APACHE HTTP SERVER 2.2.8

HTTP://HTTPD.APACHE.ORG

SUMMARY

Apache HTTP Server is an open source web server application regarded as one of the most efficient, scalable, and feature-rich web servers available. Apache is also highly customizable, with numerous third-party modules available to extend its functionality. You'll find modules that add support for SSL (Secure Sockets Layer) encryption over HTTP (HyperText Transfer Protocol), PHP support (PHP is a server-side scripting language), and authentication support for password protecting a site or page.

Apache was born at the NCSA (National Center for Supercomputing Applications) in 1993 when Rob McCool developed a public domain HTTP *daemon* (background process) that would later become the foundation of the Apache project. Apache version 1.3 still contains code from the original NCSA-developed HTTP daemon, while version 2 was rewritten from scratch and contains no NCSA code.

Apache HTTP Server is installed on nearly 53 percent of the world's web servers. Microsoft's Internet Information Server is in second place, with more than a 32 percent share of the server market.[1]

The FreeBSD port of the Apache HTTP server documented here includes support for SSL over HTTP using the `mod_ssl` module. The module was created by Ralf S. Engelschall in 1998; it is based on software developed by Ben Laurie.

RESOURCES

Official Apache HTTP Server Online Documentation
http://httpd.apache.org/docs/2.2

RFC 2616 – Hypertext Transfer Protocol – HTTP/1.1
http://tools.ietf.org/html/rfc2616

REQUIRED

- ☑ FreeBSD 7.0-RELEASE (see "FreeBSD 7.0" on page 9)
- ☑ Updated ports collection (see "FreeBSD Ports Collection" on page 23)
- ☑ Internet connection

[1] Netcraft Ltd., "Netcraft: July 2007 Web Server Survey," *http://news.netcraft.com/archives/2007/07/09/july_2007_web_server_survey.html*

APACHEHTTP
FreeBSD port path: /usr/ports/www/apache22
TCP ports used – HTTP (80), HTTPS (443)

OPTIONAL

- ☑ OpenSSL with a signed SSL Certificate (if you wish to enable secure HTTP connections; see "OpenSSL 0.9.8g" on page 127)
- ☑ Registered domain name

PREPARATION

Become the superuser and then make sure your server's hostname is resolvable locally. This should already be the case if you are running your own DNS server and have it configured properly.

If you aren't running your own DNS server, make sure you have an entry in your */etc/hosts* file that points to your server's IP address; this will ensure that your server's hostname is resolvable locally. To do so, open the *hosts* file in a text editor:

```
# ee /etc/hosts
```

You should find something like the following around line 14 of the */etc/hosts* file; replace *example.com* with your domain name, *host.example.com* with your hostname, and *192.168.1.11* with your server's local IP address:

```
::1             localhost localhost.example.com
127.0.0.1       localhost localhost.example.com
192.168.1.11    host.example.com
```

INSTALL

To begin the Apache installation process, enter the following commands:

```
# cd /usr/ports/www/apache22
# make config ; make install clean
# rehash
```

A menu should appear showing options for Apache. We'll leave the settings at their defaults, so press [TAB] to highlight **OK** and then press [ENTER].

CONFIGURE

Once the installation process completes, it is time to configure Apache for use on your system.

1. Open the *httpd.conf* file located in */usr/local/etc/apache22*:

```
# ee /usr/local/etc/apache22/httpd.conf
```

2. We'll edit a few entries in *httpd.conf* to get the HTTP daemon up and running. To do so, scroll to the `ServerAdmin` declaration (~138) and replace

you@example.com with the email address of the person who will be maintaining the server. The line should appear as follows:

```
ServerAdmin you@example.com
```

3. Scroll to the ServerName declaration (~147), *uncomment* (remove the leading hash mark), and replace *host.example.com*:80 with the hostname of your server. The line should now appear as follows:

```
ServerName host.example.com:80
```

NOTE *If you do not wish to set up SSL over HTTP, save, exit, and proceed to "Testing" on page 36.*

4. To enable SSL support, uncomment the Secure (SSL/TLS) connections declaration (~449) by removing the hash mark. The line should now appear as follows:

```
Include etc/apache22/extra/httpd-ssl.conf
```

5. Save, exit, and open Apache's SSL configuration file:

```
# ee /usr/local/etc/apache22/extra/httpd-ssl.conf
```

6. Scroll to the ServerName and ServerAdmin declarations (~78) and replace *host.example.com* with your server's hostname. Replace *you@example.com* with the email address of the person who will be maintaining the server. The two lines should now appear as follows:

```
ServerName host.example.com:443
ServerAdmin you@example.com
```

7. Go to the SSLCertificateFile declaration (~99) and input the path and filename of your server's SSL Certificate. The line should now appear as follows (replacing the path and filename shown in italics with the path and filename of your SSL Certificate):

```
SSLCertificateFile /usr/local/openssl/certs/host.example.com-cert.pem
```

8. Go to the SSLCertificateKeyFile declaration (~107) and input the path and filename of your server's private SSL key. The line should now appear as follows (replacing the path and filename shown in italics with the path and filename of your private key):

```
SSLCertificateKeyFile /usr/local/openssl/certs/host.example.com-unencrypted-key.pem
```

NOTE *It is highly recommended that you specify your unencrypted key file here. An encrypted key file will cause Apache to prompt for a password that may interfere with the startup of other critical services. See "OpenSSL 0.9.8g" on page 127 for details on decrypting your key file if you haven't already done so.*

9. Save and exit.

APACHEHTTP
FreeBSD port path: /usr/ports/www/apache22
TCP ports used – HTTP (80), HTTPS (443)

TESTING

In this section, we'll perform some basic tests to confirm that Apache answers HTTP requests properly.

1. Apache includes a utility called apachectl that is capable of testing your configuration files for syntax errors. Let's run this program to check for syntax errors:

```
# apachectl configtest
```

If apachectl returns `Syntax OK`, continue below. If apachectl finds a problem, it will list the filename, line number, and possible reasons for the error. Be sure to resolve any issues prior to continuing below.

2. We'll configure Apache to start automatically at boot time. To do so, open the *rc.conf* file located in */etc*:

```
# ee /etc/rc.conf
```

Then add the following lines in */etc/rc.conf*:

```
apache22_enable="YES"
apache22_http_accept_enable="YES"
```

Save, exit, and start Apache with this command:

```
# /usr/local/etc/rc.d/apache22 start
```

3. You can choose to conduct your tests via a standard web browser, but the instructions below will conduct tests via the command line. Enter the following commands to connect directly to the HTTP service listening on port 80:

```
# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

The `GET` command below is case sensitive. Be sure to press [ENTER] twice and input a space before and after the first slash:

```
GET / HTTP/1.0
HTTP/1.1 200 OK
Date: Sat, 01 Mar 2008 02:00:30 GMT
Server: Apache/2.2.8 (FreeBSD) mod_ssl/2.2.8 OpenSSL/0.9.8g DAV/2
Last-Modified: Sat, 20 Nov 2004 20:16:24 GMT
ETag: "cf597-2c-4c23b600"
Accept-Ranges: bytes
Content-Length: 44
Connection: close
Content-Type: text/html
<html><body><h1>It works!</h1></body></html>Connection closed by foreign host.
```

If you see the `It works!` text in the last line of output then Apache does indeed work!

4. If you configured Apache SSL support, enter the following command to connect to the HTTP server via SSL:

```
# openssl s_client -connect localhost:443
```

The `GET` command is case sensitive; press [ENTER] twice:

```
GET / HTTP/1.0
```

The output should be identical to the output you received over the unencrypted connection.

UTILITIES

Following is brief information on the apache22 script that should be used to control the Apache daemon on FreeBSD.

apache22

This script is used to control the HTTP daemon.

Command `/usr/local/etc/rc.d/apache22`

Syntax `/usr/local/etc/rc.d/apache22` *option*

Options

`start` Launches the Apache HTTP server

`stop` Stops the Apache server

`configtest` Parses the configuration files for syntax errors

`restart` Restarts the Apache server

Example

To start the HTTP daemon, issue the following command at the prompt:

```
# /usr/local/etc/rc.d/apache22 start
```

NOTE *`apache22_enable="YES"` must exist in* /etc/rc.conf *in order for this script to function.*

CONFIG FILES

/usr/local/etc/apache22/httpd.conf
The Apache HTTP server's main configuration file

/usr/local/etc/apache22/extra/httpd-ssl.conf
The Apache HTTP server's SSL configuration file

APACHEHTTP
FreeBSD port path: /usr/ports/www/apache22
TCP ports used – HTTP (80), HTTPS (443)

LOG FILES

/var/log/httpd-access.log
Contains a log of IP addresses, times, and activity on the HTTP server

/var/log/httpd-error.log
Contains a log of error messages produced by the HTTP server

/var/log/httpd-ssl_request.log
Contains a log of IP addresses, times, and activity on the HTTPS server

NOTES

- The HTTP server uses the default port 80 for non-secure communications. The SSL-enabled HTTP server uses the default port 443 for secure communications. If you are behind a NAT router, be sure to forward these ports to your server. (Refer to your router's documentation for details on port forwarding.)
- The default root folder of the HTTP server is */usr/local/www/apache22/data*. Web content must be placed here unless you change the document root directory in *httpd.conf*.