

# Apache and...

**Virtual Hosts ---- aliases  
mod\_rewrite ---- htaccess**

**GHNOG  
Cape Coast, Ghana  
July 2009**

Michuki Mwangi  
Based on Materials developed by  
Hervey Allen



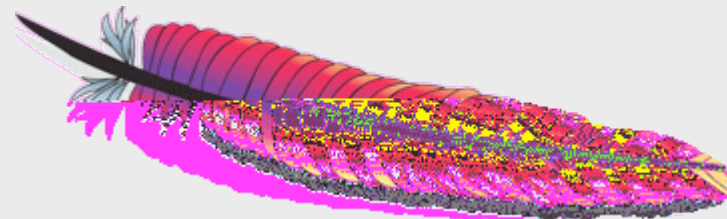
# What is Apache?

**Very good overview here:**

[http://en.wikipedia.org/wiki/Apache\\_web\\_server](http://en.wikipedia.org/wiki/Apache_web_server)

The Apache web site is an excellent source of information as well:

<http://www.apache.org/>



# Quick Facts

- Initially released in 1995
- Used on over 100 million web sites
- 46% market share. Microsoft is 29%.
- One million busiest sites, Apache 68%, Microsoft 19%
- Runs on Unix, Linux, FreeBSD, Solaris, Netware, Mac OS X, Windows, OS/2 and more.
- Licensed under the Apache License. Incompatible with GPL version 2, compatible with version 3.
- Originally designed by Robert McCool who was involved with the original web server, NCSA's HTTPd.
- Named “Apache” either because it involved many patches to the original NCSA server, or after the American Indian Apache tribe.

# What is a Virtual Host?

There are two types:

- Name-based
- IP-based

We will be configuring named-based virtual hosts.

This allows a single IP address to serve many web sites from a single server. This is possible because the web client sends the name of the site it wishes to connect to as part of its initial connection request.

# Issues

- Originally with HTTP/1.0 headers the hostname was not required to be included. Some browsers, notably Internet Explorer did not include the site name. This caused name-based hosting to fail.
- HTTP/1.1 released in 1999 requires the hostname to be part of the header. So, this is no longer an issue.
- SSL fails with name-based hosting as the hostname is not part of the initial TLS/SSL handshake – thus you cannot match the correct certificate to use for each site.

# IP-based Hosting

- This requires a separate IP address for each hostname on a web server.
- IP-based hosting works with current SSL implementations.
- IP-based hosting (can) works even if DNS has failed.
- However, requires an IP address for each site. This may not be possible and requires more effort to implement.

# Configuration Considerations: Apache

- Directory naming conventions. Decide upon one from the start:
  - /usr/local/www/share/?? (FreeBSD)
  - /var/www/share/?? (Linux)
- What to do about default actions? We'll give an example in our exercises.
- Must deal with directory permissions in more detail.

# Questions?

?

# Other Popular Apache Items

**Three include:**

- aliases
- mod\_rewrite
- htaccess

# Aliases

Allows you to specify a web directory name that maps to a separate directory *outside* the file structure of a web site.

**For example:**

Your site is

The site resides in \_\_\_\_\_, but  
you want the files in \_\_\_\_\_  
available at \_\_\_\_\_ to be

**How would you do this?**

# Aliases continued

In the file ...

But, you must set Directory permissions as well. For instance:

Remember, case counts in Apache configuration files!

# mod\_rewrite

Allows you to redirect requests from a page, or a pattern of pages to another page, or another pattern of pages.

- Extremely powerful
- Uses regular expression language
- Can save you if

In order to use `mod_rewrite` the rewrite module must be part of your Apache install (it is in FreeBSD 7.2 and Apache 2.2), and it must be loaded in the `httpd.conf` file:

# mod\_rewrite continued

Here is some sample code where `is`  
actually used (from `is`):

The end result of this is the redirect reforms the requests  
in to the form:

# htaccess

Perhaps the most common use of `mod_rewrite` is to force the use of `https` for a set of pages – such as a site login page.

**Here is an example:**

# htaccess continued

Then you must create a file “.htaccess” in the directory you wish to protect. In that file you might have something like this:

Note the file “`htpasswd`” above. This is where you store user/password information. You do this by running and using the `htpasswd` command.

# htpasswd command

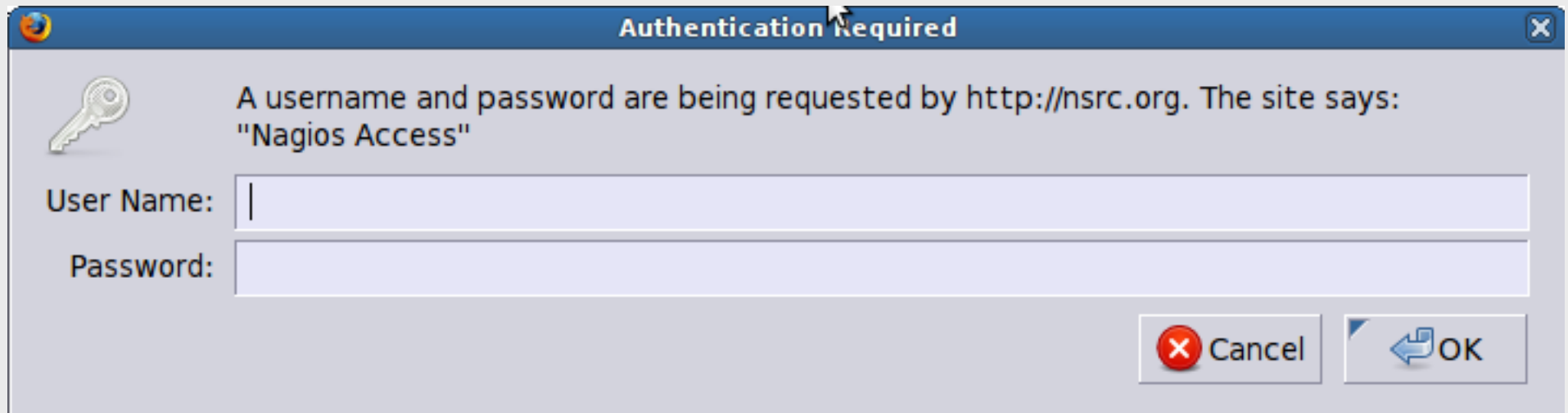
To create an initial .htpasswd file with a user and password you do:

The “-c” parameter says to create the file. Enter in the password when prompted. For the next user do:

To change a password just run the command again.

And, in the end you'll see a prompt like this...

# htaccess



## Questions?