SY110
Client-Side Scripting
Non-Event Driven

Major Brian Hawkins, USMC

U.S. Naval Academy

Fall AY 2018

Outline

Review

What we talked about last time

- The WWW
- HTML
- HTTP
- A day in the life. . .

Static vs. Dynamic

Static web pages

- Page content doesn't change
- Changes occur by loading new pages only
- All the web pages we've seen thus far in class

Dynamic web pages

- User interactions cause page to change
- Pages that look different to different users
  - Facebook, Google, YouTube

Event-Driven vs. Non-Event-Driven

- Subsets of dynamic web pages
  - Event-Driven – Requires user interaction or a timer before dynamic content displayed
  - Non-Event-Driven – Requires no interaction; the script just gets executed when the browser reads the HTML file

Client-side vs Sever-side

Dynamic content can occur in two ways

- Client-side – JavaScript runs on the client machine (web browser)
- Server-side – JavaScript runs on the server hosting the content

Consequences

- Security
  - if the code is flawed or vulnerable, the machine running the code incurs the risk.
- CPU cycles
- Network communication – server-side scripts require connectivity to the server to view content changes

The scripts in this lecture are client-side, written in JavaScript, and non-event driven. This means that they require no specific trigger (other than navigating to that page or refreshing) before they're executed.

Functionality vs Security

Inherent tension between functionality and security

Functionality vs. Security

- JavaScript can be disabled in your browser, which neutralizes attacks that use JavaScript running on your machine as an attack vector...
- ...but this causes a lot of useful dynamic content to be unavailable.

Document Object Model (DOM)

Dynamic Content w/JavaScript

- Recall the `<tag> </tag>` concept in HTML
- We can also embed JavaScript scripts into HTML files
  - `<script type="text/javascript"> ...</script>`
- These scripts can manipulate any part of the page
  - Changes the HTML, which is what is **rendered** by the browser
  - JavaScript can write to the page with the function `document.write()`

Examples

Embedded scripts in separate files

Oftentimes it is convenient to reuse the same code in multiple pages, or to keep code separate. We can save our JavaScript in a separate file with the `.js` extension, and reference it like:

```
<script type="text/javascript" src="ex.js"> </script>
```

which is identical to:

```
<script type="text/javascript">document.write("hello")</script>
```

assuming the file `ex.js` just contains `document.write("hello")`.

JavaScript `src`

The source of the JavaScript can come from any URL – why could this be dangerous?

Redirection Examples

DOM: document.location

- The variable that holds the URL for the current page.
- Can be manipulated with JavaScript.

See example in course notes for this lecture.

`http://rona.academy.usna.edu/~sy110/lec/wwwClntNonEvent/lec.html`

HTML & Email

HTML-formatted email

- Most email clients allow for
  - Plaintext formatted email
  - HTML formatted email
- Mail clients executing JavaScript can be dangerous – why?
- Images in HTML formatted email allow for determining whether email has been opened and when – how?

HTML email attachments

- HTML email attachments are opened by the browser
- Can automatically redirect recipient to any page an attacker wants

Questions

Questions?