Apache Web Server

bdNOG 6 – Bogra, Bangladesh

Outline

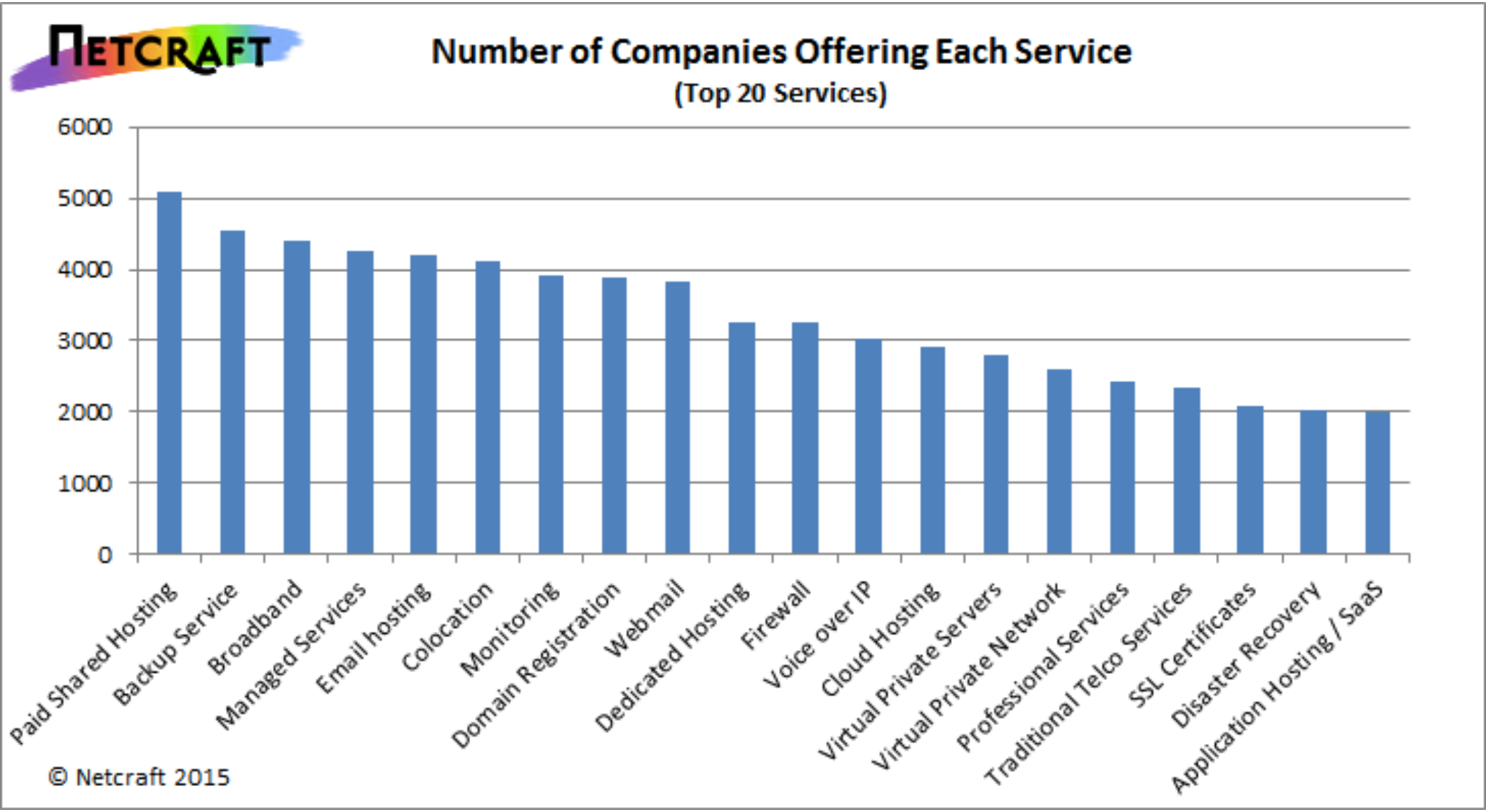
- Introduction to Apache httpd web server
- Basic Compilation, Installation and Configuration
- Apache File system
- Apache Logging & Status
- Security & Performance Features
- Virtual Hosting
- Apache Applications

About Apache

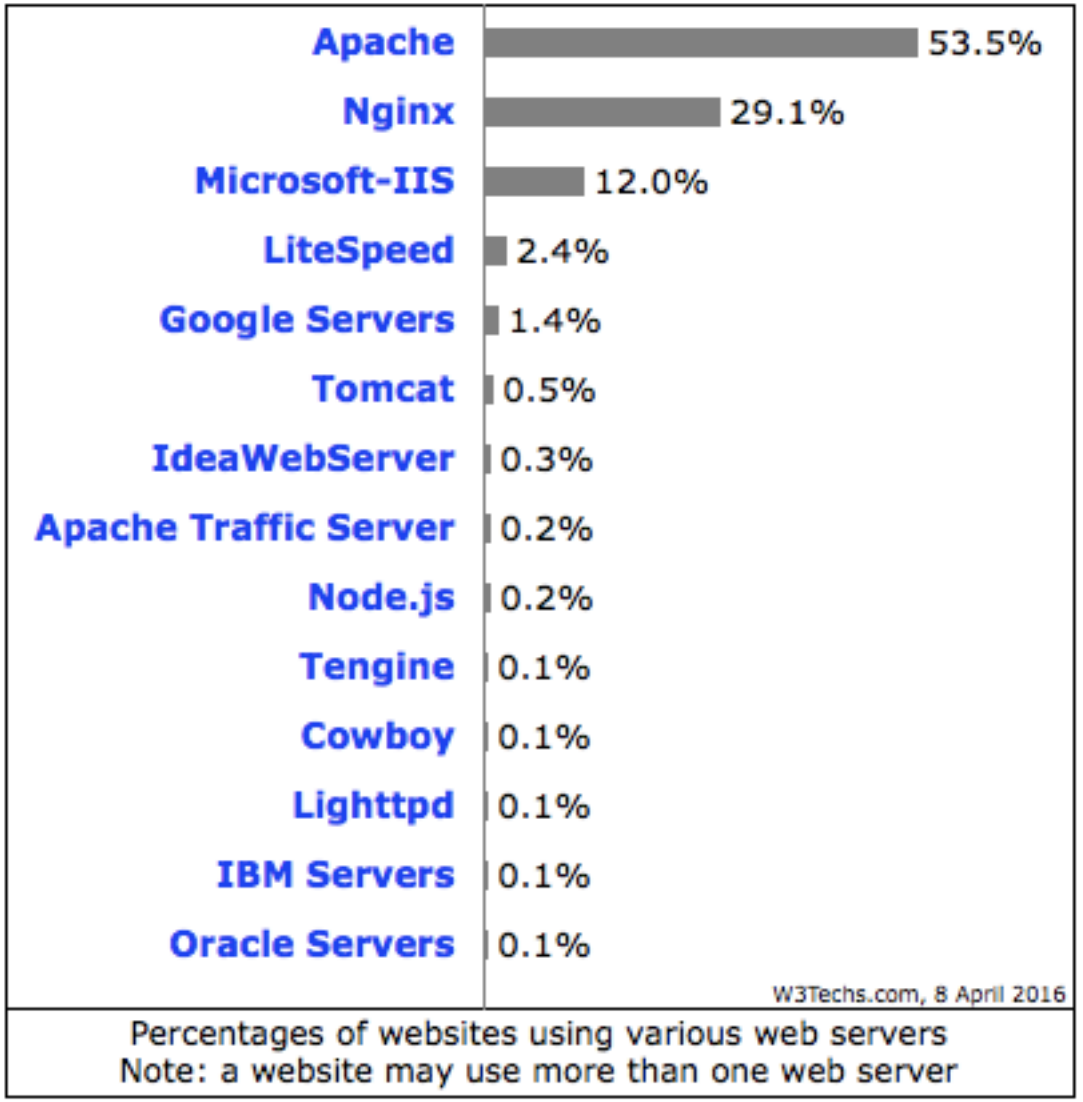
- A PAtCHy server: developed by the Apache group formed 2/95 around by a number of people who provided patch files for NCSA httpd 1.3 by Rob McCool.
- Apache HTTP server project http://httpd.apache.org
- History-http://httpd.apache.org/ABOUT_APACHE.html
- Apache foundation started to support the web server project, but now extends to a multitude of other projects
- First official public release (0.6.2) in April 1995
- Added adaptive pre-fork child processes (very important!).
- Modular structure and API for extensibility (Bob Thau)
- Port to multiple platforms.
- Apache 1.0 was released on 12/1/95.
  Pass NCSA httpd to be #1 server in Internet.

- Reference:
  - http://httpd.apache.org/docs/current/

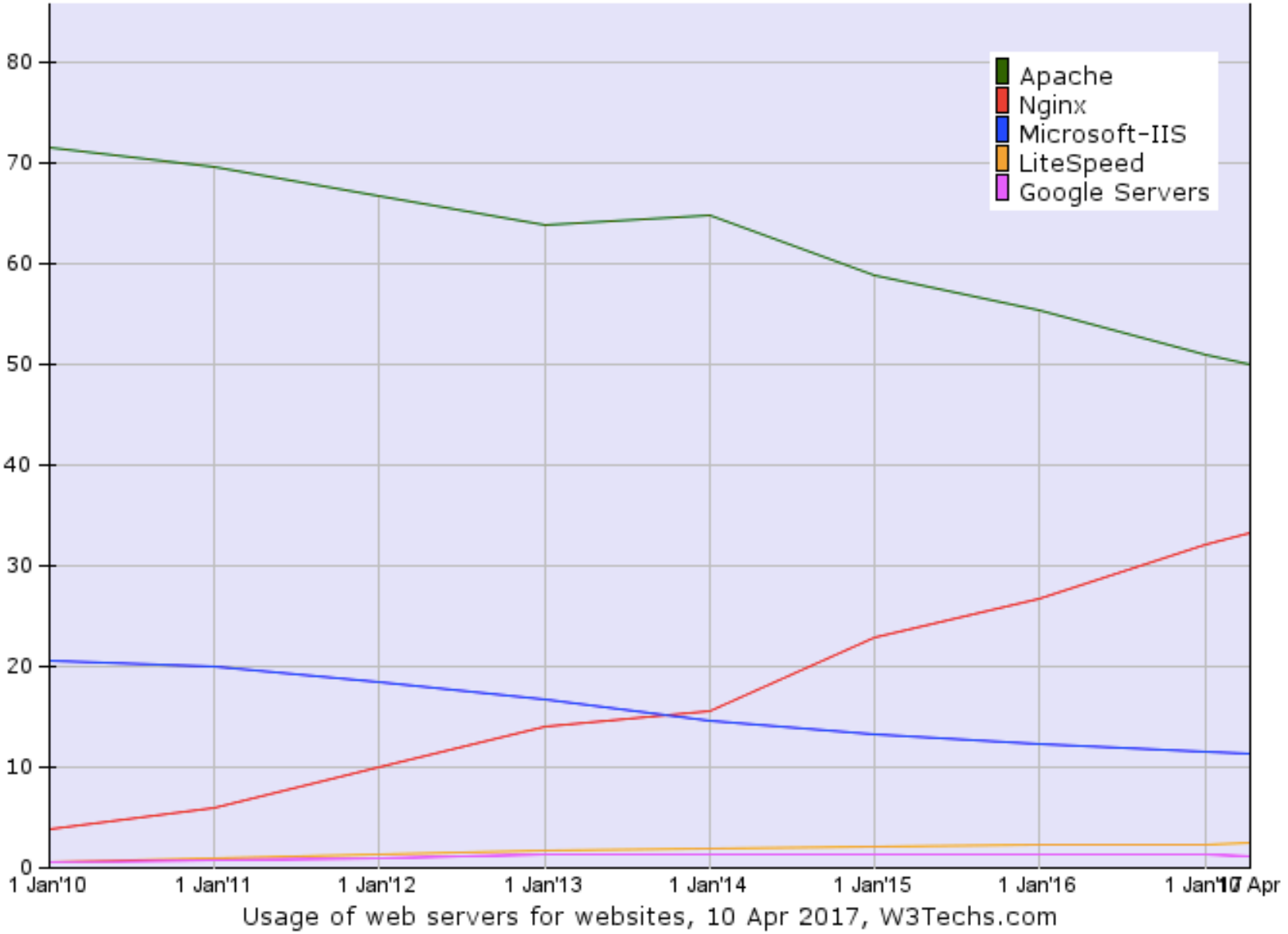
Taxonomy of Internet Services



Stats of Web Server types



Web Server Installation Statistics



- See survey statistics in → W3Techs.com &

Apache Installation

- The current stable release is Apache 2.4.25
- Can be installed via package manager
- Or custom installation from source if one requires a more recent version
- In our training machines we will install Apache 2.4.7
  - Linux Package Manager:
    - Ubuntu/Debian : apt-get install apache2
    - CentOS/Redhat/Fedora : yum install httpd
  - For win32 version, you can download from any of mirror servers. Win32 Binary including OpenSSL 0.x.x (MSI Installer). http://httpd.apache.org/download.cgi

Apache Installation : Custom

- Download httpd-2.4.x.tar.bz2 from http://httpd.apache.org/download.cgi or closer mirror sites
- $tar xjf httpd-2.4.x.tar.bz2
- $ cd httpd-2.4..x
- $ ./configure --prefix=PREFIX
- $ Make
- $ sudo make install
- $ sudo PREFIX/bin/apache2ctl start

- Here PREFIX is the prefix of the directory containing the distribution, typically it is /usr/local/apache.
- Since as a normal user, we do not have permission to install there, you need to have sudo privileges for your user.
- For configuring the apache with specific features, we can specify the corresponding features as option to the configure command. You can find the list of features by "./configure –help"

File System Layout (via Package Manager)

- config files are in

`/etc/apache2/ (Ubuntu/Debian)`
`/etc/httpd/conf (CentOS/Redhat/Fedora)`

- files the webserver will serve are in

`/var/www/html/`

- Startup script is

`/etc/init.d/apache2(Ubuntu/Debian)`

- Run

```
$ sudo /etc/init.d/apache2 start
$ sudo service apache2 start
$ sudo systemctl start apache2
```

- Restart

```
$ sudo /etc/init.d/apache2 restart
$ sudo service apache2 restart
$ sudo systemctl restart apache2
```

File System Layout (via Custom Installation)

- config files are in

`/usr/local/etc/apache2/`

- files the webserver will serve are in

`/usr/local/www/apache2/data/`

- Startup script is
  `/usr/local/etc/rc.d/apache2`
- Run
  `/usr/local/etc/rc.d/apache2 start`

- Restart
  `$ /usr/local/etc/rc.d/apache2 restart`

Apache Files (Ubuntu/Debian)

Configuration file: `/etc/apache2`

```
apache2.conf    conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars       mods-available ports.conf    sites-enabled
```

Log files: `/var/log/apache2/access_log`

`/var/log/apache2/error_log`

Modules `/etc/apache2/mods-available/`

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: Successful
  - 200 OK - means that the server did whatever the client wanted it to, and all is well.
- 3xx: Redirection Means that the resource is somewhere else and that the client should try again at a new address.
  - 301 Moved permanently - The 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 error - means that the client screwed up somehow, usually by asking for something it should not have asked for.
  - 404: Not found Seen 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 error - means that the server screwed up or that it couldn't do as the client requested.
  - 500: Internal server error - Something went wrong inside the server.

Appache 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 if 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>

  ServerName example.com

  ServerAlias www.example.com

  ServerAdmin webmaster@localhost

  DirectoryIndex index.htm index.html index.php

  DocumentRoot /var/www/html/example.com

  <Directory "/var/www/html/example.com/">

  Options FollowSymLinks

  AllowOverride All

  Allow from all

  </Directory>

ErrorDocument 404 /story.php

  ErrorLog **/var/log/httpd/example.com_error_log**

  CustomLog **/var/log/httpd/example.com_access_log combined**

- </VirtualHost>

Apache Performance Tuning

```
<IfModule mpm_prefork_module>
        StartServers            2
        MinSpareServers         5
        MaxSpareServers         10
        ServerLimit             256
        MaxClients              600
        MaxRequestWorkers       600
        MaxRequestsPerChild     1000
</IfModule>
```

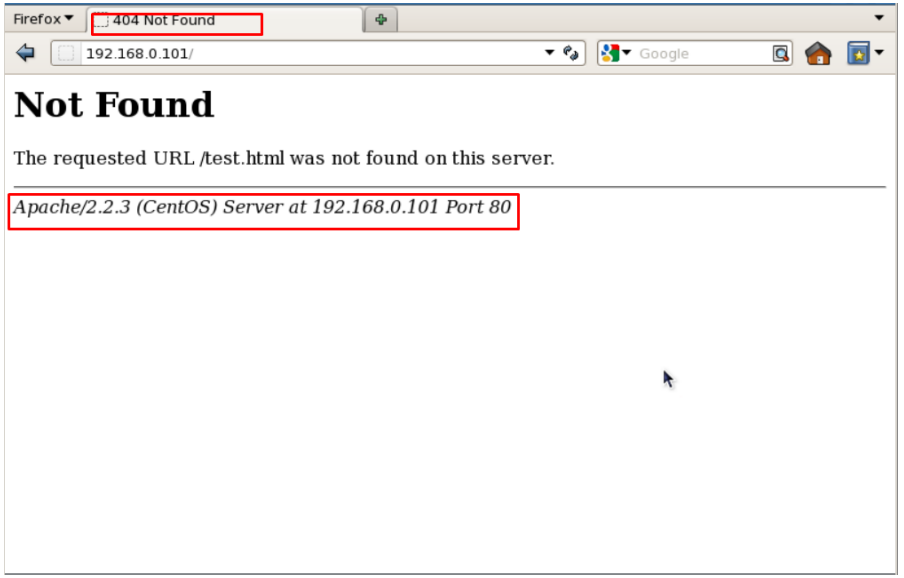
Keep Alive

Apache Performance Tuning

- Keep Alive directives
- Apache Runtime loaded modules
- Application/scripts Runtime loaded modules
- Memory mapping

Hardening apache

Hide Apache Version and OS Identity from Errors

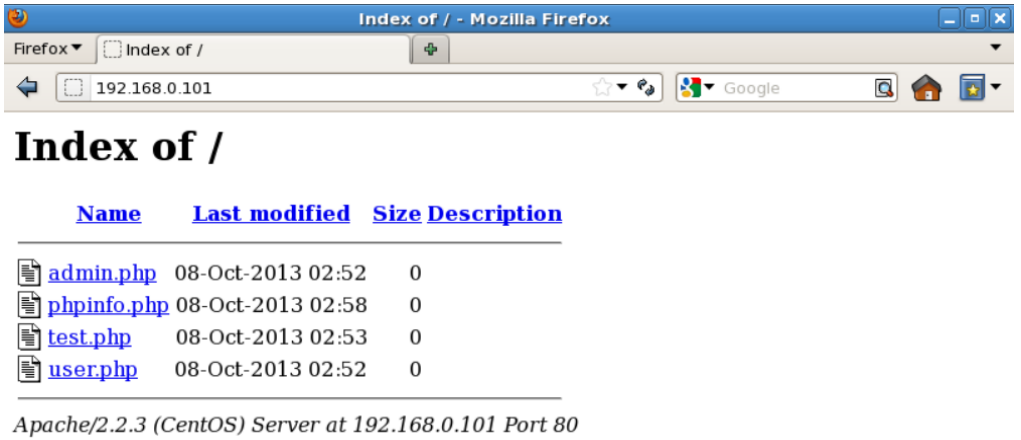


$ sudo vim /etc/httpd/conf/httpd.conf (RHEL/CentOS/Fedora)
$ sudo vim /etc/apache/mods_available/security (Debian/Ubuntu)

```
ServerSignature Off
ServerTokens Prod
TraceEnable Off
```

$ sudo service apache2 restart

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 service apache2 force-reload
```

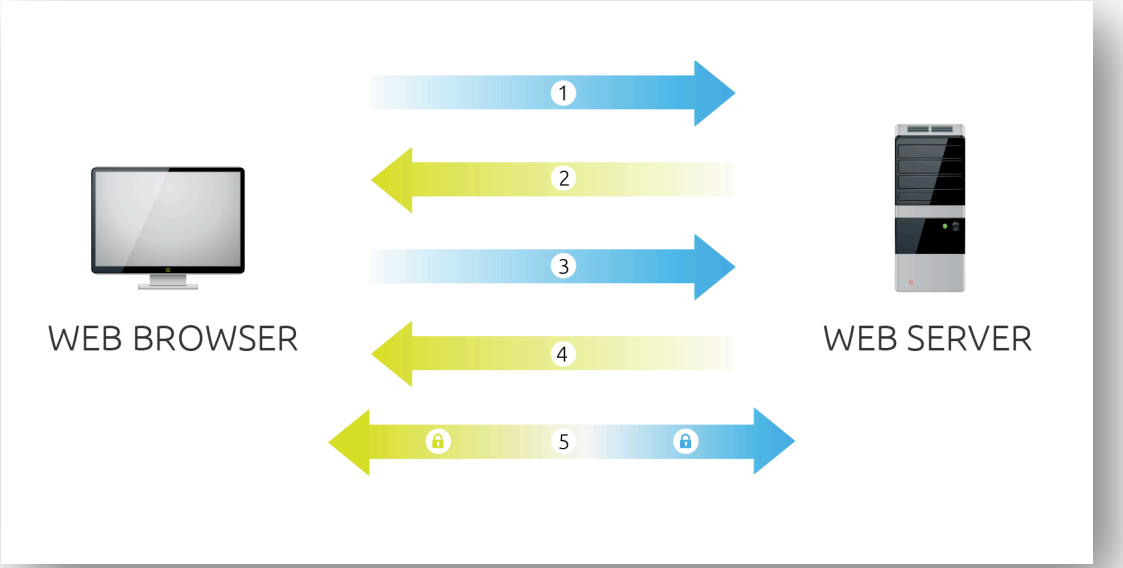
Secure Apache..

- 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 NetSol, Symantec, Comodo, etc.
- 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



1. **Browser** connects to a web server (website) secured with SSL (https). Browser requests that the server identify itself.
2. **Server** sends a copy of its SSL Certificate, including the server's public key.
3. **Browser** checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server's public key.
4. **Server** decrypts the symmetric session key using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session.
5. **Server** and **Browser** now encrypt all transmitted data with the session key.

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>
```

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

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 is relatively straight forward
- *See PHP & Mysql installation exercise section*

Apache implementations

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

Start Exercises