ISSN 2355-3286

Ancaman Keamanan pada Transport Layer Security

Muhamad Fadhli[1], Fityan Ali Munshi[2], Taufik Adi Wicaksono[3]
Jurusan Sistem Komputer, Universitas Komputer Indonesia (UNIKOM), Bandung, Indonesia
[1]moehfadhli@gmail.com, [2]fityanalimunshi@gmail.com, [3]dixblackprince93@yahoo.co.uk

Diterima 14 Desember 2015
Disetujui 30 Desember 2015

***Abstract*— Secure Socket Layer (SSL) also known as Transfer Layer Security (TLS) is de facto standard for web security. It provides confidentiality and integrity of information in transit across the public networks using their powerful cipher suites but it still contains some loopholes or flaws in its foundation. In this paper we discuss TLS standard along with various attacks found in recent years, such as BEAST, CRIME, BREACH, Lucky 13, and their proposed mitigation.**

***Index Terms*— Attack, Compression, Mitigation, Security, TLS.**

I. PENDAHULUAN

Dalam kehidupan sehari-hari, banyak kegiatan di internet seperti belanja *online*, transfer bank, dan lain sebagainya yang memerlukan masukan data bersifat rahasia. Transfer data dilakukan melalui jaringan kabel atau nirkabel. Maka diperlukan mekanisme keamanan yang kuat untuk diimplementasikan pada *transport layer* dari TCP/IP protokol *stack* yang dikenal sebagai *Transport Layer Security* (TLS) atau *Secure Socket Layer* (SSL). Pengguna internet akan merasa aman setiap ada HTTPS bukan HTTP di kolom alamat *web browser*. Tetapi ditemukan juga serangan pada TLS.

Dalam tulisan ini, pertama akan dibahas mekanisme keamanan yang disediakan oleh TLS, kemudian akan dibahas beberapa kerentanan TLS terhadap serangan, dan diakhiri dengan antisipasi yang telah dilakukan terhadap serangan pada TLS.

II. TRANSPORT LAYER SECURITY (TLS)

Secure Socket Layer (SSL), yang kini dikenal sebagai *Transport Layer Security* (TLS), pertama kali dikembangkan oleh Netscape. SSL Versi 1.0 tidak pernah dipublikasikan, sedangkan SSL Versi 2.0 dirilis resmi pada tahun 1995. TLS dikenalkan pada tahun 1999 dan diperbaharui melalui RFC 5246 pada Agustus 2008 dan RFC 6176 pada Maret 2011 [1].

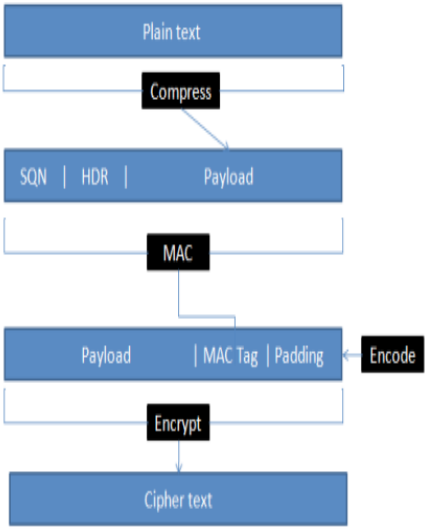
A. Mekanisme Keamanan TLS

TLS adalah kumpulan dari 3 kriptografi, yaitu :

- *Authentication*
- *Confidentiality*
- *Integrity*

Protokol ini terdiri dari berbagai macam *cipher* untuk komunikasi yang aman. *Authentication* diperoleh dengan menggunakan *asimetric cipher* seperti RSA, Diffie-Helman, dan lain-lain. *Confidentiality* diperoleh dengan melakukan enkripsi *simetric* dari *plaintext* melalui transfer jaringan. Secara umum *simetric cipher* yang kuat diimplementasikan di TLS seperti AES, DES-3, RC4, dan sebagainya. *Intergity* diperoleh dengan menghitung *Message Authentication Code*

(MAC) dari paket MD5 atau SHA-1. Secara keseluruhan semua proses ditunjukan pada gambar 1. Hal ini juga dikenal sebagai HEE.

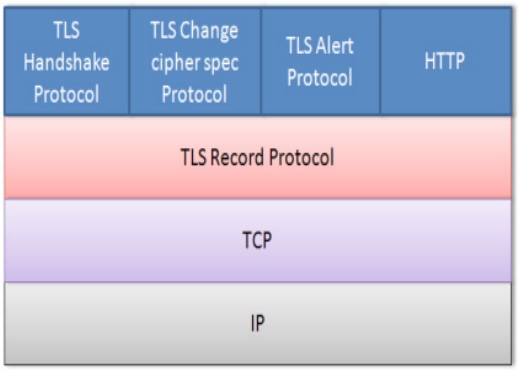


Gambar 1. Mekanisme Keamanan dari TLS

HEE merupakan singkatan *Hash-Encode-Encrypt*. Pertama-tama, *plaintext* dari lapisan aplikasi dikompres oleh HTTP. Algoritma untuk melakukan kompresi adalah Deflate dan GZIP. Teks yang dikompres dari lapisan aplikasi merupakan masukan untuk lapisan protokol TLS. Pada tingkat TLS, teks yang telah dikompres menjadi bagian dari paket data. MAC merupakan kalkulasi dari kelebihan paket dan ditambahkan ke data asli. Setelah itu *encoding* dilakukan. Hal ini merupakan pilihan apabila blok *cipher* diimplementasikan. Setelah *encoding*, enkripsi simetris dilakukan terhadap data atau paket. Proses ini dilakukan pada kedua sisi yaitu *client* dan *server*.

B. TLS *Protocol Stack*

TLS/SSL dirancang agar layanan TCP aman dan handal untuk digunakan oleh pengguna. TLS/SSL bukan protokol tunggal, melainkan protokol yang terdiri dua lapisan seperti yang ilustrasikan pada gambar 2.



Gambar 2. TLS Protocol Stack

The Handshake Protocol : Protokol ini digunakan untuk inisialisasi sesi antara dua pihak. Di dalam pesan pada protokol ini berbagai parameter seperti, algoritma dan kunci yang digunakan untuk enkripsi data dinegosiasikan. Karena adanya protokol ini, maka otentifikasi dan negosiasi parameter yang sesuai antara pihak dapat terlaksana.

The Change Cipher Spec Protocol : Merupakan protokol TLS sederhana dan terdiri dari satu pesan yang mengandung nilai 1. Tujuan utama dari pesan ini adalah untuk menyebabkan keadaan sesi yang awalnya tertunda menjadi keadaan tetap, misalnya dalam mendefinisikan set penggunaan protokol. Pesan ini harus dikirim oleh klien ke *server* dan sebaliknya. Setelah pertukaran pesan, keadaan sesi dianggap setuju. Pesan ini dan pesan TLS lainnya dikirim menggunakan catatan protokol TLS.

The Alert Protokol : digunakan untuk menyampaikan pesan sesi yang berhubungan dengan pertukaran data dan fungsi protokol. Setiap pesan dalam protokol peringatan terdiri dari dua *byte*. *Byte* pertama selalu berisi nilai "peringatan" (1) dan *byte* ke-2 berisi nilai "fatal" (2), yang menentukan tingkat keparahan dari pesan yang dikirim.

The Record Protocol : ini merupakan inti dari TLS. Protokol ini memberikan *Confidentiality* dan *Integrity* oleh enkripsi MAC.

III. SERANGAN-SERANGAN TERHADAP TLS/SSL

Selama beberapa tahun terakhir, ada beberapa serangan terhadap TLS/SSL, antara lain :

A. BEAST

BEAST merupakan singkatan dari *Browser Exploit Against* SSL/TLS [2]. BEAST pertama kali diperkenalkan oleh Thai Doung dan Juliano Rizzo pada konferensi keamanan Ekoparty bulan September 2011. Mereka mendemonstrasikan serangan terhadap Paypal dengan cara mengeksploitasi kelemahan TLS/SSL. TLS menggunakan 2 mekanisme untuk keamanan. Pertama adalah *Initialization Vectors* (IV) dan mode *Cipher Block Chaining* (CBC). BEAST mengeksploitasi mekanisme CBC. Pada mode CBC, IV dan *plaintext* merupakan dua masukan. Tetapi untuk setiap blok pada CBC, teks *cipher* dari blok sebelumnya merupakan IV untuk blok berikutnya. Berdasarkan hal ini, maka penyerang dapat memperkirakan masukan untuk enkripsi selanjutnya dengan menerapkan MITM (*Man In The Middle Attack*). IV akan mengacak aliran enkripsi. Hal-hal yang diperlukan agar serangan ini berhasil adalah :

- *Passive Network Eavesdropping*
- *Chosen boundry Format Privilege*
- *Chosen block wise plaintext injection*

B. CRIME

CRIME merupakan singkatan dari *Compression Info-Leak Mass Exploitation* [3]. CRIME akan menampilkan *plaintext* dengan menggunakan informasi dari kompresi TLS. Serangan ini menarget HTTP *header* dan informasi yang bergantung terhadap HTTP *header* seperti *cookies, Session ID*, dan lain sebagainya. Hal ini dapat berujung kepada penyerang mengambil alih sesi pada koneksi *web-browser*. Hal yang diperlukan untuk melakukan serangan ini adalah aktifnya kompresi TLS pada *web-server* dan MITM (*Man In The Middle Attack*) untuk mengambil data dari *web-browser*.

C. BREACH

BREACH merupakan singkatan dari *Browser Reconnaissance and Ex-filtration via Adaptive Compression of Hypertext* [4]. BREACH menyerang respon HTTP dari aplikasi *web* pada kanal kompresi. BREACH diperkenalkan pada konferensi BlackHat USA 2013 oleh Yoel Gluck, Neal Harris dan Angelo Prado. Seperti halnya CRIME, BREACH mengeksploitasi kombinasi dari kompresi dan enkripsi yang digunakan oleh pengguna dan *web-server*. Mekanisme kerja BREACH hampir sama dengan CRIME, dengan perbedaan CRIME menarget kompresi TLS sedangkan BREACH menarget kompresi HTTP. Tipe serangan BREACH akan memberikan informasi kepada penyerang tentang ukuran sesi *cookies*, dan penyerang dapat memasukkan *plaintext* yang dipilihnya ke permintaan HTTP pengguna.

CRIME dan BREACH sama-sama menargetkan serangan pada kompres. CRIME menargetkan kompresi TLS, sedangkan BREACH menargetkan kompresi HTTP. Kompresi respon HTTP hanya melakukan kompresi pada *body of responses* tetapi tidak informasi *header*. Algoritma yang digunakan pada kompresi HTTP terdiri dari dua komponen, LZ77 dan kode Huffman. LZ77 mengganti tiga atau lebih karakter menjadi nilai "*pointer*" untuk mengurangi ukuran data. Kode Huffman menggantikan karakter menjadi simbol dengan tujuan meng-optimalisasi ukuran data. BREACH bekerja dengan menyerang kompresi LZ77 dan meminimalisasi efek dari kode Huffman.

Dikarenakan BREACH memfokuskan kepada kompresi HTTP, maka memungkinkan untuk memanggil semua versi SSL/TLS dan tidak memerlukan kompresi pada lapisan TLS. Ada 3 hal yang diperlukan untuk BREACH, yaitu :

- Aplikasi harus mendukung kompresi HTTP.

- Respon harus mencerminkan masukan penggunaan.
- Respon harus mempunyai informasi rahasia yang tertanam padanya.

D. Lucky 13

Serangan *Lucky* 13 pertama kali dilaporkan pada bulan Februari 2013 oleh pengembangnya, Nadhem J. AlFardan dan Kenneth G. Paterson, dari *Information Security Group* di Universitas London [5]. *Lucky* 13 merupakan serangan kriptografi yang berdasarkan waktu terhadap penerapan protokol TLS. Secara umum sebuah *Message Authentication Code* (MAC) digunakan untuk meng-autentifikasi dan menyediakan integritas dari sebuah pesan. Cara yang paling baik adalah dengan meng-enkripsi pesan terlebih dahulu dan menerapkan MAC pada *plaintext*. Tetapi pada TLS, dilakukan hal yang berbeda. Pesan dimasukkan kedalam blok, sebuah MAC diterapkan pada *plaintext* dan lapisan hingga 255 *bytes* ditambahkan untuk membuat ukuran pesan sesuai dengan besar blok *cipher* (8 atau 16 *byte*). Akhirnya blok pesan ini akan di-enkripsi menggunakan mode CBC. Setelah di-dekripsi, lapisan blok akan divalidasi. Apabila proses validasi berhasil, maka integritas data yang ada didalamnya akan di-cek terhadap MAC.

Namun metode enkripsi menggunakan mode CBC pada TLS memiliki masalah pada perlindungan terhadap lapisan yang ditambahkan blok pesan. Lapisan ini tidak dilindungi oleh MAC, hanya *plaintext* yang dilindungi oleh MAC. Selama proses dekripsi, lapisan ini akan di-cek terlebih dahulu. Apabila lapisan ini valid, maka MAC akan di-cek setelahnya. Ketika lapisan ini tidak valid, maka *server* akan mengirimkan pesan kesalahan. Penyerang mengubah-ubah pesan yang ter-enkripsi berdasarkan pesan kesalahan ini. Dan setelah beberapa permintaan yang terus-menerus, maka penyerang dapat memperoleh pesan yang telah di-dekripsi tanpa memerlukan kunci enkripsi.

E. POODLE

POODLE (*Padding Oracle On Downgraded Legacy Encryption*) merupakan serangan yang akan mengakibatkan sistem keamanan pengguna yang sebelumnya adalah TLS versi 1.0 atau TLS versi 1.2 akan turun menjadi SSL versi 3.0. Kerentanan TLS terhadap serangan ini pertama kali ditemukan oleh Bodo Moller, Thai Duong dan Krzysztof Kotowicz pada September 2014 [6]. Kelemahan TLS terhadap serangan ini muncul karena tidak adanya kesamaan versi TLS/SSL yang diterapkan diantara sisi *server* dan *client*. Serangan ini berbahaya dikarenakan ketika sistem keamanan turun menjadi SSL versi 3.0, lalu lintas paket akan sangat mudah untuk ditangkap dan dianalisa oleh penyerang. Pada sistem SSL 3.0, enkripsi yang digunakan adalah RC4 *stream cipher* atau blok *cipher* pada mode CBC. Maka ketika ada data yang telah di-enkripsi menggunakan RC4 disebarkan pada jaringan, kemungkinan kebocoran data akan semakin besar.

F. FREAK

FREAK (*Factoring RSA Export Keys*) ditemukan oleh Karthikeyan Bhargavan di INRIA Paris dan tim peneliti dari miTLS pada 3 Maret 2015 [7]. Serangan FREAK terjadi ketika *web-browser* yang lemah terkoneksi ke *web-server* yang mengimplementasikan enkripsi kelas "*export*" dari Amerika Serikat. Enkripsi yang termasuk ke dalam kelas "*export*" dari Amerika Serikat hanya memiliki kunci RSA sebesar 512 bits. Hal ini bertujuan agar NSA (*National Security Agency*) dapat dengan mudah membongkar enkripsi tersebut, tetapi organisasi lain yang tidak memiliki sumber daya komputer yang mumpuni tidak dapat melakukannya. Namun, ketika harga komputer yang memiliki kecepatan tinggi sudah mulai terjangkau, enkripsi yang menggunakan kunci RSA sebesar 512 bits dapat dengan mudah dibongkar.

G. *Logjam*

Logjam merupakan celah kemanan pada TLS yang secara khusus menyerang pertukaran kunci Diffie-Hellman mulai dari kunci sebesar 512 bits sampai dengan 1024 bits [8]. Pertukaran kunci Diffie-Hellman merupakan algoritma kriptografi yang populer dan digunakan hampir oleh semua protokol, termasuk HTTPS, SSH, IPsec, SMTP, dan protokol lainnya yang bergantung kepada TLS. Algoritma ini akan mengizinkan protokol-protokol Internet menyetujui kunci-kunci yang akan digunakan bersama-sama untuk memulai koneksi yang aman.

Serangan *Logjam* akan mengakibatkan versi TLS yang digunakan pada koneksi aman untuk turun menjadi kelas kriptografi yang memiliki kunci sebesar 512 bit. Hal ini hampir sama dengan serangan FREAK, dengan perbedaan, pada FREAK yang diserang adalah pertukaran kunci RSA sedangkan pada *Logjam* yang diserang adalah pertukaran kunci Diffie-Hellman. *Logjam* akan memiliki efek pada *server* yang mendukung DHE_EXPORT *cipher*, dan akan mempengaruhi *web-browser* [9].

IV. ANTISIPASI SERANGAN PADA TLS/ SSL

Untuk mengantisipasi BEAST, pada sisi *server* digunakan RC4 (*Rivest Cipher* 4). Tetapi pada tahun 2013, peneliti menemukan kelemahan pada RC4, sehingga RC4 tidak lagi digunakan pada *server* [10]. Untuk menahan serangan pada sisi *web browser*, Mozilla meng-update NSS (*Network Security Service*) mereka. NSS digunakan oleh Mozilla Firefox dan Google Chrome. Untuk pengguna *web browser* Internet Explorer, pada buletin Microsoft tanggal 10 Januari 2012 [11], Microsoft mengubah cara komponen *Windows Secure Channel* (*Schannel*) mengirimkan paket enkripsi dari sisi *server*. Sedangkan untuk pengguna *web browser* Safari pada MacOS, pada tanggal 22 Oktober 2013 Apple mengimplementasikan pembagian 1/n-1 dan menjadikannya *default* pada OS X Mavericks [12].

Untuk CRIME dan BREACH, dikarenakan kedua jenis serangan ini menyerang kompresi, maka untuk mengantisipasinya adalah dengan menon-aktifkan level kompresi TLS pada *web browser*. Hal ini tidak terlalu mengganggu performa dari *web browser* sehingga *server* TLS tidak rentan terhadap serangan ini.

Sedangkan antisipasi serangan *Lucky 13* adalah dengan menggunakan mode enkripsi AES-GCM [13]. Dengan menggunakan mode enkripsi ini, selain dapat menangkal *Lucky 13*, dapat juga menangkal BEAST.

Untuk mengantisipasi terjadiya serangan POODLE, salah satu cara adalah dengan menon-aktifkan SSL versi 3.0 pada sisi *client* dan *server*. Pengimplementasian TLS_FALLBACK_SCSV [6] pada *browser* akan mencegah terjadinya serangan POODLE.

Antisipasi terhadap FREAK, dari sisi *client*, dapat dilakukan dengan memperbaharui *web-browser* yang digunakan. Dari sisi *server*, diharuskan untuk menonaktifkan dukungan terhadap TLS *export cipher* [14].

Untuk mengantisipasi serangan *Logjam*, dari sisi *server* harus mematikan dukungan terhadap *export cipher* dan menggunakan kunci Diffie-Hellman sebesar 2048 bit. Dan dari sisi *client*, selalu memperbaharui versi *web-browser* yang digunakan [9].

V. SIMPULAN

Untuk keamanan *online*, pada saat ini belum ada pilihan lain selain TLS. Tetapi TLS memiliki banyak kelemahan, dan beberapa serangan pada TLS dirancang secara khusus terhadap kelemahan-kelemahan tersebut. Maka perlu adanya perubahan yang penting pada implementasi TLS, baik pada sisi *server* ataupun sisi *web browser*, dan fondasi TLS itu sendiri. Dari sisi *client*, selalu memperbaharui *web-browser* yang digunakan merupakan salah satu metode pencegahan serangan-serangan terhadap

TLS. Perubahan pada fondasi TLS dapat dengan melakukan desain ulang terhadap struktur TLS [15].

DAFTAR PUSTAKA

[1] Freier, P. Karlton, P. Kocher (Agustus 2011). "The Secure Sockets Layer (SSL) Protocol Version 3.0". Alamat situs : https://www.tools.ietf.org/html/rfc6101.

[2] Duong, Thai, and Juliano Rizzo. "Here come the□ Ninjas." belum terbit.

[3] Rizzo, Juliano, and Thai Duong. "The CRIME attack". ekoparty Security Conference. Vol. 8. 2012.

[4] Gluck, Yoel, Neal Harris, And Angelo Ángel Prado. "BREACH: Reviving The Crime Attack". (2013).

[5] AlFardan, Nadhem J., and Kenneth G. Paterson. "Lucky thirteen: Breaking the TLS and DTLS record protocols. "IEEE Symposium on Security and Privacy. 2013.

[6] Möller, Bodo; Duong, Thai; Kotowicz, Krzysztof (September 2014). "This POODLE Bites: Exploiting The SSL 3.0 Fallback".

[7] B. Beurdouche & al (18 Mei 2015). "A Messy State of the Union: Taming the Composite State Machines of TLS". IEEE Security and Privacy 2015.

[8] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann 22nd ACM Conference on Computer and Communications Security (CCS '15), Denver, CO, Oktober 2015.

[9] "Weak Diffie-Hellman and the Logjam Attack". Alamat situs : https://www.weakdh.org/

[10] Ristic, Ivan (10 September 2013). "Is BEAST Still a Threat?". Alamat situs : https://www.community.qualys.com/blogs/securitylabs/2013/09/10/is-beast-still-a-threat

[11] "Vulnerability in SSL/TLS Could Allow Information Disclosure (2643584)". Buletin Microsoft (10 Januari 2012). Alamat situs : https://www.technet.microsoft.com/library/security/ms12-006

[12] Ristic, Ivan (Oct 31, 2013). "Apple Enabled BEAST Mitigations in OS X 10.9 Mavericks". Alamat situs : https://www.community.qualys.com/blogs/securitylabs/2013/10/31/apple-enabled-beast-mitigations-in-os-x-109-mavericks

[13] P. Gutmann (September 2014). "Encrypt-then-MAC for Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)". Alamat situs : https://www.tools.ietf.org/html/rfc7366

[14] "Tracking the FREAK Attack". Alamat situs : https://www.freakattack.com/

[15] Corella, Francisco, and Karen Lewison. "It Is Time to Redesign Transport Layer Security". (2013).