

The Same Origin Policy

Tamara Rezk
joint work with
Zhengqin Luo



December 17th , 2014

Web Client Mashup

- Client-side web application
- Integrating third-party gadgets
- Great way for code reuse!

For Rent [For Sale](#) [Rooms](#) [Sublets](#)

City: Price: [Show Filters](#) ^{New} [Refresh](#) [Link](#)

Powered by [craigslist](#) and [Google Maps](#)
(this site is in no way affiliated with craigslist or Google)
[About / Feedback](#)

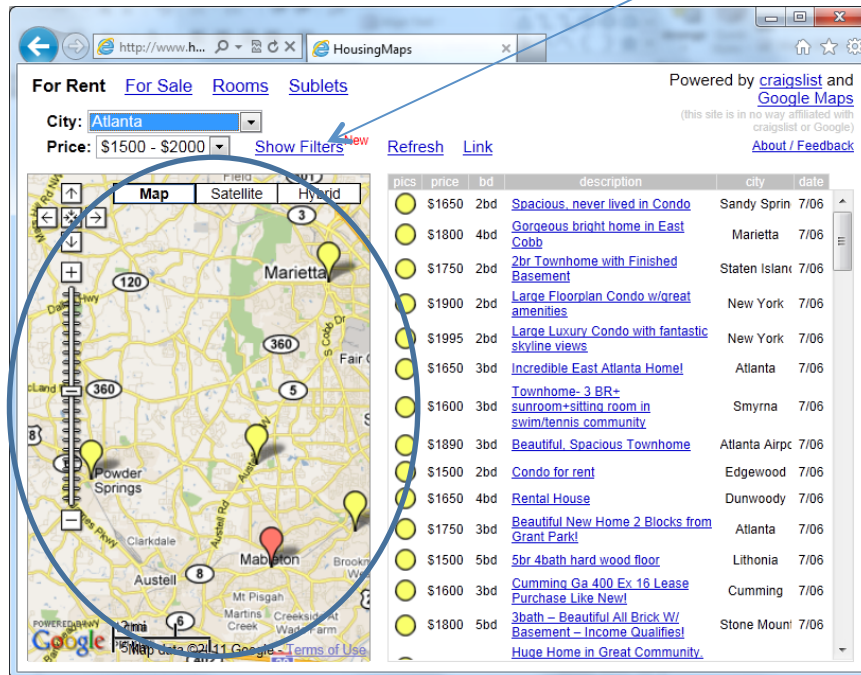
pics	price	bd	description	city	date
	\$1650	2bd	Spacious, never lived in Condo	Sandy Sprin	7/06
	\$1800	4bd	Gorgeous bright home in East Cobb	Marietta	7/06
	\$1750	2bd	2br Townhome with Finished Basement	Staten Islan	7/06
	\$1900	2bd	Large Floorplan Condo w/great amenities	New York	7/06
	\$1995	2bd	Large Luxury Condo with fantastic skyline views	New York	7/06
	\$1650	3bd	Incredible East Atlanta Home!	Atlanta	7/06
	\$1600	3bd	Townhome- 3 BR+ sunroom+sitting room in swim/tennis community	Smyrna	7/06
	\$1890	3bd	Beautiful, Spacious Townhome	Atlanta Airpc	7/06
	\$1500	2bd	Condo for rent	Edgewood	7/06
	\$1650	4bd	Rental House	Dunwoody	7/06
	\$1750	3bd	Beautiful New Home 2 Blocks from Grant Park!	Atlanta	7/06
	\$1500	5bd	5br 4bath hard wood floor	Lithonia	7/06
	\$1600	3bd	Cumming Ga 400 Ex 16 Lease Purchase Like New!	Cumming	7/06
	\$1800	5bd	3bath - Beautiful All Brick W/ Basement - Income Qualifies!	Stone Moun	7/06
			Huge Home in Great Community.		

Google Maps Gadget

Integrator's Housing Data

Different kind of Mashups

gadget



The integrator uses the gadget as a library



The gadget reads state from the integrator

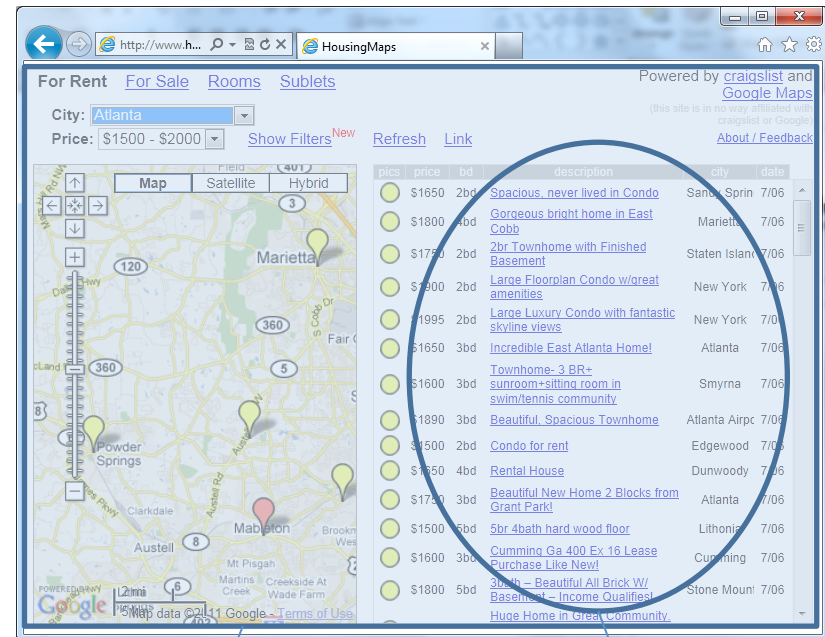
Programming model

- Mashups written in HTML + Javascript;
- Two ways to include external gadgets:
 - Using script tag
 - Using frame tag

Script tag - oversharing

```
<script src="http://maps.google.com/maps/api/js?sensor=true"></script>
```

- Gadget has integrator privileges
- Full exposure JS execution environment



Google Maps Gadget

Integrator's
Housing Data

Script tag – security violations

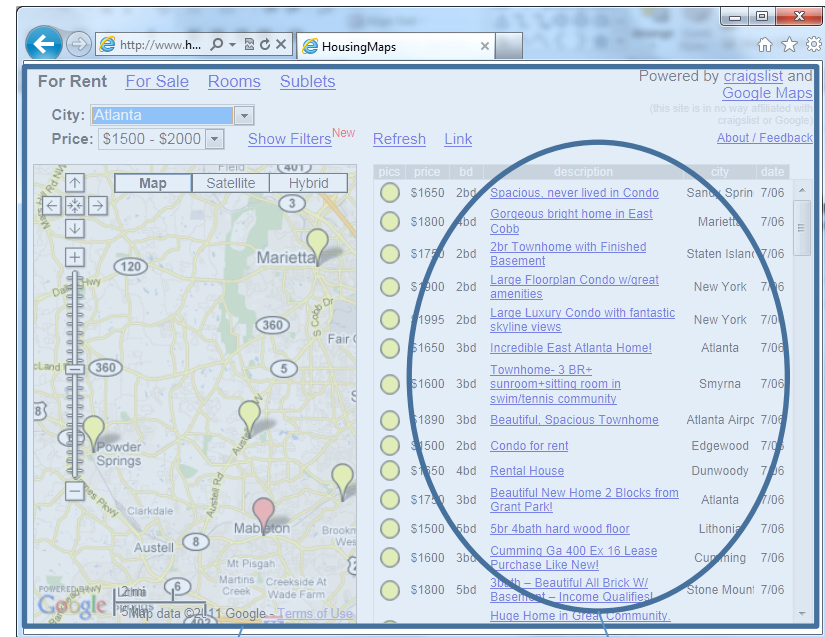
```
<script src="http://maps.google.com/maps/api/js?sensor=true"></script>
```

- Confidentiality violation

```
var steal =  
  window["integratorSecret"]
```

- Integrity violation

```
window["price"] = newPrice
```



Google Maps Gadget

Integrator's Housing Data

Script tag – integrity violation

```
<html>
<head> <script src=http://untrusted.com/untrustedAD.js></script> </
head>
<body>
<script>
  var fac= function(x) {
    if (x <= 1) {
      return 1;
    }
    return x*fac(x-1);
  }
  r = fac(3);
  s = "alert("+r+")"
  setTimeout(s, 100)
</script>
</body></html>
```

Script tag - integrity violation

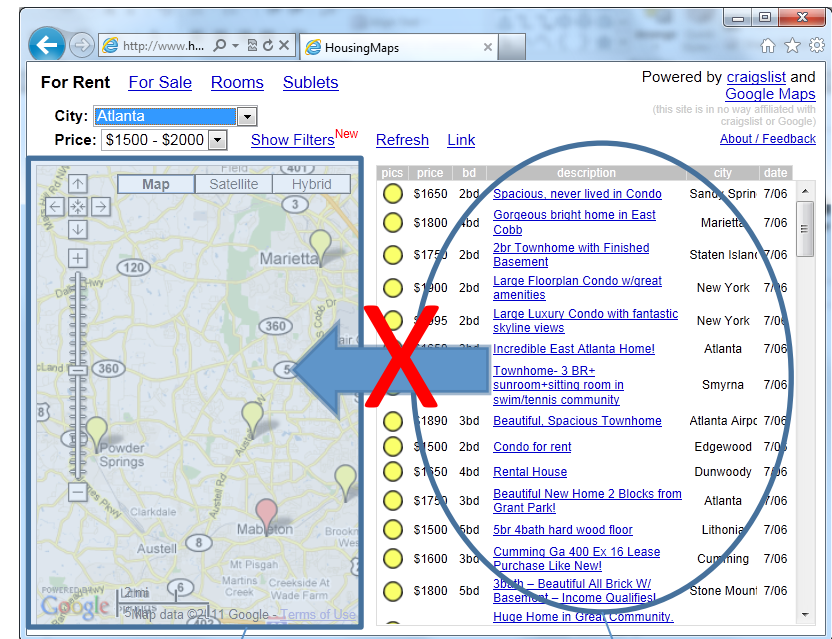
Content of <http://untrusted.com/untrustedAD.js>

setTimeout = function() { EVIL CODE HERE }

Embedded frame – undersharing

```
<iframe src="http://mapservice.com/maps.html"></iframe>
```

- Gadget has no privileges
- Isolation of JS execution environment between **integrator** and **gadget**
- Same Origin Policy



Google Maps Gadget

Integrator's
Housing Data

Same Origin Policy (SOP)

- Other frames cannot access resources from other origins

Example:

`<!-- This is allowed -->`

`<iframe src="sameDomainPage.html"> </iframe>`

`alert(frames[0].contentDocument.body); //works fine`

`<!-- This is **NOT** allowed -->`

`<iframe src="http://google.com"> </iframe>`

`alert(frames[0].contentDocument.body); //throws error`

What happens with the global object?

Embedded frame – undersharing

```
<iframe src="http://mapservice.com/maps.html"></iframe>
```

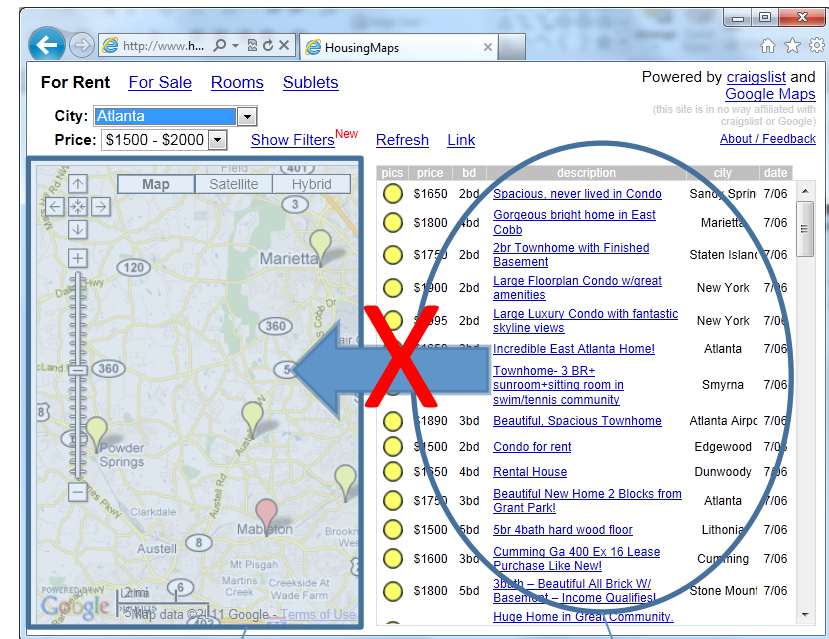
Isolation of JS execution:

- Confidentiality violation

*var steal =
window["IntegratorSecret"]*

- Integrity violation

window["price"] = newPrice



Google Maps Gadget

Integrator's
Housing Data

Script versus Frame Tag

- Script tag:
 - exposes the integrator's JS environment
 - unlimited communication
 - not secure
- Frame tag:
 - isolates the environment of gadget and integrator
 - communication is limited
 - more secure than script

HTML 5: inter-frame communication

- Cross-origin client side communications
- Postmessage channel between frames



Example of PostMessage

'http://destination.com

```
window.addEventListener('message', function(event) {  
    if(event.origin !== 'http://originExpected.com')  
    {return;}  
    {console.log('received response: ', event.data);  
  
event.source.postMessage('received', event.origin)});
```

frame at 'http://originExpected.com

```
var domain = 'http://destination.com';  
var iframe =  
document.getElementById('myIFrame').contentWindow;  
var message = 'Hello!';  
iframe.postMessage(message, domain);
```

Security considerations postmessage

- Do not configure target origin (recipient) to “*”
- Sensitive data can be leaked to unknown widgets
- Always check for sender’s origin
- Always validate data before use
- Do not consume data directly with `eval()` or `innerHTML`

Extended JS Syntax for Mashups

$M ::= \langle \text{html} \rangle F J \langle / \text{html} \rangle$

HTML page

$F ::= \langle \text{iframe src} = u \rangle \langle / \text{iframe} \rangle \mid E$

a frame or empty

$J ::= \langle \text{script } l \rangle s \langle / \text{script} \rangle J \mid E$

sequence of scripts

$s ::= e \mid \dots$

Javascript programs

$l ::=$

security labels

JS Semantics + Frames + Postmessage

DINIT

$$\frac{P = \langle \text{html} \rangle FJ \langle / \text{html} \rangle}{(\square, \varepsilon, \text{null}, P, Q_{\text{init}}) \rightarrow (\square, h_{\text{in}}, \# \text{global}_i, FJ, Q_{\text{init}})}$$

DSRIPT

$$\frac{\text{VD}(h, \ell, s, \Delta) = h'}{(\square, h, \ell, \langle \text{script} \Delta \rangle s \langle / \text{script} \rangle, Q) \rightarrow (\Delta, h', \ell, s, Q)}$$

DFRAME

$$\frac{\text{Web}(u) = J \quad h' = h \oplus h_{\text{in}}^f}{(\square, h, \ell, \langle \text{iframe src} = u \rangle \langle / \text{iframe} \rangle, Q) \rightarrow (\square, h', \# \text{global}_f, \langle \text{iframe} \rangle J \langle / \text{iframe} \rangle, Q)}$$

DPOSTMSGI

$$\frac{\ell' = \# \text{postmessage}_i \quad Q = \langle \ell_i, mq_i \rangle \parallel \langle \ell_f, mq_f \rangle}{(\square, h, \ell, \ell'(m), Q) \rightarrow (\square, h, \ell, \text{undefined}, \langle \ell_i, mq_i \rangle \parallel \langle \ell_f, mq_f + m \rangle)}$$

DADDLISTENERI

$$\frac{\ell' = \# \text{addlistener}_i \quad Q = \langle \ell_i, mq_i \rangle \parallel \langle \ell_f, mq_f \rangle}{(\square, h, \ell, \ell'(\ell_0), Q) \rightarrow (\square, h, \ell, \text{undefined}, \langle \ell_0, mq_i \rangle \parallel \langle \ell_f, mq_f \rangle)}$$

DCALLBACKI

$$\frac{\ell_f \neq \text{null} \quad Q = \langle \ell_i, mq_i \rangle \parallel \langle \ell_f, m + mq_f \rangle}{(\square, h, \ell, \varepsilon, Q) \rightarrow (\square, h, \# \text{global}_f, \ell_f(m), \langle \ell_i, mq_i \rangle \parallel \langle \ell_f, mq_f \rangle)}$$

DASGN-NEW-PROPERTY

$$\frac{m \notin h(\ell_1) \quad h(\ell_1.m_{\{\square\}} = v) = h_1}{(\square, h, \ell, \ell_1[m] = v, Q) \rightarrow (\square, h_1, \ell, v, Q)}$$

SOP Property

$$(\spadesuit, \varepsilon, \text{null}, \dot{M}_c(P_i, P_g, \mathcal{V}), Q_{\text{init}}) \rightarrow^* (\heartsuit, h, \ell, s, Q)$$

if

$$(\heartsuit, h, \ell, s, Q) \rightarrow (\heartsuit, h', \ell', s', Q')$$

then we have

- 1) (*integrity.*) $h|_{\spadesuit} = h'|_{\spadesuit}$;
 - 2) (*confidentiality.*) For any h_0 such that $h_0|_{\heartsuit} = h|_{\heartsuit}$, we have $(\heartsuit, h_0, \ell, s, Q) \rightarrow (\heartsuit, h'_0, \ell', s', Q')$, and $h'_0|_{\heartsuit} = h'|_{\heartsuit}$.
-

Conclusions

- Characterization of the Same Origin Policy as a property: now comparison to e.g. noninterference is possible
- Missing: cookies, inconsistencies (Oakland'11), navigation policies (Usenix Security'08), ...
- Extension in JSCert?