SSL (Secure Socket Layer)

Imam Prasetyo
imp.masiv@gmail.com
http://superman-kartini.blogspot.com

Pendahuluan

SSL adalah protokol keamanan yang digunakan pada hampir semua transaksi aman pada internet. SSL mengubah suatu protokol transport seperti TCP menjadi sebuah saluran komunikasi aman yang cocok untuk transaksi yang sensitif seperti Paypal, Internet Banking, dan lain-lain. Keamanan dijamin dengan menggunakan kombinasi dari kiptografi kunci publik dan kriptografi kunci simetri bersamaan dengan sebuah infrastruktur sertifikat. Sebuah sertifikat adalah sebuah kumpulan data identifikasi dalam format yang telah distandardisasi. Data tersebut digunakan dalam proses verifikasi identitas dari sebuah entitas (contohnya sebuah web server) pada internet.

SSL menyediakan otentikasi (pada sisi client, dan opsional pada sisi server) terhadap pihak-pihak yang berkomunikasi. SSL dapat mengamankan koneksi antara dua titik, dan tidak ada pihak yang dapat melakukan hal-hal yang bersifat destruktif atau mengakses informasi yang bersifat sensitif. SSL menyediakan sebuah saluran komunikasi yang aman tanpa perlu adanya pertemuan kedua pihak yang berkomunikasi untuk melakukan proses pertukaran kunci.

Fungsi SSL pada komunikasi aman sama seperti fungsi TCP pada komunikasi normal, yaitu menyediakan sebuah infrastruktur komunikasi standar di mana sebuah aplikasi dapat menggunakannya dengan mudah dan hampir tidak dapat terlihat (invisible). SSL menyediakan sebuah komponen penting pada sistem yang aman.

Mekanisme otentikasi dasar seperti *password* Telnet dan otentikasi HTTP dasar menjadi sangat kuat ketika dieksekusi dengan SSL dibandingkan dengan TCP, di mana pada SSL *password* tidak lagi dikirim dalam bentuk plainteks. SSL mengenkripsi koneksi, bukan data pada kedua pihak yang berkomunikasi, dan tidak mengandung mekanisme untuk otentikasi user ataupun perlindungan password (hanya koneksi yang diotentikasi, keamanannya akan gagal jika mesin pada kedua pihak yang berkomunikasi compromised).

Implementasi SSL paling pertama dikembangkan oleh Netscape Communications Corporation pada awal tahun 1990-an untuk mengamankan HTTP. Pada akhir tahun 1990-an, semakin terlihat dengan jelas bahwa SSL 2.0 tidaklah aman. Netscape memulai untuk membangun SSL 3.0. Dengan bantuan Netscape, Internet Engineering Task Force (IETF, badan yang mengatur untuk standar internet) memulai untuk menstandardisasi SSL, sebuah proyek yang kemudian dikenal dengan nama TLS (Transport Layer Security). SSL 3.0 tidak dikembangkan seteliti TLS, sehingga SSL 3.0 dapat dirilis lebih dahulu dan menggantikan SSL 2.0 sebagai standar industri. TLS yang akhirnya diselesaikan pada tahun 2000, menyediakan protokol terstandardisasi yang pertama untuk SSL. Walaupun SSL 3.0 masih digunakan secara luas, untuk pengembangan terbaru termasuk sudah tertinggal karena saat ini hampir semua browser modern mendukung TLS.

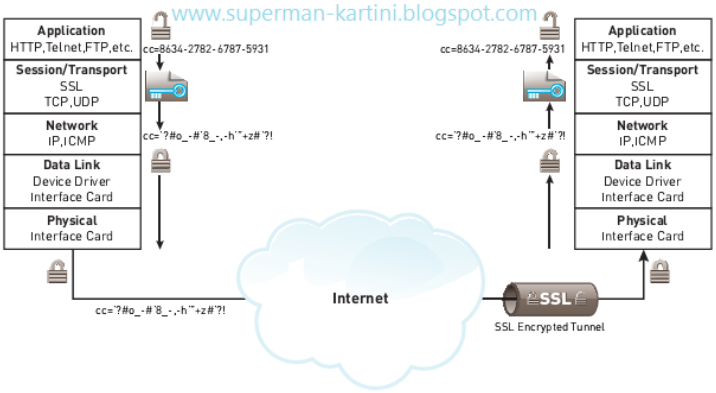
SSL Transaction

1. Handshake dimulai ketika klien terhubung ke SSL-enabled server, meminta sambungan aman, dan menyajikan daftar cipher dan versi yang didukung.
2. Dari daftar ini, server mengambil cipher terkuat dan memberitahu klien. Selain itu, server akan mengirimkan identifikasi dalam bentuk sertifikat digital. Sertifikat biasanya berisi nama server, *certificate authority* (CA) terpercaya, dan

kunci enkripsi publik server.

3. Klien memverifikasi bahwa sertifikat tersebut valid dan *certificate authority* (CA) server sesuai dengan CA list klien. Sertifikat CA ini biasanya dikonfigurasi secara lokal.
4. Setelah menentukan bahwa sertifikat tersebut valid, klien menghasilkan master secret, mengenkripsi dengan kunci publik server, dan mengirimkan hasilnya ke server.
5. Ketika server menerima master secret itu, server mendekripsi dengan kunci pribadi. Hanya server yang bisa mendekripsinya menggunakan kunci pribadi.
6. Klien dan server kemudian mengubah master secret untuk satu set kunci simetris disebut keyring atau kunci sesi. Kunci simetris ini adalah kunci umum yang digunakan server dan browser untuk mengenkripsi dan mendekripsi data. Ini adalah fakta yang membuat kunci tersebut tersembunyi dari pihak ketiga, karena hanya server dan klien yang memiliki akses ke kunci pribadi.
7. Data transfer siap dilakukan melalui saluran yang aman. Jika salah satu dari langkah di atas gagal, jabat tangan SSL gagal, dan sambungan tidak diciptakan.

Meskipun proses otentikasi dan enkripsi mungkin tampak agak rumit, hal itu terjadi dalam waktu kurang dari satu detik. Umumnya,pengguna bahkan tidak tahu prosesnya atau bagaimana hal itu terjadi. Namun, pengguna dapat mengetahui koneksi aman telah terbentuk (SSL) sejak web browser SSL-enabled menampilkan kunci tertutup kecil di bagian kanan atas browser (biasanya samping kiri URL). Pengguna juga dapat mengidentifikasi situs web yang aman dengan melihat alamat situs web, alamat situs web aman biasanya dimulai dengan https. Berikut ini ilustrasi bagaimana SSL bekerja pada OSI Layer.



Bagian SSL

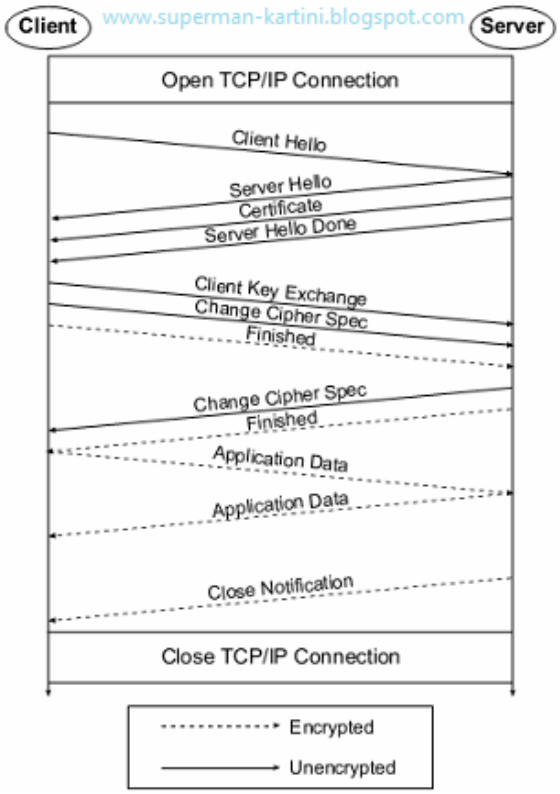
Berikut ini beberapa bagian penting dalam membangun sebuah koneksi SSL dan berkomunikasi menggunakan koneksi tersebut.

Alert

Salah satu komponen terpenting dari SSL adalah sistem penanganan errornya. Error pada SSL disebut dengan alert dan merupakan representasi dari kemungkinan serangan-serangan. Alert adalah pesan-pesan yang dikirim melalui saluran komunikasi SSL, dan kadang juga dienkripsi. Spesifikasi SSL menjelaskan ecara mendetil tentang 20 alert yang berbeda dan memberikan petunjuk bagaimana menanganinya ketika alert tersebut diterima, serta kapan waktu yang tepat untuk membangkitkan dan mengirimkannya.

Handshake

Komunikasi SSL diadakan pada sebuah SSL session. SSL ini dibangun menggunakan sebuah proses handshake yang mirip dengan TCP 3-way handshake. Keseluruhan proses handshake, termasuk pembangunan soket TCP/IP, dapat dilihat pada gambar dibawah.



Seperti yang dapat dilihat pada gambar, koneksi TCP/IP dibangun terlebih dahulu, kemudian proses handshake SSL dimulai. Session SSL dimulai ketika client dan serverberkomunikasi menggunakan parameter dan cipher yang telah dinegosiasikan. Session SSL diakhiri ketika kedua pihak selesai mentransmisikan data aplikasi dan memberitahu mesin lainnya bahwa pengiriman data telah selesai.

- *Client Hello* dan operasi kunci public

  Semua sessionpada SSL dimulai dengan sebuah pesan Client Hello. Pesan ini dikirim oleh clien tkepada server yang ingin dituju untuk berkomunikasi. Pesan ini berisi versi SSL dari client, sebuah bilangan acak yang akan digunakan selanjutnya pada penurunan kunci, dan juga sebuah kumpulan cipher suite offer. Offer ini merupakan penanda yang menunjukkan cipher dan algoritma hash yang ingin digunakan oleh client. ada saat membangun koneksi inisial, server memilih sebuah offeryang ingin digunakan, dan menyampaikan kembali offertersebut kepada clientbersama dengan certificatedan sebuah bilangan acak yang dimilikinya. Client kemudian melakukan verifikasi server menggunakan sertifikat dan mengekstraksi kunci publik server. Dengan menggunakan kunci publik, client mengenkripsi rahasia premaster, sebuah nilai acak yang akan digunakan untuk membangkitkan kunci simetri, dan mengirim pesan terenkripsi tersebut kepada server, yang kemudian mendekripsi pesan menggunakan kunci privatnya.

- Penurunan kunci simetri

  Setelah server menerima rahasia premaster dari client, server dan clientsama-sama membangkitkan kunci simetri yang sama menggunakan rahasia premaster dan juga membangkitkan bilangan acak yang telah dipertukarkan sebelumnya menggunakan TLS pseudo-random function (PRF), yang mengekspansi rahasia dan beberapa data menjadi sebuah blok dengan panjang tertentu. Dengan cara ini, yang hanya mengenkripsi rahasia premaster kecil menggunakan kriptografi kunci publik, membatasi kemungkinan mahalnya operasi pada performansi.

- Finish handshake

  Segera setelah kunci dibangkitkan, client dan server bertukar pesan-pesan “change cipher spec” untuk mengindikasikan bahwa mereka telah memiliki

kunci simetri dan komunikasi selanjutnya dapat dilaksanakan menggunakan algoritma simetri yang dipilik pada tahap inisial proses handshake. Pada tahap ini, serve rdan client menggunakan semua pesan-pesan handshakeyang diterima dan dikirim, dan membangkitkan sebuah blok data yang digunakan untuk melakukan verifikasi bahwa handshaketidak terganggu. Data ini, yang dibangkitkan menggunakan TLS PRF, dikirimkan pada pesan handshake terakhir disebut Finish. Jika data pada pesan finishyang dibangkitkan tidak cocok dengan data finishyang dibangkitkan secara lokal, maka akan koneksi diterminasi oleh pihak manapun yang gagal melakukan tes verifikasi.

Session

Ketika sebuah proses handshake selesai, client dan server mulai berkomunikasi dengan menggunakan saluran komunikasi aman yang baru. Setiap pesan di-hash, dienkripsi dan kemudian dikirim. Setiap kali ada kegagalan, baik itu pada proses dekripsi, enkripsi, hash, verifikasi, tau komunikasi, SSL alert akan dikirimkan (menggunakan enkripsi kunci simetri) oleh entitas yang mengalami kegagalan. Kebanyakan alert bersifat atal, dan menyebabkan komunikasi harus dihentikan sesegera mungkin.

Mengakhiri Session

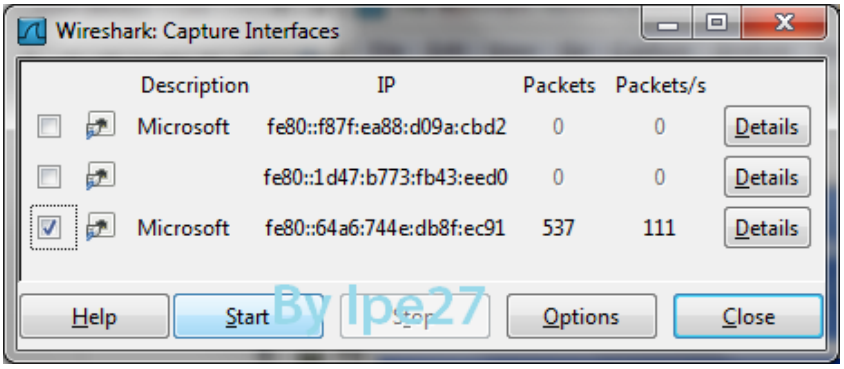
Ketika clientatau server selesai berkomunikasi, sebuah alert khusus, close_notify, dikirimkan untuk memastikan semua komunikasi telah dihentikan dan koneksi dapat ditutup. Alertini digunakan untuk mencegah pihak yang tidak bertanggung jawab melakukan sebuah serangan pemotongan, yang akan menipu serveratau client agar berpikir bahwa semua data yang ingin dipertukarkan telah berhasil terkirim, padahal sebenarnya masih ada data yang masih belum terkirim (hal ini dapat menjadi masalah pada situasi seperti ransaksi perbankan, di mana semua informasi harus terkirim).

Pengamatan SSL Dengan Wireshark

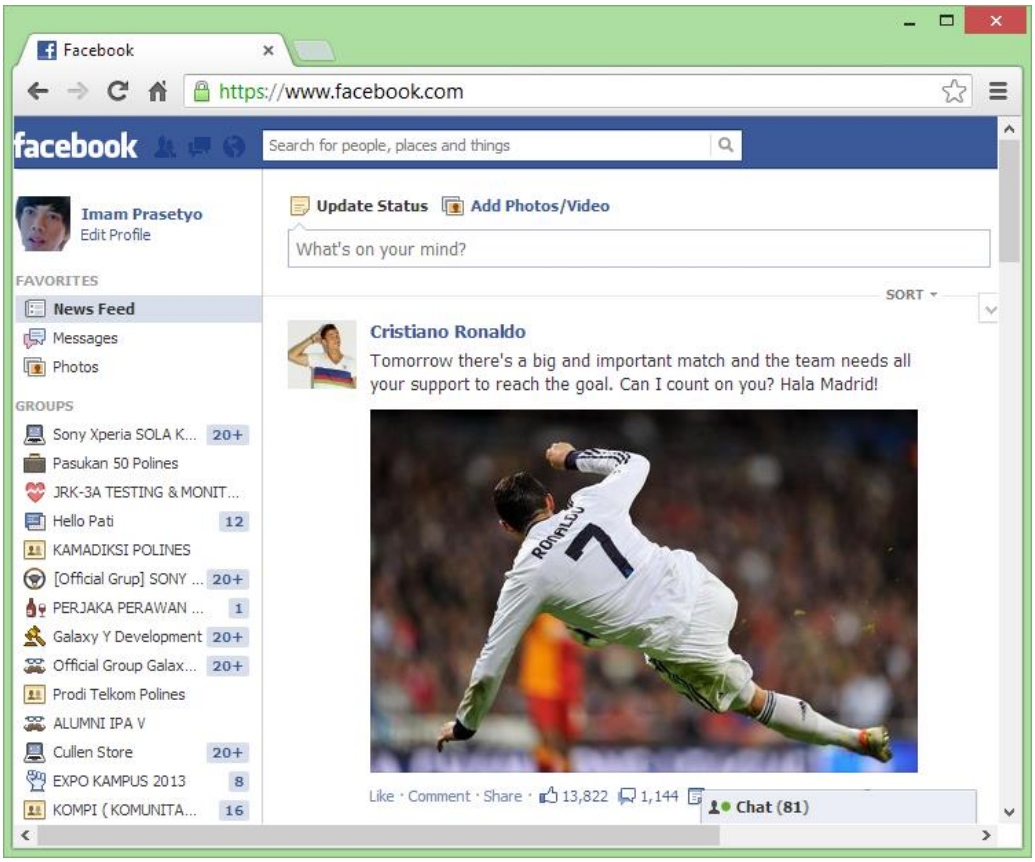
Untuk mengamati SSL transaction bukalah sebuah website seperti internet banking, paypal, ataupun beberapa website yang memakai https. Pada pengamatan yang dilakukan adalah proses akses website https://www.facebook.com/. Lakukan capture lalu lintas paket saat meload website tersebut menggunakan wireshark. Berikut ini detail

langkahnya.

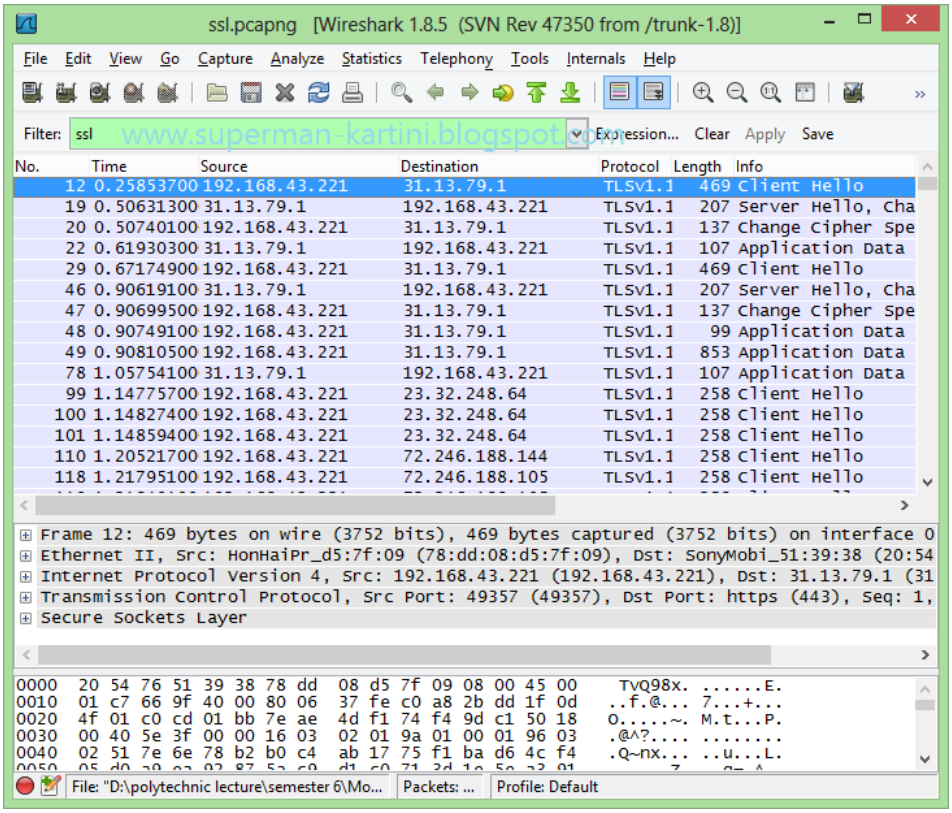
1. Buka wireshark dan start capture pada interface yang digunakan untuk akses internet.



2. Segera buka browser dan akses website https://www.facebook.com/.



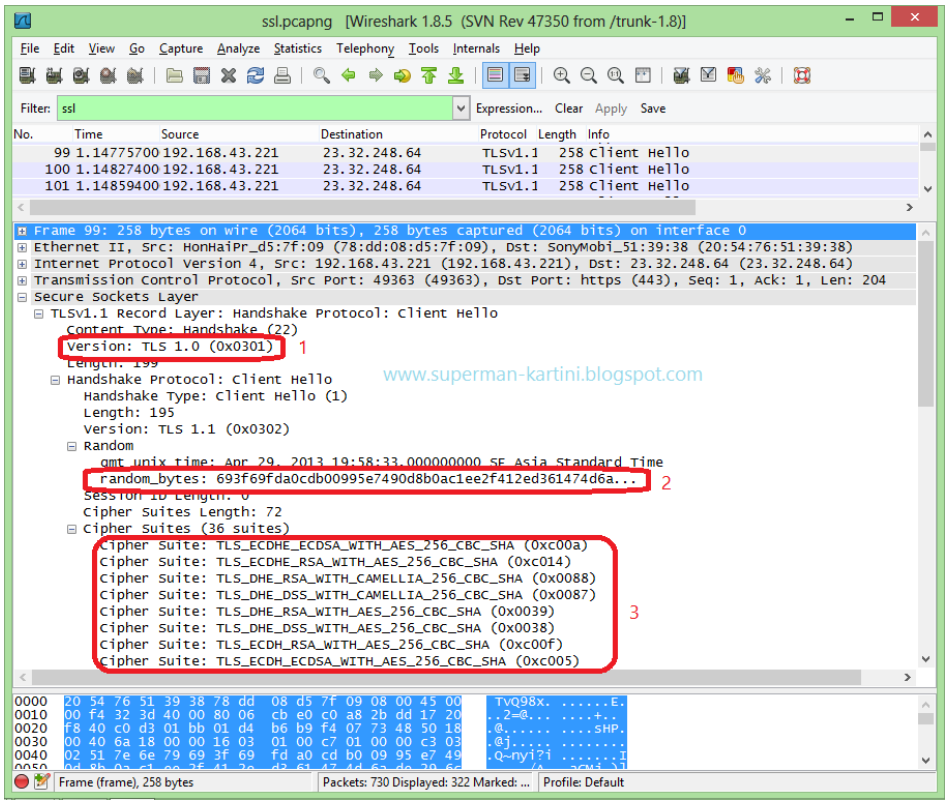
3. Lakukan filtering “ssl” pada wireshark filter.



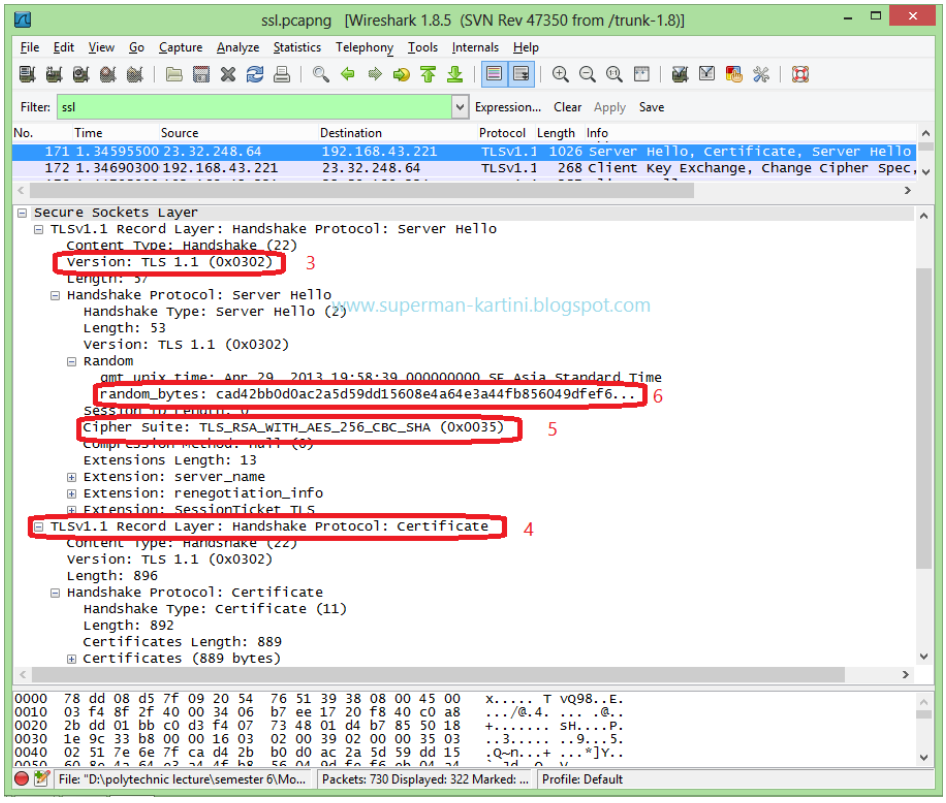
4. Lakukan analisis handshaking yang akan dibahas dibawah.

Analisis SSL Dengan Wireshark

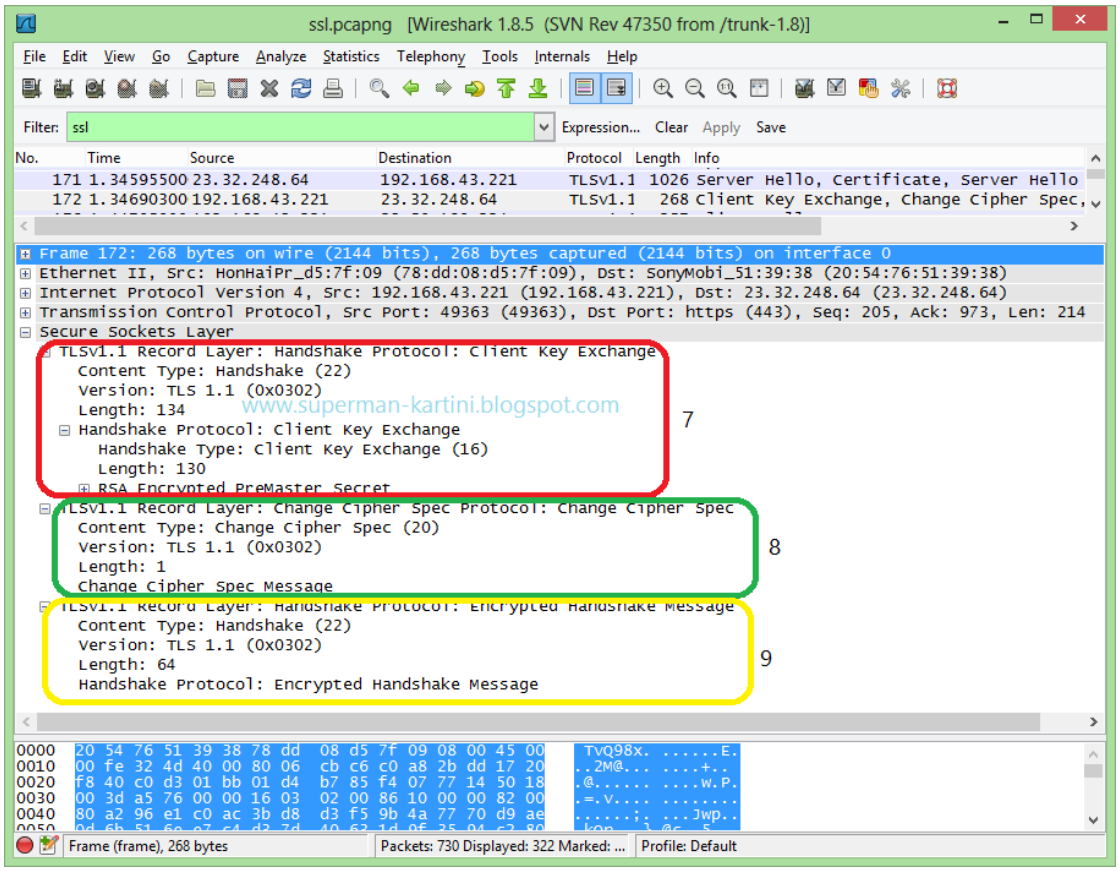
Dapat dilihat bahwa dengan filter "ssl" yang muncul saat mengakses website https://www.facebook.com/ adalah protocol TSLV1.1. Sebagaimana yang dijelaskan diawal bahwa TSLV itu merupakan pengembangan dari SSL. Handshake dimulai saat klien (IP : 192.168.43.221) mengirimkan pesan *Client Hello* kepada server facebook (IP : 23.32.248.64). Pesan ini berisi versi SSL dari client [1], sebuah bilangan acak yang akan digunakan selanjutnya pada penurunan kunci [2], dan juga sebuah daftar cipher suite offer [3].



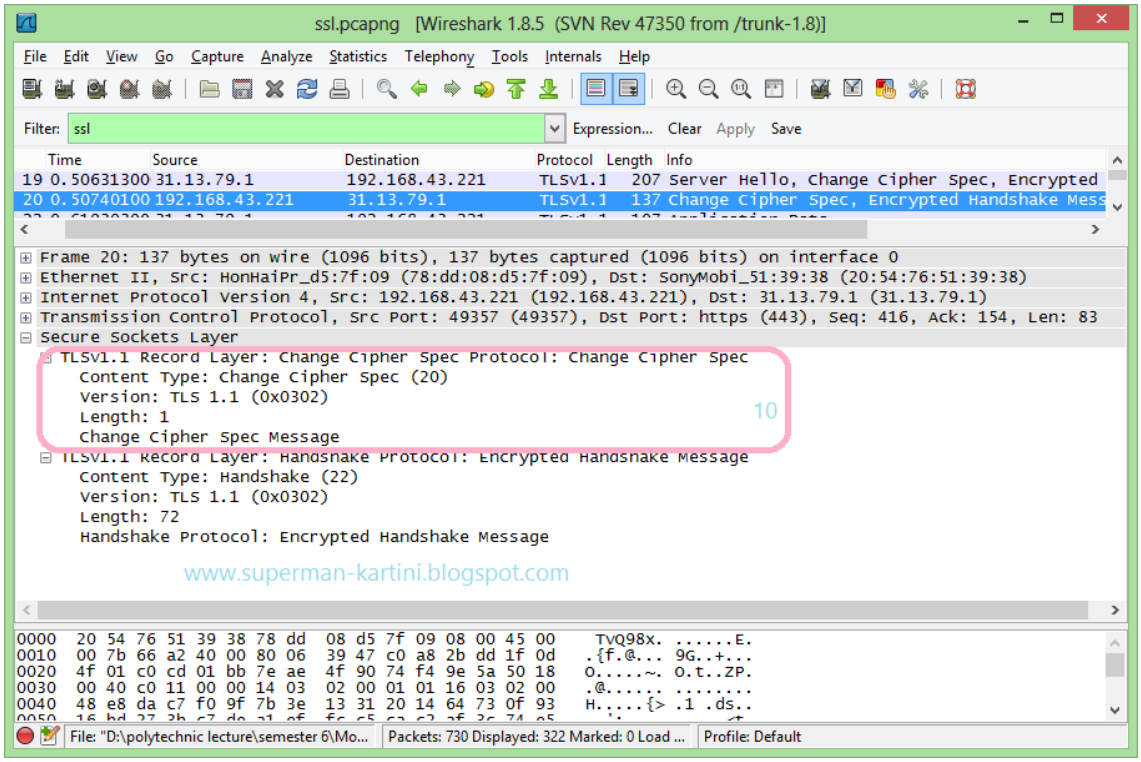
Kemudian server akan mengirimkan balasan berupa pesan *server hello*. Dapat dilihat pesan ini berisi versi SSL yang disepakati [3], certificate [4], Cipher suite terkuat yang dipilih server dari offer cipher klien tadi [5], sebuah bilangan acak yang dimilikinya [6].



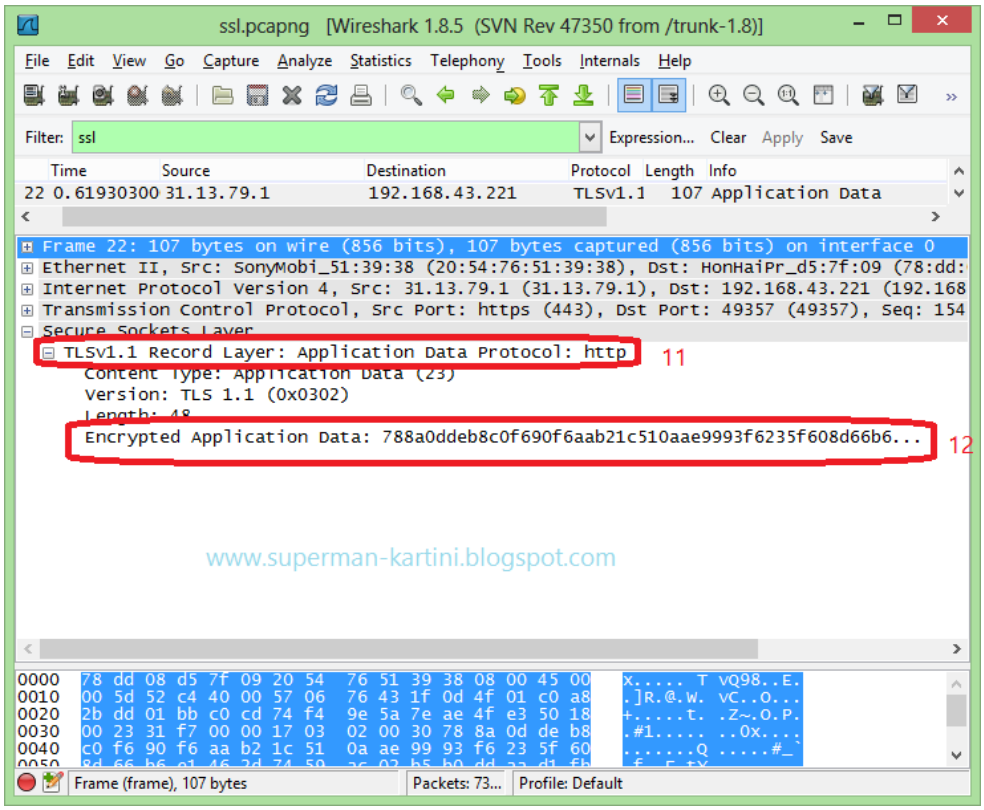
Client kemudian melakukan verifikasi server menggunakan sertifikat dan mengekstraksi kunci publik server melalui pesan client key exchange[7] (Pesan ini akan digunakan untuk membangkitkan kunci simetri, dan mengirim pesan terenkripsi tersebut kepada server) , change cipher spec[8] dan encrypted handshake message [9].



Setelah server menerima rahasia premaster dari client, server dan client sama-sama membangkitkan kunci simetri yang sama menggunakan rahasia premaster dan juga membangkitkan bilangan acak yang telah dipertukarkan sebelumnya menggunakan TLS pseudo-random function (PRF), yang mengekspansi rahasia dan beberapa data menjadi sebuah blok dengan panjang tertentu. Hal tersebut dilakukan dengan pesan change cipher spec dari server [10].



Setelah proses jabat tangan selesai dilakukan application data protocol [11] dapat dilakukan dengan jalur yang aman karena terenkripsi[12]



Biografi Penulis

Imam Prasetyo. Kuliah D4 Teknik Telekomunikasi di Politeknik Negeri Semarang. Lulusan SMA Negeri 1 Pati tahun 2010 dan SMP Negeri 1 Pati tahun 2007. Dari kecil sangat tertarik pada ilmu pengetahuan alam dan teknologi. Untuk informasi maupun tulisan menarik lain dapat diakses di situs blog *http://www.superman-kartini.blogspot.com*