Gentoo – Protecting Web Directories With .htaccess

In this guide I will show you how to protect your localhost web root with .htaccess and .htpasswd files. By doing this a username and password will be required to enter the site. Let’s get started! As root do the following:

Ok now we need to generate an encrypted user/passwd combo for our /etc/apache2/httpd-passwords file so do the following:

This will prompt you for a password and then again for a confirmation of the password. Once it is finished it will automatically create the entry in the /etc/apache2/http-passwords file.

Here is an example entry for the user root with the password of 12345:

Now type and insert the following lines into the file:

Thats it! Now when you go to http://your_server_ip you will be prompted for a username and password. There is so much more that you can do with .htaccess files, this is just a start. Try doing a google search on .htaccess and do some experimenting!

Gentoo – PHP

In a previous guide I discussed how to set up an Apache web server on your Gentoo box. In this part I will show you how to set up PHP so it works with Apache and MySQL which will allow you to host dynamic content rich websites on your web server. Plus by having MySQL, Apache, and PHP installed and working together you have access to a whole slew of web based tools that make managing your server much simpler.

In order to install PHP we must do the following as root:

Save the file and type:

PHP will take a while to compile even on a very fast PC so take a break and go do something for a half hour to an hour …

It's done? Ok, now we need to make sure that apache is set up to use PHP. So as root type:

In this file you will find a line that reads:

At the end of this line, still in the quotes add: -D PHP5 if it is not there already. The newer PHP ebuilds have been adding this automatically as of lately, but check it just to be sure. Once you have added or verified that it is there then save the file and exit. Then restart apache:

Now lets test to make sure that PHP is installed correctly. We will put a test file in our localhost folder that will give us a ton of information about our PHP setup. So as root again do the following:

- In the editor type the following
- Save the file and exit.
- In a web browser on another PC on the local network type

the following: http://server_ip_address/phpinfo.php
- If PHP is installed correctly you should see a fancy PHP info page with a lot of information.

Thats it! Simple aye? Try installing PHP and get it working that fast on a Windows/IIS server, hehe good luck!

Gentoo – Apache Web Server

In a previous guide I discussed how to set up MySQL on your Gentoo Linux box. One great use for MySQL is to combine it with an Apache web server and PHP to create content rich, dynamic websites. In this guide I will discuss how to set up an Apache web server with vhosts support.

Installing Apache is pretty straight forward, there are a few configuration files that need to be edited but nothing to difficult. First lets install apache:

Once apache finishes compiling and installs star the service by typing:

Then set it to the default runelvel:

We also need to add the apache2 use flag to our /etc/make.conf file. As root type:
and add apache2 to your list of use flags. Then check to see if any packages need to be recompiled with the new use flag.

Now we need to make some changes to the config files. Still as root type:

At the very end of this file add the line to stop apache from spamming about the server name every time the service is started. Restart the apache service:

By default apache looks in /var/www/localhost/htdocs for your web files. You should now be able to open a web browser on another pc on your network and type: http://enter_your_server's_ip_here, if you see the apache test page you are good to go. You can now delete all of the files in the /var/www/localhost/htdocs folder and replace them with some .html or .htm files.

This is fine if you only want to host one website on your server, but apache is much more powerful than that. I prefer to use the default location for development and custom tools, and set up virtual hosts for all of my websites. Apache has a config file called 00_default_vhosts.conf which is where you can specify the locations of other websites you wish to host on your server.

At the end of this file you can add as many virtual hosts as you would like. Each one would correspond to a domain name that you own that points to your IP address. This way you can keep the localhost folder access limited to the internal network only (cause you don’t want everyone to have access to your web server stats and other tools), and specify virtual hosts that will be accessed from the web via port 80.

Here is how you do it:

- First you will need to create the directory for your new virtual host in the /var/www folder. You will also want to include the same folders that are in the localhost folder such as cgi-bin, icons, htdocs, and error.
- Log in as root and do the following:
- Create the directory for the new domain:

- Copy the necessary directories from localhost
- If you had any files in htdocs that copied over just delete them in the new htdocs directory.

Now you need to edit the /etc/apache2/vhosts/00_default_vhosts.conf and add your new virtual host entry to the end of the file. So as root type:

Here is an example of what a virtual host entry would look like:

Once you finish adding your new vhost entry you need to restart apache:

Now, if you have your domain name registered and pointing to your public IP address, and port 80 is forwarded to your server in your router, you should be able to open a web browser and type http://somedomain.com and see the website that you put in that folder on your server.

Rinse and repeat to add more domains.