Chapter 3 - Web Application Technologies

The HTTP protocol

- HTTP uses a message-based model
- essentially connectionless

HTTP requests

- consist of one or more headers, each on a new line
- followed by the *mandatory* black line
- followed by an optional message body
- The first line consists of: -
  - A verb indicating the HTTP method
  - The requested URL
  - The HTTP version being used (usually either 1.0 or 1.1)
    - in 1.1 the host header is mandatory
- The `Referer` header is used to indicate the URL from which the request originated
- The `User-Agent` header is used to provide information about the browser of the client software that generated the request
- The `Host` header specifies the hostname that appeared in the full URL being associated
- The `Cookie` header is used to submit additional parameters

HTTP responses

- The first line consists of three items:
  - HTTP version
  - numeric status code indicating the result of the request
  - a textual “reason phrase” further describing the status of the response
- The `Server` header contains a banner indicating the web server software used (may or may not be accurate)
- The `Set-Cookie` header issues the browser a further cookie
- The `Pragma` header instructs the browser not to store the response in its cache
- The `Expires` header indicates that the response content expired in the past and therefore should not be cached
- The `Content-Type` header indicates that the body of this message contains a HTML document
- The `Content-Length` header indicate the length of the message body in bytes

HTTP Methods

- GET is designed to retrieve resources
- POST is designed to perform actions
- HEAD same as GET except the server should not return a message body in its response
- TRACE is designed for diagnostic purposes. The server should return in the response body the exact contents of the request message it received
  - can be used to detect the effect of any proxy servers between the client and server that may manipulate requests
- OPTIONS asks for available methods, the server returns this in an `Allow` header
- PUT tries to upload a resource

URLs

- Uniform resource locator

- Format : `protocol://hostname[:port]/[path/]file[?param=value]` (anything in brackets is optional)
- can also have relative urls that dont have `protocol://hostname[:port]`

REST

- Representational state transfer
- “REST-style URL” used to signify a URL that contains its parameters within the URL file path. rather than a query string
  - URL containing a query string: `http://wahh-app.com/search?make=ford&model=pinto`
  - URL containing “REST-style” parameters : `http://wahh-app.com/search/ford/pinto`

HTTP headers

General Headers

- `Connection` tells the other end of the communication whether it should close the TCP connection after the HTTP transmission has completed or keep it open for further messages
- `Content-Encoding` specifies what kind of encoding is being used for the content contained in the message body
- `Content-Length` specifies the length of the message body, in bytes
- `Content-Type` specifies the type of content contained in the message body, such as `text/html` for HTML documents
- `Transfer-Encoding` specifies any encoding that was performed on the message body to facilitate its transfer over HTTP

Request headers

- `Accept` tells the server what kinds of content the client is willing to accept
- `Accept-Encoding` tells the server what kinds of content encoding the client is willing to accept
- `Authorization` submits credentials to the server for one of the uil-in HTTP auth types

- `Cookie` submits cookies to the server, ones that were issued by the server
- `Host` specifies the hostname that appeared in the full URL being requested
- `If-Modified-Since` specifies when the browser last received the requested resource
- `If-None-match` specifies an entity tag, which is an identifier denoting the contents of the message body.
- `Origin` is used in cross-domain Ajax requests to indicate the domain from which the request originated
- `Refer` Specifies the URL from which the current request originated
- `User-Agent` provides information about the browser or other client software that generated the request

Response Headers

- `Access-Control-Allow-Origin` indicates whether the resource can be retrieved via cross-domain Ajax requests
- `Cache-Control` passes caching directives to the browser, eg `no-cache`
- `ETag` specifies an entity tag.
- `Expires` tells the browser for how long the contents of the message body are valid, the browser will get resources from its cache until this time
- `Location` is used in redirection responses to specify the target of the redirect
- `Pragma` passes caching directives to the browser
- `Server` provides information about the web server software being used
- `Set-Cookie` issues cookies to the browser that it will submit back to the server in future requests
- `WWW-Authenticate` is used in responses that have a 401 status code to provide details on the type(s) of authentication that the server supports
- `X-Frame-Options` indicates whether and how the current response may be loaded within a browser frame

Cookies

- Cookies are sent in the following request without user knowledge/action
- In addition to the cookie's value, `Set-Cookie` can handle additional attributes:
  - `expires` sets the data for which the cookie is valid
  - `domain` specifies the domain for which the cookie is valid
  - `path` specifies the URL path for which the cookie is valid
  - `secure` only transmit cookie over HTTPS
  - `HttpOnly` the cookie cannot be directly accessed via clientside js

Status codes

- 1xx = informational
- 2xx = The request was successful
- 3xx = The client is redirected
- 4xx = The request contains an error of some kind
- 5xx = The server encountered an error fulfilling the request
- `100 Continue` The initial request was successful, keep sending data
- `200 OK` all good
- `201 Created` successful PUT request
- `301 Moved Permanently` will always redirect
- `302 Found` redirects to value in `Location` header
- `304 Not Modified` use your cached copy, the server cba
- `400 Bad Request` invalid HTTP request (format/syntax)
- `401 Unauthorized` Not authenticated
- `403 Forbidden` No one is allowed access, regardless of authentication
- `404 Not found` resource does not exist
- `405 Method Not allowed` bad method
- `413 Request Entity Too Large` server cant handle such a large request
- `414 Request URI Too Long` The URL itself is too long to handle

- `500 Internal Server Error` Caused a server side unhandled error
- `503 Service Unavailable` The web server is up, but the service the web server tried to use had a problem

HTTPS

- HTTP + SSL
- SSL has been superseded by TLS but we still use old name

HTTP Proxies

- A proxy sits between a client and a server and mediates responses and requests

HTTP Authentication

- Types of authentication mechanisms built into HTTP protocol:
- Basic = sends base64 encoded credentials in each request
    - NTLM = challenge-response mechanism
    - Digest = challenge-response mechanism uses MD5 checksums of a nonce with user's creds

Web Functionality

Server-Side Functionality

- HTTP requests can be used to send parameters to the application in three main ways:
    - In the URL query string
    - in the file path of REST-style URLs
    - In HTTP cookies
    - In the body of requests using the POST methods
- Web applications employ a wide range of technologies on the server-side to deliver their functionality:
    - Scripting languages such as PHP, VBScript, and Perl

- Web application platforms such as ASP.NET and Java
- Web servers such as Apache, IIS, and Netscape Enterprise
- Databases such as MS-SQL, Oracle, and MySQL
- Other back-end components such as filesystems, SOAP-based web services, and directory services

The Java Platform

- In the past, the Java Platform, Enterprise Edition (formerly J2EE) was a standard for large-scale enterprise applications
- Java-based web applications' confusing terms:
  - An Enterprise Java Bean (EJB) is a relatively heavyweight software component that encapsulates the logic
  - A Plain Old Java Object (POJO) ordinary Java object, used to denote objects that are user-defined. Simple and lightweight
  - A Java Servlet, object that resides on an application server and recieves HTTP requests and returns responses
  - A Java Web Container is a platform or engine that provides a run time environment for Java-based web apps. Eg Apache Tomcat, BEA WebLogic, and JBoss.
- Many Java web apps employ third-party and open source components:
  - Authentication - JAAS, ACEGI
  - Presentation layer - SiteMesh, Tapestry
  - Database object relational mapping - Hibernate
  - Logging - Log4J

ASP.NET

- Microsoft's web application framework
- Direct competitor to Java platform
- ASP.NET has inbuilt xss protection, but not all protection

PHP

- used in conjunction with other free techs known as LAMP stack
- LAMP stack = Linux as OS, Apache as the web server, MySQL as DB server, PHP as the programming language
- open source components:
    - Bulletin boards - PHPBB, PHP-Nuke
    - Administrative front ends - PHPMyAdmin
    - Webmail - SquirrelMail, IlohaMail
    - Photo galleries - Gallery
    - shopping carts - osCommerce, ECW-Shop

Ruby on Rails

- Strong emphasis on Model-View-Controller architecture
- Rails can autogenerate a model for database content, controller actions for modifying it, and default views for the application user
- ruby has alot of reported vulns

SQL

- SQL uses queries to perform common tasks such as reading, adding, updating and deleting data from a relational database

XML

- specification for encoding data
- tag-based
- XML docs often include a Document Type Definition (DTD), which defines the tags and attributes used in the document
- Used alot on the web (before JSON)

Client-Side Functionality

HTML

- tag-based

-
- XHTML is a development of HTML that is based on XML and has stricter specifications than HTML
- XHTML is more secure than HTML

Hyperlinks

- frequently contains URL parameters
- sends data without user knowing what the value of the parameters are (often)

Forms

- HTML forms allow for arbitrary input
- specifies method
- the request contains `x-www-form-urlencode`, parameters are represented in the message body as name/value pairs

CSS

- describes the presentation
- CSS is increasingly relevant both as a source of security vulns and as a means of delivering exploits

Javascript

- Stuff can be carried out on the client side without even notifying the server
- enhance useability
- often used to:
  - validate user-entered data
  - Dynamic modifying UI
  - Querying and updating the document object model (DOM)

VBScript

- JS alternative supported only in the Internet Explorer browser
- Less powerful and developed than JS

Document Object Model

- DOM is an abstract representation
- can be acquired and manipulated
- DOM includes an event model, allowing code to hook events such as form submission, navigation via links and keystrokes

Ajax

- Ajax is a collection of programming techniques
- Ajax is a way of loading more of a website smoothly by sending minimal data to the server and updating as much as possible on the clientside
- “Asynchronous Javascript and XML”
- no longer need to be asynchronous and no longer requires XML
- Ajax is still beneficial in providing a more seamless experience by avoiding the need to reload an entire page.

JSON

- Javascript Object Notation
- data transfer format that can be used to serialize arbitrary data.
- used nowadays in Ajax as an alternative to XML

Same-Origin Policy

- made to ensure content recieved from one website is only allowed to read and modify the site it came from
- Key features:
    - A page residing on one domain can cause an arbitrary request to be made to another domain. But it cannot itself process the data returned from that request
    - A page residing on one omain can load a script from another domain and execute this within its own context.
    - A page residing on one dmain cannot read or modify the cookies or other DOM data belonging to another domain.

HTML5

- it introduces new tags, attributes and APIs that can be leveraged to deliver XSS and other attacks
- it modifies the core Ajax technology, `XMLHttpRequest`, to enable two-way cross-domain interactions. This can lead to new cross-domain attacks
- it introduces new mechanisms for client-side data storage which can lead to user privacy issues, and new catagories of attack such as client-side SQL injection

“Web 2.0”

- Buzz word
- Heavy use of Ajax
- Increased cross-domain integration
- Use of new techs on client side
- More functionality supporting user-generated content, info sharing, and interaction
- new vulns

Browser Extension Technologies

- can be deployed as byte code or install native executables
- Thick-client techs you will encounter when attack a web app:
  - Java applets
  - ActiveX controls
  - Flash objects
  - Silverlight objects

State and sessions

- individual user’s data is held wit5hin a server-side structure called a session
- sessions used to keep track of users, this is done by issuing each client a token corresponding with their session

Encoding schemes

URL encoding

- uses US-ASCII character set
- several characters are restricted, `=`, `&`, `?` etc
- useful ones to know `%3d` correlates to `=`, `%2d` correlates to `%`, `%20` correlates to `space`, `%0a` correlates to `New line`, `%00` correlates to `Null byte`
- `+` can also be used as `%20`

Unicode encoding

- designed to accept the whole worlds characters
- For transmission of HTTP, it uses the format `%u` + a hexadecimal code
- eg, `%u2215` correlates to `/`

HTML encoding

- used to represent problematic chars in html docs
- eg `<` correlates to `<`

Base64 Encoding

- allows binary data to be saftley represented using only ascii chars
- Base64 encoding processes input data in blocks of three bytes. Each is divided into four chunks of six bytes. Six bits of data allows for 64 different possible permutations, each chunk can be represented using a set of 64 chars.
- Base64 uses this char set
  `ABDEFGHIJKLMNOPQRSTUVXYZabcdefghijklmnopqrstuvwxyz0123456789+/`
- many web apps use base64 to transmit binary data within cookies and other parameters, and to hide senstivie data

Hex encoding

- Base16 encoded

Remoting and Serialization Frameworks

- Allows developers to abstract away from the nature of distributed nature of web apps
- examples:
    - Flex and AMF
        - Silverlight and WCF
        - Java serialized objects