

MODUL 3

KRIPTOGRAFI PADA WWW MENGGUNAKAN SSL

3.1 Tujuan Praktikum

1. Mengetahui konsep dari protocol otentikasi TLS
2. Memahami konsep dari protocol otentikasi SSL dan dapat mengimplementasikan pada wujud otentikasi client dan server pada Operating System Ubuntu

3.2 Alat Praktikum

Virtual Box, Operating System Ubuntu

3.3 Dasar Teori

3.3.1 TLS: Handshake & Record Protocol

Merupakan protokol keamanan dari *Internet Engineering Task Force* (IETF) sebagai pengganti protokol SSL v3.0 yang dikembangkan oleh Netscape. Protokol ini bertujuan untuk memberikan layanan privasi dan integritas data antara dua aplikasi yang berkomunikasi. Terdapat dua layer dalam *Transport Layer Security* adalah sebagai berikut.

1. *TLS Handshake protocol*

protokol yang melakukan autentikasi antara server dan klien lalu melakukan negosiasi terhadap algoritma enkripsi dan kunci kriptografi yang akan digunakan dalam transaksi sebelum protokol aplikasi mengirimkan atau menerima data. Pada prinsipnya TLS handshake yang melakukan pertukaran kunci, dapat menggunakan algoritma kunci asimetrik seperti RSA atau Diffie-hellman antara klien dan server.

Secara garis besar protokol handshake yaitu:

- a) klien mengirim ke server yaitu nomor versi TLS yang digunakan klien, parameter enkripsi, data yang dibuat secara acak dan informasi lain yang dibutuhkan oleh server untuk berkomunikasi dengan klien. Jika dibutuhkan, client juga meminta server *certificate*.

- b) Server mengirim ke klien yaitu nomor versi TLS yang digunakan server, parameter enkripsi, data yang dibuat secara acak, dan juga informasi lain untuk berkomunikasi dengan server. Jika dibutuhkan, server juga mengirim server certificate dan server meminta *client certificate*.
- c) Jika klien meminta server *certificate* maka klien akan melakukan server *authentication* terlebih dahulu, menggunakan server certificate dan informasi lain yang didapat. Jika sukses, server *certificate* tidak diminta, atau pengguna mengizinkan maka client akan melanjutkan proses dan jika tidak sesi akan dihentikan.
- d) Dengan data yang telah didapat, klien membuat suatu *premaster secret* untuk sesi. Tergantung jenis enkripsi yang digunakan, dapat dilakukan dengan partisi server. *Premaster secret* dienkripsi menggunakan kunci *public server* (diambil dari *server certificate*), lalu dikirim ke server.
- e) Jika server meminta *client certificate*, maka client menandatangani secara digital data yang unik untuk sesi yang diketahui oleh klien dan server. Data tersebut adalah *digital signature* dan *client certificate* yang dikirim oleh klien ke server.
- f) Jika server meminta *client certificate*, maka server melakukan *client authentication*. Jika *authentication* diminta dan gagal, maka sesi dihentikan.
- g) Klien dan server membuat *master secret* menggunakan *premaster secret*. *Master secret* digunakan oleh klien dan server untuk membuat kunci sesi yang merupakan kunci enkripsi simetris.
- h) Klien memberikan informasi pada server bahwa kunci sesi akan digunakan untuk mengenkripsi komunikasi lebih lanjut. Klien kemudian mengirim pesan yang dienkripsi ke server yang mengatakan bahwa ia selesai dengan handshake.
- i) Server memberikan informasi pada Klien bahwa kunci sesi akan digunakan untuk mengenkripsi komunikasi lebih lanjut. Server kemudian mengirim pesan yang dienkripsi ke client yang mengatakan bahwa ia selesai dengan handshake.
- j) *Handshake* selesai.
Server authentication dilakukan dengan memeriksa *server certificate*. Dalam *server certificate* terdapat informasi seperti berikut.
 - kunci public server.
 - masa berlaku sertifikat.

- domain name untuk server.
- *domain name* untuk pembuat *certificate* (biasanya sertifikat dibuat oleh suatu *certificate authority*).

2. *TLS Record Protocol*

Protokol yang berfungsi untuk memastikan bahwa koneksi yang dibuat aman dan dapat diandalkan. Protokol ini bertugas untuk menghasilkan blok data yang akan dikirimkan. Lalu dipecah menjadi beberapa fragmen yang setiap fragmennya dikompresi dan ditambahkan hasil perhitungan dari *Message Authentication Code* (MAC) untuk mendeteksi proses data tampering. Data yang di terima di decrypt, di verifikasi, di dekompresi dan dipasang Kembali, kemudian di kirim ke klien yang tingkatnya lebih tinggi.

TLS record protocol mencakup dua hal yaitu koneksi yang bersifat rahasia (confidential) dengan menggunakan algoritma kriptografi simetris (AES) dan reliable dengan menyediakan jaminan integritas data menggunakan fungsi hash (SHA-256).

3.3.2 SSL

SSL (Secure Socket Layer) adalah protokol untuk keamanan komunikasi antara *Server* web dan browser *Client*. Protokol ini menggunakan sebuah badan yang biasa disebut CA (Certificate Authority) untuk mengidentifikasi memverifikasi pihak-pihak yang bertransaksi. SSL digunakan oleh jutaan situs web di seluruh dunia untuk memastikan keamanan data sehingga data tetap rahasia dan utuh. Salah satu ciri ketika SSL telah ditambahkan pada website, maka URL Website akan berubah menjadi HTTPS. SSL memungkinkan informasi sensitif seperti data kartu kredit, username, password dan informasi penting ditransmisikan dari server ke client atau sebaliknya dengan aman karena data yang dikirim akan diacak (dienkripsi).

Protokol SSL memenuhi aspek keamanan berikut:

1) Confidentiality

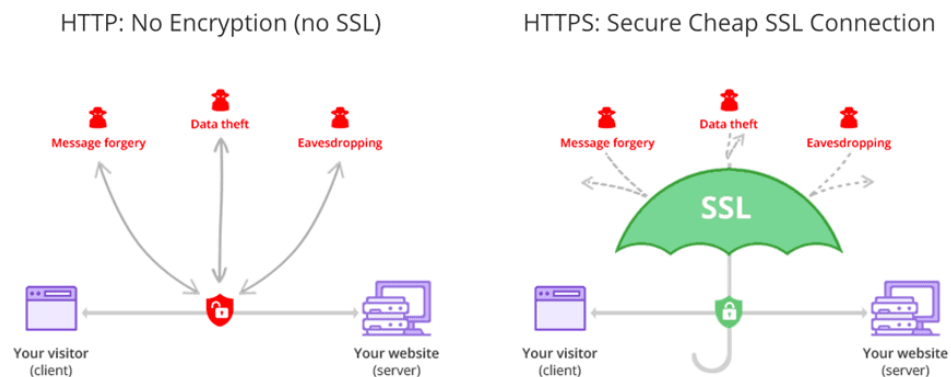
Pada protokol SSL informasi sensitif yang dikirim antara client dan server bersifat rahasia, karena plaint-text dienkripsi mejadi ciphertext dengan bantuan public key dan private key,

2) Message Integrity

Peran alert protocol pada SSL adalah guna mencegah tindakan intercept, serta pengubahan pada pesan yang dikirim. Pada ujicoba yang dilakukan message alert terlihat ketika client melakukan pembatalan konfirmasi certificates dan dimana server akan melakukan pembatalan komunikasi dengan client, dan client tidak dapat melakukan akses terhadap aplikasi web pada server.

3) Authentication.

Peran digital certificates pada protokol SSL adalah untuk memastikan jalur komunikasi SSL yang dibangun hanya yang berhak saja yang boleh memberikan informasi atau dapat dikatakan komunikasi dilakukan dengan rekan yang tepat.



Pada modul kali ini materi yang akan kita praktikan adalah materi SSL.

3.4 Langkah Praktikum

Buka VM dengan OS Ubuntu pada virtualbox lalu ikuti langkah di bawah ini :

1. Install web server apache2 dengan command `sudo apt install apache2 -y`



```
sudo apt install apache2 -y
```

2. Install open ssl dengan command `sudo apt install openssl -y`



```
sudo apt install openssl -y
```

3. Buat private key dan certificate ssl dengan command `sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/mbc.key -out /etc/ssl/certs/mbc.crt`



```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -  
keyout /etc/ssl/private/mbc.key -out /etc/ssl/certs/mbc.crt
```


Isi data sertifikat seperti di bawah ini :

```
Output
Country Name (2 letter code) [AU]:ID
State or Province Name (full name) [Some-State]:Jawa Barat
Locality Name (eg, city) []:Bandung
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:MBC Laboratory
Organizational Unit Name (eg, section) []:Cyber Security
Common Name (e.g. server FQDN or YOUR name) []:IP_address
Email Address []:mbc@laboratory.com
```

4. Buat direktori mbc pada direktori /var/www dengan command `sudo mkdir /var/www/mbc`

```
sudo mkdir /var/www/mbc
```

5. Buat file dan edit file index.html pada direktori /var/www/mbc dengan command sudo nano /var/www/mbc/index.html



```
sudo nano /var/www/mbc/index.html
```

Isi file index.html dengan <h1> WELCOME TO MBC LABORATORY </h1>



```
<h1> WELCOME TO MBC LABORATORY </h1>
```

6. Ubah kepemilikan direktori /var/www/mbc/ ke www-data dengan command sudo chown -R www-data:www-data /var/www/mbc/. Selanjutnya ubah hak akses direktori /var/www/mbc/ menjadi rwx-rwx-r-x atau Read, Write, and Execute - Read, Write, and Execute – Read and Execute dengan command sudo chmod -R 775 /var/www/mbc/

```
sudo chown -R www-data:www-data /var/www/mbc/  
sudo chmod -R 775 /var/www/mbc/
```

7. Salin konfigurasi asli yaitu file default-ssl.conf menjadi file mbc.conf dengan perintah sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-available/mbc.conf

```
sudo cp /etc/apache2/sites-available/default-ssl.conf  
/etc/apache2/sites-available/mbc.conf
```

Atur konfigurasi virtual host file dengan command sudo nano /etc/apache2/sites-available/mbc.conf

```
sudo nano /etc/apache2/sites-available/mbc.conf
```


8. Buatlah konfigurasi seperti berikut ini :

```
<VirtualHost *:80>
    Redirect "/" "https://IP_address/"
</VirtualHost>
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost.com
        ServerName IP_address
        DocumentRoot /var/www/mbc
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        SSLEngine on
        SSLCertificateFile /etc/ssl/certs/mbc.crt
        SSLCertificateKeyFile /etc/ssl/private/mbc.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>
    </VirtualHost>
</IfModule>
```

```
<VirtualHost *:80>
    Redirect "/" "https://IP_address/"
</VirtualHost>
<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost
        ServerName IP_address

        DocumentRoot /var/www/mbc

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile      /etc/ssl/certs/mbc.crt
        SSLCertificateKeyFile /etc/ssl/private/mbc.key

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>

    </VirtualHost>
</IfModule>
```

9. Hidupkan modul ssl dan header dengan perintah `sudo a2enmod ssl` dan `sudo a2enmod headers`

```
sudo a2enmod ssl  
sudo a2enmod headers
```

Hidupkan site pada konfigurasi file `mbc.conf` dengan perintah `a2ensite mbc`

```
sudo a2ensite mbc
```

10. Matikan site pada `default-ssl.conf` dan `000-default.conf` dengan command `sudo a2dissite default-ssl.conf` dan `sudo a2dissite 000-default.conf`

```
sudo a2dissite default-ssl.conf  
sudo a2dissite 000-default.conf
```

11. Tes konfigurasi dengan perintah sudo apache2ctl configtest

```
sudo apache2ctl configtest
```

Jika output seperti di bawah ini maka lakukan perintah selanjutnya

```
Output
AH00558: apache2: Could not reliably determine the server's
fully qualified domain name, using 127.0.1.1. Set the
'ServerName' directive globally to suppress this message
Syntax OK
```

Masuk ke konfigurasi file apache2.conf dengan perintah sudo nano /etc/apache2/apache2.conf lalu tambahkan konfigurasi ServerName IP_address pada file tersebut

```
sudo nano /etc/apache2/apache2.conf
```



```
ServerName IP_address
```

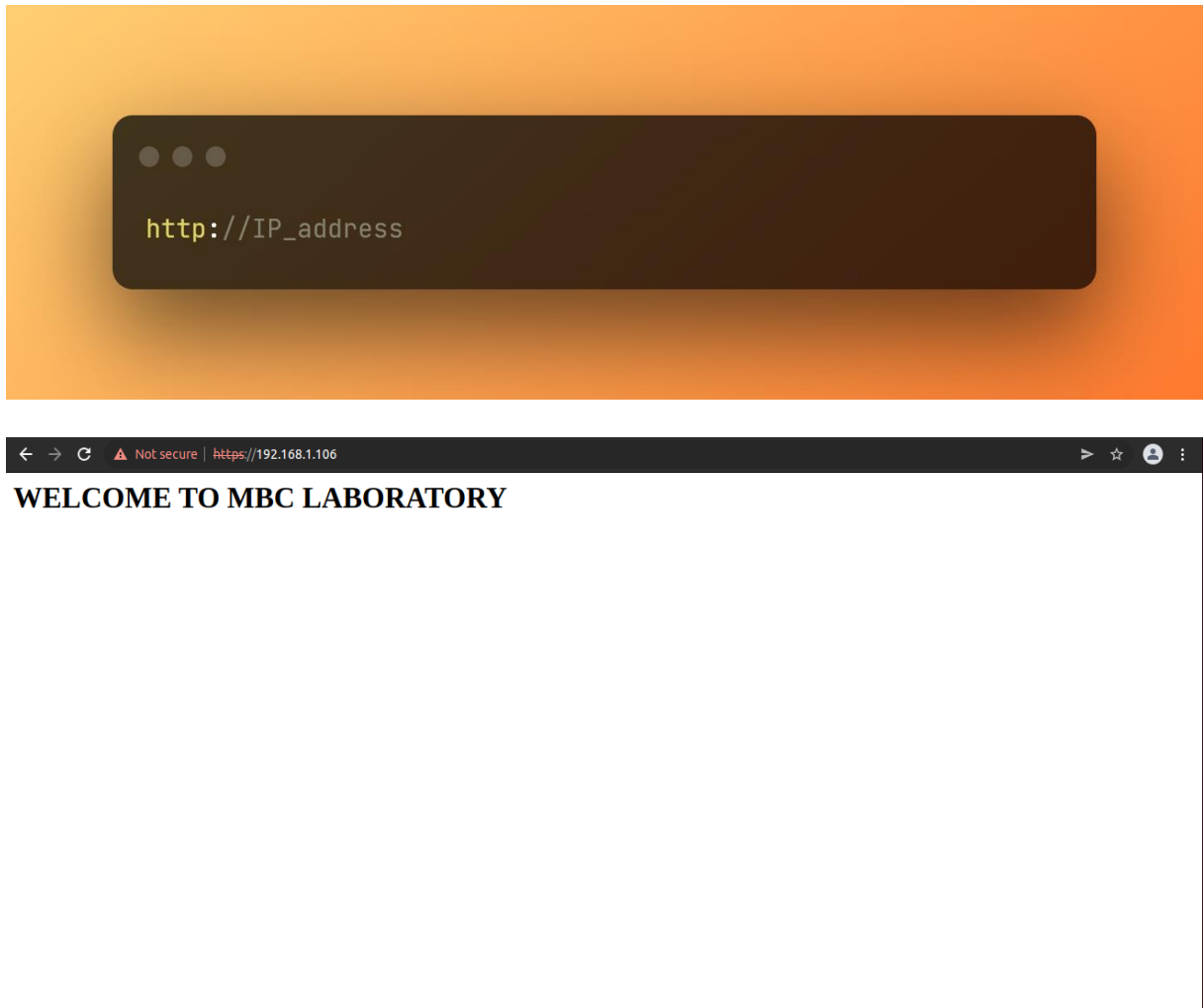
Lakukan tes lagi dengan perintah `sudo apache2ctl configtest` jika hasilnya sudah menunjukkan “Syntax OK” maka dapat dilanjutkan ke langkah selanjutnya

12. *Restart* dan *reload* apache2 dengan perintah `sudo systemctl reload apache2` dan `sudo systemctl restart apache2`



```
sudo systemctl reload apache2  
sudo systemctl restart apache2
```

13. Buka web browser dengan alamat http://IP_address maka akan muncul tampilan seperti berikut:



1. Detail Sertifikat

Certificate

10.0.2.15

Subject Name

Country	ID
State/Province	Jawa Barat
Locality	Bandung
Organization	MBC Laboratory
Organizational Unit	Cyber Security
Common Name	10.0.2.15
Email Address	mbc@laboratory.com

Issuer Name

Country	ID
State/Province	Jawa Barat
Locality	Bandung
Organization	MBC Laboratory
Organizational Unit	Cyber Security
Common Name	10.0.2.15
Email Address	mbc@laboratory.com

Validity

Not Before	Sat, 12 Feb 2022 14:19:48 GMT
Not After	Sun, 12 Feb 2023 14:19:48 GMT

Public Key Info

Algorithm	RSA
Key Size	2048
Exponent	65537
Modulus	EF:07:15:57:99:A7:E7:71:40:C2:E3:EB:AB:09:40:AD:49:45:B8:3B:EA:44:06:CF:...