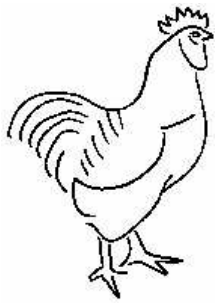
**CLTC Documentation Sheet 6:**

# Configuring the CLTC Web Server



Developed by The 'Free Range' Community-Linux Training Centre Project -
Version 1.0, January 2003. http://www.fraw.org.uk/cltc/

"*Cyberspace: A consensual hallucination experienced daily by billions of legitimate operators, in every nation. A graphic representation of data abstracted from the banks of every computer in the human system. Unthinkable complexity. Lines of light ranged in the nonspace of the mind, clusters and constellations of data. Like city lights, receding...*"
William Gibson, *Neuromancer*

**This sheet looks at the configuring the CLTC's web server; both the primary web server, and the configuration of virtual servers. It also considers the enabling of the various processing and scripting hooks, such as *server includes*, *CGI*, *Perl* and *PHP*, that will be used later to extend the functionality of the web applications used on the system.**

## The Net's second killer app...

For the military-industrial complex it was email that made the Internet interesting enough to put money into its development. The 'killer application' that was the 'Net's raison d'être However, it was the World-Wide Web, randomly linking data on thousands of computers with a really simple and friendly user interface, that made the Internet a public media.

Before the Web, in the early 90s, those using the Internet had to be geeks, or people paid to do a job, to have the time and determination to master the terse language of network communications protocols. Now many people think that the Web *is* the Internet – a misconception that limits many people's ability to use online services.

Web servers work by using 'mark up' tokens inserted into plain text files. These tokens – collectively called 'Hyper-Text Markup Language' (HTML) – format the text. The change the size, the font and the colour. It also lays out the with images, tables and other graphical objects. But that makes web pages so effective is linking. Information, images or other multimedia objects can be linked together into networks or 'webs' of information. This enables information, that appears to in one place, to actually be located across many machines, or even in different countries or continents.

A development from the early (and boring) web pages was 'scripting' – the insertion of programming or scripting languages that make the pages 'dynamic', able to react the conditions that they are displayed under.

The Web is a very important part of the Internet. However, we must be careful not to work as if it were the Internet. Hypertext Transfer Protocol (HTTP) is just one of a number of protocols that have an important role in providing information services electronically.

## The Apache server

Apache is one of the popular web servers. It's a large and very powerful program. To configure it well for handling important web services is difficult. But it is fairly simple to

configure it for use on small-scale projects like the CLTC.

The first important thing to consider is how Apache handles a web site. Every web site is located on a server. That server has an IP address. But there are three ways we can configure a web site to be accessed from a network:

- *A single server* – a single web site that works on a server, responding to any requests for HTTP services on that server.
- *A number-based virtual server* – a 'virtual' site, existing as perhaps one of many other virtual sites on the same server, but which responds to requests for a HTTP services to a specific IP number.
- *A name-based virtual server* – a virtual server that responds to a different name sent to a single IP address.

Most commercial web services work on the latter system, the virtual name-based site. This is because IP addresses in the real world are very expensive, whilst configuring another virtual server takes a couple of hours of a person's time. On the CLTC we use all three of these systems in order that we can teach the use of them all.

Every web site has a 'document root' – a directory on a file system that contains the files that make up the web site. In addition to this there are other directories associated with the document root that can contain 'dynamic content' – the programs or scripts that give greater functionality to web sites. These can be small scripts using languages such as CGI or Perl, or extensions to the HTML language using PHP (which is another large programming language working as part of the Apache server).

## Initial Configuration

To configure the server we first need to start it up. Firstly, use the `ntsysv` program to enable the `httpd` daemon at boot time. Next, we start up the daemon using the command:

```
/etc/rc.d/init.d/httpd start
```

Provided that DNS is working (if it isn't you'll have to use the numeric IP address of the server) you can now access the web server using a browser to the address `http://www.cltc.lan/`. What you get is a 'test page' telling you that the server has not been configured.

The Apache server is controlled by the file `/etc/httpd/conf/httpd.conf`. This is a large file that controls the functions of the 'single' server, as well as containing the configuration for all the name- and number-based and virtual web sites.

To configure the server we need something to point the sever at. Therefore we have to configure a few web pages. All we need is just a blank page with a heading, the heading being changed to identify each site we save the file to. The location of each web site on the CLTC's file system is:

| | |
|---|---|
| CLTC main site | `/var/www/html` |
| Alphaweb site | `/var/www/alphaweb` |
| Betaweb site | `/var/www/betaweb` |
| Gammaweb site | `/var/www/gammaweb` |
| Deltaweb site | `/var/www/deltaweb` |
| FRAW site | (see below) |
| Examples site | (see below) |

The CLTC site operates as a 'single' server. The 'user' web sites are number-based virtual sites. The 'FRAW' and 'Examples' sites are name-based sites that work off the general purpose IP address, '192.168.66.14. These have to be configured first as we have to make some changes the domain name system (we could have done this whilst setting up DNS, but it's more illustrative to outline it here).

First of all we need to create the directory for the FRAW virtual site, and set up the correct permissions for it:

```
mkdir /var/www/fraw
chown -R mobbsey.users
          /var/www/fraw
chmod -R 611 /var/www/fraw
```

Repeat this process again to create the 'Examples' site, but use the directory name 'examples' and the user ID 'epsilon'.

Now we have to make a forward zone file, and an entry into *named.conf*, in order to create a DNS entry for the domain '*fraw.lan'*. The forward zone file can be created by copying

and editing the forward zone file of one of the other virtual servers. We do the same to create a forward zone for '*examples.lan*'.

There is no reverse zone file, since the subnet of the IP address already belongs to the domain '*cltc.lan*'. But we do add another line to the CLTC's reverse zone file to point '14' back at *cltc.lan*. When we've finished these changed we restart the name server, and check that the FRAW domain functions, using the commands:

```
/etc/rc.d/init.d/
   named restart
host www.fraw.lan
```

...and you should get a the address 192.168.66.14 (likewise if you try this with *www.examples.lan*).

**'Alphaweb' virtual host configuration**

```
<VirtualHost 192.168.66.10>
     ServerAdmin   alpha@cltc.lan
     DocumentRoot  /var/www/alphaweb
     ServerName    www.alphaweb.lan
     ErrorLog      /var/log/httpd/alphaweb_error
     CustomLog     /var/log/httpd/alphaweb_log
     <Directory />
       Options Indexes Includes FollowSymLinks
       AllowOverride None
     </Directory>
     ScriptAlias /cgi/ "/var/www/alphaweb/cgi/"
     <Directory "/var/www/alphaweb/cgi">
       AllowOverride None
       Options ExecCGI
       Order allow,deny
       Allow from all
     </Directory>
</VirtualHost>
```

## Creating the virtual servers

The best introduction to setting up the Apache server is to configure the virtual servers first. Open *httpd.conf* and scroll about four-fifths of the wat through to '*Section 3: Virtual Hosts*'.

Ignore the examples given in the file. Just before the next part of the file (that deals with SSL), open up a few blank lines by pressing return. Then insert the virtual host example for '*alphaweb*' shown in the box above.

Also, before activating this configuration, you need to create the scripting directory for the 'alpha' virtual server:

```
mkdir /var/www/alphaweb/cgi
chown -R alphaweb.users
                /var/www/alphaweb/cgi
chmod -R 777
/var/www/alphaweb/cgi
```

Now you can activate the new configuration by restarting the Apache server:

```
/etc/rc.d/init.d/
      httpd restart
```

Using a browser to access the address *http://www.alphaweb.lan/* you should see the *alphaweb* index page created earlier. If you copy a CGI test program (see left for an example) into the `alphaweb/cgi` directory you can also test that the scripting alias has been correctly configured.

**Example of a CGI test program**

```
#!/usr/bin/perl
print "Content-type: text/html", "\n\n";
print "<HTML>", "\n";
print "<HEAD></HEAD>", "\n";
print "<BODY><H1>CGI Test Program</H1>", "\n";
print "<HR><PRE>";
print "Name: ", $ENV{'SERVER_NAME'},
          "<BR>", "\n";
print "Port: ", $ENV{'SERVER_PORT'},
          "<BR>", "\n";
print "CGI Rev.: ", $ENV{'GATEWAY_INTERFACE'},
          "<BR>", "\n";
print "<HR></PRE>", "\n";
print "</BODY></HTML>", "\n";
exit (0);
```

You can now repeat the process above to create the virtual web servers for *betaweb*, *gammaweb* and *deltaweb*.

Note that each of the virtual server sections contains all the important configuration options that are set in the configuration of the main CLTC web site. Next we need to change the setting in '*section 2*' of the *httpd.conf* file to setup *www.cltc.lan* (which starts about a quarter of the way into to the file).

First, find the line '*ServerAdmin*'. Reset the email address to `root@cltc.lan`. A little later you'll find the line '*#ServerName*'. You need the remove the # at the beginning of the line, and then change the server name from `localhost` to `www.cltc.lan`.

Finally, we need to create the name-based server for FRAW. Go to the beginning of the 'virtual servers' section that you have just create and insert the following line:

```
NameVirtualHost 192.168.66.14
```

Now copy the virtual server configuration after this line, renaming the directory location to `/var/www/fraw`, the logs to '*fraw_*' and the email address to `mobbsey@cltc.lan`. Then set the IP address in the virtual host tag to `192.168.1.14:80`, and the server name to `www.fraw.lan`. Do not copy the lines related to the script alias and the CGI directory as they are not required.

Now copy the section you created for FRAW, and paste in into the file after FRAW. Now rename the '*fraw*' parts to '*examples*', and finally change the server name to `www.examples.lan`.

Save the configuration file, restart the Apache server again, and this time browse the address `http://www.fraw.lan`. If all's well you should see the index page you created earlier. Then try `http://www.examples.lan/`.

What all this gives us is a main CLTC site that runs off a single address, four number-based virtual servers that run off their own addresses, and two named-based virtual servers that run off the same address. That completes the configuration of the virtual servers.

**An example *phpinfo* file**

```
<HTML>
<HEAD><TITLE>PHP Info Page</TITLE></HEAD>
<BODY BGCOLOR="#ffffff" TEXT="#000000">
<!php phpinfo(); ?>
</BODY>
<HTML>
```

## Configuring web-based services

Various dynamic functions have been automatically configured with Apache. The server-side includes have been enabled by the use of the keyword '*Includes*' in the '*options*' line for each virtual server.

The last thing to check with the pre-set configuration is whether PHP has been installed correctly. To do this we create a page to get PHP to spit out its configuration data (see the example in the box above). We save this file at `/var/www/html/info.php`, and then view the content by browsing the address `http://www.cltc.lan/info.php`. If all's well you'll see the configuration of the PHP system.

Now we are going to configure `Webalizer`. This is a program that generates usage statistics for web servers. The complication is that we must generate statistics for each individual virtual server – but by default `Webalizer` runs stats for all the sites configured for the Apache server. Therefore we need to modify a number of files. For this process we will use the *www.fraw.lan* virtual server as an example.

Firstly, we create a directory in the FRAW files area to hold the web logs:

```
mkdir /var/www/fraw/webalizer
```

Now we have to create a `Webalizer` configuration file. There is an example of a

configuration file for the FRAW site in the box on the right. We save this in the *webalizer* directory we created as `webalizer.conf`.

**An example `webalizer.conf` file for the FRAW virtual server**

```
LogFile         /var/log/httpd/fraw_log
LogType         web
OutputDir       /var/www/fraw/webalizer
HistoryName     /var/www/fraw/webalizer/webalizer.hist
ReportTitle     Usage Statistics for
HostName        www.fraw.lan
PageType        htm*
PageType        cgi
PageType        shtm*
PageType        php*
PageType        pl
HourlyGraph     no
HourlyStats     no
```

We now have a basic configuration to run `Webalizer` from the command line (this has been split to fit in the line):

```
webalizer -n www.fraw.lan -c
/var/www/fraw/webalizer/
          webalizer.conf -Y
```

If this works you should get no errors reported, and in the *webalizer* directory you should find a number of new files – including a file called `index.html` which contains the index file for the `Webalizer` statistics data for *www.fraw.lan*.

`Webalizer` needs to run automatically. It does this using a '*cron*' job. We now have to edit the *daily.cron* file to make it run `Webalizer` for the FRAW server only, and not its default settings.

First we have to open the file `/etc/cron.daily/00webalizer`. Then we replace the line `/usr/bin/webalizer`, substituting it for the command we used above to test run `Webalizer`. Then we save the file. `Cron` will now run `Webalizer` for the FRAW virtual server.

We now repeat this process for the other virtual servers. All that's required is changing the directory paths in the *webalizer.conf* files to point at the `Webalizer` stats directory and log files of the other virtual servers. Then we have to change the name of the virtual server and the directory path to the *webalizer.conf* file when we run `Webalizer` from the command line. Finally, we open the `Webalizer` *cron.daily* file and copy the current line, then paste it in as a new line after the previous command. Then we just amend this new line to point at the correct files and give the correct virtual server name.

This completes the configuration of `Webalizer`. Each virtual server on the CLTC system now has a web logs system.