

MODUL 11

WEB SERVER SECURITY

TUJUAN PEMBELAJARAN:

1. Mengenalkan pada mahasiswa tentang konsep Web Security di linux
2. Mahasiswa memahami berbagai konsep SSL yang ada di linux
3. Mahasiswa mampu melakukan analisa terhadap koneksi berbasis http dan https

DASAR TEORI

Saat ini *web* merupakan salah satu layanan informasi yang banyak diakses oleh pengguna internet di dunia. Sebagai salah satu layanan informasi maka perlu dibangun *web* yang mampu menangani permintaan (*request*) dari banyak pengguna dengan baik (*reliable*) tanpa meninggalkan aspek keamanannya. Masalah keamanan merupakan salah satu aspek yang penting dalam pembangunan *web* karena kelalaian dalam menangani keamanan *web* server dapat berakibat fatal.

Apache merupakan salah satu distribusi *web server* yang populer dengan dukungan *feature* yang sangat banyak. Perhitungan statistik yang ada saat ini menunjukkan bahwa *Apache* menjadi *web server* yang paling banyak digunakan dalam dunia internet, yaitu mencapai nilai 60 % dari seluruh *web server* yang ada.

Keberhasilan *Apache* mencapai kepopuleran saat ini selain dikarenakan memiliki banyak *feature* yang sering tidak dijumpai pada *web server* yang lain, juga dikarenakan *Apache* merupakan aplikasi gratis yang berjalan dalam berbagai sistem operasi.

Ada beberapa aspek yang perlu diterapkan dalam mengamankan *web server*, antara lain:

1. Layanan *web server* dengan *low privillages*
2. Pengaturan akses terhadap *web server*
3. Meminimalkan layanan publik pada mesin yang menjalankan *web server*
4. Menyediakan *filesystem* khusus untuk layanan *web server*

Layanan *web server* dengan *low privillages*

Pada sistem operasi berbasis UNIX semacam Linux, FreeBSD, OpenBSD setiap proses memiliki berbagai properti seperti nomor proses, pemilik proses, dan alokasi memori yang digunakan. *Apache web server* merupakan layanan publik, sehingga sangat dianjurkan proses-proses yang dijalankan oleh *Apache* dimiliki oleh user dengan hak akses terhadap sistem yang ada serendah mungkin (*low privillages*). Idealnya perlu dibuat user yang khusus menjalankan *web server*, misal user dengan nama *www*.

Dari perintah Linux di atas tampak bahwa *home* direktori dari user **www** adalah '/' yang secara *default* tidak dimiliki oleh user **www** melainkan dimiliki oleh **root** sehingga user **www** tidak memiliki akses tulis terhadap *home* direktorinya sendiri dan *shell* yang dimiliki oleh user **www** adalah '/bin/true'

sehingga secara otomatis *user www* tidak dapat menggunakan fasilitas-fasilitas semacam **telnet**, **rlogin**, **rsh**.

Langkah selanjutnya adalah melakukan setting pada *Apache web server* sehingga *web server* akan dijalankan oleh *user www* yang baru saja dibuat. File konfigurasi utama yang digunakan *Apache* adalah file **httpd.conf**. Pada file tersebut ditambahkan line semacam ini :

Pengaturan akses terhadap *web server*

Pengaturan akses pada *Apache web server* dapat dibedakan menjadi 2 macam :

- Pengaturan akses berdasarkan alamat IP dari *client*
- Pengaturan akses dengan proses autentikasi pengguna

Pengaturan akses berdasarkan *IP Address client*

Model pengaturan akses *web server* yang diterapkan disini didasarkan pada informasi *IP address* dari pengguna. Aturan dapat dibuat sehingga untuk file atau direktori tertentu akses dari *IP address* pengguna diterima atau ditolak. Untuk menerapkan aturan ini perlu dilakukan perubahan pada file konfigurasi *Apache* yaitu **httpd.conf**. Ada 3 kata kunci yang berkaitan dengan pengaturan ini yaitu **Order**, **Deny**, dan **Allow**. Kata kunci **Order** dapat diikuti oleh **deny**, **allow** atau **allow**, **deny** yang menunjukkan urutan evaluasi pengaturan berdasarkan aturan **Deny** dan aturan **Allow**. Kata kunci **Allow** maupun **Deny** diikuti oleh kata kunci **from** dan :

- **all** : menunjukkan akses untuk semua *host* diperbolehkan (**Allow**) atau ditolak (**Deny**)
- *IP address* : yaitu alamat IP yang diperbolehkan atau ditolak semisal :
- **167.205.25.6**
- **167.205.25.** (berarti berlaku untuk alamat IP **167.205.25.0 –167.205.25.255**) atau **167.205.** atau **167.**
- **167.205.25.0/27** (berlaku untuk **167.205.25.0-167.205.25.31**)
- **167.205.0.0/255.255.0.0** (berlaku untuk **167.205.0.0-167.205.255.255**)

Pengaturan akses *Apache* dapat dilakukan pula dengan membuat aturan-aturan pada file **.htaccess** pada direktori yang bersangkutan baik untuk mengatur akses terhadap file maupun akses terhadap direktori dimana file ini berada.

Pengaturan akses dengan proses autentikasi pengguna

Model pengaturan akses dengan proses autentikasi pengguna lebih fleksibel dibandingkan dengan pengaturan akses berdasarkan *IP address* pengguna.

Ketika anda mencoba untuk mengakses suatu *resources* yang dilindungi oleh model autentikasi user semacam ini maka anda harus memasukan informasi login dan password yang sesuai dalam suatu form semacam ini.

Untuk mengimplementasikan mekanisme autentikasi ini perlu dilakukan perubahan-perubahan pada file **httpd.conf** ataupun pada file **.htaccess** yang diletakkan pada direktori yang bersangkutan. Informasi data login dibuat dengan menggunakan program **htpasswd**.

Meminimalkan layanan publik pada mesin yang menjalankan *web server*

Pada umumnya mesin-mesin komputer yang terhubung dengan internet melayani beberapa program layanan sekaligus. Ada beberapa alasan yang mendasari hal tersebut antara lain :

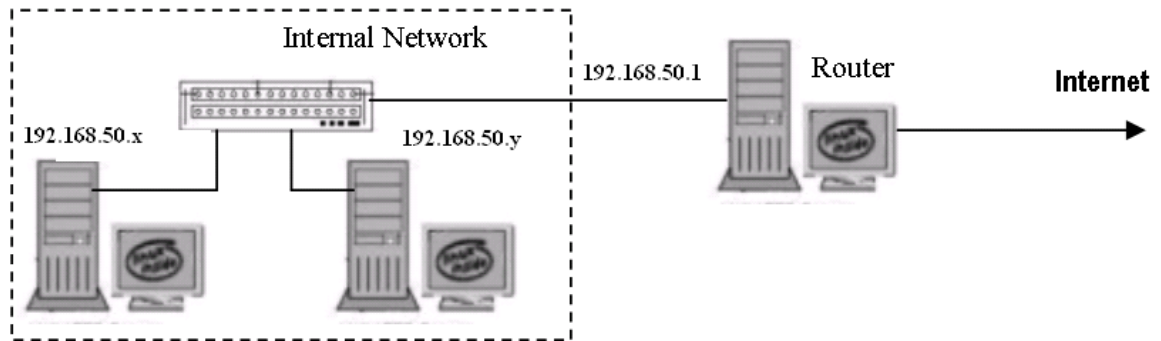
- Keterbatasan alamat IP saat ini
- Keterbatasan sumber daya mesin komputer
- Keinginan memaksimalkan sumber daya komputer dengan menjalankan

TUGAS PENDAHULUAN

1. Sebutkan dan jelaskan dengan singkat apa kegunaan dari htaccess?
2. Sebutkan pula kegunaan dari SSL pada HTTP

PERCOBAAN

1. Bangunlah jaringan seperti berikut :



Gambar 1 Jaringan Percobaan

NB:

Gunakan dhclient di masing-masing PC untuk mendapatkan IP dari router.

192.168.50.x & y : IP dari router

Pilih 192.168.50.x sebagai PC Server

Pilih 192.168.50.y sebagai PC Client

A.KONFIGURASI PADA PC SERVER

2. Install webserver Apache
apt-get install apache2
3. Buat direktori untuk menyimpan file https
mkdir /var/www/secure
4. Instalasi openssl
apt-get install openssl ssl-cert
mkdir /etc/apache2/ssl

Enable apache2 module :
a2enmod ssl
a2enmod auth_basic
5. Tambahkan konfigurasi ports SSL
vim /etc/apache2/ports.conf
Listen 80
Listen 443

6. Edit konfigurasi apache

```
# vim /etc/apache2/sites-enabled/000-default

<VirtualHost *:443>
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/server.cert
    SSLCertificateKeyFile /etc/apache2/ssl/server.key
    ServerName Komdig
    ServerAdmin zenhadi@eepis-its.edu
    DocumentRoot /var/www/secure

    Alias /secure "/var/www/secure"
    <Directory "/var/www/secure">
        AuthUserFile /home/student/.htpasswd
        AuthName EnterPassword
        AuthType Basic
        require valid-user
        Order Deny,allow
        Allow from all
    </Directory>
</VirtualHost>
```

7. Buat sertifikat di sisi server

```
# cd /etc/apache2/ssl
# openssl req -new > server.csr => membuat public key (Certificate Signing Request)
```

Generating a 1024 bit RSA private key

.....++++++

.....++++++

writing new private key to 'privkey.pem'

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:**ID**

State or Province Name (full name) [Some-State]:**East Java**

Locality Name (eg, city) []:**Surabaya**

Organization Name (eg, company) [Internet Widgits Pty Ltd]:**PENS-ITS**

Organizational Unit Name (eg, section) []:**Lab Komdig**

Common Name (eg, YOUR name) []:**komdig**

Email Address []:**zenhadi@eepis-its.edu**

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:mypass
An optional company name []:**PENS-ITS**

```
# openssl rsa -in privkey.pem -out server.key  
# openssl x509 -in server.csr -out server.cert -req -signkey server.key -days 365
```

8. Buat password untuk masuk ke webserver (apache2)
 # htpasswd -c /home/student/.htpasswd zenhadi
9. Restart apache
 # /etc/init.d/apache2 restart

B. TES KONFIGURASI DARI PC CLIENT

10. Ujicoba dengan menggunakan browser, catat dan analisa
 - a. Uji dengan http
 URL : http://<no_ip_server>
 - b. Uji dengan https, akan muncul sertifikat dari server dan autentikasinya
 URL : https://<no_ip_server>
 Tuliskan informasi certificate yang diperoleh.
11. Copy dan edit file index.html, dan simpan di /var/www/secure
 # cp /var/www/index.html /var/www/secure
 # vim /var/www/secure/index.html => tambahkan keterangan untuk membedakan dengan http
 <html>
 <body>
 <h1>It works!</h1>

 Ini di https
 </body>
 </html>
12. Uji kembali dengan https, catat dan analisa
 URL : https://<no_ip_server>
13. Analisa dengan menggunakan wireshark baik ketika mengakses menggunakan http dan https.
 Bandingkan hasil proses antara http dan https, gambarkan dalam bentuk blok diagram.

Daftar Pertanyaan

1. Berikan kesimpulan hasil praktikum yang anda lakukan.
2. Jelaskan pengertian TLS dalam sistem https.

LEMBAR ANALISA

Praktikum Network Security (Web Server Security)

Tanggal Praktikum :

Kelas :

Nama dan NRP :

A. Gambar topologi jaringan beserta dengan IP Addressnya, tunjukkan mana PC Server dan Client

B. Catat proses pembuatan sertifikat pada poin 7.

C. Catat hasil pada poin 8.

D. Catat hasil tes konfigurasi pada poin 10

E. Catat hasil pada poin 12.

F. Gambarkan blok diagram proses yang terjadi pada http dan https (poin 13)