“Some cities, when confronted with fun, think, ‘I know, I’ll fuck with the DNA Lounge’. Now they have two problems.”
E Is for ECMAScript

• Cross-platform, vendor-neutral liability
• Event-driven, asynchronous bug generator
• Easy to use, easier to misuse
• Challenging to maintain
try {
    security();
}
let me = count(ways);

**jsfunfuzz**
Over five years of fuzzing Mozilla's browser to find JavaScript-related bugs.
var Pwn2Own = $money;

address book, photos, videos and browsing history from a fully patched iPhone 4S.

The hack, which netted a $30,000 cash prize at the mobile Pwn2Own contest here, exploited a WebKit vulnerability to launch a drive-by download when the target device simply surfs to a booby-trapped web site.

"It took about three weeks, starting from scratch, and we were only working on our private time,"
<script>
var arrr = new Array();
arrr[0] = window.document.createElement("img");
arrr[0]["src"] = "L";
</script>
<iframe src="child.html">

<head><script>
function funcB() { document.execCommand("selectAll"); }
function funcA() {
    document.write("L");
    parent.arrr[0].src="YMjf\\u0c08\\u0c0cKDogjsiIejengNEkoPDjfiJDIWUAzdfghjAAuUFGBSIPPPUDFJKSOQJG H";
}
</script></head>
<body onload='funcB();' onselect='funcA()'>
<div contenteditable='true'>a</div>
Internal Browser Security

- Process separation
- Sandboxing plugins
  - HTML5 does away with plugins altogether
- XSS Auditors
  - Only for the simplest scenarios
- Phishing warnings
  - Primarily for known sites
  - Some behavioral patterns, e.g. URL authority abuse
- Auto-updating
Dangerous Territory
Subtle and Quick to Anger

• Scope, blocks, & var
• Types & type coercion

```javascript
typeof null == "object";
typeof undefined == "undefined"
null == undefined;
null === undefined; // nope!
```

```javascript
```
JavaScript Crypto

• Use TLS for channel security
  • Better yet, use HSTS and DNSSEC.

• There is no trusted execution environment
  • ...in the current prototype-style language
  • ...in an intercepted HTTP connection
  • ...in an exploitable HTML injection vuln
JavaScript Crypto

- Math.random()
- sjcl.random
  - Fortuna-like generator
  - Entropy estimator
  - Exceptions

```javascript
sjcl.random.addEntropy([x,y], 2, "mouse")
sjcl.random.addEntropy((new Date()).valueOf(), 2, "loadtime");
sjcl.random.addEntropy(ab, 1024, "crypto.getRandomValues"); // WebKit
```
HTML Injection

• The 20+ year-old vuln that refuses to die.

• But JavaScript makes the situation better!

• No, JavaScript makes the situation worse!

• HTML5 to the rescue! (?)
Here are some things recently added to GeoCities

GeoCities now gives you 2MB of space for your home page!
GeoCities says "Happy Holidays" to our homesteaders with a gift of disk space. All homesteaders are able to use up to 2 megabytes of space for their personal home page. What will you do with your extra space?

PRODUCTION ASSISTANT - HTML Programmer (Full Time Freelance) Entry level. Assist online production department in adaptation and creation of content for the Web. Primary responsibilities include uploading and inputting information for daily maintenance, and turning larger projects into top-notch HTML. Some research, assisting with live chat and being patient is involved. Could involve creation and development of new content areas as time permits. You must be excellent with HTML 3.0 and all the Netscape extensions (tables, frames, and image maps are crucial). It would be superb if you can implement forms and know what CGI is all about. It would be even better if you have a knowledge of Java, Shockwave and a strong desire to learn and implement new technologies. Don't even think about it unless you're good with Macintosh, Photoshop, UNIX, graphic formats, know your MIME types, and IRC. It goes without saying that you need a good head for design and know your MTV. It will be helpful if you are a hardcore music/pop culture fan. Please include your URL with your email.

<!doctype html>
Capability, Security, Privacy

“In a world with one eye on privacy, the blind browser is king.”

- AppCache
- Battery Status
- Geolocation
- Web Storage
- WebGL
- WebPerf APIs
- Browser Fingerprinting
- Device Fingerprinting
- Usage Statistics
- User Tracking

* choose two (one?)
Oh, No! XSS Is Worse!

http://web.site/vuln?foo=xss"...

<input type="text" name="foo" value="xss" autofocus onchange="window.alert(9);"/>

(yawn)
XSS Blacklisting Is Worse

- New elements
- New attributes
- Didn’t work in the first place
  - `<img src="">"onerror=alert(9)>
  - `<img src="\"a\"">"onerror=alert(9)>
  - `<a href=""&<img&amp;/onclick=alert(9)>foo</a>`
We’ll Always Have Parsers

```html
<input type="text" name="foo" value="''" onmouseover="alert(9)"">

<script%20!–%20–>alert(9)</script>

<script/%a>alert(9)</script>

<script/%a>alert(9)</script> <a>foo</a>
```
We’ll Always Have People

<table>
<thead>
<tr>
<th>Keyword</th>
<th>State</th>
<th>Data type</th>
<th>Control type</th>
</tr>
</thead>
<tbody>
<tr>
<td>hidden</td>
<td>Hidden</td>
<td>An arbitrary string</td>
<td>n/a</td>
</tr>
<tr>
<td>text</td>
<td>Text</td>
<td>Text with no line breaks</td>
<td>A text field</td>
</tr>
<tr>
<td>search</td>
<td>Search</td>
<td>Text with no line breaks</td>
<td>Search field</td>
</tr>
<tr>
<td>tel</td>
<td>Telephone</td>
<td>Text with no line breaks</td>
<td>A text field</td>
</tr>
<tr>
<td>url</td>
<td>URL</td>
<td>An absolute URL</td>
<td>A text field</td>
</tr>
<tr>
<td>email</td>
<td>E-mail</td>
<td>An e-mail address or list of e-mail addresses</td>
<td>A text field</td>
</tr>
<tr>
<td>password</td>
<td>Password</td>
<td>Text with no line breaks (sensitive information)</td>
<td>A text field that obscures data entry</td>
</tr>
<tr>
<td>datetime</td>
<td>Date and Time</td>
<td>A date and time (year, month, day, hour, minute, second, fraction of a second) with the time zone set to UTC</td>
<td>A date and time control</td>
</tr>
<tr>
<td>date</td>
<td>Date</td>
<td>A date (year, month, day) with no time zone</td>
<td>A date control</td>
</tr>
<tr>
<td>month</td>
<td>Month</td>
<td>A date consisting of a year and a month with no time zone</td>
<td>A month control</td>
</tr>
<tr>
<td>week</td>
<td>Week</td>
<td>A date consisting of a week-year number and a week number with no time zone</td>
<td>A week control</td>
</tr>
<tr>
<td>time</td>
<td>Time</td>
<td>A time (hour, minute, seconds, fractional seconds) with no time zone</td>
<td>A time control</td>
</tr>
<tr>
<td>datetime-local</td>
<td>Local Date and Time</td>
<td>A date and time (year, month, day, hour, minute, second, fraction of a second) with the time zone set to UTC</td>
<td>A date and time control</td>
</tr>
<tr>
<td>number</td>
<td>Number</td>
<td>A numerical value</td>
<td>A numerical value</td>
</tr>
<tr>
<td>range</td>
<td>Range</td>
<td>A numerical value</td>
<td>A numerical value</td>
</tr>
<tr>
<td>color</td>
<td>Color</td>
<td>An sRGB color</td>
<td>An sRGB color components</td>
</tr>
<tr>
<td>checkbox</td>
<td>Checkbox</td>
<td>A set of zero or one value</td>
<td>A set of zero or one value</td>
</tr>
<tr>
<td>radio</td>
<td>Radio Button</td>
<td>An enumerated set of values</td>
<td>An enumerated set of values</td>
</tr>
<tr>
<td>file</td>
<td>File Upload</td>
<td>Zero or more files, optionally a file</td>
<td>Zero or more files, optionally a file</td>
</tr>
<tr>
<td>submit</td>
<td>Submit Button</td>
<td>An enumerated set of values</td>
<td>An enumerated set of values</td>
</tr>
<tr>
<td>image</td>
<td>Image Button</td>
<td>A coordinate, the extra semantic is selected and</td>
<td>A coordinate, the extra semantic is selected and</td>
</tr>
</tbody>
</table>
Same Attacks, New Mechanisms

• Origin
• Referer
• X-Moz: prefetch
Improving SOP

• Granular access control
  • What happened to least privilege?

• Make the `<iframe>` more useful for isolating Origins
  • seamless
  • sandbox
### Sandbox Evaluation

<table>
<thead>
<tr>
<th>Sandbox Setting</th>
<th>Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>(empty)</td>
<td>JavaScript not executed</td>
</tr>
<tr>
<td><code>sandbox</code></td>
<td>JavaScript not executed</td>
</tr>
<tr>
<td><code>sandbox=&quot;allow-scripts&quot;</code></td>
<td>JavaScript executed</td>
</tr>
<tr>
<td><code>text/html-sandboxed</code></td>
<td>Waiting for browser support</td>
</tr>
</tbody>
</table>
On the Other Hand...

...if you’re not using X-Frame-Options: DENY.

```javascript
function killFrames()
{
if (top.location!=location)
{
if (document.referrer)
{
var a=get_hostname_from_url(document.referrer);
var b=a.length;
if (b==11&&a!="web.site")
{top.location.replace(document.location.href)}
else
if (b!=11&&a.substring(a.length-12)!=".web.site")
{top.location.replace(document.location.href)}
}
if (top.frames.length!=0) top.location=self.document.location;
}
function get_hostname_from_url(a)
{
return a.match(/://[/\]*(.[^/\?] +)/)[1];
}
killFrames();
```
Content Security Policy

- Granular access for retrieving resources
- Header only
  - Probably requires code changes, or unsafe-eval
  - (http-equiv has lower precedence)
- Waiting for universal implementation
  - X-Content-Security-Policy
  - X-WebKit-CSP
- http://www.w3.org/TR/CSP/
Selective Resource Control

X-CSP: default-src 'self'; frame-src 'none'

```html
<!doctype html>
<html>
<body>
    <iframe src="./infected.html"></iframe>
</body>
</html>
```
Defeat Exploits, Not Vulns

```
X-CSP: default-src 'self'

<input type="text" name="q" value="foo" autofocus onfocus=alert(9)//"/>
```

```
X-CSP: default-src 'self' 'unsafe-inline'

<input type="text" name="q" value="foo" autofocus onfocus=alert(9)//"/>
```
https://web.site/page#<img src="" onerror=alert(9)>

<!DOCTYPE html>
<html>
<head>
<script src="jquery-1.8.2.min.js"></script>
<script>
$(document).ready(function() {
    var x = (window.location.hash.match(/^[^!/].+$/) || []).[1];
    var w = $('a[name="" + x + '"", [id="" + x + '""']);
});
</script>
</head>
<body>
    <div id="main">foo</div>
</body>
</html>
https://web.site/page#<img src="" onerror=alert(9)>

```html
<!DOCTYPE html>
<html>
<head>
<script src="jquery-1.8.2.min.js"></script>
<script src="main.js"></script>
</head>
<body>
  <div id="main">foo</div>
</body>
</html>

$(document).ready(function() {
  var x = (window.location.hash.match(/^[^/].+$/) || [[]])[1];
  var w = $('a[name="' + x + '", [id="' + x + '"]');
});
```
As Easy as ABC

• Move intrinsic events from HTML to JavaScript

• Avoid “inline” event handler attributes

```javascript
$('#main').attr('onclick', 'alert(9)');
```

• Use event managers

```javascript
$('#main').bind("click", function(e) {
  alert(9) });
```

```javascript
$('#main').click(function(e) {
  alert(9) });
```

```javascript
$('#main').on("click", function(e) {
  alert(9) });
```
On the Other Hand...

...an awesome XSS DoS payload if injectable into a `<head>` section.

```html
<meta http-equiv="X-WebKit-CSP" content="default-src 'none'">
```
On the Other Hand...

...another way to forge POST method for CSRF.

```html
<!doctype html><html><head>
<meta http-equiv="X-WebKit-CSP"
    content="img-src 'none'; report-uri 'https://csrf.target/page?a=1&b=2&c=3'"> 
</head><body>
<img alt="" src="whatever"> 
</body></html>
```
Partial CSRF Influence

POST /page?a=1&b=2&c=3 HTTP/1.1
Host: csrf.target
User-Agent: Mozilla/5.0 ...
Content-Length: 116
Accept: */*
Origin: null
Content-Type: application/x-www-form-urlencoded
Referer: http://web.site/HWA/ch3/csrf.html
Cookie: sessid=12345
Connection: keep-alive

document-url=http%3A%2F%2Fcsrf.target%2FHWA%2Fch3%2Fcsrf.html&violated-directive=default-src+%27none%27
<script>
WebSocket.prototype._s = WebSocket.prototype.send;
WebSocket.prototype.send = function(data) {
//  data = ".";
    console.log("\u2192 " + data);
    this._s(data);
    this.addEventListener('message', function(msg) {
        console.log("\u2190 " + msg.data);
    }, false);
    this.send = function(data) {
        this._s(data);
        console.log("\u2192 " + data);
    }
};
</script>
Data = “.”

[22:49:57][*] BeEF server started (press control+c to stop)
   /opt/local/lib/ruby1.9/gems/1.9.1/gems/json-1.7.5/lib/json/common.rb:155:in `initialize': A JSON text must at least contain two octets! (JSON::ParserError)
“And what does it say now?” asked Arthur.
“Mostly harmless,” admitted Ford with a slightly embarrassed cough.
Programming

• Origin is an identity hint, not access control attribute
  • The return of X-Forwarded-For
• JSON serializes, not sanitizes, data
• Avoid string concatenation
  • Review, refactor, refine
• use strict
CORS

• Defines read-access trust of another Origin
  • Expresses trust, not security
  • But still contributes to secure design

• Principle of Least Privilege
  • Beware of Access-Control-Allow-Origin: *
  • Short Access-Control-Max-Age
  • Minimal Access-Control-Allow-{Methods | Headers}

• Check the Origin
What to Think About

• Start with an established library

• Abstracting vs. annotating
  • Closure or TypeScript

• Emscripten (!?)

Underscore JS
Angular
Batman JS
ObjectiveJ (Cappucino)
Google Closure
CoffeeScript
Dojo
Ember JS
Ext JS
Facebook Connect
jQuery
Knockout
Midori JS
Modernizr
MooTools
MooTools More
Prototype
Pusher
Qooxdoo
Raphael
Rico
Sammy
Scriptaculous
Socket.io
Spine
Spry
TypeKit
twttr
twttr
jsmd
UIZE
YUI
YAHOO
What’s Coming

• Steps towards a trusted environment
  • Freeze & Seal an Object
  • Object.hasOwnProperty()
  • Modular libraries
  • toStaticHtml()*
• More complexity
Here, There, Everywhere

- jQuery [ http://jquery.com ]
- sjcl.js [ http://crypto.stanford.edu/sjcl/ ]
- BeEF [ http://beefproject.com ]
- Screen Shots [ https://github.com/niklasvh/html2canvas ]
Code Like It’s Not 1999

- Encourage users to update browsers
  - Legacy support is a pain anyway
- Start with an established JavaScript library
  - Pure development vs. patch management
- Adopt HTML5 security features
  - ...to protect users with HTML5-enabled browsers
Thank You!
Questions?

• @CodexWebSecurum

• http://deadliestwebattacks.com

• Hacking Web Apps