Studi Serangan Kriptografi pada protokol SSH

Ditto Narapratama (13504132)
Jurusan Teknik Informatika ITB, Bandung, email: dittonara@yahoo.com

Abstrak – *Seringkali kita membutuhkan cara untuk mengakses atau mengendalikan komputer dari jarak jauh. Beberapa aplikasi yang sering digunakan untuk kebutuhan ini adalah ftp, rlogin, telnet dan rsh. Tetapi aplikas-aplikasii tersebut memiliki kekurangan dalam segi keamanan*

SSH (Secure Shell) adalah protokol jaringan yang memungkinkan pertukaran data secara aman antara dua komputer. SSH dapat digunakan untuk mengendalikan komputer dari jarak jauh, mengirim file, membuat tunnel yang terenkripsi, dan lain-lain. Protokol ini mempunyai kelebihan dibanding protokol lain yang sejenis seperti telnet, ftp, dan rsh, karena SSH memiliki sistem otentikasi, otorisasi, dan enkripsinya sendiri. Dengan begitu, keamanan sebuah sesi komunikasi melalui bantuan SSH ini menjadi lebih terjamin.

SSH memang lebih aman dibandingkan dengan protokol sejenis, tetapi protokol SSH tetap rentan terhadap beberapa jenis serangan tertentu. Pada umumnya serangan-serangan ini ditujukan pada SSH versi pertama (SSH-1) yang memang memiliki tingkat keamanan yang lebih lemah daripada SSH versi kedua (SSH-2). Salah satu serangan pada SSH versi pertama adalah serangan Man in the Middle pada saat pertukaran kunci.

Pada makalah ini penulis membahas tentang protokol SSH serta algoritma yang digunakan pada kedua versi SSH, lalu serangan-serangan yang terjadi pada SSH dan bagaimana SSH mengatasinya. Pada makalah ini penulis juga akan memberikan usulan untuk meningkatkan tingkat keamanan dari protokol SSH, yaitu dengan cara menggunakan kartu kriptografi untuk autentifkasi.

Kata Kunci*: SSH, man in the middle attac, denial of service, cryptographic card*

1. PENDAHULUAN

Sekarang ini banyak orang yang mempunyai lebih dari satu akun komputer. Ada yang mempunyai akun di *Internet Service Provider* (ISP), akun di jaringan lokal tempat kerja atau kuliah, maupun akun di komputer rumah. Bisa jadi juga seseorang memiliki izin untuk menggunakan akun anggota keluarga atau temannya. Jika seseorang memiliki lebih dari satu akun, adalah hal yang wajar bila ingin dibentuk koneksi diantara akun-akun tersebut. Koneksi ini biasanya dibuat dengan tujuan agar seseorang bisa mentransfer fileke komputer lain melalui jaringan. Koneksi ini juga bisa dimanfaatkan untuk *login* atau memasuki akun melalui komputer lain secara jarak jauh, atau bisa juga untuk memberi *command* atau perintah ke komputer lain secara jarak jauh untuk dilakukan eksekusinya. Banyak program yang dibuat untuk tujuan-tujuan ini, misalnya *ftp* dan *rcp* untuk mentransfer file, *telnet* dan *rlogin* untuk *login* jarak jauh, dan *rsh* untuk eksekusi perintah secara jarak jauh. Sayangnya, kebanyakan program ini mempunya masalah fundamental, yaitu kurangnya aspek keamanan. Jika seseorang mentransfer file sensitif melalui internet, para penjahat bisa menangkap dan membaca file tersebut. Begitu juga jika seseorang mencoba memasuki komputer lain secara jarak jauh dengan menggunakan program seperti telnet, *username* dan *password* orang tersebut bisa ditangkap saat melewati jaringan. Tentunya hal ini sangat berbahaya. Beberapa hal yang bisa dilakukan untuk mencegah hal-hal tersebut adalah dengan cara menggunakan program enkripsi untuk mengubah plainteks menjadi cipherteks agar orang lain tidak bisa membacanya, atau dengan cara memasang *firewall*, yang berguna untuk melindungi sebagian dari jaringan komputer dari orang lain. Solusi-solusi ini bisa digunakan secara sendiri-sendiri atau dikombinasikan, dengan kompleksitas dan biaya yang bervariasi. Solusi yang mudah, murah, tersedia luas dan dapat diandalkan adalah dengan cara menggunakan SSH.

1.1 Pengertian SSH

SSH adalah singkatan dari *Secure Shell*. SSH adalah nama sebuah protokol, tetapi ada juga produk yang mengimplementasikan SSH yang juga bernama SSH, sehingga menimbulkan ambiguitas. Terdapat dua jenis protokol SSH yaitu SSH-1 dan SSH-2. Masing-masing protokol memiliki arsitektur yang berbeda dan menggunakan algoritma kriptografi yang berbeda.

1.2 Fitur SSH

Protokol SSH, baik SSH-1 maupun SSH-2 memiliki fitur sebagai berikut :

1. Enkripsi
   SSH melindungi data saat melewati jaringan dengan cara mengenkripsinya. Saat data dikirim ke jaringan, SSH secara otomatis melakukan enkripsi terhadap data tersebut, lalu setelah data tersebut sampai ke pihak penerima, SSH secara otomatis melakukan dekripsi terhadap data tersebut. SSH menyediakan beberapa algoritma enkripsi, yaitu ARCFOUR, Blowfish, DES, IDEA, dan triple-DES (3DES).

2. Integritas
   Menjamin data yang dikirim melalui jaringan sampai ke penerima dalam keadaan utuh dan tidak termodifikasi. Protokol SSH-2 menggunakan algoritma hash MD5 dan SHA-1, sedangkan protokol SSH-1 menggunakan metode yang lebih lemah, yaitu 32-bit *cyclic redudancy check* (CRC-32) dalam data yang tidak terenkripsi pada setiap paket.

3. Autentifikasi
   Semua koneksi koneksi SSH melibatkan dua autentifikasi, yaitu saat klien mengautentifikasi server SSH (autentifikasi server) dan server mengautentifikasi user yang meminta akses (autentifikasi user). Autentifikasi server menjamin bahwa server SSH asli, dan melindungi dari serangan *man in the middle*. SSH-1 menggunakan RSA untuk autentifikasi server, sedangkan SSH-2 menggunakan DSA. Autentifikasi user pada SSH menggunakan dua cara, yaitu dengan menggunakan sistem public key dan sistem password yang terenkripsi.

1.3 Algoritma yang digunakan pada SSH

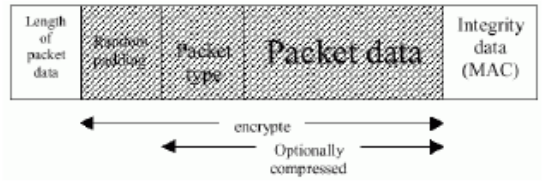
SSH-1 dan SSH-2 menggunakan algoritma public key, hash, simetri, dan kompresi yang berbeda-beda. Berikut adalah tabel yang berisi algoritma yang digunakan pada kedua protokol SSH.

Algoritma/Versi	SSH-1	SSH-2
Public Key	RSA	DSA, DH
Hash	MD5, CRC-32	SHA-1, MD5
Simetri	3DES, IDEA, ARCFOUR, DES	3DES, Blowfish, Twofish, Cast-128, IDEA, ARCFOUR
Kompresi	zlib	zlib

Algoritma enkripsi pada SSH adalah algorima cipher block. Pembagian blok paket pada SSH berbeda untuk SSH-1 dan SSH-2. Berikut adalah perbedaannya :

SSH-1

Berikut adalah struktur paket pada SSH-1 :



Paket pada SSH-1 terdiri dari :

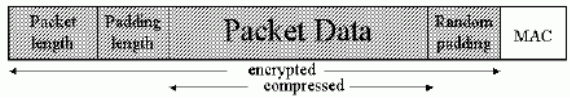
1. Panjang paket data
2. Random padding
3. Jenis paket
4. Paket data
5. Integrity protection data(MAC)

Enkripsi paket dilakukan dengan menggunakan algortma cipher block. Nomor tiga dan empat (jenis paket dan paket data) dikompres menggunakan *gzip*

sebelum enkripsi. Panjang paket dibulatkan ke kelipatan delapan byte dengan menggunakan random padding. Algoritma cipher block tersebut relatif kuat dan sulit dilakukan kriptaanalisisnya.

SSH-2

Berikut adalah struktur paket pada SSH-2



Paket pada SSH-2 terdiri dari :

1. Panjang Paket
2. Panjang Padding
3. Paket Data
4. Random Padding
5. MAC

Pada SSH-2, panjang paket juga ikut dienkripsi

2. SERANGAN PADA SSH

Eavesdropping

Serangan ini termasuk jenis serangan pasif. *Eavesdropping* dilakukan dengan cara membaca *network traffic* tanpa mengubahnya. Enkripsi pada SSH mmencegah *eavesdropping*, karena walaupun data yang dikirim berhasil dibaca, orang yang melakukan *eavesdropping* tidak akan mengerti isinya, karena berupa cipherteks yang sulit didekripsi karena algoritma enkripsi yang kuat (ARCFOUR, Blowfish, DES, IDEA, dan 3DES).

Name Service dan IP Spoofing

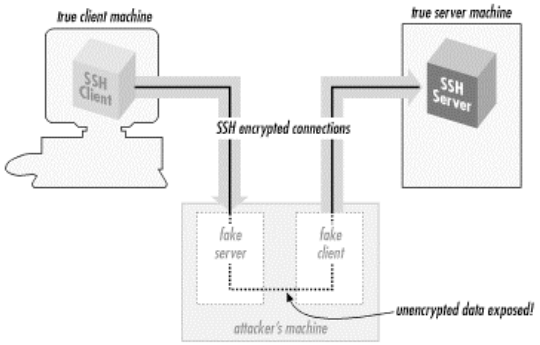
Jika penyerang mengubah nama service (DNS, NIS dan sebagainya), bisa saja program terpaksa terkoneksi dengan komputer yang salah. Penyerang bisa menyamar menjadi host dengan mencuri IP address. Bila hal ini terjadi, program klien akan tersambung dengan server palsu yang bisa mencuri password anda. SSH mencegah serangan ini dengan cara memverifikasi identitas server host (*strict host key checking*). Ketika memulai sebuah sesi, SSH memvalidasi host key dari server dengan mencocokannya dengan daftar lokal yang berisi nama server dan alamat beserta host keynya. Jika host key tidak cocok dengan host key yang ada di daftar, SSH akan memberi peringatan. Namun hal ini bisa menjadi mengganggu karena pesan peringatan bisa jadi sering muncul karena host key tidak tercatat di daftar lokal.

Connection Hijacking

Serangan ini termasuk ke jenis serangan aktif. Terkadang penyerang tidak hanya membaca *network trafffic*, tetapi juga memasukan data ke dalamnya. Hal ini memungkinkan penyerang untuk membacak koneksi TCP. Hal ini bisa jadi sangat berbahaya, karena sehebat apapun metode autentifikasi, penyerang bisa menunggu sampai seseorang melakukan *login*, lalu mencuri koneksi dan kemudian memasukan perintah berbahaya ke dalam session. SSH tidak bisa mencegah *hijacking*, karena ini adalah kelemahan dari TCP, yang beroperasi di bawah SSH. Namun, SSH bisa membuat serangan ini menjadi tidak efektif. Sistem pengecekan integritas dari SSH dapat mendeteksi jika sebuah session dimodifikasi selama perjalanan, bila terdeteksi modifikasi, SSH akan menutup koneksi dengan segera.

Serangan Man in the Middle

Serangan ini termasuk ke dalam jenis serangan aktif. Pada serangan ini, penyerang mengintersepsi komunikasi antara dua pihak yang berkomunikasi dan kemudian menyerupai salah satu pihak dengan cara bersikap seolah-olah ia adalah salah pihak yang berkomunikasi. Pihak lainnya tidak menyadari bahwa ia berkomunikasi dengan pihak yang salah. Tujuan dari serangan ini adalah untuk mendapatkan informasi berharga seperti kunci atau nilai rahasia lainnya. Caranya, penyerang memutus komunikasi antara dua pihak lalu menempatkan dirinya di antara keduanya. Seluruhi informasi dari pengirim dan penerima akan diterima oleh penyerang. Penyerang mengubah pesan yang dikirim oleh pengirim sebenarnya dengan pesannya sendiri, lalu mengirimkannya ke pihak penerima. Penerima mengira informasi tersebut berasal dari pengirim sebenarnya.



SSH mencegah serangan ini dengan dua cara :

1. Autentifikasi server host
   Jika penyerang belum memiliki private host key server, maka penyerang tidak bisa menyamar menjadi server. Agar proteksi ini berhasil, adalah hal yang penting untuk mencocokkan public host key server terhadap daftar host. Jika tidak, maka tidak ada jaminan bahwa server tersebut asli. Jika seseorang terkoneksi untuk pertama kalinya ke server baru dan membiarkan program klien SSH menerima host key, orang tersebut menjadi rentan terhadap serangan *man in the middle*. Tetapi, jika orang tersebut tidak terkena *man in the middle*, koneksi ke server tersebut aman selama penyerang tidak mencuri host key server.
2. Membatasi metode autentifikasi yang rentan terhadap serangan ini
   Metode autentifikasi dengan sistem password cenderung lemah, tetapi autentifikasi dengan menggunakan public key kuat. Penyerang tidak bisa mengetahui session key hanya dengan mengobservasi pertukaran kunci.

Serangan Insertion

SSH-1 memiliki mekanisme pengecekan integritas yang lemah (CRC-32). Kelemahan ini ditemukan dalam serangan yang dilakukan oleh Ariel Futoransky dan Emiliano Kargieman pada bulan Juni 1998. Serangan ini memungkinkan penyerang untuk memasukan data tambahan ke dalam plainteks. Algoritma CRC-32 tidak bisa mendeteksi perubahan ini. Tetapi hal ini bisa dicegah pada SSH-2, karena SSH-2 menggunakan algoritma yang lebih kuat, yaitu triple-DES.

Password Cracking

Sistem password di SSH mempunyai kelebihan dibanding pada telnet atau rsh, karena SSH mengenkripsi password sebelum dikirim ke jaringan. Walaupun begitu, sistem autentifikasi dengan menggunakan password tetaplah bentuk autentifikasi yang lemah. Seseorang harus memilih password yang bagus dan mudah diingat, tetapi tidak mudah ditebak. Password juga tidak boleh dicuri, sehingga harus disimpan di tempat yang aman. Password dapat dengan mudah dicuri dengan cara mengintip saat seseorang mengetikkan password, menggunakan program keylogger, maupun dengan memanfaatkan *social engineering*, yaitu dengan memanfaatkan psikologi manusia untuk memberikan password secara sukarela. Untuk itu, sebaiknya pilih metode autentifkasi pada SSH dengan menggunakan public key.

Serangan pada TCP dan IP

SSH beroperasi di atas TCP, jadi SSH rentan terhadap serangan yang menyerang TCP dan IP. Hasilnya adalah serangan *denial of service*, karena bila terjadi serangan pada TCP dan IP, SSH akan bereaksi dengan cara menutup koneksi.

Analisa Trafik

Walaupun data yang melewati jaringan tidak bisa dibaca oleh penyerang karena terenkripsi dengan baik, namun penyerang tetap bisa mengambil informasi dari aliran data. Penyerang bisa mendapatkan informasi mengenai ke mana data tersebut dikirim, dari mana data tersebut dikirim, dan besar dari data yang dikirim. Penyerang juga bisa mengetahui kapan kira-kira aliran data yang besar terjadi, sehingga bisa diketahui waktu di mana terjadi transaksi data yang besar. Pola-pola yang diketahui ini bisa memberi informasi kapan terjadi operasi backup dan saat-saat di mana mudah dilakukan serangan *denial of service*.

4. KESIMPULAN

Dari sisi enkripsi, SSH memiliki algoritma enkripsi yang kuat dan sulit dikriptaanalisis, seperti triple-DES,

Blowfish, dan lain-lain. Hal ini menyebabkan SSH cukup kebal dari serangan yang menyerang enkripsi. SSH juga cukup handal dalam mengecek integritas data, kecuali protokol SSH-1 yang masih menggunakan algoritma CRC-32 untuk pengecekan integritas. Dari segi autentifikasi, SSH memiliki kelemahan bila menggunakan password sebagai metode autentifikasi, karena password adalah metode autentifikasi yang lemah bila dibandingkan dengan public key.

Serangan-serangan yang bisa dilakukan pada SSH cukup banyak, tetapi sebagian bisa ditangani oleh SSH. Serangan yang bisa ditangani oleh SSH adalah *eavesdropping, name service dan ip spoofing, connection hijacking, man in the middle* (dengan pengecualian SSH-1), dan *insertion attack*(kecuali SSH-1). Tetapi ada juga serangan yang tidak bisa dicegah oleh SSH seperti *password cracking,* serangan pada TCP/IP (*denial of service*) dan analisa trafik. Sebagian serangan tidak bisa dicegah karena SSH berjalan di atas TCP/IP yang mempunyai beberapa kelemahan.

4. USULAN ATAU SARAN

Penulis menyarankan untuk menggunakan sistem kartu kriptografi (*cryptographic card*) untuk metode autentifikasi. Sistem ini menggunakan *smart card* dan smart card reader untuk proses autentifikasi. *Smart card* tersebut menyimpan public key yang berguna untuk proses autentifikasi. Sistem ini lebih aman dibandingkan dengan sistem password. Seseorang tinggal membawa *smart card* tersebut dan menggunakannya di komputer mana saja tanpa perlu merasa khawatir. Kelemahan dari sistem ini adalah *smart card* dan *smart card reader* belum tersedia luas dan belum umum dipakai. Penulis juga menyarankan agar keamanan pada TCP/IP diperbaiki, karena secanggih apapun sistem keamanan SSH, bila TCP/IP yang berada di bawahnya mempunya kelemahan, maka SSH juga mempunyai kelemahan.

DAFTAR REFERENSI

1[1] Daniel Barret, “SSH the Secure Shell, Definife Guide”, 2001

2[2]http://www.afina.com.mx/download/docs/rsa/SecurIDSmartCard.pdf

[3] http://www.stanford.edu/~mlustig/ssh_report.pdf