Assignment 2

Web server and Web technology

Nguyen Quoc Trong Nghia – s3343711
Nguyen Vinh Linh – s3410595
Date: December 27, 2013

I. Compile and install the Apache web server on Ubuntu (virtual host)

Make related directory for the requirement of assignment

Firstly, I logged in as super user mode in order not to re-type password when the system needed authentication. I used the following command:

sudo –i

Then I typed password. I went to the /home directory.

cp /home

I created folder SimpleLife and accessed this folder

mkdir SimpleLife

cd SimpleLife

I created multiple folders by using this command

mkdir –p Research secure src phpinfo

I went to Research and secure folder to create directories as assignment requirements

cd /home/SimpleLife/Research

mkdir –p Books Thesis Journal

cd /home/SimpleLife/secure

mkdir wswt

cd /home/SimpleLife/secure/wswt

mkdir –p Course Lab SimpleLife Tute

cd /home/SimpleLife/secure/wswt/SimpleLife

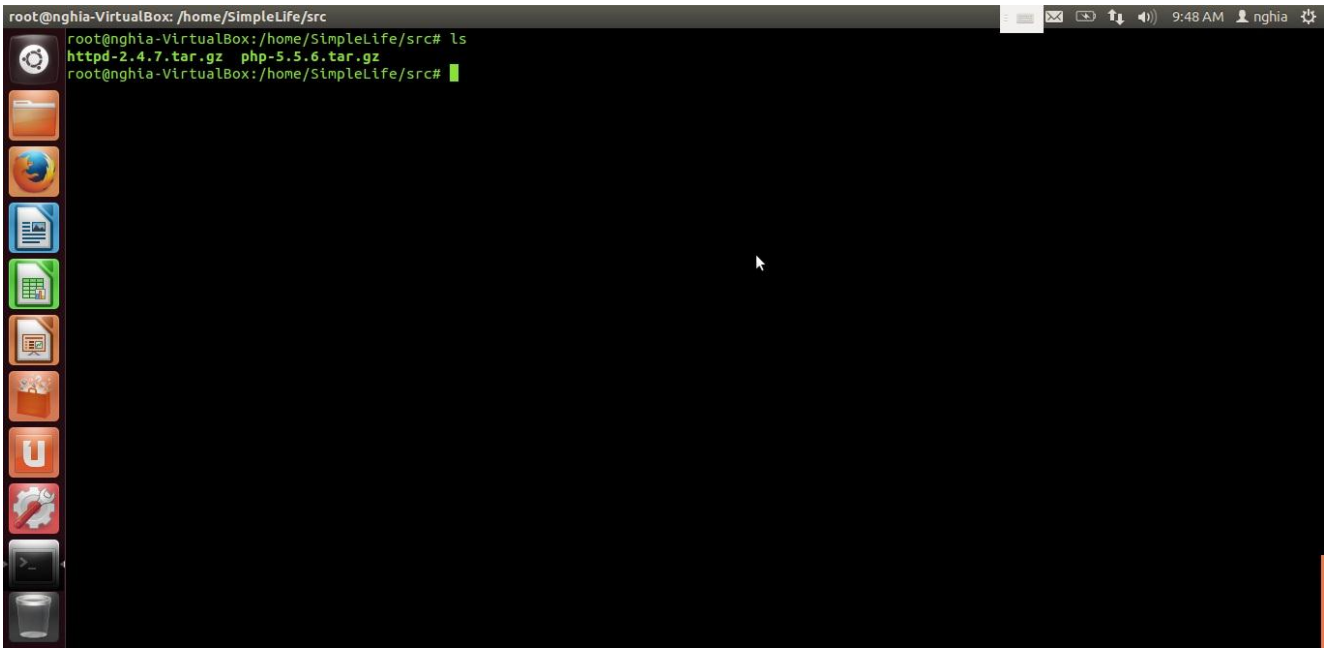
mkdir –p Assinment1 Assignment2

-p option refers to make multiple folders within current working directory

Apache non-secure Installation

I went to src folder and downloaded the latest Apache and PHP setup files

cd /home/SimpleLife/src



I uncompressed the file using this command:

tar -xvzf httpd-2.4.7.tar.gz

cd httpd-2.4.7

Done uncompressing, I need to config the port so that I can use this port later.

./configure --prefix=/home/SimpleLife/apache2-secure \--with-port=53711 \--enable-so \--with-mpm=prefork --with-apr=/usr/local/apr/ --with-pcre=/usr/local/pcre

--prefix=/home/SimpleLife/apache2: this will tell the ternimal where to install the Apache.

--with-port=: it defines a port.

53711: This is the port for me. The format of the port is: 5 + last 4 digits of my student ID

--enable-so: This means to enable shared object or DSO

--with-mpm=prefork:

--with-apr=/usr/local/apr/ and **--with-pcre=/usr/local/pcre** were the 2 libraries that the latest Apache server needs

The installation will be processed after I type these commands:

make

make install

In order to run the server, I need to go to the bin folder inside the apache folder that just installed and execute command:

cd /home/SimpleLife/apache2/bin

./apachectl start

To check that the Apache server works properly, I access to the URL: **localhost:53711**. If the page shows the words "It works!" it proves that the server is working.

PHP installation

Just like the Apache installing process, I executed commands so that the Mekong can extract and install PHP:

cd /home/SimpleLife/src

tar –xvzf php-5.5.6.tar.gz

cd php-5.5.6

Then I execute my configuration:

./configure --with-apxs2=/home/username/apache2/bin/apxs

\--prefix=/home/username/php5

--with-apxs2=/home/SimpleLife/apache2/bin/apxs: It tells PHP that it uses Apache apxs executable.

--prefix=/home/SimpleLife/php5: It will install in the php5 folder with the path to this folder

Next step is compilation:

make

make install

After installing Apache and PHP, I added few lines of code in the httpd.conf in the **"/home/SimpleLife/apache2/conf**" using this executing command

gvim /home/SimpleLife/apache2/conf/httpd.conf

In this file I added

AddType application/x-httpd-php .php: line 302

LoadModule php5_module modules/libphp5.so: line 148

Final step is executing 2 commands
cd /home/SimpleLife/apache2/bin
./apachectl graceful

I created and modified **info.php** file in **cd /home/SimpleLife /apache2/htdocs**
I added this line to info.php: **<?php phpinfo(); ?>**
Then I accesses Internet and check for PHP information.

System	Linux nghia-VirtualBox 3.8.0-29-generic #42~precise1-Ubuntu SMP Wed Aug 14 15:31:16 UTC 2013 i686
Build Date	Dec 17 2013 23:59:58
Configure Command	'./configure' '--with-apxs2=/home/SimpleLife/apache2/bin/apxs' '--prefix=/home/SimpleLife/php5.5.6'
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/home/SimpleLife/php5.5.6/lib
Loaded Configuration File	(none)
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API220121212,NTS
PHP Extension Build	API20121212,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled

II. Apache secure Installation

First, I remove the compiled Apache folder and unzip the new folder to create and compile new secure-Apache.

cd /home/SimpleLife/src
rm -R httpd-2.4.7
rm -R php-5.5.6
tar -xvzf httpd-2.4.7.tar.gz
cd httpd-2.4.7
./configure --prefix=/home/SimpleLife/apache2-secure \--with-port=50595 \--enable-so \--with-mpm=prefork --enable-ssl --enable-cgi --with-apr=/usr/local/apr/ --with-pcre=/usr/local/pcre

--prefix=/home/SimpleLife/apache2-secure: this will tell the ternimal where to install the Apache.

--with-port=: it defines a port.

50595: This is the port for secured server. The format of the port is: 5 + last 4 digits of my student ID

--enable-so: This means to enable shared object or DSO

--with-mpm=prefork:

--enable-ssl: This means the server includes the Secure Socket Layers mode

--enable-cgi: This means the server includes the Common Gateway Interface mode

--with-apr=/usr/local/apr/ and **--with-pcre=/usr/local/pcre** were the 2 libraries that the latest Apache server needs

The installation will be processed after I type these commands:

make

make install

In order to run the server, I need to go to the bin folder inside the apache folder that just installed and execute command:

cd /home/SimpleLife/apache2-secure/bin

./apachectl start

To check that the Apache server works properly, I access to the URL: **localhost:50595**. If the page shows the words “It works!” it proves that the server is working.

III. Configure non-secure Apache server

httpd.conf

Listen 53711

#make virtual host for root directory

NameVirtualHost *:53711

<VirtualHost *:53711>

DocumentRoot **/home/SimpleLife**

ServerName **home2171.bit.rmit.edu.vn**

CustomLog "**logs/home2171_access_log**" combined

#set permission for root directory

<Directory "**/home/SimpleLife**">

Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

</VirtualHost>

#make virtual host for **research 2171.rmit.edu.vn**

<VirtualHost *:53711>

DocumentRoot **/home/SimpleLife/Research**

ServerName **research2171.bit.rmit.edu.vn**

CustomLog "**logs/research2171_access_log**" combined

#set permission to Research folder

<Directory "**/home/SimpleLife/Research**">

Options Indexes FollowSymLinks

AllowOverride None

Order allow,deny

Allow from all

</Directory>

</VirtualHost>

#make virtual host for **cosc2171.bit.rmit.edu.vn**

<VirtualHost *:53711>

DocumentRoot **/home/SimpleLife/secure/wswt**

```
        ServerName cosc2171.bit.rmit.edu.vn
        CustomLog "logs/cosc2171_access_log" combined

        #set permission to wswt folder
        <Directory "/home/SimpleLife/secure/wswt">
            Options Indexes FollowSymLinks
            AllowOverride None
            Order allow,deny
            Allow from all
        </Directory>
    </VirtualHost>

    #make virtual host for phpinfo2171.bit.rmit.edu.vn
    <VirtualHost *:53711>
        DocumentRoot /home/SimpleLife/phpinfo
        ServerName phpinfo2171.bit.rmit.edu.vn
        CustomLog "logs/phpinfo2171_access_log" combined

        #set permission to phpinfo folder
        <Directory "/home/SimpleLife/phpinfo">
            Options Indexes FollowSymLinks
            AllowOverride None
            Order allow,deny
            Allow from all
        </Directory>
    </VirtualHost>

    #disable one sub folder of Research directory
    <Directory "/home/SimpleLife/Research/Thesis">
        Options Indexes FollowSymLinks
        AllowOverride None
        Order allow,deny
        Allow from none
    </Directory>

    #Allow html, php, jpg, png file extensions
    <FilesMatch "\.(html|php|jpg|jpeg|png)$">
        Order allow,deny
        Allow from all
    </FilesMatch>
```

Explanation

Firstly, I create 4 virtual hosts which are **home2171.bit.rmit.edu.vn**, **research2171.bit.rmit.edu.vn**, **cosc2171.bit.rmit.edu.vn** and **phpinfo2171.bit.rmit.edu.vn.**

home2171.bit.rmit.edu.vn holds the index.html file to display the links to go to the **cosc2171**, **research2171** and **phpinfo2171**. The root is **/home/SimpleLife.**

research2171.bit.rmit.edu.vn is the host for the Research directory.

cosc2171.bit.rmit.edu.vn is the host for the wswt directory.

phpinfo2171.bit.rmit.edu.vn is the host to display the PHP server information.

DocumentRoot is the tag to specify the directory location for the virtual host.
ServerName is the link for user to type in the browser.
CustomLog is the command to record all activities when the users access the directory.
Directory tag is used to specify further options for users.
FollowSymLinks makes Apache follow system symbolic links/shortcuts .
Indexes allows access to open folders within server directory
Order allow,deny Allow from all is the command to allow all user to access this directory.
Allow from none: is to disallow user to access this directory
FilesMatch tag is to filter all extensions that matched with the regularexpression.

IV. Configure secure Apache server

httpd.conf

Include **conf/extra/httpd-ssl.conf**

Explanation

conf/extra/httpd-ssl.conf is the URL for the httpd-ssl.conf. Because we decided that the entire configuration is stored in the httpd-ssl.conf, this httpd.conf does not have any code. It just needs to include the httpd-ssl.conf in order to run the server.

httpd-ssl.conf

```
Listen 50595
<VirtualHost *:50595>
    #   General setup for the virtual host
    DocumentRoot "/home/SimpleLife/secure/wswt"
    ServerName homesecure2171.bit.rmit.edu.vn
    #   SSL Engine Switch:
    SSLEngine on
    #   Server Certificate:
    SSLCertificateFile "/home/SimpleLife/apache2-secure/ssl-key/assignment2.crt"
    #   Server Private Key:
    SSLCertificateKeyFile "/home/SimpleLife/apache2-secure/ssl-
    key/assignment2.key"
</VirtualHost>
<Directory "/home/SimpleLife/secure">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

#DocumentRoot "/home/SimpleLife/secure/wswt/SimpleLife"
<Directory "/home/SimpleLife/secure/wswt/SimpleLife/Assignment2">
    AuthType Basic
    AuthName "SimpleLife Group Only"
    AuthBasicProvider file
    AuthUserFile "/home/SimpleLife/apache2-secure/password/pwd_log"
    AuthGroupFile "/home/SimpleLife/apache2-secure/password/group"
    Require group SimpleLife
```

```
        AllowOverride None
    </Directory>
    <Location /home/SimpleLife/Research/server-status>
        AuthType Basic
        AuthName "SimpleLife Group Only"
        AuthBasicProvider file
        AuthUserFile "/home/SimpleLife/apache2-secure/password/pwd_log"
        AuthGroupFile "/home/SimpleLife/apache2-secure/password/group"
        Require group SimpleLife
        SetHandler server-status
        Order Deny,Allow
        Deny from none
    </Location>

    ServerTokens ProductOnly
    ServerSignature Off

    <FilesMatch "\.(jpg|gif|ping)$">
        # 21 days = 21days * 24hours * 60minutes * 60seconds = 1814440 (seconds)
        Header set Cache-Control "max-age=1814440, public"
    </FilesMatch>

    <FilesMatch "\.(html)$">
        # 8 hours = 8hours * 60minutes * 60seconds = 28800 (seconds)
        Header set Cache-Control "max-age=28800, public"
    </FilesMatch>

    <FilesMatch "\.(php)$">
        Header set Cache-Control "private"
    </FilesMatch>
```

Explanation

SSLEngine on: This mode to turn on the Secure Socket Layers Engine on.

SSLCertificateFile: This code is used to include the certificate file when user access to the secured link on browser.

"/home/SimpleLife/apache2-secure/ssl-key/assignment2.crt":this is the link to the certificate file when the user needs.

SSLCertificateKeyFile: This code to include the key file

"/home/SimpleLife/apache2-secure/ssl-key/assignment2.key": this is the link to the certificate key file when starting the secured server

AuthType Basic: This is the mode to require authenticated username and password of the user when access authenticated directory. **Basic** means that the user's password is unencrypted when sending to the server.

AuthName "SimpleLife Group Only": This is a message that would be display when web browser prompt username and password.

AuthBasicProvider file: This means that the server uses file as authenticated basic provider.

AuthUserFile: The command to specify which file is used to store the username and password of authenticated users

"/home/SimpleLife/apache2-secure/password/pwd_log": This is the path file that stored username and password.

AuthGroupFile: a file to store all authenticated accounts. This also links to the AuthUserFile to get the password.

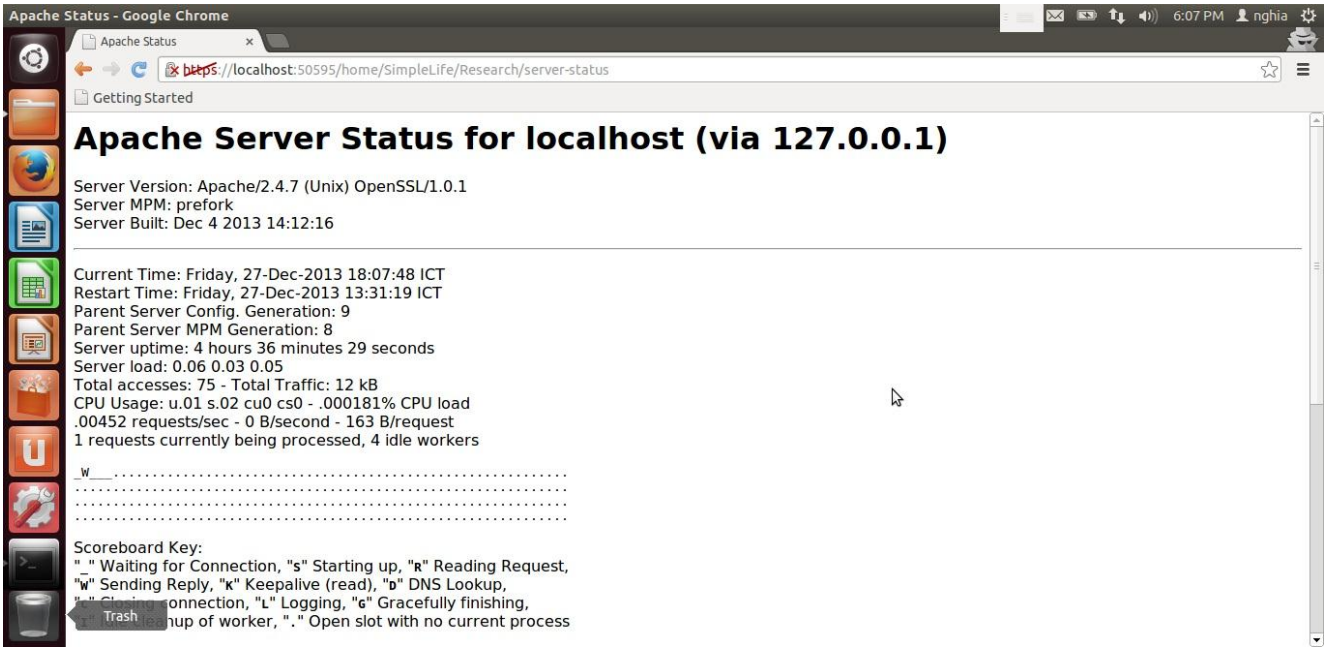
"/home/SimpleLife/apache2-secure/password/group": This is the path file that stored authenticated usernames

Require group SimpleLife: Only this group can access specified directory, other users cannot log in to this directory

ServerTokens ProductOnly: This will show only the Server name, not the version of the server.

ServerSignature Off: is used to configure the footer on server-generated documents. **Off** means that the server does not display this documents

SetHandler server-status: This code is to display the Apache server status



Make Basic authentication

Make username & password file

./bin/htpasswd -c /home/SimpleLife/apache2-secure/password/pwd_log admin

Then, it will prompt you to type password for user named admin, we decide to use 1234 as a password of admin

Make accessing group for number of users including: admin, s3410595, s3343711

gvim /home/SimpleLife/apache2-secure/password/group

In this stage, we wrote:

SimpleLife: s3410595, s3343711, admin

It means that group name is SimpleLife, and this group have three members, they have ids which are “admin”, “s3410595”, “s3343711”.

Add new username to username & password file (file pwd_log)

./bin/htpasswd -c /home/SimpleLife/apache2-secure/password/pwd_log s3410595

./bin/htpasswd -c /home/SimpleLife/apache2-secure/password/pwd_log s3343711

Modify httpd-ssl.conf

<Directory "**/home/SimpleLife/secure/wswt/SimpleLife**">

#Set basic authentication for folder SimpleLife

AuthType Basic

AuthName "SimpleLife Group Only"

AuthBasicProvider file

AuthUserFile "/home/SimpleLife/apache2-secure/password/pwd_log"

AuthGroupFile "/home/SimpleLife/apache2-secure/password/group"

Require group SimpleLife

AllowOverride None

</Directory>

V. The PKE key pair you created for your secure server and the certificate you created for your secure server.

Generate private key for web server, this key is used to decrypt message from client. In order to generate this key in Linux, use the following command:

openssl genrsa -des3 -out assignment2.key 2048

genrsa: generation of RSA Private Key.

des3: Triple-DES Cipher

assignment2.key: the output file after generating process

2048: the number of characters within the private key file.

Making certificate signing request (.csr), this process will make a .csr file which will be sent to Certificate Authority (CA) to check the identity of website, After checking completely, CA will give back a certificate (.crt file) to website owners. However, in this assignment, we make our own certificate and play a role as a CA.

Make certificate signing request (.csr) file

openssl req -new -key assignment2.key -out assignment2.csr

req: Certificate Signning Request

-key assignment2.key: the private key

-out assignment2.csr: set the output for generating process

make certificate (.crt) file

openssl x509 -req -day 365 -in assignment2.csr -signkey assignment2.key

-out assignment2.crt

Remove pass parse of private key (optional, this part can help you run server without pass parse)

cp assignment2.key assignment2.key.org

openssl rsa -in assignment2.key.org -out assignment2-nopass.key

Configure server with certificate and private key

Within apache2-secure folder, we make a new folder which is used to store private key and certificate

Make new folder within apache2-secure

mkdir ssl-key

Store certificate and private key inside ssk-key folder

cp assignment2.key /home/SimpleLife/apache2-secure/ssl_key

cp assignment2.crt /home/SimpleLife/aapche2-secure/ssl_key

Regard to configuration, we decide to use extra file – it is httpd-ssl.conf instead of only using httpd.conf.

+ Within httpd.conf, because we use extra config file, hence we include the extra file in the htttpd.conf by adding the following directive.

Include extra/httpd-ssl.conf

+ Within httpd-ssl.conf, we add and modify following directive:

Listen 50595

<VirtualHost *:50595>

DocuemntRoot "/home/SimpleLife/secure/wswt"

ServerName homesecure2171.bit.rmit.edu.vn

SSLEngine on

SSLCertificateFile "/home/SimpleLife/apache2-secure/ssl-key/assignment2.crt"

SSLCertificateKeyFile "/home/SimpleLife/apache2-secure/ssl-key/assignment2.key"

</VirtualHost>

This part above belongs to httpd-ssl.conf, and it only indicates how we set up SSL for our web server, besides that part, there are some more directives which we used to set basic authentication. But it is not belong to this paragraph.

VI. Describe the steps that you took to compile and install the Apache modules.

Firstly, I went to the **/home/SimpleLife/src** and unzip the **httpd-2.4.7.tar.gz** file

cd /home/SimpleLife/src

tar -xvzf httpd-2.4.7.tar.gz

Then I copy all modules in **httpd-2.4.7** folder into new folder called lib in the apache2-secure directory

cd /home/SimpleLife/apache2-secure/

mkdir lib

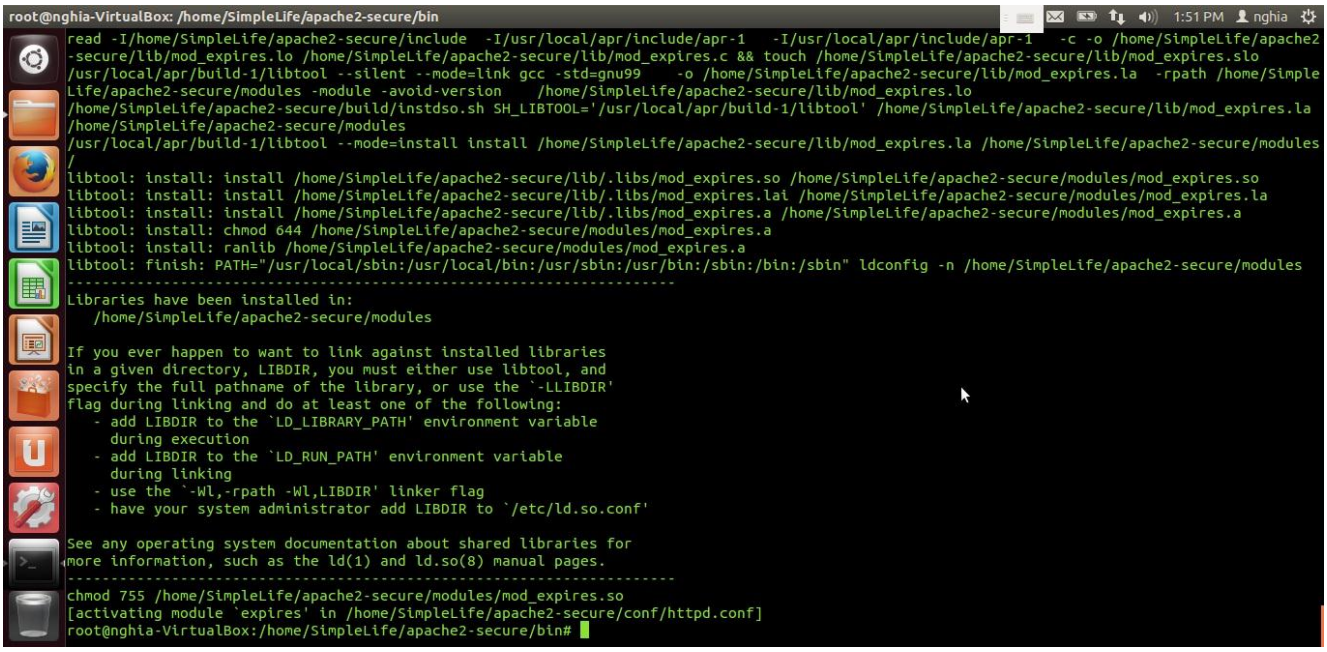
cd lib

cp /home/SimpleLife/src/httpd-2.4.7/modules/* /home/SimpleLife/apache2-secure/lib

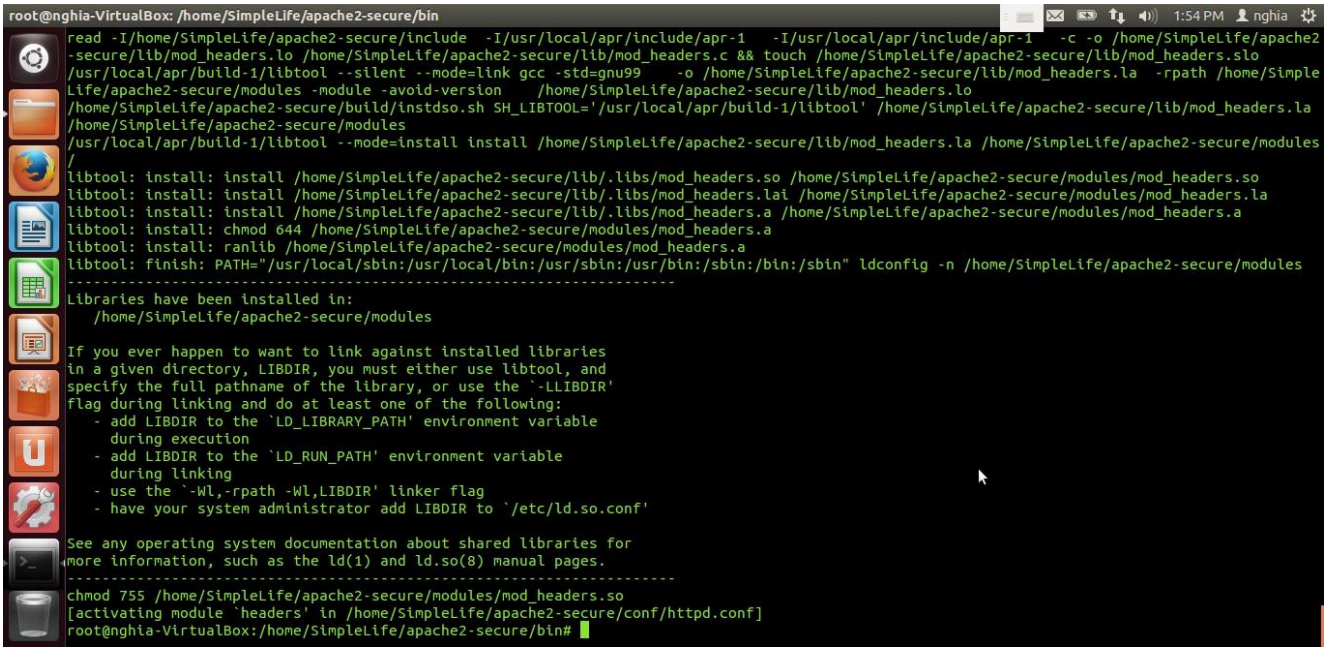
Then I went to the bin folder and compile the header mod and expire mod.

cd /home/SimpleLife/apache2-secure/bin

./apxs -i -a -c /home/SimpleLife/apache2-secure/lib/mod_expires.c



./apxs -i -a -c /home/SimpleLife/apache2-secure/lib/mod_headers.c



Then I restarted the server

./apachectl graceful

VII. Observation using Virtual OS on Virtual box

This is not the first time I use virtual Linux OS so that I don't have any problem using this OS. When I got any problems, I just looked up in the Internet to find the solutions. Especially, our group use Ubuntu as avirtual host which has a very big and strong community. Prior to go further with this assignment, we have trouble in installing apache web server. There are lacks of apr, pcre which are needed to compile apache web server. In order to solve this problem, it need time to work around, because as a ordinary Window users, from a single click or two, people can easily install program, it is not that difficult which is in Linux Operating System. Besides that, we also have trouble in using text editor, it is VIM, and it is not like other editors with IDE and took time to get cope with. The VIM editor is hard to use because the configuration for it to work wisely is awful if the programmer is a newbie in coding. Furthermore, the key binding and checking errors before compiling do not exist in

VIM so that it is very hard to debug.After that, we have no technical problem regarding to the requirement of this assignment.