

Comp 7006 - Lab #2

Configuring the Apache Server

Objective: To learn how configure the Apache server.

- The world's most used HTTP server; it is used by more Internet web sites than all other commercial web servers combined.
- Apache is based on free NCSA code, which was "patched" so heavily it was referred to as "apache webserver".

Concepts and Background

- The standard document route is:

/var/www/html

- The document root can also be specified in:

/etc/httpd/conf/httpd.conf

- Place your web content in this root directory.
- ***/etc/rc.d/init.d/httpd*** is the command used to control the Apache daemon.

Configuration Issues

- In ***/etc/httpd/conf/httpd.conf*** you will find the main configuration file for Apache:
- Examine the ***/etc/httpd/conf/httpd.conf*** file. Some key parameters that can be changed are:
 - You can turn host name lookups on.
 - The listening port can be changed from 80.
 - You can change the number and format of the logs.
- Apache has an access control scheme that can restrict which users can get to a particular web page.
- Look at the configuration information in ***/etc/httpd/conf/httpd.conf*** .

Step 1. Getting Started

- Install the Apache rpm if it is not installed yet.
- Configure the service and make sure the apache daemon is running:

/etc/rc.d/init.d/httpd status

- If necessary start it with:

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

- Direct Netscape to <http://localhost>. The page served by **httpd** is a generic page included in the Red Hat Linux distribution.
- Direct your web browser to your neighbor's website. See what you get back.
- Examine **httpd.conf**. the **DocumentRoot** directive specifies where the documents for the main website reside.
- Examine the stanza that governs access to different web directories. Notice the access control scheme that is implemented.
- Try modifying the generic page and then access it.

Step 2. Creating Web Site Accounts

- Note that in order for user account web access to function properly you will have to comment out the "**UserDir disable**" macro and uncomment the "**UserDir public_html**" macro in **httpd.conf**.
- Create a user account (**foo**) that will be used a web site from which to distribute documents.
- Log in as the user and create a directory called "**public_html**".
- This will now be the default document root directory. Create a document called "**index.html**" and place it in that directory.
- Test access to that web site from another machine as follows (assume my user account is called "**foo**"):

<http://192.168.0.xxx/~foo/>

- You should see the default document that you created.

Step 3. Adding password access to your site.

- To request authentication for your document tree to users within your network, just modify your **httpd.conf** configuration file to have a stanza like this:

```
<Directory /home/foo>
AllowOverride None
AuthUserFile /var/www/html/passwords/foobar
# Group authentication is disabled
AuthGroupFile /dev/null
AuthName course_index
AuthType Basic
<Limit GET>
    require valid-user
    order deny,allow
    deny from all
    allow from all
</Limit>
</Directory>
```

- The **order**, **deny**, and **allow** directives limit who will get a login panel.
- If you want users to be able to use your server from outside your network, just omit these directives.
- Otherwise, just replace **domain** with the domain name for your organization, or better yet, specify your domain by using an IP address notation.
- If you replace **domain** with **all**, every user will get a password panel displayed on their browser.
- Next, you will need to create the password directory on your **httpd** tree:

mkdir /var/www/html/passwords

- Make sure the passwords directory is readable by user or group your server runs under.
- The tools you use to manage the password file depend on the type of authentication you use. If you are using flat files, you will use the **htpasswd** program. If you are using database files, you'll use the **dbmmanage** script.

- The **htpasswd** program has following syntax:

htpasswd [-c] passwordfile username

- The **-c** flag creates the password file **passwordfile**. Here's a sample session:

```
# cd /var/www/html/passwords
# htpasswd -c foobar foo
Adding password for foo.
New password:
Re-type new password:
```

- The passwords won't be displayed on the terminal as you type, so as a security measure, **htpasswd** will ask for the password twice.
- You can create as many password files as you like. However, you'll have to use different filenames to reference them.
- Test and show that your password access is functional.