Exercises: FreeBSD: Apache and SSL: pre SANOG VI Workshop

January 12, 2005

Exercises

Note: The text '*pc#*' represents your command prompt. Commands are the text after '*pc#*'.

1.) Install Apache with SSL support [Top]

For this exercise you need to be root user.

We are going to install the Apache web server with Secure Socket Layer (ssl) support. We'll do this by going to the ports collection, finding Apache with the mod_ssl extension, using make and make install to compile the port and install it on your machine.

Steps to install Apache

```
pc# cd /usr/ports/www [Apache with ssl port is here]

pc# ls | more [Look for the Apache port we want]

pc# cd apache13-modssl [This is the exact Apache port we want]

pc# make
```

At this point you will see multiple screenfulls of information scroll by as the Apache web server with SSL support is being compiled on your machine. The process will take several minutes.

The final make message you will see is:

```
===>  Creating Dummy Certificate for Server (SnakeOil)
      [use 'make certificate' to create a real one]
```

The make process creates an initial digital certficate that is not signed and installs this for use with Apache.

Finish installing Apache

To complete the Apache with SSL installation process on your machine type:

```
pc# make install
```

The final message you receive indicates that some important Apache files can be found here:

- /usr/local/sbin/httpd [Apache server binary]
- /usr/local/etc/rc.d/apache.sh [Apache startup scripts]

Type the following:

```
pc# man httpd
```

And read until the end of the man page. At the end is a list are some more files that are important to the Apache web server. These include:

```
/usr/local/apache/conf/httpd.conf
/usr/local/apache/conf/srm.conf
/usr/local/apache/conf/access.conf
/usr/local/apache/conf/mime.types
/usr/local/apache/conf/magic
/usr/local/apache/logs/error_log
/usr/local/apache/logs/access_log
/usr/local/apache/logs/httpd.pid
```

At this point Apache is installed, but not running. Our next exercise will be to configure your machine to automatically start Apache at boot, and to start and stop Apache manually. We'll be using some of the files listed above for these exercises.

2.) Configure Apache to start at boot [Top]

For this exercise you need to be root.

Remember on Day 1 that you learned that third party software might install their startup scripts in /usr/local/etc/rc.d? This is the case for the Apache web server. If you look in /usr/local/etc/rc.d/ you will see a file named "apache.sh". This file is the startup script for the Apache web server.

Remember that startup scripts look in your various rc.conf files to see whether they have been enabled. This is the same with the startup script /usr/local/etc/rc.d/apache.sh. First have a look at this script to understand where it is looking to see if it should execute at system startup:

```
pc# less /usr/local/etc/rc.d/apache.sh
```

The first screen of information includes the follwing:

```
# Define these apache_* variables in one of these files:
#       /etc/rc.conf
#       /etc/rc.conf.local
#       /etc/rc.conf.d/apache
#
# DO NOT CHANGE THESE DEFAULT VALUES HERE
#
apache_enable="NO"
apache_flags="-DSSL"
apache_pidfile="/var/run/httpd.pid"
```

As you can see this script will check in /etc/rc.conf then /etc/rc.conf.local (deprecated), and then /tec/rc.conf.d/apache to see it Apache has been enabled.

Now let's edit the file /etc/rc.conf and add the line that will tell Apache to start each time you boot your machine:

```
pc# vi /etc/rc.conf
```

And, in this file you should add a line that reads:

```
apache_enable="YES"
```

To do this move your cursor down to the bottom of your file and add the line. Then save your file and exit. Here are the steps for this beginning after you have typed "vi /etc/rc.conf":

1. Move down to bottom of the file using the DOWN-ARROW
2. Press the "o" key to place yourself in input mode and add a new line
3. Type in "apache_enable="YES"
4. Press the ESCape key
5. Press colon, then 'w' then 'q' then ENTER, or ":wq: [ENTER]

Now you should be back at your system prompt (pc#).

Now that the file /etc/rc.conf includes the line apache_enable="YES" you should be able to use the script apache.sh in /usr/local/etc/rc.d/ to start, stop, verify, etc. the Apache server. Let's start the Apache web server by doing:

```
pc# /usr/local/etc/rc.d/apache.sh start
```

If you want to see the various things you can do with this script just type:

```
pc# /usr/local/etc/rc.d/apache.sh
```

And you should see:

```
Usage: /usr/local/etc/rc.d/apache.sh [fast|force|one](start stop restart rcvarstatus poll)
```

3.) Verify Apache and SSL functionality are working [Top]

For this exercise you can use your standard *username* account.

At the most simple level let's verify that the Apache web server daemon appears to be running. We can use the *ps* command to do this:

```
pc# ps auxw | grep httpd
```

Remember that Apache uses the actual binary file /usr/local/sbin/httpd to start the Apache web server as indicated by the final message during installation. That's why we grep'ed on "httpd" instead of "apache".

The output you should see will look something like this:

```
root     54884  0.0  0.9  5224 3296  ??  Ss   12:43PM   0:00.22 /usr/local/sbin/httpd -DSSL
www      54885  0.0  0.9  5264 3328  ??  I    12:43PM   0:00.01 /usr/local/sbin/httpd -DSSL
www      54886  0.0  0.9  5264 3328  ??  I    12:43PM   0:00.01 /usr/local/sbin/httpd -DSSL
www      54887  0.0  0.9  5248 3324  ??  I    12:43PM   0:00.01 /usr/local/sbin/httpd -DSSL
www      54888  0.0  0.9  5248 3328  ??  I    12:43PM   0:00.01 /usr/local/sbin/httpd -DSSL
www      54889  0.0  0.9  5248 3324  ??  I    12:43PM   0:00.01 /usr/local/sbin/httpd -DSSL
www      54890  0.0  0.9  5240 3308  ??  I    12:43PM   0:00.00 /usr/local/sbin/httpd -DSSL
root     54951  0.0  0.2  1476  796  p5  S+   12:59PM   0:00.01 grep httpd
```

Note that Apache runs with multiple instances of the httpd daemon. This is so that the web server can rspond to multiple requests more efficiently. Also notice that the first httpd daemon that starts runs as root, but subsequent daemons use the user "www" - This is to make the web server less vulnerable to attacks that might gain root access.

So, this shows you that Apache is running, but is it accessible to users with web browsers? To check for this you can take advantage of the "Lynx" text-based web browser that you installed on the first day. To do this type:

```
pc# lynx 127.0.0.1
```

To exit from Lynx you press "q" and then you answer "y" to the prompt "Are you sure you want to quit?"

You could also type any of the following:

```
pc# lynx localhost

pc# lynx pcN.presanog.org.bt

pc# lynx 202.144.139.N
```

Where "N" (in both cases) is the number of your machine. The address 127.0.0.1, by definition, is your machine. The name "localhost" should resolve to 127.0.0.1 and should be defined in the file /etc/hosts. The other two addresses work because you have been assigned an IP address and we are using DNS to resolve these addresses to a name. Clearly using "127.0.0.1" removes several layers of complexity and is the easiest to use for initial testing.

Now, what if you did not have the Lynx text-based web browser package installed? When you first installed FreeBSD this was not installed. It's possible you might be on a machine in the future and not have a web browser available, even though the machine is running a web server. From your exercises on Tuesday remember you can use telnet to verify if the web server is available. To do this type:

```
pc# telnet 127.0.0.1 80
```

If you get back something like:

```
Trying 127.0.0.1...
Connected to localhost.presanog.org.bt.
Escape character is '^]'.
```

This is a good indication that you have a web server working. Still, to be sure that this is not some other server running on port 80 you should go a step further. You could repeat our exercise from yesterday and view the initial page by doing:

```
^]                        [press CTRL key and ']' character to exit]

pc# cd                    [to go your home directory]

pc# script apache.txt

pc# telnet 127.0.0.1 80

GET / HTTP/1.0            [press ENTER]

host: localhost           [press ENTER twice]
```

```
pc# exit                 [to leave your script shell]
```

And you will see the initial Apache welcome page scroll by on your screen. Now that you saved the output of this session to the file ~/apache.txt (the '~' character is short for "home" in Unix) we can get some additional information.

Type the file apache.txt to your screen by doing:

```
pc# cd                   [to go your home directory]

pc# less apache.txt
```

In the first page of information presented you should see:

```
HTTP/1.1 200 OK
Date: Tue, 11 Jan 2005 05:40:20 GMT
Server: Apache/1.3.33 (Unix) mod_ssl/2.8.22 OpenSSL/0.9.7d
Content-Location: index.html.en
Vary: negotiate,accept-language,accept-charset
TCN: choice
Last-Modified: Mon, 10 Jan 2005 02:41:25 GMT
ETag: "41f6f-a71-41e1eb55;41e1ec2c"
Accept-Ranges: bytes
Content-Length: 2673
Connection: close
Content-Type: text/html
Content-Language: en
Expires: Tue, 11 Jan 2005 05:40:20 GMT
```

Notice that you can now see exactly what version of Apache is running, that it appears to be ssl-enabled and it is using OpenSSL 0.9.7d and mod_ssl to do this.

So, it appears that Apache is ssl-enabled on this machine, but how can we prove this? A web server with ssl support means that you can go to URL addresses that start with "https" (http secure). We could try this:

```
pc# lynx https://localhost/
```

But, you will receive an error message that reads:

```
Alert!: This client does not contain support for HTTPS URLs.

lynx: Can't access startfile https://noc/
```

That is because we did not install the Lynx package that includes ssl support. We could do this now, but instead we'll use a tool that comes with OpenSSL to allow us to make ssl connections, verify encryption in use, view certificates, etc.You can simply type "openssl" and then you will get a prompt where you can use the multiple openssl tools, or you can combine the command "openssl" with the various tools on your command line. This is what we will do using the openssl s_client tool. Try typing these commands:

```
pc# cd

pc# script ssltest.txt

pc# openssl s_client -connect localhost:443

pc# exit

pc# less ssltest.txt
```

And you will get several screens of information about your Apache web server, the ssl certificate that was installed by default and it's detailed information, what protocols are in use, and more.

As you can see a certificate installed by the "Snake Oil" organization has been installed on this server and the protocol in use is TLSv1.

Now that we know that Apache is up and running with both secure (SSL/TLS) and insecure support we can make a change to our default configuration, which we'll do in the next exercise.

4.) Change the default DocumentRoot for Apache [Top]

For this exercise you need to be root.

Now that Apache is installed maybe you don't want your web server to use the default directory location and files that come with Apache. By default if you go to http://localhost/ Apache will look in the directory /usr/local/www/data under FreeBSD. If you look in this directory you will see a number of files:

```
pc# ls /usr/local/www/data
```

These files are designed to display the initial Apache welcome page in the language you are using on your machine.

Let's use a different directory as the default directory that Apache will use. To do this we need to edit the file /usr/local/etc/apache/httpd.conf and change the location where the configuration setting DocumentRoot points. We'll create a directory /u/apache for this exercise and create an index.html file in this directory. First let's do this:

```
pc# mkdir /u/apache

pc# echo "Welcome to my new home page!" > /u/apache/index.html
```

Now let's update the Apache httpd.conf configuration file so that the web server will automatically look in /u/apache and display the file /u/apache/index.html when you go to http://localhost/.

```
pc# vi /usr/local/etc/apache/httpd.conf

press "/" and type in "DocumentRoot" then press ENTER
```

Scroll down a few lines and change the line that reads:

```
DocumentRoot "/usr/local/www/data"
```

To read:

```
DocumentRoot "/u/apache"
```

Remeber it's "i" for input mode, ESCape to get out of input mode, and "x" to delete characters.

Make sure you don't include a trailing "/". Now you need to save and exit the file:

```
Press ESCape and then ":" then wq and [ENTER]
```

Now you need to restart the Apache web server so that it will read /usr/local/etc/apache/httpd.conf and recognize the change you just made. To do this type:

```
/usr/local/etc/rc.d/apache.sh restart
```

This will restart the server (current connections are dropped). Now go to your machine's new default home page by using Lynx and typing:

```
lynx http://localhost/
```

And, you should see "Welcome to my new home page!" in your Lynx window. You can exit Lynx by pressing "q" and answering "y" to the prompt Are you sure you want to quit?"

5.) View the online Apache documentation [Top]

When Apache is installed it includes a fair amount of documentation that can be quite useful. You can view this documentation in your Lynx web browser by doing this:

```
pc# lynx http://localhost/manual/index.html
```

And, since we have installed SSL you may want to view the online mod_ssl documenation for Apache. You can do this by typing:

```
pc# lynx http://localhost/manual/mod/mod_ssl/
```

The physical address for the Apache manuals collection is is /usr/local/share/doc/apache/. When you go to http://localhost/manual/index.html you are redirected to this directory. There is a setting in your Apache configuration file (/usr/local/etc/apache/httpd.conf) that tells the web server to map this directory to the URL http://localhost/manual/. If you type this:

```
pc# grep manual /usr/local/etc/apache/httpd.conf
```

You will see this:

```
# This Alias will project the on-line documentation tree under /manual/
Alias /manual/ "/usr/local/share/doc/apache/"
```

This is a configuration setting in Apache known as an "Alias" to redirect a URL directory to a different physical address.

Hervey Allen
January 2005