IT350 Web and Internet Programming

SlideSet #16: HTTP and HTTPS
Chapter 21 4th edition or Chapter 17 5th edition
http://www.garshol.priv.no/download/text/http-tut.html

Client / Server Big Picture

Language for these communications?
HTTP Protocol Stack

<table>
<thead>
<tr>
<th>HTTP</th>
<th>FTP</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Various</td>
<td></td>
</tr>
</tbody>
</table>

What does HTTP request look like?

- User enters URL:
  http://www.usna.edu/
- Browser sends request to www.usna.edu:
  GET / HTTP/1.0
  User-Agent: Mozilla/3.0 (compatible; Opera/3.0; Windows 95/NT4)
  Accept: */*
- What would GET line be if URL were…
  http://www.usna.edu/cs/news.html
What does HTTP response look like?

- If okay, server sends back response:
  HTTP/1.1 200 OK
  content-length: 4303
  accept-ranges: bytes
  server: Apache/2.0.54 (Unix) PHP/5.0.4
  last-modified: Tue, 13 Aug 2013 13:18:07 GMT
  connection: close
  etag: "328f-10cf-1c8181c0"
  x-pad: avoid browser bug
  date: Wed, 14 Aug 2013 17:58:32 GMT
  content-type: text/html

  <!DOCTYPE html>
  <html> <head> …

Lab Exercise

- How does the HTTP request look like for

  - ssh into zee.academy.usna.edu
  - Use netcat (nc) to request the page above
    - nc www.usna.edu 80
    - [The HTTP request here + 1 empty row]

- What is the answer?
Variants of the HTTP request

HEAD / HTTP/1.0

GET /cgi-bin/query.pl?key=value HTTP/1.0

POST /cgi-bin/query.pl HTTP/1.0
   Content-Type: application/x-www-form-urlencoded
   Content-Length: 16

str=value

GET /img1.jpg HTTP/1.1
Host: www.host1.com

GET /img6.jpg HTTP/1.1
Host: www.host1.com
Connection: close


Variants of the HTTP response

- Status codes
  200 OK
  301 Moved permanently
  400 Bad request
  403 Forbidden
  404 Not found
  500 Internal server error
  503 Service unavailable
Exercise: How do the HTTP request and response look like?

http://zee.academy.usna.edu/~adina/welcome.pl?username=ac

welcome.pl

#!/usr/bin/perl
use strict;
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

my $username = param("username");

print "Set-Cookie: Username=$username; 
";
print header();
print start_html();
print h1("Welcome $username");
print end_html();

Lab Exercise:
1. Create Lab09 on your W drive. Create welcome.pl with the code below
2. How do the HTTP request and response look like? Use nc to verify!

http://zee.academy.usna.edu/~mXXXX/Lab09/welcome.pl?username=ac

welcome.pl

#!/usr/bin/perl
use strict;
use CGI qw(:standard);
use CGI::Carp qw(warningsToBrowser fatalsToBrowser);

my $username = param("username");

print "Set-Cookie: Username=$username; 
";
print header();
print start_html();
print h1("Welcome $username");
print end_html();
Lab Exercise

• Open Chrome ➔ Tools ➔ Developer Tools ➔ Network tab

• Type
  http://zee.academy.usna.edu/~mXXXX/Lab09/welcome.pl?username=ac in the address bar (XXXX –your alpha)
  – What http request is made?
  – Check the content of the request by right-clicking on the path in the list and choose “Copy the request headers” – paste it into Notepad++

• Type www.google.com in the address bar
  – What http requests are made? Why?

• Type www.amazon.com in the address bar
  – How many requests are made?

HTTPS: HTTP over SSL

<table>
<thead>
<tr>
<th>HTTP</th>
<th>FTP</th>
<th>…</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL</td>
<td>TCP</td>
<td>IP</td>
</tr>
<tr>
<td></td>
<td>Various</td>
<td></td>
</tr>
</tbody>
</table>
Encryption

- Encode / decode the data
- Need:
  - algorithm
  - key

- Symmetric encryption

- Asymmetric encryption
  - Interesting property: you can switch encryption and decryption key and get the same results

SSL Protocol Highlights

- Browser connects to SSL-enabled server
- Computers agree on encryption method
- Server sends its digital certificate (contains the public key)
- Browser and server generate session key
- Further communications are encrypted using the session key

Some from: http://video.google.com/videoplay?docid=7130470471741831613
HTTP over SSL: How to use it?

- Example: need to submit login information securely; script to execute is login.pl
- Secure invocation:

Digital Certificates

- Bob got a public key from Amazon. Is it really Amazon’s key?