Web Security II: Basic Concepts and UI attacks

Chengyu Song

Slides modified from Raluca Ada Popa and Dan Boneh
Administrivia

- Homework2
What is Web?

Web is a platform for deploying applications and sharing information, **portably and securely**
Hypertext Transfer Protocol

HTTP: a common data communication protocol on the web.
Uniform Resource Locator

URL: global identifiers of network-retrievable resources

Example:

http://safebank.com:81/account?id=10#statement
HTTP request

Sending commands to the sever side, like system call.

**GET**: no side effect

**POST**: possible side effect

<table>
<thead>
<tr>
<th>Method</th>
<th>Path</th>
<th>HTTP version</th>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/index.html</td>
<td>HTTP/1.1</td>
<td></td>
</tr>
<tr>
<td>Accept:</td>
<td>image/gif,</td>
<td>image/x-bitmap,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>image/jpg,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accept-Language:</td>
<td>en</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Connection:</td>
<td>Keep-Alive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User-Agent:</td>
<td>Chrome/21.0.1180.75 (Macintosh; Intel Mac OS X 10_7_4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Host:</td>
<td><a href="http://www.safebank.com">www.safebank.com</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Referer:</td>
<td><a href="http://www.google.com?q=dingbats">http://www.google.com?q=dingbats</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data – none for GET
Blank line
HTTP response

Retrieving results from the server side, like system call returns.

```
HTTP/1.0 200 OK
Date: Sun, 12 Aug 2012 02:20:42 GMT
Server: Microsoft-Internet-Information-Server/5.0
Connection: keep-alive
Content-Type: text/html
Last-Modified: Thu, 9 Aug 2012 17:39:05 GMT
Set-Cookie: ...
Content-Length: 2543

<html>
  This is web content formatted using html
</html>
```
HyperText Markup Language

**HTML**: a markup language to create structured documents that can embed images, objects, create interactive forms, etc.

```html
<html>
  <body>
    <div>foo <a href="http://google.com">Go to Google!</a></div>
    <form>
      <input type="text" />
      <input type="radio" />
      <input type="checkbox" />
    </form>
  </body>
</html>
```
Web security: a historical perspective

• Similar to Internet, web is an example of "bolt-on security"

• Originally, the World Wide Web (www) was invented to allow physicists to share their research papers
  • Only textual web pages + links to other pages
  • No security model to speak of
Web security: nowadays

• The web became complex and adversarial quickly

• Web pages become very complex with embedded images, JavaScript, dynamic HTML, AJAX, CSS, frames, audio, video, sensors, VR, ... from different servers
  • Today, a web site is a distributed application

• Web applications also become very diverse, news, shopping, videos, social network, banking, gaming, ...
  • Attackers have various motivations
Desirable security goals

• **Integrity**: malicious websites should not be able to tamper with the integrity of my computer or my information on other web sites

• **Confidentiality**: malicious websites should not be able to learn confidential information from my computer or other web sites

• **Privacy**: malicious websites should not be able to spy on me or my activities online
How to achieve these goals?
How to achieve these goals?

- Reference monitor (access control)
  1. How to name/identify subject and object?
  2. What would be the access control policy?
- What about network level?
  - One layer at a time
  - TLS, DNSSEC, etc
Same-origin policy

• The most important access control policy for web applications

  1. Each site in the browser is isolated from all others
  2. Multiple pages from the same site are not isolated
Same-origin policy: different sites

browser:

security barrier

wikipedia.org

mozilla.org
Same-origin policy: same site

browser:

No security barrier

wikipedia.org

wikipedia.org
What is an Origin?

- Origin = protocol + hostname + port

http://coolsite.com:81/tools/info.html
How to define the origin?

- The origin of a resource is derived from the URL it was loaded from

http://en.wikipedia.org

http://upload.wikimedia.org
How to define the origin?

- Special case: Javascript runs with the origin of the page that loaded it.
## Exercises

<table>
<thead>
<tr>
<th>Originating document</th>
<th>Accessed document</th>
</tr>
</thead>
</table>
## Exercises

<table>
<thead>
<tr>
<th>Originating document</th>
<th>Accessed document</th>
</tr>
</thead>
</table>
Cross-origin communication

- Similar to IPC, different origins can communicate through a narrow API: `postMessage`
- Receiving origin decides if to accept the message based on origin.

Check origin, and request!
UI attacks

• Use visual tricks to lure users to perform unintended bad operations

• Address bar attack
  • Exploitation where the URL displayed in the address bar is not the one you visited

• Clickjacking attacks
  • Exploitation where a user's mouse click is used in a way that was not intended by the user
Safe to type your password?
Safe to type your password?
Safe to type your password?
Status bar

- Trivially spoofable

  `<a href="http://www.paypal.com/
      onclick="this.href = 'http://www.evil.com/';">PayPal</a>`
Cursorjacking

- Can customize cursor!

  CSS example:
  ```css
  #mycursor {
    cursor: none;
    width: 97px;
    height: 137px;
    background: url("images/custom-cursor.jpg")
  }
  ```

- Javascript can keep updating cursor, can display shifted cursor

  Fake cursor, but more visible  
  | Real cursor |
Cursorjacking

Fake, but more visible  real
Cursorjacking

You will be redirected to the requested page in 60 seconds.

Fake cursor

Real cursor
For next class ...

- Web Security III: Server Side Attacks