

CLIENT-SIDE RUNTIME ANALYSIS AND ENFORCEMENT

Ben Livshits, Microsoft Research

Overview of Today's Lecture

2

- Background (rehash)
- Language restrictions
 - AdSafe
 - FBJS
- Extensive rewriting
 - Caja
 - WebSandbox
- Better runtimes
 - CSP
 - HTML5 Sandbox
- Tradeoffs of different containment strategies and going forward

JavaScript Security Model

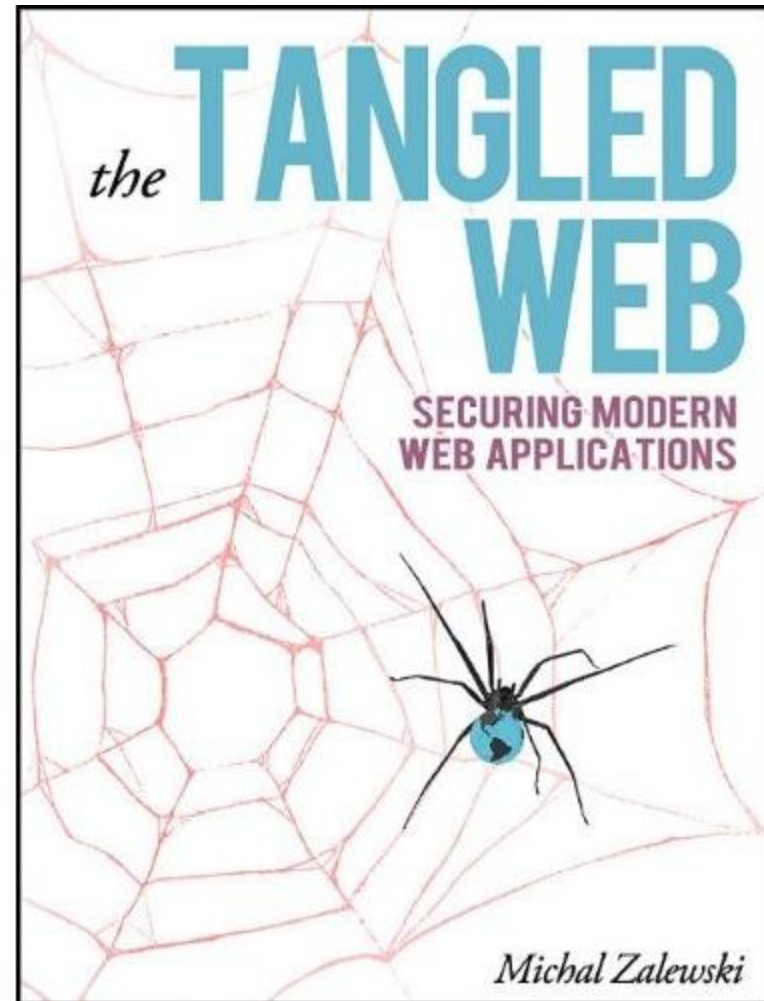
slide 3

- Script runs in a “sandbox”
 - ▣ No direct file access
 - ▣ Restricted network access
- Same-origin policy
 - ▣ Code can only access properties of documents and windows from the same origin
 - ▣ Gives a degree of isolation
 - ▣ Origin roughly is the URL, but not quite
 - If the same server hosts unrelated sites, scripts from one site can access document properties on the other
 - Is the origin always representative of content?

This is Just the Beginning...

4

- Browser Security Handbook
 - ▣ ... DOM access
 - ▣ ... XMLHttpRequest
 - ▣ ... cookies
 - ▣ ... Flash
 - ▣ ... Java
 - ▣ ... Silverlight
 - ▣ ... Gears
 - ▣ Origin inheritance rules



XmlHttpRequest

5

- XmlHttpRequest is the foundation of AJAX-style application on the web today
- Typically:

```
01.  var request = new XMLHttpRequest();
02.  request.open('GET', 'file:///home/user/file.json', false);
03.  request.send(null);
04.
05.  if (request.status == 0)
06.      console.log(request.responseText);
```

Virtually No Full Compatibility

6

Test description	MSIE6	MSIE7	MSIE8	FF2	FF3	Safari	Opera	Chrome	Android
Banned HTTP methods	TRACE	CONNECT TRACE ⁺	CONNECT TRACE ⁺	TRACE	TRACE	CONNECT TRACE	CONNECT TRACE ^{**}	CONNECT TRACE	CONNECT TRACE
XMLHttpRequest may see httponly cookies?	NO	NO	NO	YES	NO	YES	NO	NO	NO
XMLHttpRequest may see invalid HTTP 30x responses?	NO	NO	NO	YES	YES	NO	NO	YES	NO
XMLHttpRequest may see cross-domain HTTP 30x responses?	NO	NO	NO	YES	YES	NO	NO	NO	NO
XMLHttpRequest may see other HTTP non-200 responses?	YES	YES	YES	YES	YES	YES	YES	YES	NO
May local HTML access unrelated local files via XMLHttpRequest?	NO	NO	NO	YES	NO	NO	YES	NO	n/a
May local HTML access sites on the Internet via XMLHttpRequest?	YES	YES	YES	NO	NO	NO	NO	NO	n/a
Is partial XMLHttpRequest data visible while loading?	NO	NO	NO	YES	YES	YES	NO	YES	NO

Why is lack of compatibility bad?

Active Research and Development

7

Computer

Security Vulnerabilities in the Same-Origin Policy: Implications and Alternatives

September 2011 (vol. 44 no. 9)

pp. 29-36

Hossein Saiedian, University of Kansas

Dan S. Broyles, Sprint Nextel

DOI Bookmark: <http://doi.ieeecomputersociety.org/10.1109/MC.2011.226>

ABSTRACT

The same-origin policy, a fundamental security mechanism within Web browsers, overly restricts Web application development while creating an ever-growing list of security holes, reinforcing the argument that the SOP is not an appropriate security model.

ADDITIONAL INFORMATION

Index Terms:

Security, Web browsers, Web applications, Same-origin policy (SOP), Cross-site request forgery (CSRF), Cross-site scripting (XSS)

Citation:

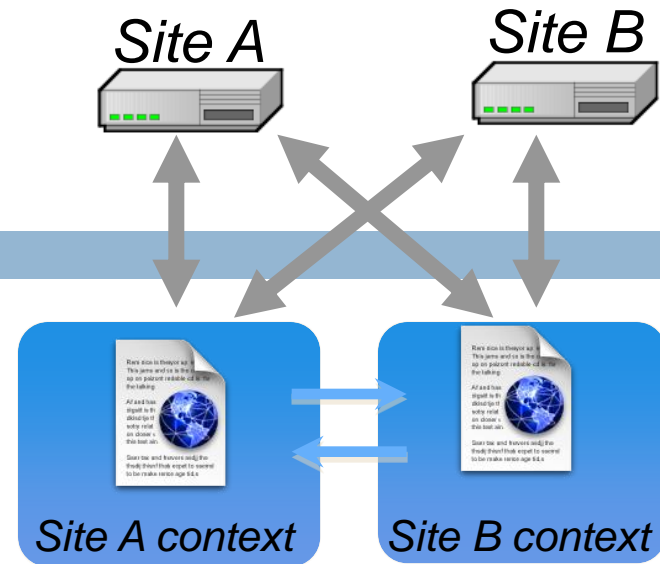
Hossein Saiedian, Dan S. Broyles, "Security Vulnerabilities in the Same-Origin Policy: Implications and Alternatives," *Computer*, vol. 44, no. 9, pp. 29-36, July 2011, doi:10.1109/MC.2011.226

How Do We Do Cross-Domain XHR?

8

- Server-side proxying
 - ▣ Is this a good idea?
- Alternatives abound, no consensus
 - ▣ XMLHttpRequest in IE8
 - ▣ XMLHttpRequest
 - ▣ CS-XHR

Recent Developments



- ❑ Cross-origin network requests

Access-Control-Allow-Origin: <list of domains>

Access-Control-Allow-Origin: *

- ❑ Cross-origin client side communication

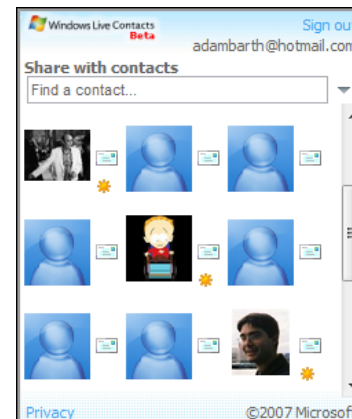
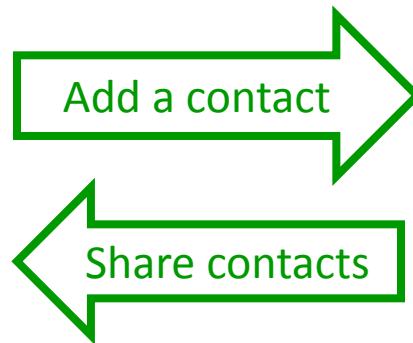
- Client-side messaging via **postMessage**

window.postMessage

- New HTML5 API for inter-frame communication
 - ▣ Supported in latest betas of many browsers



- ▣ A network-like channel between frames



Facebook Connect Protocol

11

- SOP policy does not allow a third-party site (e.g TechCrunch), called *implementor*, to communicate with facebook.com
- To support this interaction, Facebook provides a JavaScript library for sites implementing Facebook Connect
- Library creates two hidden iframes with an origin of facebook.com which in turn communicate with Facebook
- The cross-origin communication between hidden iframes and the implementor window are layered over postMessage

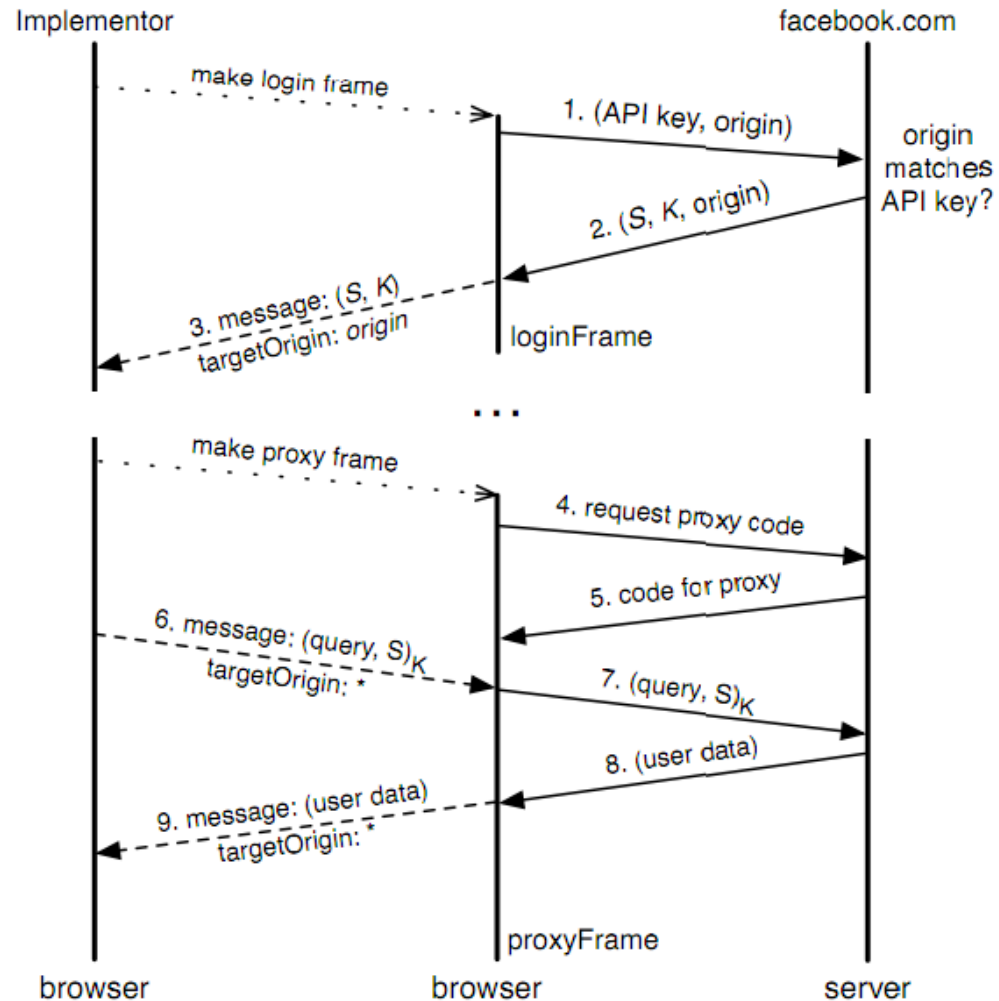
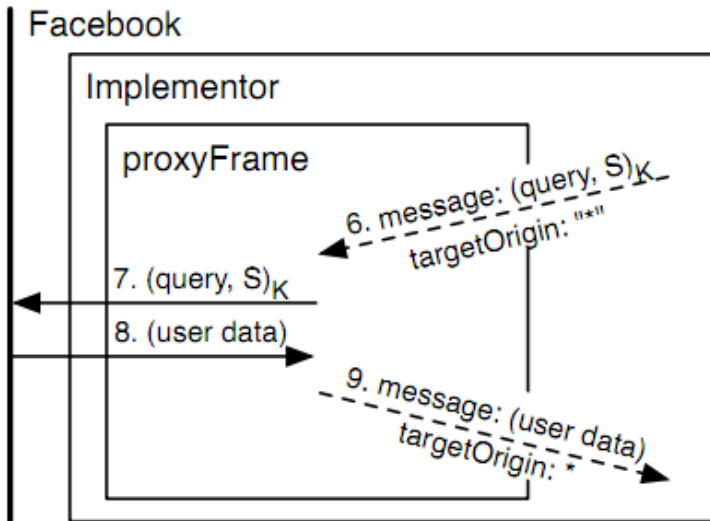
Facebook Connect

12

- Facebook Connect is a system that enables a Facebook user to share his identity with third-party sites
- Some notable users include TechCrunch, Huffington Post, ABC and Netflix
- After being authorized by a user, a third party web site can query Facebook for the user's information and use it to provide a richer experience that leverages the user's social connections
- For example, a logged-in user can view his Facebook friends who also use the third-party web site, and interact with them directly there
- Note that the site now contains content from multiple principals—the site itself and facebook.com

Facebook Connect

13

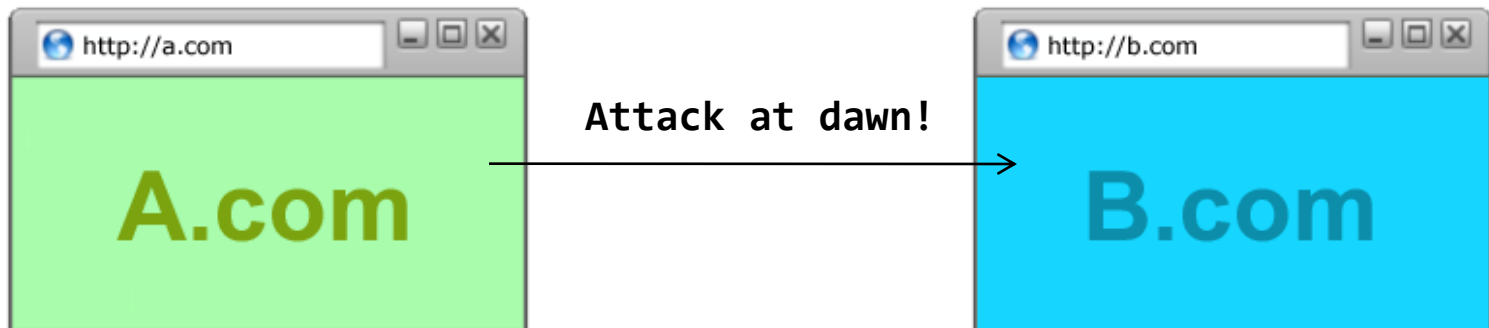


The Emperor's New APIs: On the (In)Secure Usage of New Client-side Primitives, Hanna et. al, 2010

postMessage syntax

```
frames[0].postMessage("Attack at dawn!",  
                      "http://b.com/");
```

```
window.addEventListener("message", function (e) {  
  if (e.origin == "http://a.com") {  
    ... e.data ...  
  }  
}, false);
```



Why Include The Target Origin?

- What goes wrong?

```
frames[0].postMessage("Attack at dawn!");
```

if we just do this?

- Are there other issues with the use of `postMessage`?

Trusted and Untrusted Web Content

16

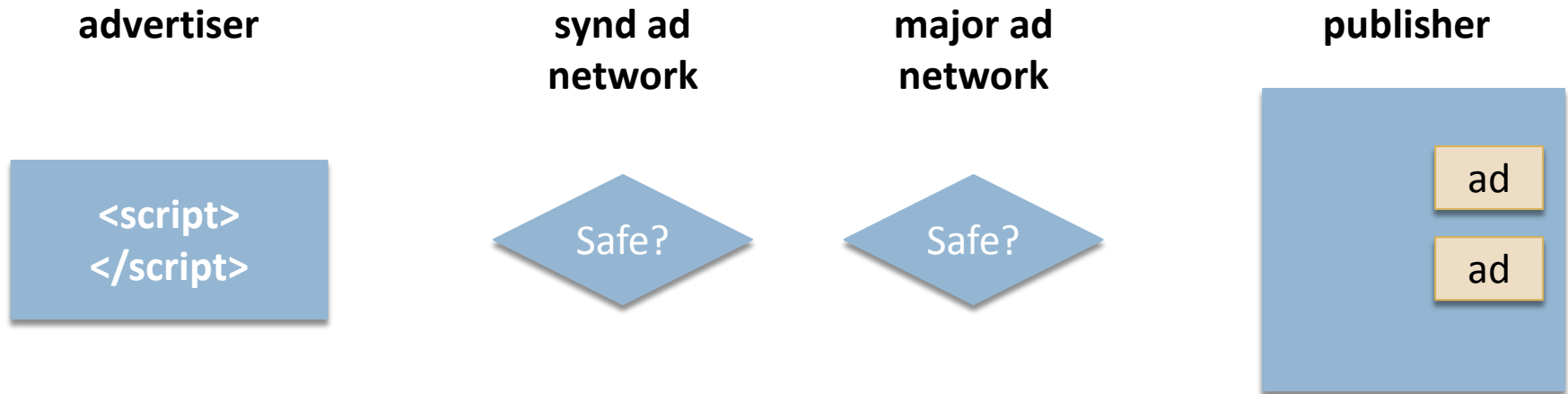
- Two trust levels:
trusted and untrusted
 - ▣ Trusted: code belonging to host.
 - ▣ Untrusted: all third-party code
- What is the issue?
 - ▣ Untrusted components are sequentially composed and placed in a trusted context
- Model fits the case of web pages with advertisements, iGoogle, Facebook Apps

17

JavaScript Language Restrictions

Ad Scenario: Why ADsafe?

18



- Ensure safety of ads containing JavaScript
- Always a good idea?

ADsafe Example

19

Making JavaScript

JavaScript, the programming language. Any script in a page and relationships of the page advertising unacceptably r

ADsafe makes it safe to place advertising or widgets on a page. JavaScript that is powerful enough to create complex interactions, while at the same time causing damage or intrusion. The ADsafe tools like [JSLint](#) so that no code for safety. The ADsafe increasing the likelihood th

The ADsafe subset blocks scripts from directly accessing the page. Instead, ADsafe gives the scripts by the page's server, giving elements and other page s

ADsafe does not modify scripts or alter their behavior. ADsafe determine that script is saf

And because ADsafe verifies every stage of the deployment compliance testing.

```
18 <script>
19 ADSAFE.go("ROMAN_", function (dom, lib) {
20     "use strict";
21     var roman = (function () {
22         var table = [
23             ['', 'I', 'II', 'III', 'IV', 'V', 'VI', 'VII', 'VIII', 'IX'],
24             ['', 'X', 'XX', 'XXX', 'XL', 'L', 'LX', 'LXX', 'LXXX', 'XC'],
25             ['', 'C', 'CC', 'CCC', 'CD', 'D', 'DC', 'DCC', 'DCCC', 'CM']
26         ];
27
28         return function (n) {
29             var result = '', i;
30
31             n = +n;
32             for (i = 0; i < table.length; i += 1) {
33                 result = table[+i][+(n % 10)] + result;
34                 n = Math.floor(n / 10);
35             }
36             for (i = 0; i < n; i += 1) {
37                 result = 'M' + result;
38             }
39             return result;
40         };
41     })();
42
43     var input = dom.q("input_text");
44     input
45         .on('enterkey', function (e) {
46             dom.q("#ROMAN_RESULT").value(roman(input.getValue()));
47             input.select();
48         })
49         .focus();
50 });
51 </script>
52 </div>
```

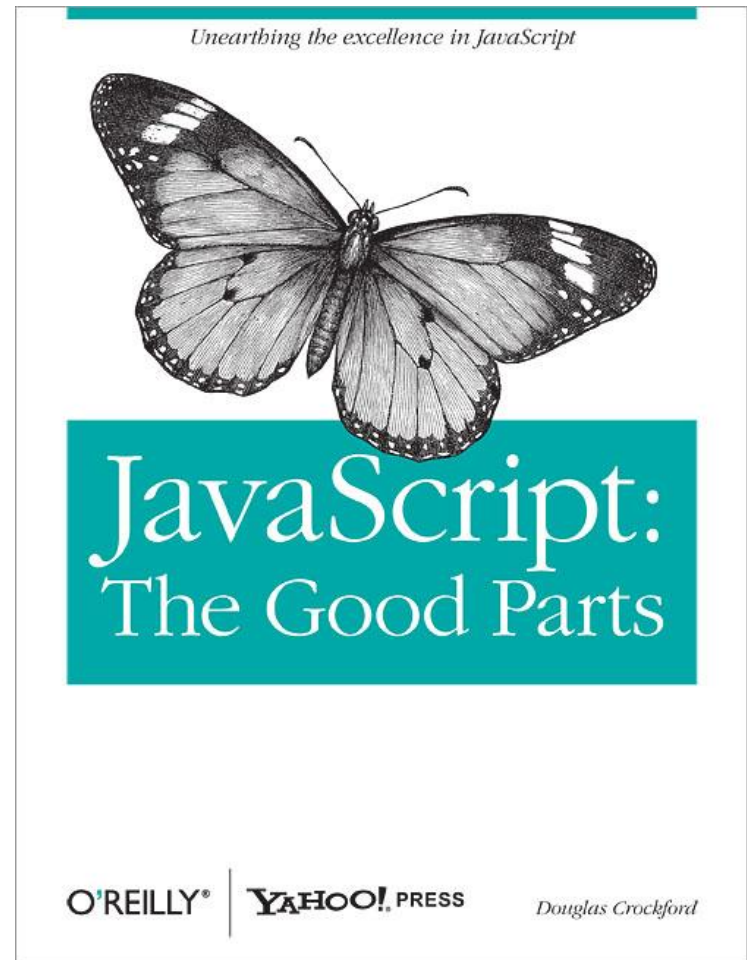
the box and press the [enter] key.

meral in the box and press the [enter]

ADsafe Goals

20

- ADsafe removes features from JavaScript that are either *unsafe* or grant uncontrolled access to *unsafe browser components* or that contribute to *poor code quality*



ADsafe Restrictions

21

- Global variables: ADsafe's object capability model prohibits the use of most global variables.
- Limited access: Array, Boolean, etc.
- `this`: If a method is called as a function, `this` is bound to the global object. Since ADsafe needs to restrict access to the global object, it must prohibit the use of `this` in guest code.
- `arguments`: Access to the arguments pseudo-array is not allowed.
- `eval`: The `eval` function provides access to the global object.
- `with` statement: The `with` statement modifies the scope chain, making static analysis impossible.
- Dangerous methods and properties: arguments callee caller constructor eval prototype stack unwatch valueOf watch
 - Capability leakage can occur with these names in at least some browsers, so use of these names with `.` notation is prohibited.
- Names starting or ending with `_`: Some browsers have dangerous properties or methods that have a dangling `_`.
- `[]` subscript operator except when the subscript is a numeric literal or string literal or an expression that must produce a number value: Lookup of dynamic properties could provide access to the restricted members. Use `ADSAFE.get` and `ADSAFE.set` instead
- `Date` and `Math.random`: Access to these sources of non-determinism is restricted in order to make it easier to determine how widgets behave

Trade-offs

22

ex

```
ADSAFE.go("AD_", function (dom, lib) {
  var myWindow, fakeNode, fakeBunch, realBunch;

  fakeNode = {
    appendChild: function(elt) {
      myWindow = elt.ownerDocument.defaultView;
    },
    tagName: "div",
    value: null
  };

  fakeBunch = {"__nodes__": [fakeNode]};

  realBunch = dom.tag("p");
  fakeBunch.value = realBunch.value;
  fakeBunch.value(""); // calls phony appendChild

  myWindow.alert("hacked");
});
```

safety

ADsafe

INTERNET
WayBack

2011

FBJS: How FB Apps are Programmed

23

- Basics
 - ▣ Facebook apps are either IFRAMEd or integrated
 - ▣ Integrated Facebook applications are written in FBML/FBJS
- FBJS: Facebook subsets of HTML and JavaScript
 - ▣ FBJS is served from Facebook, after filtering and rewriting
 - ▣ Facebook libraries mediate access to the DOM
- Security goals
 - ▣ No direct access to the DOM
 - ▣ No tampering with the execution environment
 - ▣ No tampering with Facebook libraries
- Isolation approach
 - ▣ Blacklist variable names that are used by containing page
 - ▣ Prevent access to global scope object

FBJS By Example

24

```
function foo(bar) {  
  var obj = {property: bar};  
  return obj.property;  
}
```

```
function a12345_foo(a12345_bar) {  
  var a12345_obj = {property: a12345_bar};  
  return a12345_obj.property;  
}
```

```
obj.className = "SBGGiftItemImage";
```

```
obj.setClassName("SBGGiftItemImage");
```

```
obj.onmouseout = function() {  
  this.className = "SBGGiftItemImage";};
```

```
obj.addEventListener("mouseout",  
  function()  
    {this.setClassName('SBGGiftItemImage');});
```


FBJS Restrictions

25

`o[e] -> a12345_o[$FBJS.idx(e)]`

- Other, indirect ways that malicious content might reach the window object involve accessing certain standard or browser-specific predefined object properties such as `__parent__` and `constructor`
- Therefore, FBJS blacklists such properties and rewrites any explicit access to them in the code into an access to the useless property `unknown`

More on FBJS

26

- Facebook Application Directory:
 - <http://www.facebook.com/apps/directory>
 - But also FBML and FBQL

- Subject of much research in 2009-2011
 - *Designing Malicious Applications in Social Networks*
 - *Preventing Capability Leaks in Secure JavaScript Subsets*
 - *Isolating JavaScript with Filters, Rewriting, and Wrappers*

27

Question of the Day

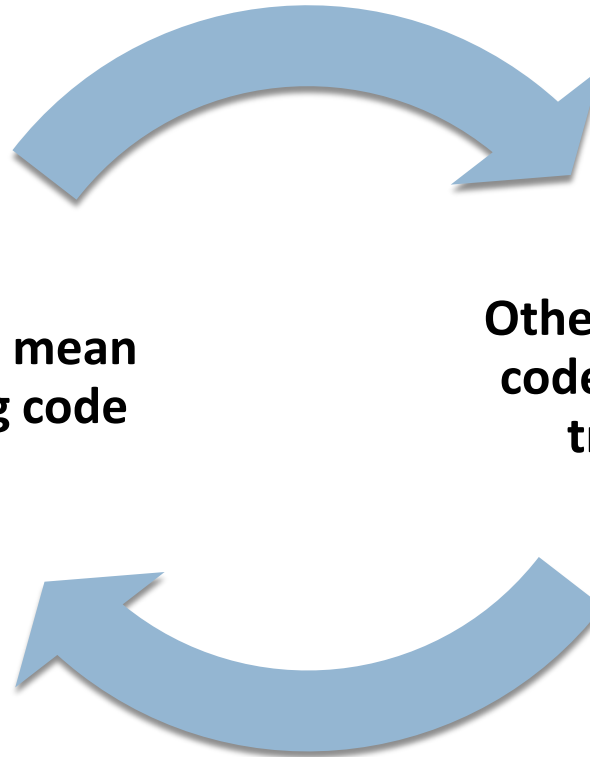
What Are the Pros/Cons of Static Restriction vs. Code Rewriting

Mashup Scenario: Developer's Dilemma



**Mashups mean
including code**

**Other people's
code can't be
trusted**



Typical Mashup: Yelp + Google Maps

29

The screenshot shows a Windows Internet Explorer browser window displaying a Yelp search for "indian" in Palo Alto, CA. The search results are filtered to show 1 to 10 of 144 results. The page includes a search bar, navigation links, and a list of restaurant results. A Google Map is overlaid on the right side of the results, showing the location of the restaurants in Palo Alto, CA. The map includes labels for various neighborhoods like Menlo Park, Palo Alto, and Mountain View, and major roads like 101 and 280.

Search for (e.g. taco, salon, Max's) **Near** (Address, City, State or Zip)
indian Palo Alto, Ca **Search**

Welcome About Me Write a Review Find Reviews Invite Friends Messaging Talk Events Member Search Account Log In

indian Palo Alto Browse Category: Indian/Pakistani 1 to 10 of 144 - Results per page: 10

Hide Filters

Sort By	Cities	Distance	Features	Price	Category
» Best Match Highest Rated Most Reviewed	<input type="checkbox"/> Palo Alto <input type="checkbox"/> Mountain View <input type="checkbox"/> Sunnyvale <input type="checkbox"/> Redwood City ... More Cities »	» Bird's-eye View Driving (5 mi.) Biking (2 mi.) Walking (1 mi.) Within 4 blocks	<input type="checkbox"/> Good for Groups <input type="checkbox"/> Take-Out <input type="checkbox"/> Takes Reservations ... More features »	<input type="checkbox"/> \$\$\$\$ <input type="checkbox"/> \$\$\$ <input type="checkbox"/> \$\$ <input type="checkbox"/> \$	<input type="checkbox"/> Indian/Pakistani <input type="checkbox"/> Restaurants <input type="checkbox"/> Specialty Food <input type="checkbox"/> Bakeries ... More categories »

Sponsored Result

New Saffron
Category: Indian/Pakistani
★★★★★ 8 reviews
2700 W El Camino Real
Mountain View, CA 94040
(650) 948-0123

Two words. DINNER BUFFET. Where have you been all my adult life? \$12.95. All you can eat Indian food goodness. For vegetarians and carnivores alike. Also, no gnats. A plus. My go-to Indian...

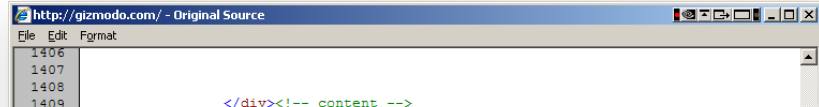
▼ **1. Darbar Indian Cuisine**
Category: Indian/Pakistani
★★★★★ 110 reviews
129 Lytton Avenue
Palo Alto, CA, 94301
(650) 321-6688

even have tamarind chutney, which is weird/annoying for a restaurant that is trying to be authentic (as opposed to Junnoon which is modern/fusion indian). Their paneer do pizza and their onion kulcha

▼ **2. Junnoon Restaurant**
Category: Indian/Pakistani
★★★★★ 92 reviews
150 University Avenue

Done Internet 100%

Web-based Counter



```
<div id="sitemeter" class="plain">
<!--WEBBOT bot="HTMLMarkup" startspan ALT="Site Meter" -->
<script type="text/javascript" language="JavaScript">var
site="s15gizmodo"</script>
<script type="text/javascript" language="JavaScript1.2"
src="http://s15.sitemeter.com/js/counter.js?site=s15gizmodo">
</script>
```



Failure Should Not Be An Option



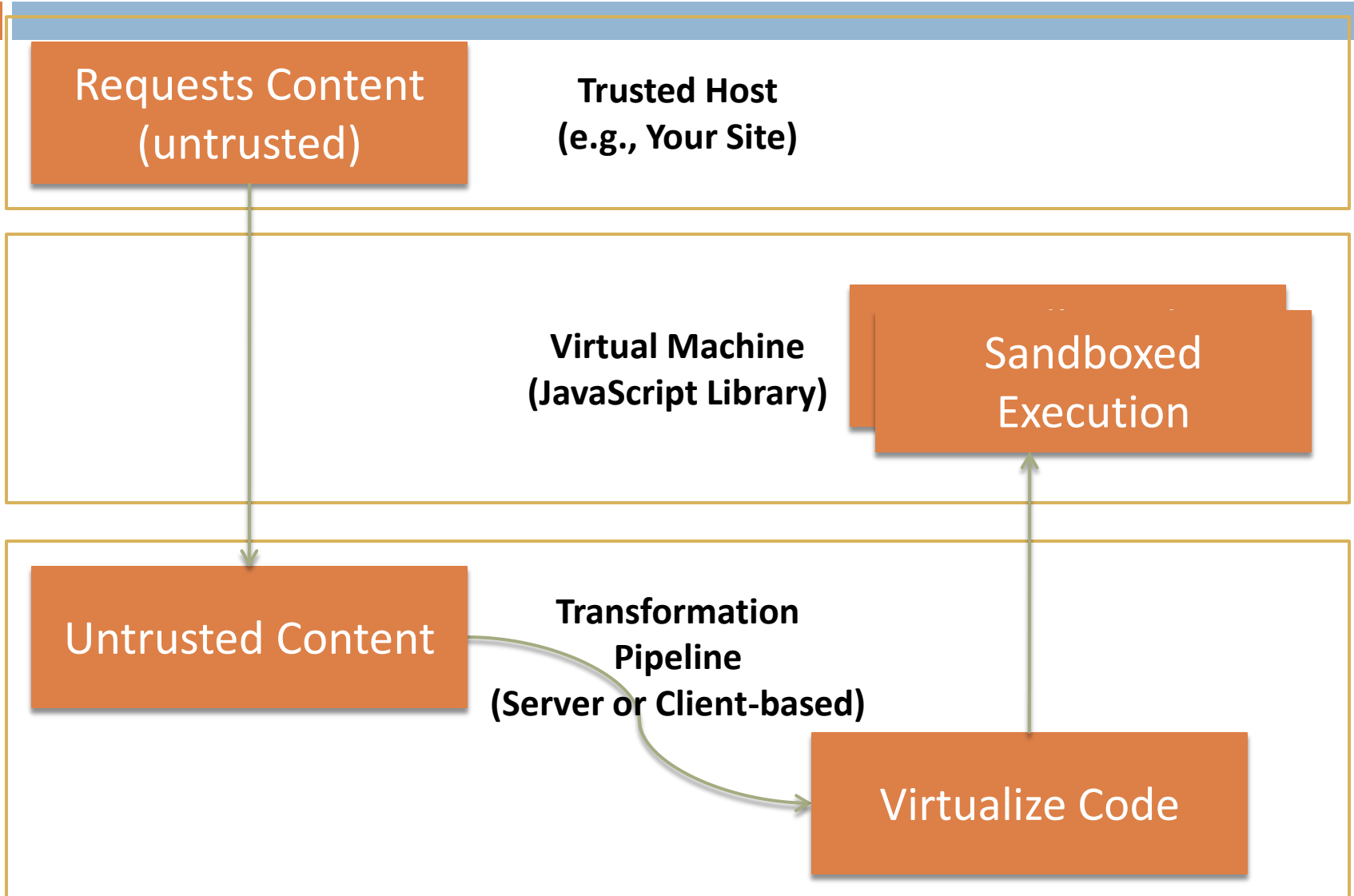
Sandboxing through Source-level Rewriting

32

- Browser offers iFRAMEs as an isolation mechanism
 - ▣ Every iFRAME has (an isolated) global object
 - ▣ SOP prevents arbitrary cross-frame communication
- Not bad, but sometimes too restrictive
 - ▣ Framed applications are confined to pre-determined screen regions
 - ▣ Interactions with other iFrames require message passing using the `postMessage` API


Google Caja and Microsoft WebSandbox

Web Sandbox: The Big Picture



Web Sandboxed Gadget

Web Sandboxed Gadget

 Clock Sample

Clock Sample

Tue Oct 11 2011 15:04:26 GMT-0400 (Eastern Daylight Time)

```
<html>
  <head>
    <title>Clock Sample</title>
    <base href="http://www.websandbox.org/" />
    <link href="Images/favicon.ico" rel="icon" />
    <style>
      .sampleTitle
        {font-family: Segoe UI, Tahoma; font-size: 11pt; font-weight:
bold; color: #07519A; }
      .clockSample { height: 130px; border: solid 1px lightgrey;
background: white; background-repeat: repeat-x; background-
position: left top; padding: 10px; overflow-y: auto;}
    </style>
  </head>
  <body>
    <div id="sample" class="clockSample">
      <div class="sampleTitle">Clock Sample</div>
      <br />
      <span id="currentTime"></span>
      <script type="text/javascript">
        window.setInterval(function() {
          document.getElementById("currentTime").
            innerText = new Date();
          }, 999)
      </script>
    </div>
  </body>
</html>
```

Web Sandbox Rewriting

```
var settings = { css : {".sampleTitle" :
{"font-family":"Segoe UI,Tahoma", ... }};

var headerJavaScript =
function(a)
{
    var b = a.gw(this),
        c = a.g,
        d = a.i,
        e = c(b,"document");
    d(e,"initializeHTML",
[[{"body":{"c":[, "
",{"div":{"a":{"id":"sample","class":"clockS
ample"},"
c":[, "
",{"div":{"a":{"class":"sampleTitle"},"c":[,
"Clock Sample"]}}," ",{"br":{}}},"
",{"span":{"a":{"id":"currentTime"}}},"
",{"script":{"__src__":"c20","a":{"type":"te
xt/javascript"}}}," " ]}}," " ]}}]]
};
```

```
var metadata =
{"author":"","description":"","imagepath":"","title":"Cloc
k Sample",...,
"scripts" : {"c20" :
function(a)
{
    var b = a.gw(this),
        c = a.g,
        d = a.s,
        e = a.i,
        f = a.n,
        g = a.f,
        h = c(b,"document");
    e(b,"setInterval",[g(function()
{
    d(e(h,"getElementById",[ "currentTime"]), "innerText", f(c(b,
"Date"), []))
    }), 999])
    }]);

    $Sandbox.registerCode(headerJavaScript, "2", settings,
metadata);

    var SandboxInstance = new
    $Sandbox(document.getElementById('g_2_0_inst'),
    $Policy.Canvas, "2");

    SandboxInstance.initialize();
```

Translation Continued

36

```
var metadata =
{"author":"","description":"","imagepath":"","title":"Your Gadget's
Title","preferredheight":0,"preferredwidth":0,"location":"","icon":"","
base":{"href":"","target":""},"scripts" : {"c00" :
function(a)
{
    var b = a.gw(this),
        c = a.g
}}};

$Sandbox.registerCode(headerJavaScript, "0", settings, metadata);

var SandboxInstance = new
$Sandbox(document.getElementById('g_0_0_inst'), $Policy.Canvas, "0");

SandboxInstance.initialize();
```

W3C CSP: Content Security Policy

37

- **Example 1:** A server wants all content to come from its own domain:
 - `X-Content-Security-Policy: default-src 'self'`

- **Example 2:** An auction site wants to allow images from anywhere, plugin content from a list of trusted media providers including a content distribution network, and scripts only from a server under its control hosting sanitized ECMAScript:
 - `X-Content-Security-Policy: default-src 'self'; img-src *;`
 - `object-src media1.example.com media2.example.com *.cdn.example.com;`
 - `script-src trustedscripts.example.com`

- **Example 3:** A site operations group wants to globally deny all third-party scripts in the site, and a particular project team wants to also disallow third-party media in their section of the site. Site operations sends the first header while the project team sends the second header, and the user-agent takes the intersection of the two headers to form the complete interpreted policy:
 - `X-Content-Security-Policy: default-src *; script-src 'self'`
 - `X-Content-Security-Policy: default-src *; script-src 'self'; media-src 'self'`

- **Example 4:** Online banking site wants to ensure that all of the content in its pages is loaded over TLS to prevent attackers from eavesdropping on insecure content requests:
 - `X-Content-Security-Policy: default-src https://*:443`

HTML5 Sandbox

38

```
<iframe src="untrusted.html"  
    sandbox="allow-scripts allow-forms">  
</iframe>
```

- ▣ allow-scripts
- ▣ allow-forms
- ▣ allow-same-origin
- ▣ allow-top-navigation
- ▣ ms-allow-popups

HTML5 Sandbox in Action

39

Information Disclosure Phishing Page Redirection Controlling Popups

Phishing
Imagine you mistal prevent th


Try "loggi
 Enable

WoodG
by Clippy

SEATTLE-Lc
pharetra ve
cursus et ac
augue non i
gravida ero
neque. Pha
blandit ven
odio accum
enim luctus
purus ac ma
porttitor alic

Ut purus odi
Donec matt
tellus quis n
Phasellus ni
nec tempor
odio ultricie
purus ut ul
ultrices blan
aliquet in, n

Page Redirection



fering what lo
protect you us

The fake ad above is attempting to redirect you to a fake malicious site (without you even clicking it). HTML5 Sandbox is preventing it from doing so.

Try disabling sandbox to see how the ad could maliciously redirect you.

Disable Sandbox

Sandboxed Page Redirection: **BLOCKED**

[Oakland S&P 2010]

ConScript

Specifying and Enforcing Fine-Grained Security Policies
for JavaScript in the Browser

Leo Meyerovich
UC Berkeley

Benjamin Livshits
Microsoft Research



Microsoft®
Research

Only Allow `eval` of JSON

- Idea for a policy:
 - Parse input strings instead of running them
 - Use ConScript to *advise* `eval` calls
- AspectJ advice for Java

```
void around call Window::eval (String s) { ... }
```

- How to do advice in JavaScript?
 - No classes to speak of

ConScript approach

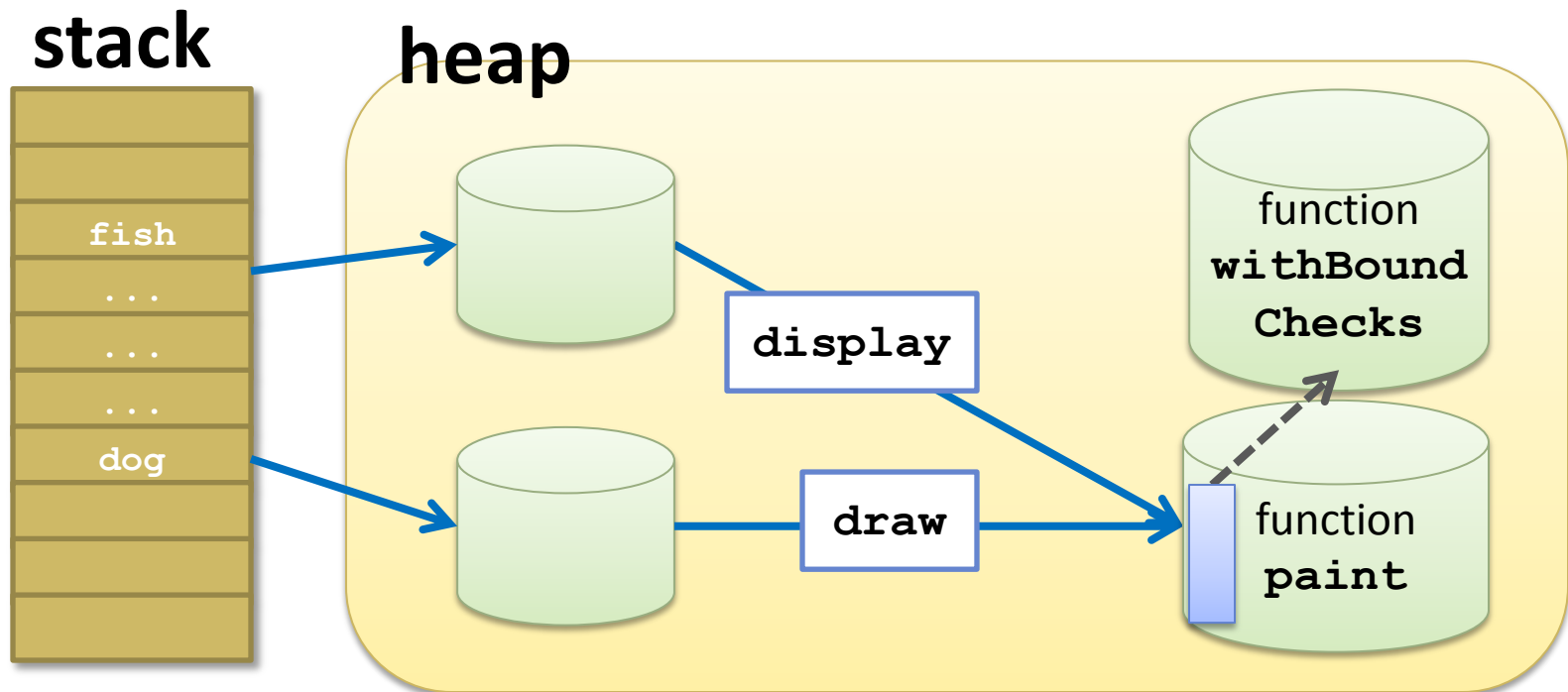
- Deep advice for complete mediation
- Implemented within the browser for efficiency and reliability

Example of Applying Advice in ConScript

```
1. <SCRIPT SRC="facebook.js" POLICY="
2.     var substr = String.prototype.substring;
3.     var parse = JSON.parse;
4.     around(window.eval,
5.         function(oldEval, str) {
6.             var str2 = uCall(str, substr, 1,
7.                 str.length - 1);
8.             var res = parse(str2);
9.             if (res) return res;
10.            else throw "eval only for JSON";
11.        } );">
```

Advising JavaScript Functions in IE8

```
around (paint, withBoundChecks) ;  
dog.draw () ;  
fish.display () ;
```



Policies are Easy to Get Wrong

toString redefinition!

```
1.
2. around (window, stub)
3.   function (post, target)
4.     if (!okOrigin[target])
5.       throw 'err';
6.     else {
7.       return stub.call(this, msg, target);
8.     }
9.
```

Function.prototype poisoning!

Object.prototype poisoning!

Paper presents

enforce public vs. private

manifest of script URLs

HTTP-only cookies

resource blacklists

no pop-ups

```
around(document.createElement,  
function (c : K, tag : U) {  
  var elt : U = uCall(document, c, tag);  
  if (elt.nodeName == "IFRAME") throw 'err';  
  else return elt; });
```

no eval

<noscript>

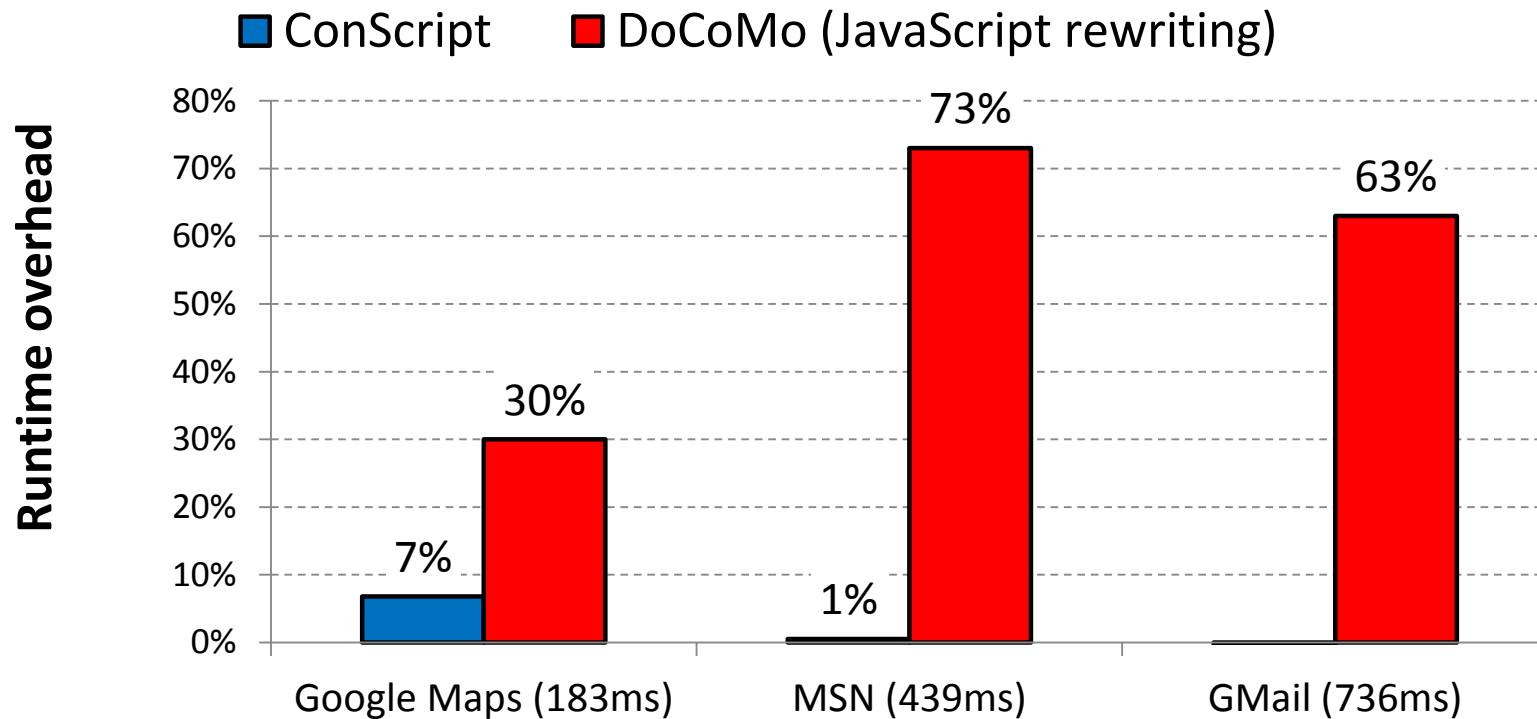
no foreign links

script whitelist

no dynamic IFRAME creation

DoCoMo Policy Enforcement Overhead

47



H. Kikuchi, D. Yu, A. Chander, H. Inamura, and I. Serikov,
"JavaScript instrumentation in practice," 2008

Summary

48

- Background on SOP
- Language restrictions
 - ▣ AdSafe
 - FBJS
- Extensive rewriting
 - ▣ Caja
 - ▣ WebSandbox
- Better runtimes
 - ▣ CSP
 - ▣ HTML5 Sandbox
- Tradeoffs of different containment strategies and going forward