 2000 sebagai proyek open source. Pada tahun 2004, SILC diajukan kepada IETF (Internet Engineering Task Force), yaitu badan yang mempromosikan standar di dunia internet, namun ditolak[1].

Protokol SILC bisa dibagi menjadi tiga bagian: SILC Key Exchange protocol, SILC Authentication Protocol, dan SILC Packet protocol[2].

SILC Packet Protocol

Bagian paling penting dari protokol SILC adalah paket SILC. SILC Packet protocol menyediakan paket binary yang *secure* dan menjamin isi paketnya aman dan terautentikasi.

Paket digunakan oleh protokol sepanjang waktu untuk mengirim channel message, private message, perintah dan informasi lainnya. Semua paket dalam jaringan SILC selalu dienkripsi dan integritas nya dijamin dengan Message Authentication Code (MAC).

Protokol ini mendefinisikan beberapa tipe paket dan *payload* paket. Tiap tipe paket biasanya memiliki *payload* paket yang spesifik yang mendefinisikan isi dari paket.

Paket SILC dibentuk dari SILC Packet Header yang disertakan dalam semua paket SILC, data area yang berisi packet *payload*, dan MAC area yang memastikan integritas dari paket. Seluruh paket SILC selalu dienkripsi, kecuali pada bagian MAC area.

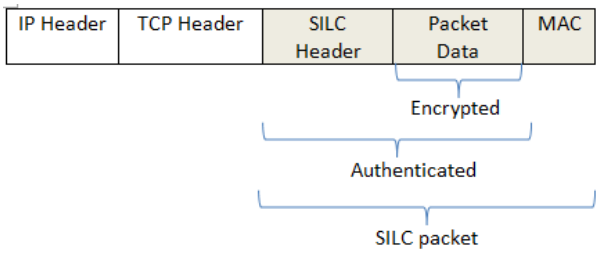

Diagram paket SILC.

SILC Key Exchange Protocol

SILC Key Exchange Protocol (SKE) digunakan untuk pertukaran informasi rahasia dari dua client yang saling terhubung. Hasil protokol ini adalah material kunci yang digunakan untuk mengamankan channel komunikasi. Protokol ini akan dijalankan ketika client terhubung ke server dan ketika server terhubung ke router. Tujuan dari protokol SKE ini adalah membuat session key untuk digunakan pada SILC session saat itu. SKE menggunakan algoritma key exchange dari Diffie-Hellman dan aman dari serangan man-in-the-middle attack dengan menggunakan digital signature.

Protokol ini merupakan protokol pertama yang dijalankan ketika membuat koneksi kepada server SILC. Semua protokol lain akan dijalankan setelah protokol ini, sehingga protokol lain akan aman karena SKE membuat session key yang digunakan untuk mengamankan semua paket yang menyusul. Session key yang dibuat hanya valid untuk suatu periode waktu tertentu atau jika suatu session berakhir.

Security properties yang digunakan dalam session SILC juga dinegosiasikan selama SKE. Protokol ini memiliki *initiator* dan *responder*. *Initiator* adalah yang pertama kali menjalankan negosiasi SKE dan *responder* adalah yang menerima negosiasi SKE. Ketika protocol dijalankan, *initiator* mengirim daftar security properties yang didukung. *Responder* kemudian memilih security properties yang didukungnya dan mengirim jawabannya kembali kepada *initiator*. Security properties termasuk ciphers, fungsi hash, algoritma kunci publik, fungsi HMAC, dan security properties lainnya.

Setelah security properties dipilih, protokol melanjutkan dengan melakukan algoritma key exchange Diffie-Hellman. Pada saat yang sama *initiator* dan *responder* juga saling bertukar kunci publik dan sertifikat. *Initiator* dan *responder* juga menghitung signature yang akan diverifikasi pihak lain. Secara default, protokol dijalankan dalam suatu mode yang disebut mutual authentication mode, dimana kedua belah pihak menghitung signature yang diverifikasi oleh satu sama lainnya secara independen. Dengan cara ini kedua pihak akan membukitikan kepemilikan kunci privat terhadap kunci publik yang disediakan pada protokol. Jika salah satu fase dari protokol gagal, maka koneksi akan langsung diputus saat itu juga.

Kunci publik atau sertifikat yang diterima selama protokol SKE harus diverifikasi, jika tidak maka akan ada celah untuk serangan man-in-the-middle attack terhadap protokol SKE. Jika yang digunakan adalah sertifikat, maka dapat diverifikasi oleh pihak ketiga Certification Authority (CA).

SILC Connection Authentication Protocol

Tujuan dari protokol SILC Connection Authentication adalah mengautentikasi pihak yang tersambung dengan server atau router. Protokol ini dijalankan ketika client terhubung ke server juga ketika server terhubung ke router. Tujuan lainnya adalah menerangkan apakah pihak ini client, server, atau router. Jika client, maka server akan membuat Client ID baru untuk client. Jika server, maka protokol ini akan meminta server untuk mengirimkan Server ID-nya.

Karena protokol SILC Connection Authentication selalu dijalankan setelah protokol SKE, session keys sudah terlebih dahulu disiapkan. Hal ini berarti semua paket yang dikirim pada protokol ini terenkripsi dan terautentikasi.

Proses autentikasi bisa berdasarkan enkripsi kunci publik atau *passphrase*. Juga memungkinkan untuk tidak mendapatkan autentikasi sama sekali. Jika autentikasi berdasarkan *passphrase* maka *passphrase* tersebut akan dikirim ke server. Jika autentikasi berdasarkan kunci publik, maka client akan menandai data dengan private key-nya dan mengirimnya ke server. Server kemudian memverifikasi signature ini dengan menggunakan kunci publik dari client. Paket juga akan dienkripsi untuk autentikasi kunci publik.

Jika autentikasi gagal, maka koneksi ke server atau router akan ditolak. Jika autentikasi sukses, koneksi akan diijinkan. Setelah itu client siap untuk berkomunikasi dengan jaringan SILC.

III. KEKURANGAN PROTOKOL SILC

Berdasarkan artikel dari riseup.net[3], protokol SILC memiliki beberapa kekurangan untuk dijadikan sebagai protokol keamanan untuk aktivitas chat. Poin-poin kekurangan yang dibahas antara lain:

1. Komunitas
   SILC dikembangkan oleh satu orang saja (Pekka Riikonen), walaupun beberapa orang kadang-kadang ikut memberi tambahan. Komunitas SILC tidak berkembang dan meskipun terdapat banyak version release, yang dibenahi hanya beberapa minor bug saja.

2. Menggunakan komponen sendiri
   Protokol SILC dibangun dengan komponen-komponen yang dikembangkan sendiri oleh pengembang dan tidak menggunakan library yang sudah ada. Hal ini memungkinkan peluang adanya bug atau kesalahan yang lebih besar daripada menggunakan library yang sudah ada. Library yang sudah ada digunakan dan dipantau oleh banyak orang dan banyak komunitas, sehingga kemungkinan adanya bug amat kecil, karena jika ada kesalahan seperti itu pasti terpantau oleh banyak penggunanya. Berbeda jika mengggunakan library sendiri.

3. Support untuk implementasi client lemah
   Sebagai akibat dari pengembangan proyek dari komponen-komponen yang dibuat sendiri, dukungan terhadap berbagai macam sistem operasi amatlah lemah. Client yang tersedia sekarang tidak *reliable*, dan memiliki implementasi protokol yang sangat terbatas, sehingga tidak semua orang dapat menggunakannya. Implementasi sebagai plugin pada berbagai client merupakan implementasi yang tidak lengkap yang akhirnya malah tidak memenuhi fungsionalitas SILC dari segi protokol keamanan.

4. Respon terhadap keamanan
   Respon pengembang terhadap isu keamanan yang dilaporkan dikatakan amat lamban. Contohnya pada tahun 2005 terdapat laporan isu keamanan pada SILC di Gento Bug Tracker, namun pihak pengembang baru merilis versi bebas *bug* tiga bulan setelah laporan tersebut masuk.

5. Autentikasi yang lemah
   Proses autentikasi SILC masih memiliki kelemahan dalam implementasinya. Contohnya adalah tidak memungkinkan untuk membedakan antara dua user yang kuncinya telah diverifikasi sebelumnya. Jika dua kunci user ini telah diverifikasi, dan salah satu user memutuskan untuk menyamar menjadi user satunya, SILC akan tetap melanjutkan tanpa memperhatikan situasi ini.

6. Sistem keamanan yang tidak perlu
   Draft spesifikasi SILC menyebutkan beberapa langkah yang diperlukan dalam proses pengamanannya yang sebenarnya tidak menambahkan apa-apa ke dalam sistem keamanan (dan oleh karena itu sebenarnya tidak perlu).
   Contohnya pada langkah koneksi client-server, ketika client terhubung ke sever, server harus melakukan *IP address lookup* dan *reverse IP address lookup* untuk memastikan host asal benar-benar sesuai klaimnya. Padahal DNS merupakan protokol yang tidak aman (rentan spoofing dan poisoning).

7. Tidak ada mekanisme fungsional untuk *key revocation* atau *re-keying*
   Walaupun protokol SILC dapat memanfaatkan penggunaan PKI (Public Key Infrastructure), tidak ada implementasi client atau server yang menggunakan sertifikat X.509v3, sertifikat OpenPGP, atau sertifikat SPKI. Juga tidak ada deskripsi jelas bagaimana pihak-pihak yang terlibat dapat mengidentifikasi satu sama lain dengan pendekatan PKI ini.

Frederick Akalin dan Siu Hong Yen dari Stanford juga melakukan analisis terhadap keamanan protokol SILC menggunakan Murphi[4]. Mereka membuat Murphi model dari protokol dan melihat apakah malicious client bisa melakukan penyadapan terhadap pembicaraan dalam channel di luar channel yang berisi client tersebut.

Murphi model yang mereka bangun dapat mencegat paket yang dikirim dan menyimpannya, meneruskan paket yang disimpan, juga bekerja dengan malicious client atau server dengan memberikan paket tersebut pada malicious client atau server untuk didekripsi.

Hasilnya Murphi dapat menemukan exploit pada protokol yang telah dimodifikasi dimana penyusup bergabung dengan channel, keluar, dan masih bisa mendekripsi paket yang dikirim walaupun dia sudah

keluar dari channel.

Exploit lain yang ditemukan adalah diasumsikan Bob bergabung dalam #foo dan Murphi juga bergabung dalam #foo. Baik Bob maupun Murphi akan mendapat kunci K1 saat ini. Murphi meninggalkan #foo dan server akan mengirim kunci K2 pada Bob. Penyusup memblok pesan, dan sejak saat itu jika Bob mengirim pesan yang dienkripsi dengan kunci K1, Murphi masih tetap bisa mendekripsinya.

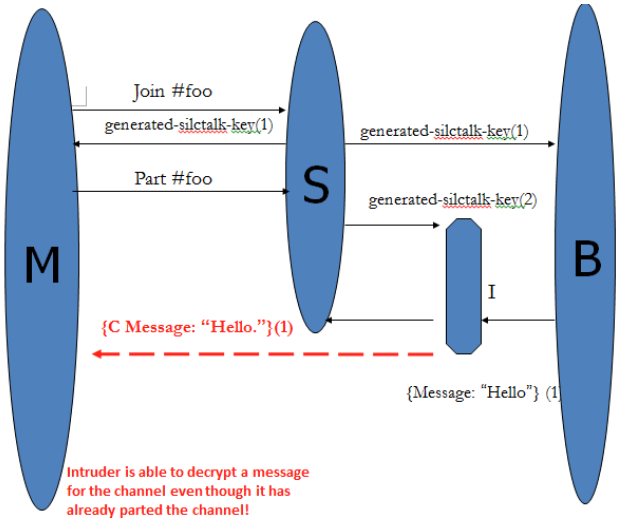

diagram exploit pada SILC oleh Murphi
ket. M = Murphi, S = Server, I = Intruder, B = Bob

Menurut mereka, exploit ini berhasil dilakukan karena:

- Tidak dilakukan *timestamping* atau *key numbering* oleh SILC. Dalam spesifikasi SILC juga tidak disebutkan masalah *timestamping* dan *numbering*.
- Penggunaan channel key. Seharusnya enkripsi menggunakan session key.

IV. Pengujian dan Analisis

Penulis mencoba menguji aplikasi client dari SILC yang sudah terintegrasi dengan chat client Pidgin. Aplikasi client ini bisa diunduh pada website resmi SILC yaitu http://silcnet.org. Versi yang bisa diunduh adalah versi 2.5.0. Penulis mencoba menguji reliabilitas dari implementasi client yang sudah diintegrasikan pada Pidgin.

Selain itu penulis mencoba menguji keamanan protokol dari SILC dengan mencoba melakukan eavesdropping terhadap paket yang dikirim dengan protokol SILC yang sudah diintegrasikan dengan Pidgin dan melihat bentuk enkripsi yang dilakukan terhadap pesan.

Tools yang akan digunakan untuk pengujian ini adalah:

- Wireshark versi 1.4.4
  Digunakan untuk melakukan packet sniffing terhadap pesan yang nantinya dikirim
- Yahoo Messenger versi 10
  Digunakan untuk membandingkan isi paket yang sudah disadap oleh Wireshark dengan isi paket dari SILC.

Sayangnya pengujian gagal dilakukan karena aplikasi client Pidgin mengalami kesulitan ketika melakukan koneksi dengan akun Yahoo (Yahoo Messenger) maupun Gmail (GoogleTalk). Bahkan ketika mencoba melakukan join pada chat room, aplikasi langsung crash dan harus ditutup. Dari situ penulis berkesimpulan terdapat bug dari aplikasi client Pidgin yang sudah diintegrasikan dengan SILC tersebut.

V. Kesimpulan

SILC dapat menjamin keamanan data yang dikirim dari satu user ke user lain dengan memberikan enkripsi terhadap pesan yang dikirim, namun melihat review dari beberapa sumber, terlihat bahwa prinsip keamanannya masih belum bisa menjamin karena masih memungkinkan terjadinya man-in-the-middle attack melalui percobaan oleh Frederick Akalin dan Siu Hong Yen.

Penulis sendiri ketika menguji aplikasi chat client Pidgin yang sudah terintegrasi dengan SILC berkesimpulan integrasi aplikasi tersebut masih memiliki bug dan masih belum menjalankan fungsionalitasnya dengan baik. Melihat tanggal rilis kabar terbaru dari website resmi nya tertanggal 10 Agustus 2009, nampaknya integrasi ini tidak akan diperbaiki lagi dan kemungkinan protokol SILC ini sudah ditingalkan pengembangnya meskipun statusnya masih tetap open source.

References

[1] http://en.wikipedia.org/wiki/SILC_(protocol)
Tanggal akses 07-05-2011

[2] Riikonen. Pekka, "SILC Protocol White Paper". Version 1.2. 22 October 2003.

[3] https://we.riseup.net/riseup+tech/problems-with-silc
Tanggal akses 07-05-2011

[4] http://www.stanford.edu/class/cs259/WWW04/projects/project07/
Tanggal akses 07-05-2011

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Mei 2011

ttd

Haryus Aminul Akbar 13507016