KEAMANAN HTTP DAN HTTPS BERBASIS WEB MENGGUNAKAN SISTEM OPERASI KALI LINUX

Adzan Abdul Zabar[1], Fahmi Novianto[2]

Program Studi Teknik Komputer – FTIK
Universitas Komputer Indonesia
Jln. Dipatiukur 122 Call. 022-2504119 Fax. 022-2533754
E-mail: dzanity@gmail.com[1], noviantofahmi@gmail.com[2]

ABSTRAK

Pada awal perkembangannya jaringan komputer hanya digunakan untuk pengiriman *e-mail* antar perguruan tinggi untuk keperluan riset dan untuk berbagi penggunaan *printer* dalam suatu perusahaan. Untuk memenuhi tujuan tersebut, aspek keamanan jaringan pada saat itu tidak mendapat perhatian penting. Seiring dengan perkembangan, jaringan komputer telah digunakan sejak lama untuk hal-hal yang lebih kompleks seperti untuk perbankan, untuk perdagangan dan masih banyak lainnya. Dan semua itu menggunakan media Internet. Aspek keamanan dalam komunikasi melalui jaringan komputer menjadi semakin penting terutama karena banyaknya aktivitas pertukaran informasi rahasia melalui Internet. Untuk menghindari penyadapan atau tindak kejahatan lainnya, maka dibuatlah jurnal ini untuk membahas perbandingan keamanan antar HTTP dan HTTPS.

1. PENDAHULUAN

Banyaknya pertukaran informasi, transaksi dan tingginya aktifitas yang dilakukan pada jaringan Internet seperti menyimpan data-data baik data pribadi, data rahasia, ataupun informasi penting lainnya terkadang membuat pengguna Internet lupa bahwa semua informasi yang ada di Internet sebenarnya bersifat umum atau terbuka. Hal ini menjadikan Internet sebagai suatu sarana media yang rentan terhadap serangan karena banyaknya lubang keamanan yang memungkinkan seseorang dapat mengetahui informasi data.

Oleh karena itu, jurnal ini akan membahas beberapa kelemahan yang ada pada Internet khususnya pada website dengan melakukan perbandingan keamanan antara *http* dan *https*. Banyaknya pengguna yang masih belum menyadari tentang pentingnya keamanan sistem informasi menjadikan hal ini sebagai salah satu alasan penulisan jurnal ini.

2. LANDASAN TEORI

2.1 Internet

Internet adalah suatu jaringan komputer yang saling terhubung untuk keperluan komunikasi dan informasi. Sebuah komputer dalam satu jaringan internet dapat berada di mana saja atau bahkan di seluruh Indonesia. Sering juga internet diartikan sebagai jaringan komputer di seluruh dunia yang berisikan informasi dan sebagai sarana komunikasi data yang berupa suara, gambar, video dan juga teks. Informasi ini dibuat oleh penyelenggara atau pemilik jaringan komputer atau dibuat pemilik informasi yang menitipkan informasinya kepada penyedia layanan internet.

Sedangkan pengertian internet menurut jika dilihat dari segi ilmu pengetahuan, internet adalah sebuah perpustakaan besar yang didalamnya terdapat jutaan (bahkan milyaran) informasi atau data yang dapat berupa teks, grafik, audio maupun animasi dan lain lain dalam bentuk media elektronik [1].

2.2 Website

Website atau situs merupakan sebuah kumpulan halaman-halaman web beserta file-file pendukungnya yang mnampilkan informasi seperti file gambar, video, dan file digital lainnya yang disimpan pada sebuah web server yang umumnya dapat diakses melalui internet. Atau dengan kata lain, website adalah sekumpulan folder dan file yang mengandung banyak perintah dan fungsi fungsi tertentu, seperti fungsi tampilan, fungsi menangani penyimpanan data, dsb [2].

2.3 Hypertext Transfer Protokol (HTTP)

HTTP adalah sebuah protokol meminta atau menjawab antara client dan server. Sebuh client HTTP seperti web browser, biasanya memulai permintaan dengan membuat hubungan TCP/IP ke port tertentu di tuan rumah yang jauh (biasanya port 80). Sebuah server HTTP yang mendengarkan di port tersebut menunggu client mengirim kode permintaan (request), seperti "GET / HTTP/1.1" (yang akan meminta halaman yang sudah ditentukan), diikuti dengan pesan MIME yang memiliki beberapa informasi kode kepala yang menjelaskan aspek dari permintaan tersebut, diikut dengan badan dari data tertentu. Beberapa kepala (header) juga bebas ditulis atau tidak, sementara lainnya (seperti tuan rumah) diperlukan oleh protokol HTTP/1.1. Begitu menerima kode permintaan (dan pesan, bila ada), server mengirim kembali kode jawaban, seperti "200 OK", dan sebuah pesan yang diminta, atau sebuah pesan error atau pesan lainnya. Pengembangan HTTP dikoordinasi oleh *Konsorsium World Wide Web* (W3C) dan grup bekerja Internet Engineering Task Force (IETF), bekerja dalam publikasi satu seri RFC, yang paling terkenal RFC 2616, yang menjelaskan HTTP/1.1, versi HTTP yang digunakan umum sekarang ini [3].

2.4 Hypertext Transfer Protocol Secure (HTTPS)

Hypertext Transfer Protocol Secure memiliki pengertian yang sama dengan *http* hanya saja *https* memiliki kelebihan fungsi di bidang keamanan (*secure*). Dengan menggunakan *Secure Socket Layer* (SSL) atau *Transport Layer Security* (TLS) sebagai sublayer di bawah *http* aplikasi *layer* yang biasa. Teknologi *https* protokol mencegah kemungkinan “dicurinya” informasi penting yang dikirimkan selama proses komunikasi berlangsung antara *user* dengan *web server* atau sebaliknya.

Secara teknis, *website* yang menggunakan *https* akan melakukan enkripsi terhadap informasi (data) menggunakan teknik enkripsi SSL. Dengan cara ini meskipun seseorang berhasil “mencuri” data tersebut selama dalam perjalanan *user web server*, orang tersebut tidak akan bisa membacanya karena sudah diubah oleh teknik enkripsi SSL. Umumnya *website* yang menggunakan *https* ini adalah *website* yang memiliki tingkat kerawanan tinggi yang berhubungan dengan masalah keuangan dan privasi dari pelanggannya seperti *website* perbankan dan investasi.

HTTPS dienkripsi dan deskripsi dari halaman yang di minta oleh pengguna dan halaman yang di kembalikan oleh web *server*. Kedua protokol tersebut memberikan perlindungan yang memadai dari serangan *eavesdroppers*, dan *man in the middle attacks*. Pada umumnya *port* yang digunakan HTTPS adalah *port 443*. Tingkat keamanan tergantung pada ketepatan dalam mengimplementasikan pada *browser web* dan perangkat lunak *server* dan didukung oleh algoritma penyandian yang aktual. Oleh karena itu, pada halaman *web* digunakan HTTPS, dan URL yang digunakan dimulai dengan https:// [2].

2.5 Cara Kerja HTTP

Https bukan protokol yang terpisah, tetapi mengacu pada kombinasi dari interaksi HTTP normal melalui *Socket Layer* terenkripsi SSL (*Secure*) atau *Transport Layer Security* (TLS) mekanisme transportasi. Hal ini menjamin perlindungan yang wajar dari penyadapan dan serangan. *Port default* TCP *https*: URL adalah 443. Untuk mempersiapkan web-*server* terkoneksi dengan *https* penerima harus menjadi *administrator* dan membuat sertifikat kunci publik untuk *server web*. Sertifikat ini dapat dibuat untuk *server* berbasis Linux dengan alat seperti open SSL. Sertifikat ini harus ditandatangani oleh otoritas sertifikat satu bentuk atau lain, yang menyatakan bahwa pemegang sertifikat adalah siapa yang mereka ajukan. *Web browser* pada umumnya didistribusikan dengan penandatanganan sertifikat otoritas sertifikat utama, sehingga mereka dapat memverifikasi sertifikat yang ditandatangani oleh mereka.

Bila menggunakan koneksi *https*, *server* merespon koneksi awal dengan menawarkan daftar metode enkripsi mendukung. Sebagai tanggapan, *client* memilih metode sambungan, dan *client* dan sertifikat *server* pertukaran untuk otentikasi identitas mereka. Setelah ini dilakukan, kedua belah pihak bertukar informasi terenkripsi setelah memastikan bahwa kedua menggunakan tombol yang sama, dan koneksi ditutup. Untuk host koneksi *https*, *server* harus memiliki sertifikat kunci publik, yang *embeds* informasi kunci dengan verifikasi identitas pemilik kunci itu. Sertifikat Kebanyakan diverifikasi oleh pihak ketiga sehingga *client* yakin bahwa kuncinya adalah aman [2].

2.6 Secure Socket Layer (SSL)

Secure Socket Layer adalah suatu protokol yang diciptakan oleh *Netscape* untuk memastikan keamanan dalam bertransaksi di internet antara *webserver* dan *browser* dari *client*. Protokol ini menggunakan sebuah badan yang biasa disebut CA (*Certificate Authority*) untuk mengidentifikasi memverifikasi pihak-pihak yang bertransaksi. Secara umum, cara kerja protokol SSL adalah sebagai berikut:

1. *Client* membuka suatu halaman yang mendukung protokol SSL, biasanya diawali dengan "https://" pada *browser*.
2. Web*server* mengirimkan kunci publiknya besertsa dengan sertifikat *server*
3. Browser melakukan pemeriksaan, apakah sertifikat tersebut dikeluarkan oleh CA *(Certificate Authority)* yang terpercaya? Apakah sertifikat terebut masih *valid* dan memang berhubungan dengan alamat situs yang sedang dikunjungi?
4. Setelah diyakini kebenaran dari *web-server* tersebut, kemudian *browser* menggunakan kunci publik dari *web-server* untuk melakukan enkripsi terhadap suatu kunci simetri yang dibangkitkan secara acak dari pihak *client*. Kunci yang dienkripsi kemudian dikirimkan ke *web-server* untuk digunakan sebagai kunci utuk mengenkripsi alamat URL (*Uniform Resource Locator*) dan data http lain yang diperlukan
5. *Web-server* melakukan dekripsi terhadap enkrispi dari *client* tadi, menggunakan kunci *private server*. *Server* kemudian menggunakan kunci simetri dari *client* tersebut untuk mendekripsi URL dan data *http* yang akan diperlukan *client*
6. *Server* mengirimkan kembali halaman dokumen HTML yang diminta *client* dan data *http* yang terenkripsi dengan kunci simetri.
7. Browser melakukan dekripsi data *http* dan dokumen HTML menggunakan kunci simetri dan menampilkan informasi yang diminta [3].

2.7 Transport Layer Security (TLS)

Protokol keamanan dari *Internet Engineering Task Force* (IETF) adalah *Transport Layer Security* sebagai pengganti untuk protokol *SSL v3.0* yang dikembangkan oleh *Netscape*. TLS didefinisikan di dalam suatu request for comment, yaitu pada RFC2246. Banyak protokol pada layer aplikasi yang menggunakan TLS untuk menciptakan koneksi yang aman, antara lain HTTP, IMAP, POP3, dan SMTP [4].

2.8 Kriptografi

Kriptografi adalah ilmu dan praktik menjaga kerahasiaan dari pihak-pihak yang tidak dikehendaki baik saat penyampaian maupun penyimpanan informasi tersebut. Informasi yang hendak dilindungi itu disamarkan dengan menggunakan cara-cara dan kunci tertentu. Kriptografi tidak hanya menjaga kerahasiaan informasi, namun juga menjaga keutuhan dan keaslian informasi yang disampaikan.

Salah satu aplikasi kriptografi di jaringan internet adalah pengamanan situs dengan menggunakan protokol HTTPS ini memungkinkan terjadinya akses dan transaksi melalui situs internet secara aman, misalnya dalam *online banking*, *online shopping*, *login* ke *email host* dan sebagainya. Ketika menggunakan koneksi HTTPS, *server* menanggapi inisiasi koneksi oleh *client* dengan menawarkan berbagai metode enkripsi yang dapat menunjang [5].

3. PEMBAHASAN

3.1 Obyek Penelitian

Obyek penelitian berupa sebuah sistem operasi, program aplikasi dan juga sebuah *device*. Sistem operasi yang akan diteliti adalah *Kali Linux 2.0*, yang merupakan sistem operasi khusus untuk *penetration testing.* Sedangkan aplikasi yang akan dipakai untuk menganalisa hasil percobaan adalah *Wireshark*, sebuah program aplikasi yang digunakan untuk memantau paket data yang terkirim maupun diterima oleh sebuah *device*, yang pada percobaan Wireshark akan digunakan sebagai aplikasi untuk *sniffing* paket data [6]. Kemudian, *device* yang akan dijadikan obyek percobaan adalah sebuah *smartphone Sony Xperia LT25i* berbasis *Android Jellybean 4.3*.

3.2 Langkah Percobaan

Percobaan dilakukan menggunakan laptop dengan system operasi *Kali Linux 2.0 (SANA)* yang terhubung dengan sebuah jaringan kabel *LAN (Local Area Network)*, dimana laptop tersebut akan digunakan sebagai *hot-spot* untuk membagi koneksi internet-nya kepada *device* lain yang terhubung melalui jaringan *wireless* (dalam hal ini *device*

adalah *smartphone android*). Adapun langkah-langkah yang harus dilakukan dalam percobaan ini yaitu:

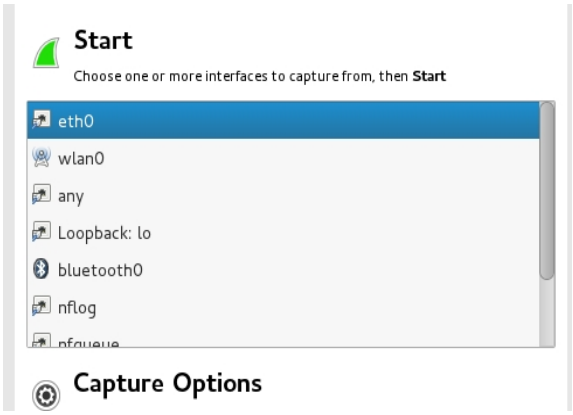
1. Mengaktifkan *wi-fi hotspot* pada laptop.
2. Hubungkan *smartphone* dengan *wifi* menggunakan fasilitas *wireless* yang sudah ada pada *smartphone.*
3. Setelah *smartphone* terhubung dengan *wifi* laptop, jalankan program *Wireshark* yang sudah ter-*install* pada Kali Linux 2.0.
4. Pada program *Wireshark* pilih *interface* yang akan di analisa paket datanya (dalam kasus ini akan dianalisa paket data yang aktif pada *interface* eth0).
5. Setelah *interface* dipilih, dapat langsung mengaktifkan *Wireshark* untuk menganalisa paket data yang aktif (pengiriman dan penerimaan) yang terjadi antara laptop, *smartphone* dan internet (proses ini disebut *sniffing*)
6. Pada saat *smartphone* melakukan aktivitasnya pada jaringan *wireless* yang disediakan oleh laptop dan berinteraksi dengan internet, beberapa aktivitasnya meninggalkan jejak yang dapat dilihat (*sniff*) pada program *Wireshark*
7. Setelah cukup, berikutnya menghentikan program *Wireshark* dan menganalisa hasil *sniffing* yang diperoleh. Akan memakan banyak waktu jika menjalankan program *Wireshark* terlalu lama, oleh karena itu beberapa detik pun cukup untuk menangkap segala aktivitas yang dilakukan *smartphone.*
8. Lakukan analisa terhadap beberapa protokol seperti *HTTP*, *UDP*, dan *TCP*. Untuk protokol tertentu seperti *TLSv1.2* dan *SSL* tidak dapat dianalisa, karena protokol tersebut merupakan protokol *secure*, dimana pada saat mencoba untuk menganalisanya, maka akan mendapatkan karakter acak yang mana itu adalah *enkripsi* pada protokol tersebut.
9. Selesai. Untuk keterangan mengenai percobaan akan dijelaskan pada Hasil Percobaan dan Pembahasan.

3.3 Hasil Pengujian

Setelah mengaktifkan *wireless hotspot* pada *Kali Linux* seperti yang terlihat pada gambar 1, maka proses selanjutnya adalah pemilihan *ethernet* pada perangkat lunak *wireshark*.

Gambar 1. Mengaktifkan *wireless hotspot* pada sistem operasi *kali linux*

Wireshark yang digunakan berfungsi sebagai *interface* untuk memantau paket data, tampilan *interface wireshark* dapat dilihat seperti yang ditunjukkan pada gambar 2 di bawah ini:

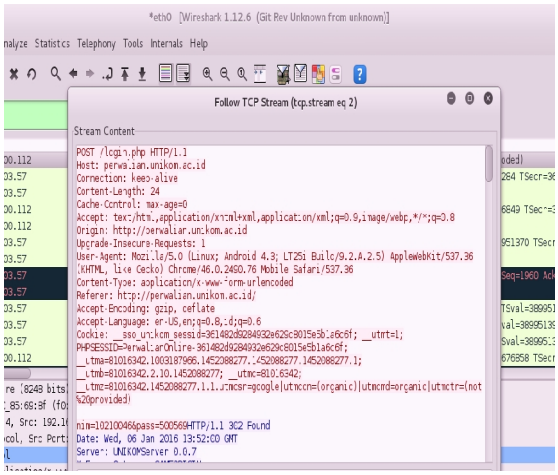


Gambar 2. Tampilan perangkat lunak *Wireshark*

Setelah melakukan sambungan terhadap eth0 pada *wireshark* maka terlihat aktifitas yang dilakukan oleh pengguna lain dalam jaringan yang sama. Melakukan teknik *sniffing* terhadap pengguna yang melakukan aktifitas internet seperti *login* atau memasukan *password* yang bersifat pribadi. Berikut adalah contoh gambar pengguna yang melakukan *login* pada salah satu *website* unikom yang ditunjukkan pada gambar 3.

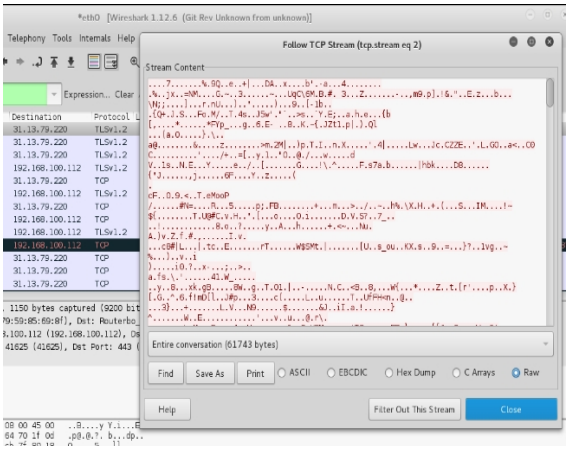
Gambar 3. Tampilan *web* yang hendak dimasuki oleh pengguna lain.

Dan berikut ini adalah hasil *sniffing* dari *web* unikom yang dimasuki oleh pengguna lain. Yang ditunjukan pada gambar 4 seperti berikut:



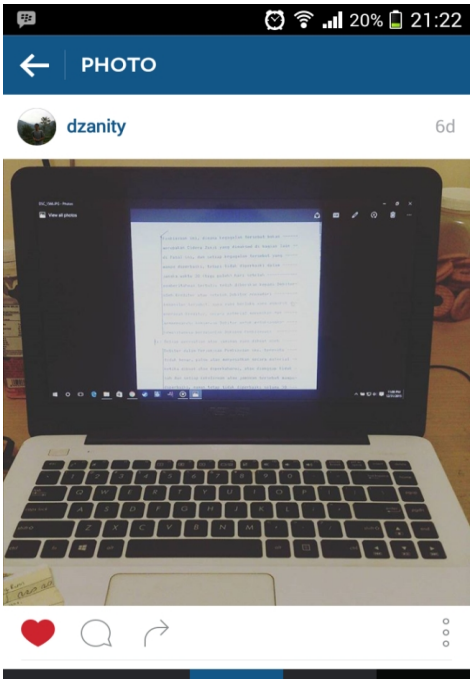
Gambar 4. Nim dan *password* pada *website* yang diakses dapat terlihat.

Selain itu teknik *sniffing* ini juga mampu melihat aktifitas yang dilakukan terhadap situs *web* lainnya baik melalui *http* atau *https* seperti misalnya teknik *sniffing* yang dilakukan di jejaring sosial *facebook*. Pada alamat *https://facebook.com* melalui *browser*. Terlihat aktifitas yang terdapat pada jaringan tersebut. Hal ini dapat terlihat pada gambar 5 berikut.



Gambar 5. Hasil *sniffing* pada *https facebook.com* melalui *browser*, data terenkripsi

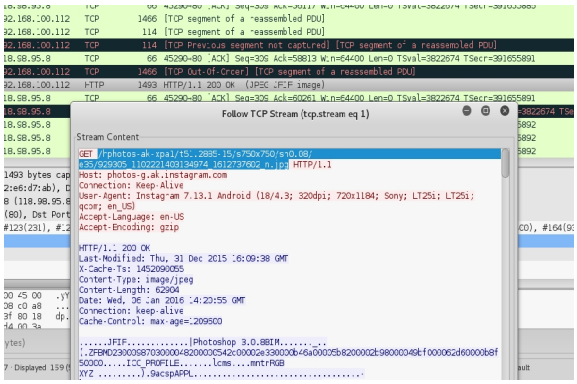
Selain *facebook,* aktifitas lainnya yang dapat terlihat adalah *instagram* sebagaimana terdapat pada gambar 6. berikut ini:



Gambar 6. Pengguna dijaringan LAN yang sama mengunggah gambar laptop pada akun *instagram*-nya.

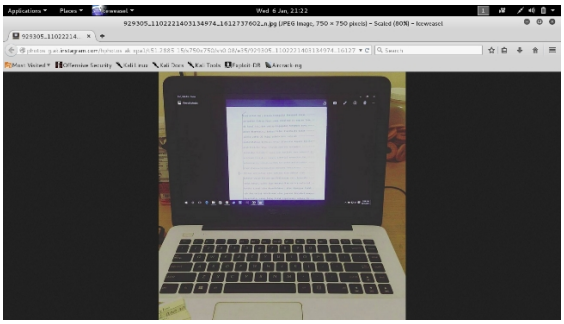
Dalam gambar dapat terlihat aktifitas yang dilakukan oleh pengunggah foto tersebut dengan melihat data yang dikirimkan dan diterima yang melewati jaringan yang telah dibuat sebelumnya. Data yang didapat berupa alamat *web* tempat dimana gambar tersebut diunggah. Di bawah ini merupakan tampilan gambar 7 yang menunjukan

proses *sniffing* pada akun *instagram* pengguna *device* di jaringan yang sama.



Gambar 7. *Sniffing* pada aplikasi Instagram di *device*, dapat terlihat alamat dari gambar

Alamat *web* yang terdapat pada *wireshark* dapat dibuka melalui *browser* dengan menulis kembali alamat *web* tersebut. Gambar 8 ini menunjukan gambar yang telah dibuka pada *browser*.



Gambar 8. Hasil gambar yang diakses oleh *device* dapat di buka melalui *web browser*

4. PENUTUP

4.1 Kesimpulan

Dari hasil percobaan dan penelitian yang dilakukan *http* lebih rentan terhadap *sniffing* dibandingkan dengan *https*, karena *http* tidak menggunakan metode enkripsi dalam pengiriman maupun penerimaan paket data yang dilakukan antara device dengan *server*. Oleh karena itu dikembangkan https guna untuk mengatasi kekurangan tersebut, dengan metode enkripsi yang lebih aman dapat mengurangi serta mencegah serangan *sniffing* oleh *hacker*.

Pada percobaan berikutnya diharapkan dilakukan penelitian terhadap kemanan *https*, karena sejauh ini protocol *https* adalah yang paling aman walaupun sebenarnya masih ada cara untuk menembus protokol tersebut dengan men-*downgrade* https menjadi *http*.

4.2 Saran

Untuk para pengguna internet, dari hasil percobaan ini diharapkan dapat lebih berhati-hati dalam menggunakan *wifi* sebagai koneksi bebas yang disediakan pada suatu tempat, karena bisa jadi semua aktivitas yang dilakukan dipantau oleh *hacker* dan terjadi hal yang tidak diinginkan, seperti pencurian akun, penyebaran gambar, informasi, dll.

DAFTAR PUSTAKA

[1]. Riska, Hari H, Agustin N, *Studi Tentang Penggunaan Internet Oleh Pelajar*, http://ejournal.sos.fisip-unmul.ac.id/site/?p=614, 25 Januari 2016 13.52

[2]. Hamzah H, *Pengertian Website Dan Fungsinya*, http://ilmuti.org/wp-content/uploads/2014/03/HamzahHartono_Pengertian_WEBSITE_Dan_Fungsinya.pdf, 25 Januari 2016 14.11

[3]. Ferdian Pramudya P, Agung Kaharesa W, *Protocol HTTP Dan Handshaking Client-Server Untuk Berkomunikasi via HTTPS*, http://blog.binadarma.ac.id/suryayusra/wp-content/uploads/2011/10/http-dan-handshake-via-https-.pdf, 25 Januari 2016 2.30

[4]. Hary F, *Studi dan Implementasi Sistem Keamanan Berbasis Web dengan Protokol SSL di Server Students Informatika ITB*, http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2009-2010/Makalah2/Makalah2_IF3058_2010_037.pdf, 25 Januari 2016 08.00

[5]. Ernestasia S, *Aplikasi Kriptografi pada Secure,* Socket Layer (SSL), http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2007-2008/Makalah/MakalahIF2153-0708-057.pdf, 25 Januari 2016 08.00

[6]. F Octavian, Kali Linux 300% Attack Computer Book, Jasakom, 2015