Introduction to the Web

CS-576 Systems Security
Instructor: Georgios Portokalidis
Fall 2018
Photographer Champions Black Cat Adoptions

This story began at an animal shelter with an adorable kitten named Imogen! In December of 2014 Los Angeles-based photographer Casey ...
The World Wide Web (abbreviated WWW or the Web) is an information space where documents and other web resources are identified by Uniform Resource Locators (URLs), interlinked by hypertext links, and can be accessed via the Internet.
Uniform Resource Locator (URL)

URL format
- Items in brackets are optional

scheme://[username:password@]hostname[:port][/path/to/resource][?query_string][#fragment]
https://www.facebook.com

scheme://[username:password@]hostname[:port]/[path/to/resource][?query_string][#fragment]

Scheme: https

No credentials

Hostname: www.facebook.com

Port: Not specified, therefore default used
  - 443 for HTTPS

Path: /

No query string, no fragment
http://example.com/foo/index.php?a=1&b=2#foo

Scheme: http

No credentials

Hostname: example.com

Port: Not specified, therefore default used
  ▪ 80 for HTTP

Path: /foo/index.php

Query string: a=1&b=2

Fragment: foo
  ▪ Fragments are not sent to the server, they are kept and used only by the client, typically to scroll to a particular location of the incoming document
    ▪ <a name="#foo"></a>
  ▪ A website can still access them via JavaScript
Step 0

The user types a URL in a browser
www.stevens.edu does not mean anything to a computer

Your browser needs to first find the IP address belonging to that domain name
nslookup

nslookup www.stevens.edu

Server: 155.246.149.79
Address: 155.246.149.79#53

www.stevens.edu  canonical name = www.stevens.edu.cdn.cloudflare.net.

Name: www.stevens.edu.cdn.cloudflare.net
Address: 104.16.126.51
Name: www.stevens.edu.cdn.cloudflare.net
Address: 104.16.125.51
How Does DNS Work?

DNS (Domain Name System) works through a distributed hierarchical database of DNS servers.

Your computer has what is called a “stub resolver”.

- This stub resolver does two things:
  - Ask your recursive resolver (typically provided to you by your ISP) to resolve domains for it
  - Remember (cache) the answer of recent queries
Talking to the Web Server

Web browser → OS → Web server

TCP connection

Common ports: 443, 80
Talking to the Web Server

Web browser | O S | TCP connection | O S | Web server

Common protocols of communication: HTTP or HTTPS

Common ports: 443, 80
HTTP and HTTPS

HTTP (HyperText Transfer Protocol)

HTTPS (HTTP Secure)

HTTP is encapsulated within HTTPS
HTTP Basics

Stateless protocol used to send and receive data
  ▪ Text-based → Human readable

Used by many applications
  ▪ Simplicity
  ▪ Most firewalls & intrusion prevention systems allow HTTP

HTTP transactions follow the same general format
  ▪ 3-part client request / server response
    1. request or response line
    2. header section
    3. entity body
HTTP Request

Request line

<METHOD> /path/to/resource?query_string HTTP/1.1

GET /index.html?param=value HTTP/1.0
Request with a Header Section

The header contains name value pairs

GET /search?q=searchterm HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0 ... Firefox/3.5.8
Accept: text/html,...
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Request with a Body Section

In this example the body is used to send parameters

```
POST /search HTTP/1.1
Host: www.google.com

Content-Type: application/x-www-form-urlencoded
Content-Length: 12

q=searchterm
```
Other HTTP methods

HEAD

- Works like GET but the server does not send the body of a response (it only sends the appropriate headers)

TRACE

- Designed for diagnostic purposes. Returns in its response body the exact request it received.

OPTIONS

- Returns the available methods for a specific resource.

PUT

- Allows the upload of a file in certain location. This should be disabled by default.
Popular Request Headers

All request headers are meant to communicate some information to the server

**User-Agent**
- Family and version of browser, as well as the underlying environment

**Accept**
- Kind of content the client is willing to accept

**Accept-encoding**
- What type of encoding the client supports (e.g. gzip)

**Host**
- The target website of this request

**Cookie**
- Send the server all cookies the browser has for this specific website

**Referer**
- Specifies the URL from which the current request originated
- Note the misspelling. This is intentional.
HTTP Response

Response line

HTTP/1.1 <STATUS CODE> <STATUS MESSAGE>

HTTP/1.1 200 OK
Date: Fri, 09 Apr 2010 12:40:23 GMT
Content-Type: text/html; charset=UTF-8

<html><head>
<title>searchterm - Google-Search</title>
</head><body bgcolor="#e5eccc"/>
HTTP Response

Here the body is used to send the requested data compressed
Popular Response Headers

All response headers are meant to communicate some information to the client (browser)

Cache-control:
- Passing caching directives to the client (e.g. no-cache)

Expires:
- How long the content is valid (and may be cached for)

Server
- Provides information about the identity of the server

Set-Cookie
- Sets cookies for this website
The Body of the Response

The browser gets the response and starts consuming it
- Drawing on the screen according to HTML code
- Fetching additional resources
- Executing code (JS, etc.)

The content received can be classified as

Static
- Content that is stable and determined by the path of the URL

Dynamic
- Content that is changes based on user input and server state
A Web Application

“a program that runs on a server, accepts inputs via the web, processes it, and finally returns some answer”

Inputs can be supplied by (almost) anyone

Developed in a variety of languages

- Mostly type/memory safe, but not always
A Typical Web Server

- Operating System
- HTTP Server (Apache, Nginx, etc.)
- CGI
- Plugins (e.g., PHP, JSP, etc.)
- Files, scripts, etc.
- Internet
- PHP
- Bash
- Perl
- Python
A Typical Web Server

[Diagram showing the components of a web server, including:
- Files, scripts, etc.
- Plugins (e.g., PHP, JSP, etc.)
- CGI
- Web application code
- PHP, Bash, Perl, Python
- HTTP Server (Apache, Nginx, etc.)
- Operating System
- Internet]
A Typical Web Server

Transform:
- HTTP requests → Programatically accessible inputs
- Programatically generated outputs → HTTP responses

Files, scripts, etc.

Plugins (e.g., PHP, JSP, etc.)

CGI

Operating System

HTTP Server (Apache, Nginx, etc.)

Internet

PHP
Bash
Perl
Python

Transform:
- HTTP requests → Programatically accessible inputs
- Programatically generated outputs → HTTP responses
HTTP is a Stateless Protocol

User=John, password=papa

Server

Good to see you again John!

User

Can you make a money transfer for me?

Please login!

Server
HTTP Session Management

HTTP is a stateless protocol

User = john, password = papa

Session ID = sdfdk4kl70sdfpfvi0sdfok;sd

 SID, transfer_amount = 100

Done!

User = john
Group = users

SID = Session ID
Implementing Session IDs

Encoding it into the URL as GET parameter

▪ Exposed! Visible
  ▪ Even when using encrypted connections
    ▪ Stored in logs, history, visible in browser location bar

Hidden form field submitted in POST requests

▪ Lost when browser tab is closed

Cookies

▪ Preferable
▪ Survives when browser tab is closed
▪ Can be rejected by clients
Cookies

Token that is set by server, stored on client

Key-value pairs ("name=value")

Access control based on server domain
What Are Cookies Used For?

Authentication
- The cookie proves to the website that the client previously authenticated correctly

Personalization
- Helps the website recognize the user from a previous visit

Tracking
- Follow the user from site to site; learn his/her browsing behavior, preferences, and so on
Cookie Variations

Non-persistent cookies
- Only stored in memory during browser session

Secure cookies
- Only transmitted over encrypted (SSL) connections
- Only encrypting the cookie is vulnerable to replay attacks

Cookies that include the IP address
- Example: hash(IP) + nonce
- Makes cookie stealing harder
- Breaks session if IP address of client changes during that session
Crypto Systems and the Web

Crypto systems have enabled the Web to grow

They..

- keep content secret from unauthorized entities (3rd parties)
- protect content from unauthorized modification
- confirm the identity of communicating entities