Web Server với Linux
Nội dung

• Cơ chế của dịch vụ web
• Cài đặt và cấu hình webserver
• Các thao tác cơ bản của webserver
• Khái niệm SSL
• Sử dụng SSL
Coσ chẻ của dịch vụ web
URL và địa chỉ vật lý
Triển khai webservice
Qui trình cài đặt

- Installation of Apache program
- Setting of configuration file
- Registration of DNS server
- Starting of Apache, operation verification
Cài đặt Apache

1. Obtaining of Apache

```
Official site of apache
The Apache Software Foundation  http://www.apache.org/
```

2. Decompression and development

```
# gzip -cd httpd-2.0.50.tar.gz | tar xvf -
# ls -F httpd-2.0.50/
ABOUT_APACHE  InstallBin.dsp  NWGNUmakefile  apachenw.mcp.zip  docs/  os/
Apache.dsp    LAYOUT         README       build/       emacs-style  server/
Apache.dsw    LICENSE        README.platforms buildconf*  httpd.spec  src/ lib/
BuildBin.dsp  Makefile.in   VERSIONING    config.layout  include/  support/
CHANGES       Makefile.win  acconfig.h   configure*    libhttpd.dsp  test/ modules/
```

3. Compilation and installation

```
# cd httpd-2.0.50/
# ./configure --enable-ssl
    :
# make
    :
# make install
    :
```
Thành phần của Apache

Server Program (httpd)

- Standard module (static link)
  - Module included in package of Apache

Apache
core program

- Core program
  - Basic function part of Apache

- Standard module (static link)
  - The module of a static link is linked with the program file of Apache

DSO (xxx.so)

- Module made by third party (DSO)
  - Module not included in package of Apache
- Standard module (DSO)
  - Enhancing program group which offers various functions

DSO will be built in after Apache is started, when its function is used
Các mô đun mặc định

<table>
<thead>
<tr>
<th>Mod Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mod_access</td>
<td>mod_disk_cache</td>
</tr>
<tr>
<td>mod_actions</td>
<td>mod_echo</td>
</tr>
<tr>
<td>mod_alias</td>
<td>mod_env</td>
</tr>
<tr>
<td>mod_asis</td>
<td>mod_example</td>
</tr>
<tr>
<td>mod_auth</td>
<td>mod_expires</td>
</tr>
<tr>
<td>mod_auth_anon</td>
<td>mod_ext_filter</td>
</tr>
<tr>
<td>mod_auth_dbm</td>
<td>mod_file_cache</td>
</tr>
<tr>
<td>mod_auth_digest</td>
<td>mod_headers</td>
</tr>
<tr>
<td>mod_auth_ldap</td>
<td>mod_imap</td>
</tr>
<tr>
<td>mod_autoindex</td>
<td>mod_include</td>
</tr>
<tr>
<td>mod_cache</td>
<td>mod_info</td>
</tr>
<tr>
<td>mod_cern_meta</td>
<td>mod_isapi</td>
</tr>
<tr>
<td>mod_cgi</td>
<td>modldap</td>
</tr>
<tr>
<td>mod_cgid</td>
<td>mod_log_config</td>
</tr>
<tr>
<td>mod_charset_lite</td>
<td>mod_log_forensic</td>
</tr>
<tr>
<td>mod_dav</td>
<td>mod_logio</td>
</tr>
<tr>
<td>mod_dav_fs</td>
<td>mod_mem_cache</td>
</tr>
<tr>
<td>mod_deflate</td>
<td>mod_mime</td>
</tr>
<tr>
<td>mod_dir</td>
<td>mod_mime_magic</td>
</tr>
<tr>
<td>mod_negotiation</td>
<td>mod_nw_ssl</td>
</tr>
<tr>
<td>mod_proxy</td>
<td>mod_proxy_connect</td>
</tr>
<tr>
<td>mod_proxy_ftp</td>
<td>mod_proxy_http</td>
</tr>
<tr>
<td>mod_rewrite</td>
<td>mod_setenvif</td>
</tr>
<tr>
<td>mod_so</td>
<td>mod_speling</td>
</tr>
<tr>
<td>mod_ssl</td>
<td>mod_status</td>
</tr>
<tr>
<td>mod_sudo</td>
<td>mod_unique_id</td>
</tr>
<tr>
<td>mod_userdir</td>
<td>mod_usertrack</td>
</tr>
<tr>
<td>mod_vhost_alias</td>
<td>mod_vhost_alias</td>
</tr>
</tbody>
</table>

※ The module with ✓ sign is built in by default.
Cấu hình các mô đun (tính)

- Module built-in

```bash
# cd httpd-2.0.58/
# ./configure --enable-ssl --disable-userdir
# make
# make install
```

Module 'mod_ssl' is added, and module 'mod_userdir' is deleted

- Verification of module built-in

```bash
#/usr/local/apache2/bin/httpd -1
```

Compiled in modules:

- core.c
- mod_access.c
- mod_auth.c
- mod_include.c
- mod_log_config.c
- mod_negotation.c
- mod_env.c
- mod_sonvif.c
- mod_ssl.c
- prefork.c
- http_core.c
- mod_mime.c

We can see that the module 'mod_ssl' is built-in

We can see that module 'mod_userdir', which is usually built in by default, is not built-in

※ We have changed lines for module list to display for convenience of space. Originally, the execution result of 'httpd -l' is displayed continuously
Cấu hình các mô đun (động)

- Module built-in

```
# cd httpd-2.0.50/
# ./configure --enable-echo=shared

# make
# make install
```

Module `mod_echo` is added as DSO.

- Verification of module built-in

```
# cd /usr/local/apache2/modules/
# ls
```

It verifies that module `mod_echo` is installed under `/usr/local/apach2/modules`.

- Verification of module built-in (verification of `/usr/local/apache2/conf/httpd.conf` file)

```
# Dynamic Shared Object (DSO) Support

LoadModule echo_module modules/mod_echo.so
```

Verify description to build-in module `mod_echo` has been added.
Thư mục Apache

Server Root: /usr/local/apache2/

- bin/
- cgi-bin/
- conf/
- error/
- htdocs/
- icons/
- logs/
- man/
- manual/
- modules/
- httpd.conf
- mime.types
Các tệp cấu hình

• /etc/apache2.conf
• Có thể gắn thêm nhiều tệp khác
Các môi trường cấu hình

- <Directory>
- <DirectoryMatch>
- <Files>
- <FilesMatch>
- <IfDefine>
- <IfModule>
- <IfVersion>
- <Location>
- <LocationMatch>
- <Proxy>
- <ProxyMatch>
- <VirtualHost>
Các lệnh cấu hình

• Chỉ dùng một dòng, không có thẻ mở và đóng
• Có thể nằm ở ngoài hoặc trong một môi trường
• Chỉ có tác dụng trong môi trường
Cấu hình cơ bản (httpd.conf)

<table>
<thead>
<tr>
<th>ServerRoot</th>
<th>&quot;/usr/local/apache2&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Listen</td>
<td>80</td>
</tr>
<tr>
<td>User</td>
<td>nobody</td>
</tr>
<tr>
<td>Group</td>
<td>nobody</td>
</tr>
<tr>
<td>ServerAdmin</td>
<td><a href="mailto:webmaster@foo.co.jp">webmaster@foo.co.jp</a></td>
</tr>
<tr>
<td>ServerName</td>
<td><a href="http://www.foo.co.jp:80">www.foo.co.jp:80</a></td>
</tr>
<tr>
<td>DocumentRoot</td>
<td>&quot;/usr/local/apache2/htdocs&quot;</td>
</tr>
<tr>
<td>DirectoryIndex</td>
<td>index.html index.html.var</td>
</tr>
</tbody>
</table>
CGI

Editing of `/usr/local/apache2/conf/httpd.conf` file

```
ScriptAlias /cgi-bin/ "/usr/local/apache2/cgi-bin/"

#AddHandler cgi-script .cgi
```

- Specification by directory
- Specification by extension
Khai báo một thư mục

- `<directory></directory>`
- `.htaccess`
- `Allowoverride [None, Directive]`
  - Sử dụng Directive trong `.htaccess`
- `Allow, Deny`
  - Allow from all
  - Allow from 192.168.192.0/18 hut.edu.vn
- `Order Allow, Deny`
Options

- All
- ExecCGI
- FollowSymLinks
- Includes Server-side
- IncludesNOEXEC
- Indexes
- MultiViews
- SymLinksIfOwnerMatch
.htaccess

- Khai báo các thuộc tính của thư mục
- Được apache đọc và kích hoạt trực tiếp
• <Directory>
• <DirectoryMatch>
• <Files>
• <FilesMatch>
• <IfDefine>
• <IfModule>
• <IfVersion>
• <Location>
• <LocationMatch>
• <Proxy>
• <ProxyMatch>
• <VirtualHost>
Cấu hình DNS

1. Editing the zone data file

   Edit the forward lookup zone data file

   | www.foo.co.jp. | IN | A | 192.168.0.40 |

   Edit the reverse lookup zone data file

   | 40.0.168.192.in-addr.arpa. | IN | PTR | www.foo.co.jp. |

2. Restart of DNS server

   # kill `cat /var/run/named.pid`
   # /usr/local/sbin/named

3. Verification of registered information

   # host www.foo.co.jp
   www.foo.co.jp has address 192.168.0.40

   #
Khởi động webserver

- Web server startup by 'apachectl' command

```bash
# /usr/local/apache2/bin/apachectl start
```

- Verification of Web server's startup

```
# ps -ef | grep httpd
root  15633  1 16 17:26 ?  00:00:01 /usr/local/apache2/bin/httpd -k start
nobody 15634 15633  0 17:26 ?  00:00:00 /usr/local/apache2/bin/httpd -k start
nobody 15635 15633  0 17:26 ?  00:00:00 /usr/local/apache2/bin/httpd -k start
nobody 15636 15633  0 17:26 ?  00:00:00 /usr/local/apache2/bin/httpd -k start
nobody 15637 15633  0 17:26 ?  00:00:00 /usr/local/apache2/bin/httpd -k start
nobody 15638 15633  0 17:26 ?  00:00:00 /usr/local/apache2/bin/httpd -k start
#```
Bảo mật webserver

Access limitation of Apache

Host authentication

User authentication
Quản lý truy cập tài nguyên

```
<Directory "/usr/local/apache2/htdocs">
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all

</Directory>

<Files "*.ht">
  Order allow,deny
  Deny from all

</Files>
```
Kiểm soát máy tính

Editing `/usr/local/apache2/conf/httpd.conf` file

```
<Directory "/usr/local/apache2/htdocs">

    Order deny,allow
    Allow from 192.168.0.0/24
    Deny from all

</Directory>
```
Kiểm soát tài khoản

Editing ‘/usr/local/apache2/conf/httpd.conf’ file

```xml
<Directory "/usr/local/apache2/htdocs”>
    AuthType Basic
    AuthName “Protected Page on www.foo.co.jp”
    AuthUserFile /usr/local/apache2/conf/.htpasswd
    require user taro hanako
</Directory>
```

Password registration by ‘htpasswd’ command

```
# /usr/local/apache2/bin/htpasswd -c /usr/local/apache2/conf/.htpasswd taro
New password: 
Re-type new password: 
Adding password for user taro
#`
```
Liên hệ giữa tệp cấu hình và tệp kiểm soát tài khoản

```
/usr/local/apache2/conf/httpd.conf

<Directory "/usr/local/apache2/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride ALL
</Directory>

/usr/local/apache2/htdocs/.htaccess

Setting concerning directory '/usr/local/apache2/htdocs'

The name of the access control file is specified as '.htaccess'.
The access control file is referred when the access control file is effective.
```
Log

Access log

192.168.0.40 - - [12/Aug/2004:20:05:42 +0900] "GET / HTTP/1.1" 401 512

Error log

[Fri Aug 13 10:29:43 2004] [error] [client 192.168.0.40] (13)Permission denied: exec of '/usr/local/apache2/cgi-bin/printenv' failed

Agent log

Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922
Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922
Mozilla/5.0 (X11; U; Linux i686; ja-JP; rv:1.4) Gecko/20030922

Referrer log

- -> /index.html.var
http://www.foo.co.jp/ -> /apache_pb.gif
- -> /cgi-bin/printenv
Cấu hình log

HostnameLookups Off
:

ErrorLog logs/error_log
:

LogLevel warn
:

LogFormat "%h %l %u %t "%r" %>s %b "%{Referer}i" "%{User-Agent}i"
combined
LogFormat "%h %l %u %t "%r" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
:

CustomLog logs/access_log common
:

#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
Khái niệm SSL

SSL (Secure Socket Layer)

Protocol that encrypts communication root and authenticates other party of communication

Client

Internet

Encryption

Server

SSL's functions

- Encryption of data
- Server authentication
- Client authentication
- Data falsification prevention
Sự cần thiết của SSL
Các loại mã hóa

**Common key cipher**
- Use the same key for the encryption and the decryption.
- The mechanism of the cipher communication being simple, the encryption/decryption load is light.
- The problem is: how to receive and to pass the key?

**Public key cipher**
- Use a different key to encrypt and to decode.
- Make a pair of public key and private key, and open the public key to the public.
- Only the private key, that forms a pair, can decipher the information encrypted by the public key.
- Only the public key, that forms a pair, can decipher the information encrypted by the private key.
- The mechanism of the cipher communication being complex, the load is heavy.
Cơ chế sử dụng SSL
Cài đặt SSL

- Installation of SSL server
- Making of key pair (private key and public key)
- Making of certificate signature request
- Contract with Certificate authority
- Setting of SSL server
Cài đặt SSL

# gzip -cd openssl-0.9.7d.tar.gz | tar xvf -
# cd openssl-0.9.7d/
# ./config : 
# make : 
# make install :
#

• Compilation and installation of Apache

# PATH=/usr/local/ssl:/usr/local/ssl/bin:$PATH ;export PATH
#
# cd httpd-2.0.50/
# ./configure --enable-ssl :
# make :
# make install :
#
Quản lý cặp khóa

# cd /usr/local/apache2/conf/
# mkdir ssl.key
# chmod 700 ssl.key
# ls -ld ssl.key
drwx------ 2 root root 4096 Aug. 13 11:31 ssl.key
# cd ssl.key/
# cat /bin/ls /bin/cp /bin/more > /tmp/random.db
#
#/usr/local/ssl/bin/openssl genrsa -des3 -rand /tmp/random.db 1024 > server.key
144644 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
............................+++++
............................+++++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
#
# chmod 400 server.key
# ls -l
Total 4
-r-------- 1 root root 963 Aug. 13 11:42 server.key
#
Ký bằng chứng chỉ

```
# cd /usr/local/apache2/conf
# mkdir ssl.csr
# chmod 700 ssl.csr
# ls -ld ssl.csr
drwx------ 2 root root 4096 8月 13 13:12 ssl.csr
# cd ssl.csr

#/usr/local/ssl/bin/openssl req -new -key ..ssl.key/server.key -out server.csr
Enter pass phrase for ..ssl.key/server.key:

-----
Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Ohta-ku
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FLM
Organization Unit Name (eg, section) []:Learning Service
Common Name (eg, YOUR name) []:www.foo.co.jp
Email Address []:webmaster@foo.co.jp

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

Make a directory of storage for certificate signature request, and change access right
Make certificate signature request
Input passphrase of the private key
Input site information
Usually, do not input anything, and omit it with [Enter]
(Follows the instruction of certificate authority)
Change access right of certificate signature request
Xác thực

Apply for the certificate issue through HP

Mail content (certificate)

Dear VeriSign Customer,

Thank you for ordering VeriSign Digital ID. Your Server ID (Certificate) has been issued and is attached at the end of this message. Please refer to the following URL to install your Digital ID on your server and to see the details of Secure Site Seal.

Please visit:
http://www.verisign.com/jp/serverops/s_id.html

Common Name : WWW FOO.CO.JP

---BEGIN CERTIFICATE---
MIGfMAoGCSqGSIb3DQEBAQUAA4GNADCBiQKBgQYFyHk2U8q3G3f1a1CHTJHjEf1EF
3d2G8q5f7c8bUJ4GyUGPv7hUJs4QyUNxkbPMPfjUH+D7j1B2H8mYXG/RZ94Fb
---END CERTIFICATE---
Cấu hình server

Setting of certificate acquired from certificate authority

```
# mkdir /usr/local/apache2/conf/ssl.crt
#
# mv server.pem /usr/local/apache2/conf/ssl.crt/
```

Make directory for certificate authority storage
Store certificate acquired from certificate authority

Editing of '/usr/local/apache2/conf/ssl.conf' file

```conf
<IfDefine SSL>

<VirtualHost _default:443>

ServerName www.foo.co.jp:443
ServerAdmin webmaster@foo.co.jp

# SSL Engine Switch:
SSLEngine on

# Server Certificate:
SSLCertificateFile /usr/local/apache2/conf/ssl.crt/server.pem

# Server Private Key:
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server.key

</VirtualHost>
</IfDefine>
```
Kiểm tra cấu hình

Startup of Apache for SSL

```
# /usr/local/apache2/bin/apachectl sslstart
Apache/2.0.50 mod_ssl/2.0.50 (Pass Phrase Dialog)
Some of your private key files are encrypted for security reasons.
In order to read them you have to provide us with the pass phrases.

Server www.foo.co.jp:443 (RSA)
Enter pass phrase: 

Ok: Pass Phrase Dialog successful.
```

Specify https for the protocol

It locks, when SSL is communicated
Giấu mật khẩu

Signature for private key

```
# cd /usr/local/apache2/conf/ssl.key
#/usr/local/ssl/bin/openssl rsa -in ./server.key -out ./server_signed.key
Enter pass phrase for ./server.key:
writing RSA key
#
```

Set the signed private key to the server

```
SSLCertificateKeyFile /usr/local/apache2/conf/ssl.key/server_signed.key
```

Start of Apache for SSL

```
#/usr/local/apache2/bin/apachectl sslstart
#
```

Passphrase input is not requested
Cấu hình xác thực client

### Certificate Authority (CA):
Set the CA certificate verification path where to find CA certificates for client authentication or alternatively one huge file containing all of them (file must be PEM encoded).

Note: Inside SSLCACertificatePath you need hash symlinks to point to the certificate files. Use the provided Makefile to update the hash symlinks after changes.

```bash
SSLCACertificatePath /usr/local/apache2/conf/ssl.crt
SSLCACertificateFile /usr/local/apache2/conf/ssl.crt/cacert.pem
```

### Client Authentication (Type):
Client certificate verification type and depth. Types are none, optional, require and optional_no_ca. Depth is a number which specifies how deeply to verify the certificate issuer chain before deciding the certificate is not valid.

```bash
SSLVerifyClient require
SSLVerifyDepth 10
```
Chứng chỉ đơn giản

```
# cd /usr/local/ssl
# /usr/local/ssl/misc/CA.sh -newca
CA certificate filename (or enter to create)

Making CA certificate ...
Generating a 1024 bit RSA private key
.............+++++
.............+++++
writing new private key to "./demoCA/private./cakey.pem"
Enter PEM pass phrase: Verifying - Enter PEM pass phrase:

Country Name (2 letter code) [AU]:JP
State or Province Name (full name) [Some-State]:Tokyo
Locality Name (eg, city) []:Ohta-ku
Organization Name (eg, company) [Internet Widgits Pty Ltd]:FLM
Organizational Unit Name (eg, section) []:CMASTER
Common Name (eg, YOUR name) []:ca.foo.co.jp
Email Address []:camaster@foo.co.jp
```

- Make simple certificate authority (making of key pair of simple certificate authority and certificate)
- Set password to private key of simple certificate authority
- Set simple certificate authority information
- Change access right of key pair of simple certificate authority.