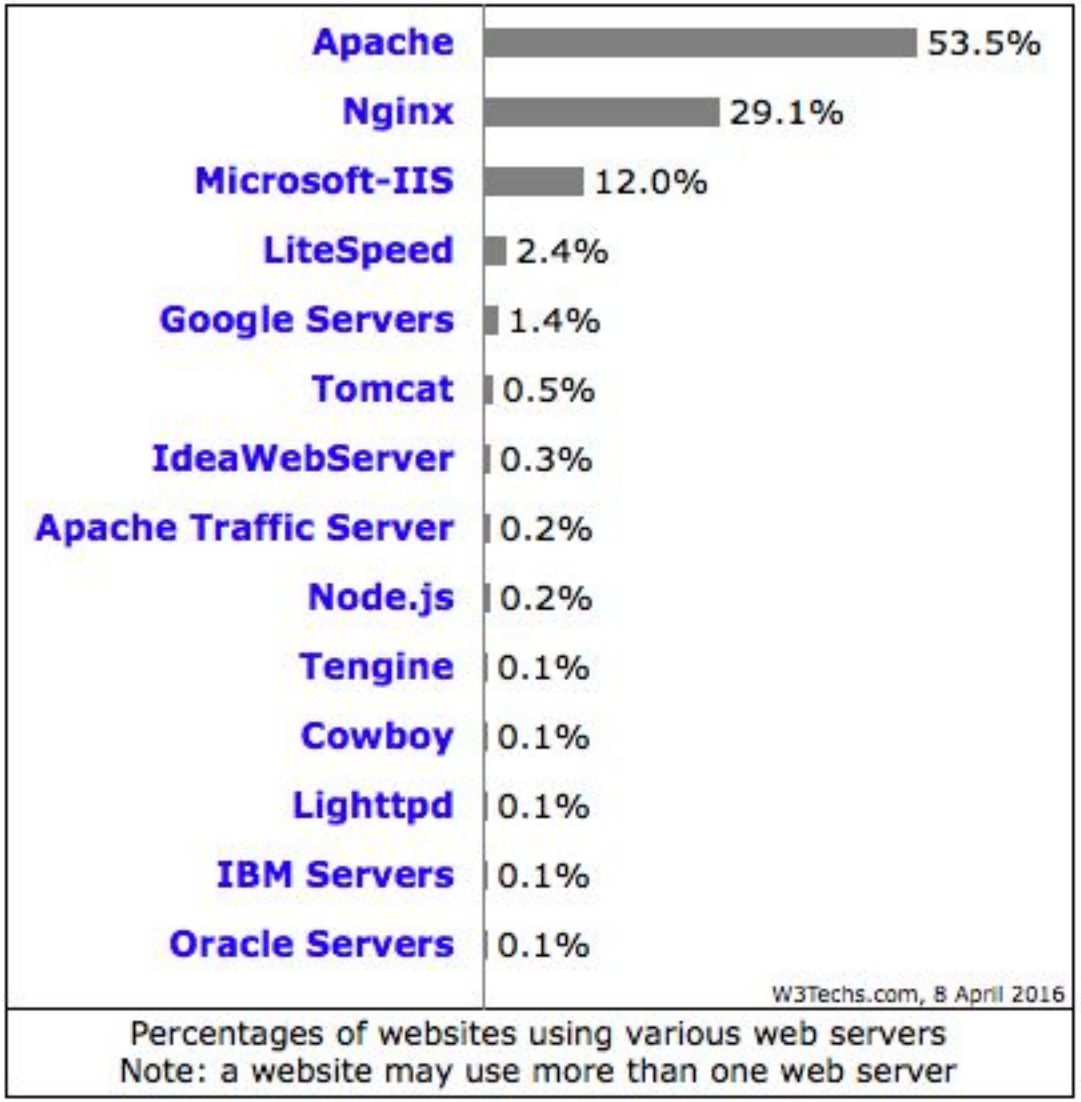
Apache Web Server

About Apache

- Apache http server project
- http://httpd.apache.org
- Apache foundation started to support the web server project, but now extends to a multitude of other projects.

Stats of Web Server types



Apache Installation on Ubuntu

- Can be installed from apt-get
- Or from source if one requires a more recent version

File System Layout

- config files are in

`/etc/apache2/,/etc/httpd/conf`

`/usr/local/etc/apache22/`

- files the webserver will serve are in

`/usr/local/www/apache22/data/`

- Startup script is

/etc/init.d/apache2

`/usr/local/etc/rc.d/apache22`

- Run

```
/usr/local/etc/rc.d/apache22 start
/etc/init.d/apache2
```

- Restart

```
$ apachectl restart
```

Apache Files

Configuration file: /etc/httpd/conf/httpd.conf

Log files: /var/log/httpd/access_log and /var/log/httpd/error_log

Modules /etc/httpd/modules

Default Document Root /var/www/html

Default CGI Root /var/www/cgi-bin

Status codes

- The status codes are all three-digit numbers that are grouped by the first digit into 5 groups.
- The reason phrases given with the status codes below are just suggestions. Server can return any reason phrase they
- 1xx: Informational
- 2xx: Successful200 OKMeans that the server did whatever the client wanted it to, and all is well.
- 3xx: RedirectionMeans that the resource is somewhere else and that the client should try again at a new address.301 Moved permanentlyThe resource the client requested is somewhere else, and the client should go there to get it. Any links or other references to this resource should be updated.

Status codes

- 4xx: Client errorMeans that the client screwed up somehow, usually by asking for something it should not have asked for.
- 404: Not foundSeen this one before? :) It means that the server has not heard of the resource and has no further clues as to what the client should do about it. In other words: dead link
- 5xx: Server errorThis means that the server screwed up or that it couldn't do as the client requested.
- 500: Internal server errorSomething went wrong inside the server.

Enable log

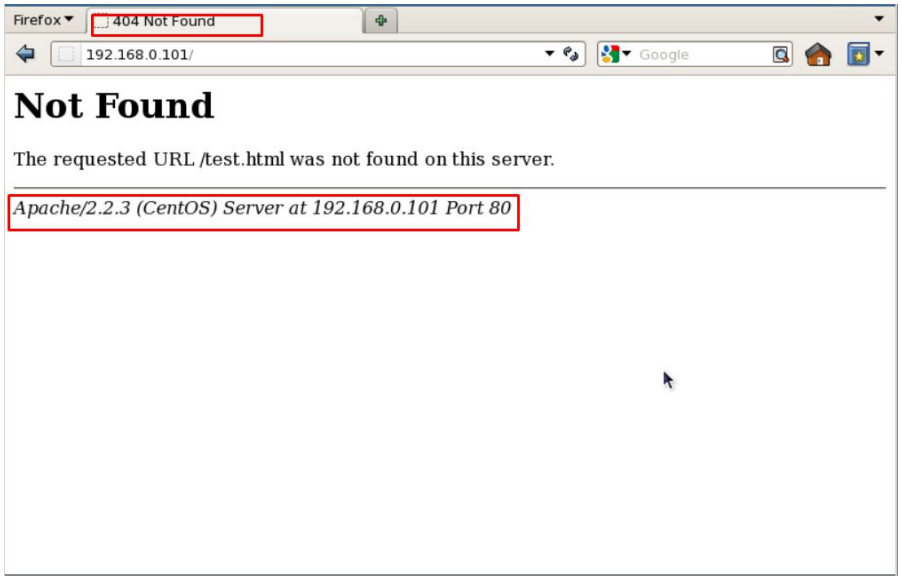
- Enable Apache Logging
- Apache allows you to logging independently of your OS logging. It is wise to enable Apache logging, because it provides more information, such as the commands entered by users that have interacted with your Web server.
- To do so you need to include the mod_log_config module. There are three main logging-related directives available with Apache.
- TransferLog: Creating a log file.
- LogFormat : Specifying a custom format.
- CustomLog : Creating and formatting a log file.
- You can also use them for a particular website it you are doing Virtual hosting and for that you need to specify it in the virtual host section. For example, here is the my website virtual host configuration with logging enabled.

Enable log

- <VirtualHost *:80>
- DocumentRoot /var/www/html/example.com/
- ServerName www.example.com
- DirectoryIndex index.htm index.html index.php
- ServerAlias example.com
- ErrorDocument 404 /story.php
- ErrorLog **/var/log/httpd/example.com_error_log**
- CustomLog **/var/log/httpd/example.com_access_log combined**
- </VirtualHost>

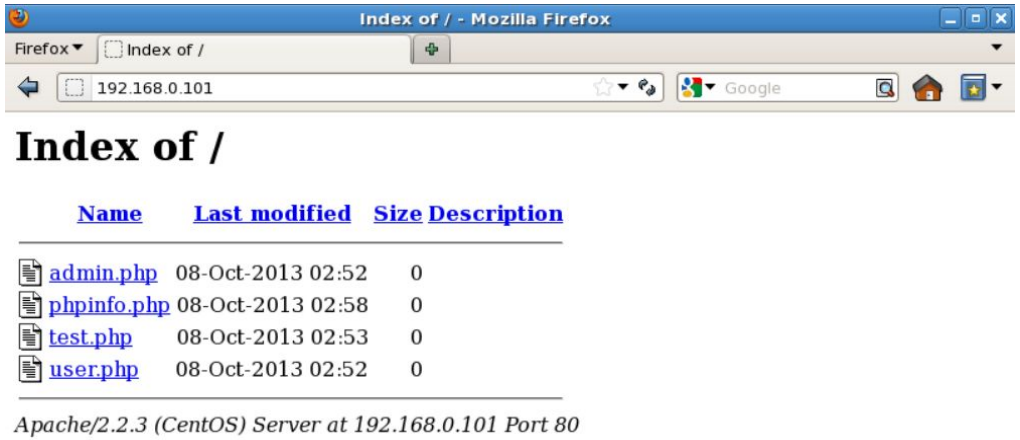
Hardening apache

Hide Apache Version and OS Identity from Errors



vim /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora)
vim /etc/apache/apache2.conf (Debian/Ubuntu)
ServerSignature Off
ServerTokens Prod

Disable Directory Listing



<Directory /var/www/html>
Options –Indexes
</Directory>

Use mod_security and mod_evasive Modules to Secure Apache

- Mod_security
- Where mod_security works as a firewall for our web applications and allows us to monitor traffic on a real time basis. It also helps us to protect our websites or web server from brute force attacks. You can simply install mod_security on your server with the help of your default package installers.
- Install mod_security on Ubuntu/Debian
- $ sudo apt-get install libapache2-modsecurity
- $ sudo a2enmod mod-security
- $ sudo /etc/init.d/apache2 force-reload

Secure Apache..

- Mod_evasive
- mod_evasive works very efficiently, it takes one request to process and processes it very well. It prevents DDOS attacks from doing as much damage. This feature of mod_evasive enables it to handle the HTTP brute force and Dos or DDos attack. This module detects attacks with three methods.
- If so many requests come to a same page in a few times per second.
- If any child process trying to make more than 50 concurrent requests.
- If any IP still trying to make new requests when its temporarily blacklisted.
- mod_evasive can be installed directly from the source. Here, we have an Installation and setup guide of these modules which will help you to set up these Apache modules in your Linux box.

Apache SSL

- Secure Socket Layer (SSL) port is 443
- SSL is important to protect communication between browser and web-server
- Requires the creation of SSL certificates and Certificate Signing Requests (CSR)
- For integrity SSL certificates are signed by a Certificate Authority's (CA) such as Verisign
- Self signed Certificates will also work but your browser will not trust it and will give a warning to users (which most don't read)
- *Refer to the Creating SSL Certificate Exercise Section*

How SSL Works

- Each SSL certificate has a Public and Private key
- The Public Key is used to encrypt the information
- The Public Key is accessible to everyone
- The private Key is used to decipher the information
- The private should be not be disclosed

Role of Certificate Authority

- There are a number of CA that certify certificates
- Most browsers have pre-included public Keys from the CA's
- A CA certified certificate will have validation information signed by the CA's private key
- The browser will decrypt the validation information using the public key and verify that the certificate is certified by the CA
- If this fails a warning is given

Virtual Hosting

- Apache Provides multiple options of virtual hosting and scales
  - Name Based virtual hosts
  - IP Based Virtual Hosts
  - Aliases
- Its recommended to use an IP address over hostnames in virtual hosting configuration

Virtual Hosting

```
NameVirtualHost    *:80

<VirtualHost       *:80>
    ServerName server-name
    DocumentRoot path-to-virtual-document-root
    </VirtualHost>

<VirtualHost    *:80>
    ServerName server-name
    DocumentRoot path-to-virtual-document-root
    </VirtualHost>
```

Installing PHP & Mysql

- PhP and Mysql implementations have increased driven mainly by development requests
- LAMP and WAMP are the most common implementations
- FreeBSD = “FAMP” ? http://geekybits.blogspot.com/2007/09/creating-famp-server.html
- Installation via ports and relatively straight forward
- *See PHP & Mysql installation exercise section*

Apache and IPv6

- Apache supports IPv4 and IPv6 by default
- Set the listen option to port 80 will listen for both IPv4 and IPv6
- listen option with IPv4 and IPv6 specific addresses will invoke different sockets for each protocol

  Listen 196.200.219.xx:80

  Listen [2001:4348:0:219:196.200.219:xx]:80

Start Apache!

- `/etc/init.d/apache2 start`
- Check that you can access `http://localhost` in your browser
- Check that you can access `https://localhost` in your browser, and that you get a certificate warning
- Click on the padlock icon in your browser and check that the certificate details are correct
- Profit!

Apache implementations

- Apache is widely used to serve many content applications
- Webmail, Blogs, Wiki’s, CMS etc

Start Exercises