Apache and...

Virtual Hosts ---- aliases
mod_rewrite ---- htaccess

AFNOG 11
Kigali, Rwanda
May 2010

Dorcas Muthoni

Courtesy: Hervey Allen

What is Apache?

Very good overview here:

http://en.wikipedia.org/wiki/Apache_web_server

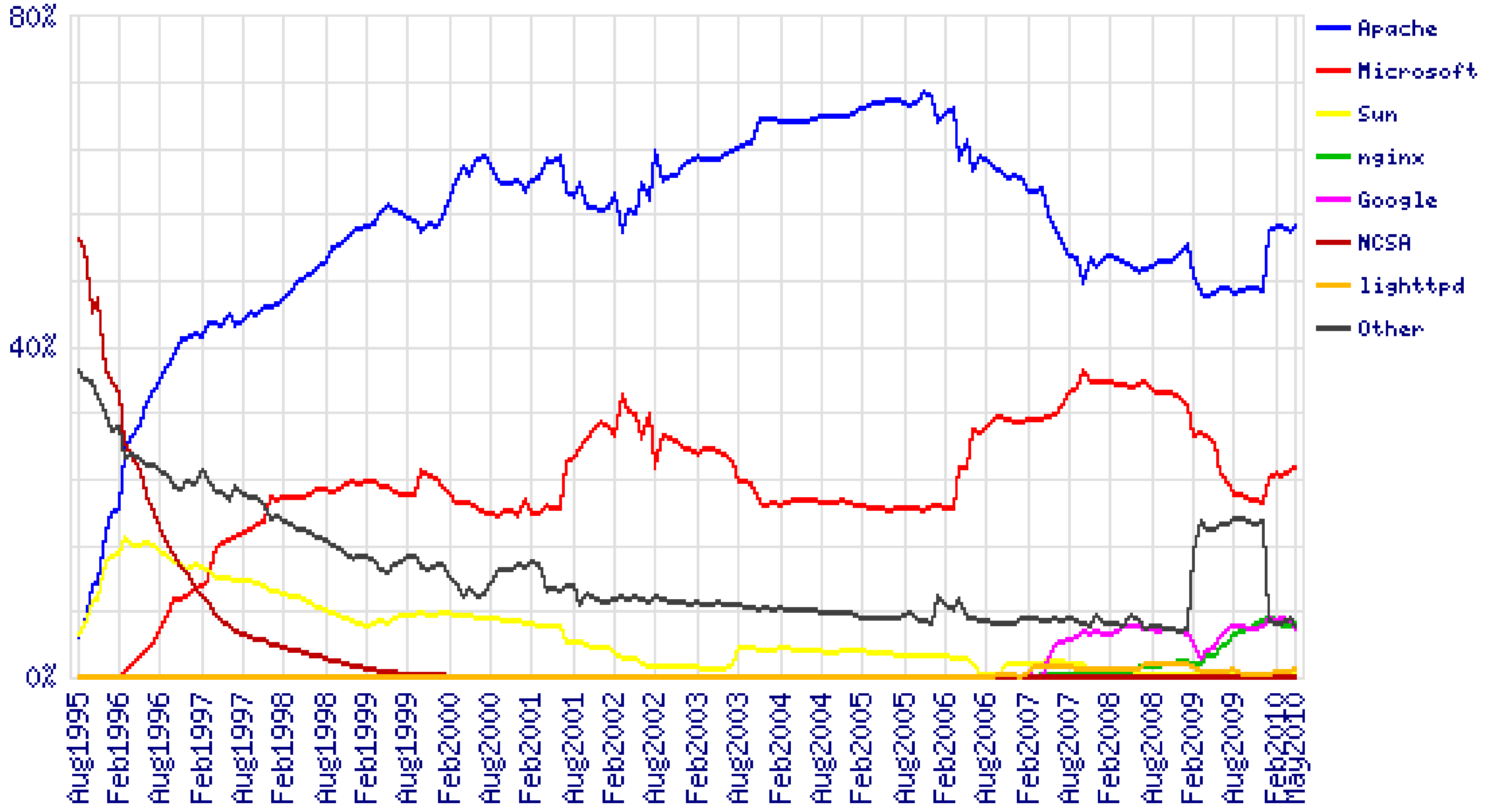
The Apache web site is an excellent source of information as well:

http://www.apache.org/

Quick Facts

- Initially released in 1995
- Used on over 100 million web sites
- 54% market share. Microsoft is 25%.
- One million *busiest sites*, Apache 66.82%, Microsoft 16.87%
- *Cross platfrom*: 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.

May 2010 Statistics



http://news.netcraft.com/archives/category/web-server-survey/

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) work 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

Primary Configuration file	**/usr/local/etc/apache22/httpd.conf**
Where your website files are stored	DocumentRoot Default is usually "/usr/local/www/apache22/data”
File that Apache will serve if a directory is requested	DirectoryIndex Default is usually **index.html** Others can be index.php or index.htm etc
Listen port	**Listen 80** You can also bind apache to a port, IP or both e.g. Listen 12.34.56.78:80
Supplemental configuration	The configuration files in the etc/apache22/extra/ directory can be included to **add extra features or to modify the default configuration** Include etc/apache22/extra/httpd-vhosts.conf

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 `http://www.example.com/`

The site resides in `/usr/local/www/share/default/`, but you want the files in `/usr/local/www/books/` to be available at `http://www.example.com/books/`

How would you do this?

Aliases continued

In the file `httpd.conf`...

```
Alias /books /usr/local/www/share/books
```

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

```
<Directory “/usr/local/www/share/books”>
      Options Indexes FollowSymLinks
      AllowOverride None
      Order allow,deny
      Allow from all
</Directory>
```

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 8.0 and Apache 2.2), and it must be loaded in the `httpd.conf` file:

```
LoadModule rewrite_module modules/mod_rewrite.so
```

mod_rewrite continued

Here is some sample code where `mod_rewrite` is actually used (from `httpd.conf`):

```
# turn on the use of the mod_rewrite module
              RewriteEngine on
# Redirect old style ISO=NN requests
               RewriteRule ^\/db\/lookup\/ISO=([A-Z])   \
               /db/lookup/redirect.php
```

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

```
   http://nsrc.org/db/lookup/country.php?ISO=eg
or
   http://nsrc.org/db/lookup/provider.php?           \
id=89733450039&fromISO=eg
```

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:

```
# Turn on the rewrite engine.
       # If we are not using port 443 (ssl) AND
       # We are trying to access something under the /trac directory AND
       # We are NOT trying to open the initial index.php file (to avoid
       # infinite redirects), THEN keep the URI and force the user to use
       # SSL. Too many passords and sensitve info are thrown around on
       # the trac project pages.
       RewriteEngine on
       RewriteCond %{SERVER_PORT} !443
       RewriteCond %{REQUEST_URI} ^/trac
       RewriteCond %{REQUEST_URI} !^/trac/index.php
       RewriteRule ^(.*)$ https://ws.edu.isoc.org$1 [R=301]
```

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:

```
AuthName "AfNOG 2010 SAE, Trac Access"
AuthType Basic
AuthUserFile /var/www/html/trac/afnog10/.htpasswd
require user afnog
```

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:

```
# htpasswd -c .htpasswd username
```

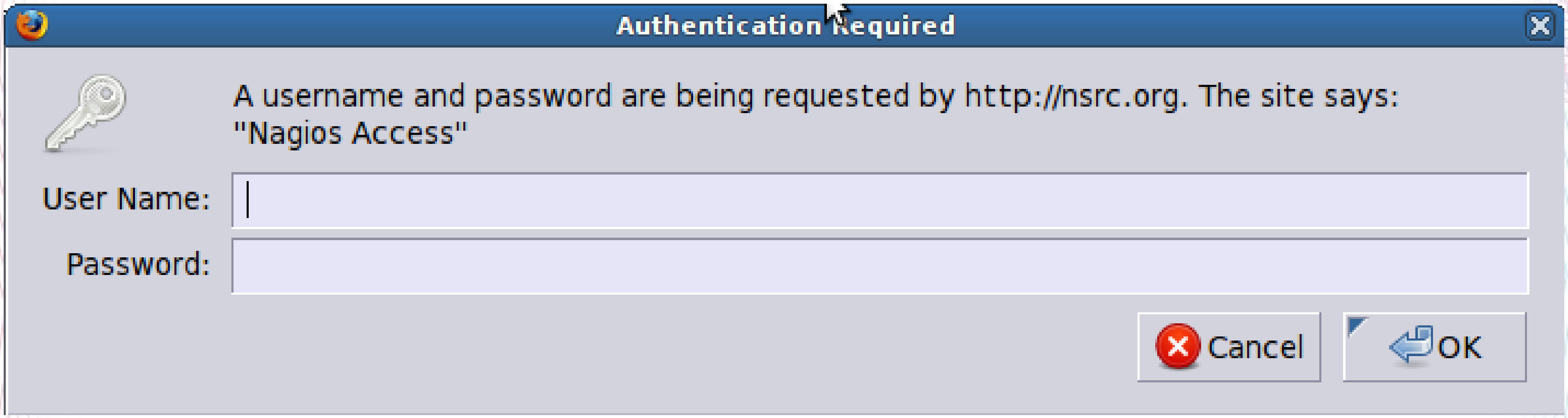
The “-c” parameter says to create the file. Enter in the password when prompted. For the next user do:

```
# htpasswd .htpasswd username
```

To change a password just run the command again.

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

htaccess


Authentication Required

A username and password are being requested by http://nsrc.org. The site says: "Nagios Access"

User Name:

Password:

Questions?