

COMP09024 Unix System Administration

Lecture 10: Web Services

Duncan Thomson/Hector Marco

UWS

Trimester 1 2020/21

Outline

10.1 Web and HTTP

- Web Protocols
- HTTP
- Web Applications

10.2 Apache

- Configuration Commands
- Configuration Files
- Configuration Directives

10.3 Common Configurations

- Virtual Hosting

- Access Control
- Securing HTTP with TLS/SSL

10.4 Scripting

- PHP
- Java Servlets and JSP

10.5 Databases

- MySQL
- PostgreSQL
- SQLite
- Database APIs

10.1 Web and HTTP

What is the Web?

- A multimedia network application providing access to hypertext pages (pages linked via selectable links)
- Client/server application, based around HTTP (Hypertext Transfer Protocol)
- Pages are defined in Hypertext Markup Language (HTML):
 - HTML pages can be static files sent directly
 - HTML pages can also be dynamically generated by programs (e.g PHP).
 - Multimedia information can be included
- Viewed using a web browser (eg Firefox, Chromium. . .)
- Content provided using a web server (eg Apache, NginX,. . .)

Overview of Web Protocols/Standards

- Hypertext Transfer Protocol (HTTP) is the main application-layer protocol by which clients request information from (and send information to) servers
- Hypertext Markup Language (HTML) is the language in which pages are written
 - May include style information with Cascading Style Sheets (CSS)
 - May include active scripts in JavaScript/DOM
- Multipurpose Internet Mail Extensions (MIME) provides a means by which content type may be negotiated (eg text/html, image/jpeg)
- RFC-2822 defines header format

HTTP

- HTTP runs over TCP (port 80)
- Client sends an HTTP request to server:
 - First line: (Method (GET/POST/...), URI, HTTP version)
 - Headers (RFC 2822 format): General, Request, Entity
 - Possibly an entity (data object — eg form data)
- Server replies by sending an HTTP response to client:
 - First line: (HTTP version, Code, Response)
 - Headers: General, Response, Entity
 - Usually an entity (data object — eg web page or image)

Web Applications and Architecture

- Typical web application is composed of three parts:
 - Web browser
 - Web server with scripting language or framework (eg PHP, JSP, Rails. . .)
 - Storage server (usually a relational database)
- Browser/web server communication using HTTP and HTML/CSS/Javascript
- Web server / database typically communicate using a database API and Structured Query Language (SQL) — a database query language
- The two server components may be distributed over two (or more) servers

10.2 Apache

The Apache Software

- Most widely used web server software for over a decade
- Originally developed for Unix/Linux, but now Windows too
- Modular server: core functionality with many modules containing additional functionality, for example:
 - Server-side scripting environments
 - Authentication mechanisms
 - TLS/SSL support
 - HTTP proxying
- Currently version 2.4 (Debian package: `apache2` contains 2.2)
- Text-based configuration files

Configuration Commands

- `apache2ctl` allows direct control of Apache, including some useful functionality:
 - `apache2ctl configtest` tests configuration files
 - `apache2ctl graceful` reloads configuration without dropping connections
- Site management (Debian)
 - Links in `sites-enabled/` link to files in `sites-available/`
 - `a2ensite` and `a2dissite` allows these links to be automatically created and deleted
- Module management (Debian)
 - Links in `mods-enabled/` link to files in `mods-available/`
 - `a2enmod` and `a2dismod` allows these links to be automatically created and deleted

Apache Configuration Files in Debian

- All configuration files are in `/etc/apache2/`
- `apache2.conf` contains main configuration files
- `httpd.conf` for backwards compatibility
- `ports.conf` specifies IP/TCP addresses and ports
- `envvars` specifies some variables used in other files
- `magic` is used for MIME content negotiation
- `sites-enabled/` holds links to files in `sites-available/`
- `mods-enabled/` holds links to files in `mods-available/`
- `conf.d/` holds config files for packages

Apache Configuration Directives

- `Include` allows inclusion of other config files
- `ServerName` specifies DNS name of server
- `ServerRoot` specifies root directory of installation
- `User` and `Group` specify UID & GID to run as
- `DocumentRoot` specifies root directory of web pages
- `Listen` lists IP address and TCP ports
- `ErrorLog` gives name of error log file
- `Alias` maps from a URL to a directory/file
- *Sectional directives*, such as `VirtualHost`, `Location`, `Directory`, `Method`, `IfModule` allow groups of directives only applicable in certain situations

10.3 Common Configurations

Virtual Hosting

- Allows multiple web sites to be served from the one server
- Can be done using multiple IP addresses or TCP port numbers, but usually done using multiple DNS addresses (all pointing to the one server)
- The configuration required is (for two virtual hosts):

```
<VirtualHost *>
    ServerName www.example.org
    DocumentRoot /home/bob/ExampleOrg
</VirtualHost>
<VirtualHost *>
    ServerName www.example.com
    DocumentRoot /home/bob/ExampleCom
</VirtualHost>
```

Access Control by Location

- Access control allows limiting access to certain content by the source IP address of the request
- May use sectional directive to specify which requests are limited
- Then `Order`, `Permit` and `Deny` directives are used to control a three pass access control check

```
<Location /intranet>  
    Order Allow,Deny          # deny non/both matches  
    Allow from 172.16.0.0/16  # allow from internal  
    Deny from 172.16.240.0/20 # but not visitor WiFi  
</Location>
```

Access Control by User

- Access may also be controlled by users providing authentication credentials
- A wide range of modules are available for authentication (against databases, LDAP, and many more)
- Simplest is using a password file
- `htpasswd` is used to manage password file (which should not be web-accessible)

```
<Directory /var/www/area51>  
    AuthType Basic  
    AuthName "Restricted Area"          # shown to user  
    AuthUserFile /home/alice/passwords51  
    Require valid-user  
</Directory>
```


Securing HTTP: TLS/SSL

- Standard HTTP (carried in IP & TCP) is plaintext
- To prevent eavesdropping of HTTP traffic (eg for online payments) it should be encrypted using Secure Sockets Layer / Transport Layer Security (SSL/TLS)
- This has several steps:
 - 1 Generate a certificate signed by a CA (who may charge for the service)
 - 2 Enable Apache's SSL or TLS module (`mod_ssl` with OpenSSL or `mod_gnutls` with GnuTLS)
 - 3 Configure Apache to use SSL/TLS as appropriate
- Here our example uses SSL

Setting up SSL

- Create certificate for DNS name in `/etc/apache2/ssl/`
 - `make-ssl-cert`
`/usr/share/ssl-cert/ssleay.cnf key.pem`
 - Type correct full DNS name in response to prompt
 - Ideally have this signed by a recognised CA
- Enable the SSL module
 - `a2enmod ssl`
 - This will load SSL module and listen on port 443

```
<VirtualHost www.example.org:443>      # 443=HTTPS
DocumentRoot /home/alice/onlineorders
SSLEngine on
SSLCertificateFile ssl/key.pem
</VirtualHost>
```

10.4 Scripting

Scripting

- Many languages can be used on client and/or server side
- Client side (eg JavaScript/ DOM, AJAX) runs in browser
- Examples of server side scripting systems include:
 - Common Gateway Interface (CGI) — flexible but slow
 - PHP Hypertext Preprocessor (PHP)
 - Java Servlets / Java Server Pages (JSP)
 - Perl (`mod_perl`)
 - Python (`mod_python`)
- Also possible to use a *web application framework* — extensions to a language providing (eg) database access, templating, and session management; examples include Ruby on Rails and Zend
- Or install a full Content Management System (CMS), eg Drupal, Joomla, Typo3, SharePoint, Moodle...

PHP

- A web application language for server-side scripts
- Supported by Apache as a module, additional functionality (eg database access) in PHP modules
- .php files are sent to the PHP module for processing
- These are typically HTML with embedded PHP code

```
<html><head><title>PHP Example</title></head><body>
<h1>PHP Example</h1>
<!-- this is HTML, now start PHP code -->
<?php
    $date=date("Y-m-d (l)"); // this is PHP
    echo("<p>Hello, today is $date.</p>");
?>
</body></html>
```

Java Servlets and JSP

- Java Servlets are server-side programs written in Java
- Java Servlets also support Java Server Pages (JSP):
 - Support embedding of Java code in HTML pages (like PHP)
 - Also provide ability of writing code for custom tags
- Tomcat is a well-known Java Servlet engine, which can be:
 - Run as a standalone webserver
 - Run as a Java Servlet engine alongside a 'standard' webserver such as Apache or NGinX with a 'connector' allowing passing of requests to the Java/JSP engine

10.5 Databases

Databases

- Generally web applications require data to be stored
- Most common data storage is using a relational database management system (RDBMS), for example:
 - MySQL or PostgreSQL — open-source databases
 - Oracle, SYBASE, DB2 — commercial databases
- Web applications access the data using:
 - A library providing an API for connecting to DB
 - Structured Query Language (SQL) queries sent over this connection
- In some cases data can be stored in other ways, known as the NoSQL approach — eg MongoDB, CouchDB, Berkeley DB, etc

MySQL

- Now owned by Oracle, though developed since 1995
- GPL licence (some features released under proprietary licence)
- Some purely GPL forks exist (eg MariaDB)
- Widely used for web applications, mainly owing to speed
- Debian package is `mysql-server` (and `mysql-client`)
- Configuration file in `/etc/mysql/my.conf`:
 - Like Windows `.ini` file, sectioned with eg `[mysqld]`, `[client]`
 - Options then specified with `option = value`
- Data stored by default under `/var/lib/mysql/`

Administering MySQL

- Admin commands include:
 - `mysqladmin` for user/database admin
 - `mysqlaccess` for access control
- ...but administration mostly done via `mysql` CLI:
 - `mysql -u user -p` allows login — the `root` user (separate from the Unix `root` user) is the DB admin
 - `GRANT ALL PRIVILEGES ON *.* TO 'username'@'hostname' IDENTIFIED BY 'password'` to add a user
 - `CREATE DATABASE dbname` to add a database

PostgreSQL

- Developed since 1985 (previously as Postgres), BSD-like licence
- An object-relational database server — relational, with some object-oriented features (eg new types with inheritance, operator overloading, etc)
- Configuration (in Debian) under (eg)
`/etc/postgresql/9.1/main/`
 - `postgresql.conf` holds server configuration
 - `pg_hba.conf` holds access control settings
- Data stored by default under (eg)
`/var/lib/postgresql/9.1/main/`

Administering PostgreSQL

- `createuser` used to create PostgreSQL users:
 - `postgres` is default DBA user
 - Unix users automatically mapped to PostgreSQL users
 - `-r` or `-R` to allow/disallow user creation
 - `-d` or `-D` to allow/disallow database creation
 - Additional users may have encrypted passwords (`-E -P`)
- `createdb` used to create databases:
 - `-O` specifies owner
- `dropuser` and `dropdb` remove users / databases
- `pg_ctl` can stop/start PostgreSQL server(s)
- `psql` provides SQL command-line access to a database
- `pg_dump` dumps SQL commands to recreate database

SQLite: The world's most widely deployed database

- A file-based SQL-accessible database — not a client/server system
- Part of software distributed in many embedded devices, including mobile phones:
 - ‘the world’s most widely deployed database’
- `sqlite` provides command-line access to DB stored in a file
- File permissions determine who can read/change data

Database APIs

- For software to access data in databases, a library is required to match the programming language to the database
- Many languages have a standardised database application programming interface (API):
 - Java: Java Database Connector (JDBC)
 - Python: DB-API
 - Perl: Database Interface (DBI)
 - PHP: PHP Data Objects (PDO)
- Drivers for each database are generally available adhering to these APIs
- Generally available as packages on Linux distributions

Summary

- The Web and WWW protocols: HTTP, MIME, RFC2822
- Web standards: HTML, CSS, JavaScript, XML, etc
- Web applications: web browsers and servers
- Apache web server software: configuration and control
- Apache configuration directives
- Virtual hosting, access control, TLS/SSL
- Server-side scripting languages: CGI, PHP, JSP
- Content-management systems
- Databases and SQL
- MySQL, PostgreSQL and SQLite
- Database APIs