

AfNOG 2012

Serekunda, Gambia – May 2012

Apache 2.2 with SSL, PHP, Mysql and Wordpress Exercises

1. Intro:

The packages to be installed are:

- MySQL 5.1
- Apache 2.2
- PHP 5

We will use ports. FreeBSD Ports and Packages Collection offers a simple way for users and administrators to install applications. There are currently 23559 ports available (as of May 2012)

2. Mysql51 Server Installation

```
$ cd /usr/ports/databases/mysql51-server/  
$ sudo make install clean
```

Edit /etc/rc.conf for MySQL to start add the line

```
mysql_enable="YES"
```

Start MySQL

```
$ sudo /usr/local/etc/rc.d/mysql-server start
```

Create root password

```
$ sudo mysqladmin -u root password afnog
```

The "newpassword" is the password of your choice. In this case we will use afnog.

3. Apache22 Installation:

Install apache. We will use a program called portmaster for this which will automatically resolve all the dependencies, present the dependency options and then proceed with downloading and installing

```
$ cd /usr/ports/ports-mgmt/portmaster  
$ sudo make install clean
```

Also install perl (which is the programming language used by Apache) using pkg_add like so:

```
$ sudo pkg_add -rv perl
```

Then install apache with -G option which means don't bring up the make config options

```
$ sudo portmaster -G www/apache22
```

Choose whatever default options are presented. Once installed, in `/etc/rc.conf`, add the following line

```
apache22_enable="YES"
```

To start apache run

```
$ sudo /usr/local/etc/rc.d/apache22 start
```

Check if the apache web server you have just installed works by pointing a browser to the server i.e. <http://<your-ip-address>>. You can also do

```
$ telnet localhost 80
```

4. Configuring Virtual Hosts

Edit the `httpd.conf` file:

```
$ sudo vi /usr/local/etc/apache22/httpd.conf
```

Uncomment the line below in the apache configuration file:

```
#include etc/apache22/extra/httpd-vhosts.conf
```

Ensure that Apache listens on port 80

```
# Listen for virtual host requests on all IP addresses  
# (both IPv4 and IPv6)  
Listen 80
```

Ensure that Apache server name is set not to bind to any DNS name listens on port 80

```
# To avoid binding to DNS Names or IP  
ServerName *:80
```

Edit `/usr/local/etc/apache22/extra/httpd-vhosts.conf` to define the virtual hosts. In the exercise “name” should be replaced with your name as in the DNS exercise.

```
NameVirtualHost *:80  
  
<VirtualHost *:80>  
  ServerAdmin webmaster@<name>.sse.ws.afnog.org  
  DocumentRoot /home/afnog/<name>  
  ServerName vm  
  <Directory /home/afnog/<name>  
    Order deny,allow  
    Allow from all  
  </Directory>  
</VirtualHost>
```

Create the directory “name” in the `/home/afnog` directory and give it the right permissions for apache to access it – in this case

```
$ cd /home/afnog
```

```
$ mkdir <name>
$ sudo chown -R www:www <name>
$ sudo chmod u+x <name>
```

NOTE: By default apaches Directory access permissions are restrictive to deny all. This requires that any directory access for apache outside the Document root should be explicitly set.

To allow from all from httpd.conf may resolve the problem, but is not recommended. The best option is as follows;

Create a file called name.conf (where name is same as “name” above for ease of management) in **/usr/local/etc/apache22/Includes/**

5. Configuring Secure Virtual Hosts (Apache+SSL)

To create a secure virtual host accessed via https rather than http, you will need to configure your Apache server to use OpenSSL for encrypting the data servered from the web server.

The following steps should do the trick.

5.1 Create the SSL Certificates for your Apache Web Server:

```
$ cd /usr/local/etc/apache22/
$ sudo openssl genrsa -des3 -out server.key 1024
```

NOTE: Password-Phrase is needed to encrypt the key. **However**, this pass-phrase will be needed at every apache restart. To get rid of the pass-phrase prompts at every apache restart and maintain the original key.

```
$ sudo cp server.key server.key.org
$ sudo openssl rsa -in server.key.org -out server.key
```

5.1.1 Create Certificate Request

```
$ sudo openssl req -new -key server.key -out server.csr
```

NOTE: The CommonName is the name of the Website you will use in this case the localhost name i.e `vmXX.sse.ws.afnog.org` where XX is your computer number

5.1.2 Self Sign your Own Certificate

```
$ sudo openssl x509 -req -days 3650 -in server.csr -signkey
server.key -out server.crt
```

5.2 Enable SSL in Apache

Edit the `httpd.conf` file and **uncomment** the line below;

```
#include etc/apache22/extra/httpd-ssl.conf
```

Edit the `httpd-ssl.conf` file and make the following changes:

```
$ vi /usr/local/etc/apache22/extra/httpd-ssl.conf
```

NOTE:

- Each virtual host must have its own certificate file see comments on “CommonName”.
- The path is where the certificate File and Keys are located in this case `/usr/local/etc/apache22/` (see virtualhost example below)

```
SSLCertificateFile /usr/local/etc/apache22/server.crt
SSLCertificateKeyFile /usr/local/etc/apache22/server.key
```

5.3 Create the Secure Virtual Host

At the end of the file (`httpd-ssl.conf`) add the virtual hosts that will be handled with SSL

```
<VirtualHost vmXX.sse.ws.afnog.org:443>
  ServerAdmin webmaster@<name>. sse.ws.afnog.org
  DocumentRoot /home/afnog/<name>
  ServerName vmXX.sse.ws.afnog.org
  ErrorLog "/var/log/<name>.sse.ws.afnog.org-error.log"
  TransferLog "/var/log/<name>.sse.ws.afnog.org-access.log"
  SSLEngine on
  SSLCertificateFile /usr/local/etc/apache22/server.crt
  SSLCertificateKeyFile /usr/local/etc/apache22/server.key
  <FilesMatch "\.(cgi|shtml|phtml|php)$">
    SSLOptions +StdEnvVars
  </FilesMatch>
  <Directory "/home/afnog/<name>">
    Options -Indexes FollowSymLinks
    AllowOverride AuthConfig FileInfo
    Order allow,deny
    Allow from all
  </Directory>
  <Directory "/usr/local/www/apache22/cgi-bin">
    SSLOptions +StdEnvVars
  </Directory>
  BrowserMatch ".*MSIE.*" \
    nokeepalive ssl-unclean-shutdown \
    downgrade-1.0 force-response-1.0
  CustomLog "/var/log/httpd-ssl_request.log" \
    "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
```

Copy the default **index.html** file to the directory from which you are going to serve your web pages.

```
$ sudo cp /usr/local/www/apache22/data/index.html  
/home/afnog/<name>/
```

Restart the Apache

```
$ sudo apachectl restart
```

Check if the apache-ssl virtualhost you have just setup works by pointing a browser to the desired URL
i.e. *https://vmXX.sse.ws.afnog.org*

6. Installing PHP & PHP Extensions

```
$ sudo portmaster lang/php5
```

**** During the installation, select Apache Option # “Build Apache Module” option**

Once its completes, proceed to add the following into the apache httpd.conf file to enable PHP in
apache

```
$ sudo vi /usr/local/etc/apache22/httpd.conf
```

Find directory index as below and add index.php to the end of the line

```
DirectoryIndex index.html index.htm index.php
```

Also find the Addtype section and add the 2 lines below

```
AddType application/x-httpd-php .php  
AddType application/x-httpd-php-source .phps
```

Copy the php initialization file that has the php features.

```
$ cp /usr/local/etc/php.ini-development /usr/local/etc/php.ini
```

Install php5-extensions, which provide support for various modules like MySQL and others.

```
$ sudo cd /usr/ports/lang/php5-extensions  
$ sudo make install clean
```

**** Select MySQL and IMAP support, once it is complete, restart Apache.**

```
$ apachectl restart
```

6.1 Test the PHP installation

Create PHP test page

```
$ vi /home/afnog/<name>/test.php

<?php
    $hostname = gethostbyaddr($_SERVER['REMOTE_ADDR']);
    echo "Your IP Address is $hostname";
    phpinfo();
?>
```

Point your browser to the following URL:

<http://vmXX.sse.ws.afnog.org/test.php>

7. Enable IPv6 in FreeBSD

As user root, edit /etc/rc.conf and add:-

```
ipv6_enable="YES"
```

Add (manual) interface configuration:-

```
ipv6_network_interfaces="em0"
```

Use the above option to limit the interface to enable IPv6. By default it's "auto", meaning to enable IPv6 on all the interfaces.

```
ipv6_ifconfig_em0="2001:43f8:0220:219::1XX"
```

(where x is your pc number, eg.1, 2, 3 ...)

Add the default router:-

```
ipv6_defaultrouter="2001:43f8:0220:219:196:200:219:254"
```

Save and exit the file then run:

```
$ sudo /etc/netstart
```

Test with the configuration changes with "tracroute6" and "ping6"

- To your neighbor's ipv6 address
- An external host e.g. tracroute6 www.afrinic.net

Open your browser and go to:

- <http://www.afrinic.net>

- [http://\[2001:610:240:a50::2\]/](http://[2001:610:240:a50::2]/)

or go to <http://ipv6.google.com> and www.kame.net

You can also do this after editing the `/etc/rc.conf`;

```
$ sudo ifconfig em0 inet6 2001:4348:0220:219::1XX prefixlen 64
$ sudo route add -inet6 default
2001:4348:0220:219:196:200:219:254

$ sudo /etc/rc.d/network_ipv6 start
```

The other option would be to reboot the system.