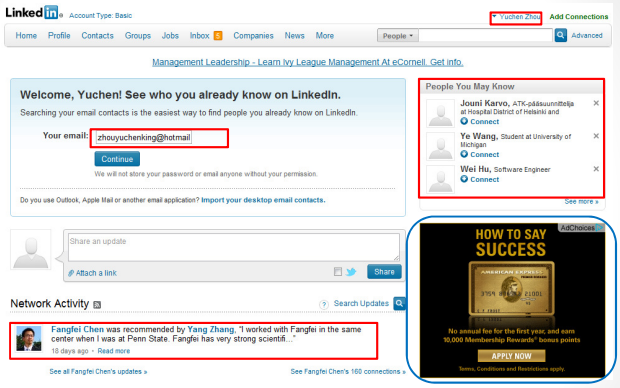
Protecting Private Web Content from Embedded Scripts

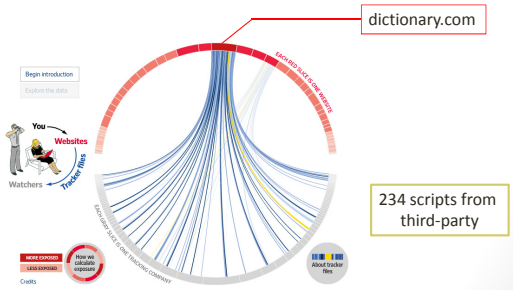
Yuchen Zhou
David Evans

http://www.cs.virginia.edu/DOMinator



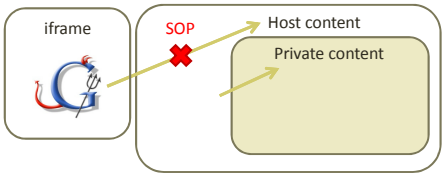
Third-Party Scripts

49.95% responsive top 1 million sites use Google Analytics as of Aug 2010. [Wikipedia]



Washington Street Journal, What They Know?

All or nothing trust model



Dilemma of current web technologies

Threat Model

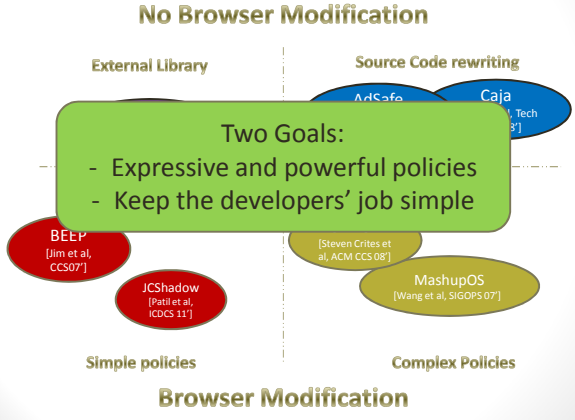
Content provider embeds third-party scripts directly in its webpages.

Adversary controls those scripts and may use any means to get confidential information.

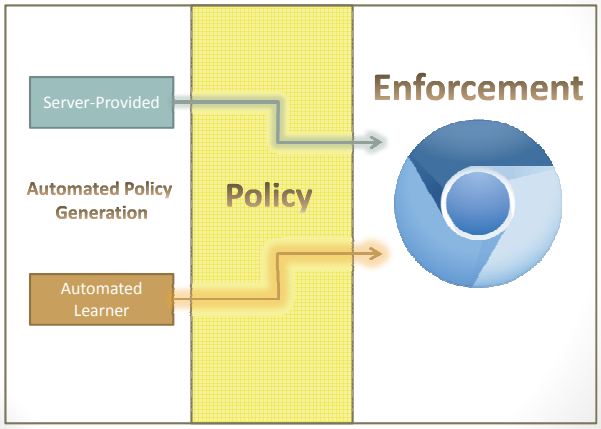
- DOM APIs
- JavaScript variables/functions

High-level goal:
- Add policies to host pages to restrict third-party scripts' privilege and prevent them from stealing private information.

Related Work



Overview



Isolation Policies

Adversary techniques	
JavaScript	DOM APIs
Access host script variables	document.getElementById.innerHTML
Call host script functions	document.cookie

JavaScript Execution Isolation

Isolation policy: 'worldID' attribute:

$< script\ worldID = "string" >$

- Scripts with the same worldID execute in the same world (context).
- Scripts without worldID is most privileged (host script).

One-way access policy: 'sharedLibID' and 'useLibID' attribute:

$< script\ sharedLibId = "string" >$

$< script\ useLibId = "string" >$

- Scripts can share their global objects by specifying 'sharedLibId' attribute.
- Scripts can use resources in a different world by specifying 'useLibId' attribute.

Isolation Policies

Adversary techniques	
JavaScript	DOM APIs
Access host script variables	document.getElementById.innerHTML
Call host script functions	document.cookie

DOM APIs Access Control

DOM node access control list:

$< div\ RACL = "worldID1; worldID2, etc.." >$

$< div\ WACL = "worldID1; worldID2, etc.." >$

Script with worldID that does not appear in a DOM node's access control list cannot perform corresponding actions on that node.

- For RACL: privileged world may read the content/attribute of this node
- For WACL: privileged world may modify the content/attribute of this node.

Annotated Page Example

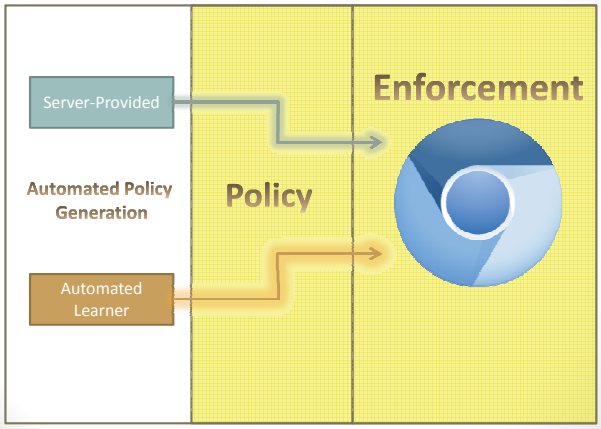
```
<html>
<body>
<div id='public'>
Hello, world!
</div>
<div id='secret'>
This is a secret
</div>
<script src='third-party.js'>
</script>
</body>
</html>
```

Original HTML

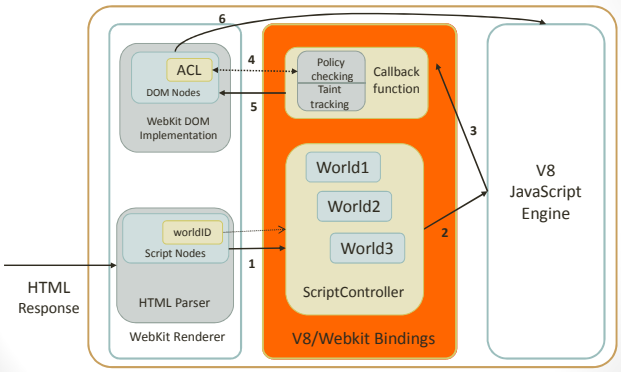
```
<html>
<body>
<div id='public' RACL='3rd-p' WACL='3rd-p'>
Hello, world!
</div>
<div id='secret'>
This is a secret
</div>
<script src='third-party.js' worldID='3rd-p'>
</script>
</body>
</html>
```

Annotated HTML

Overview

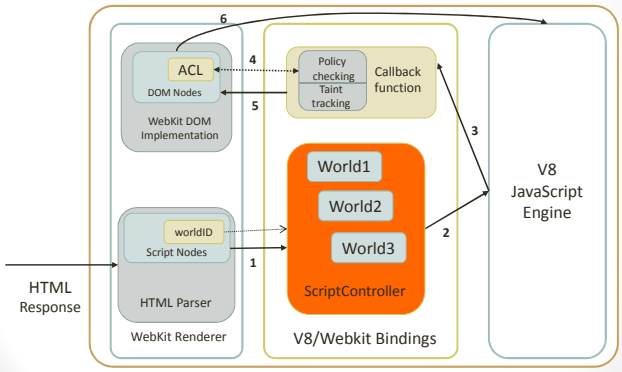


Enforcement Overview



Chromium Architecture

Enforcement Overview

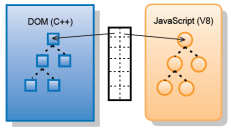


Chromium Architecture

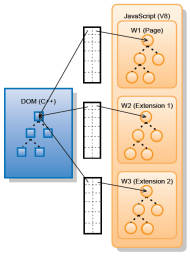
Isolated World

Our Goal: Apply this to page script isolation.

Adam Barth et al.[NDSS 2010]
- Goal: separate extension execution contexts
- Already built into Chromium's trunk code



DOM-JS 1-to-1 mapping

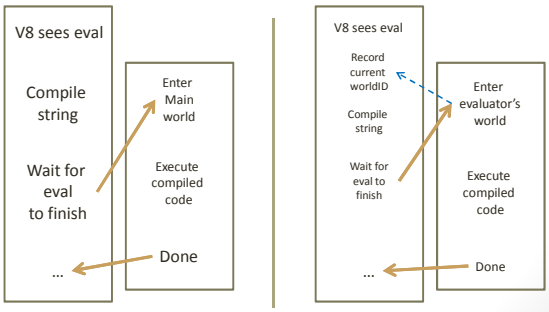


DOM-JS 1-to-n mapping

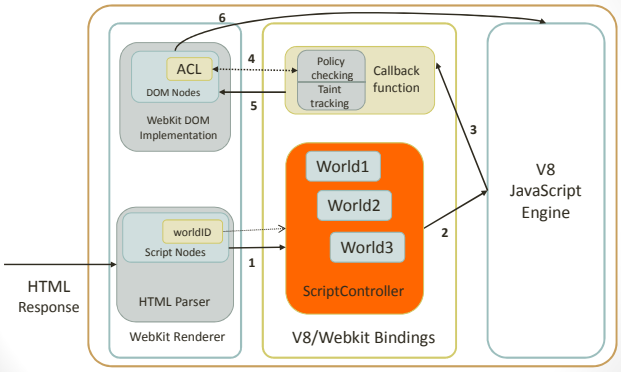
Dynamic Scripts

An eval() example:

```
<div id="secret">A secret</div>
<script worldID="untrusted1">
...
eval(alert(document.getElementById('secret').innerHTML));
...
</script>
```

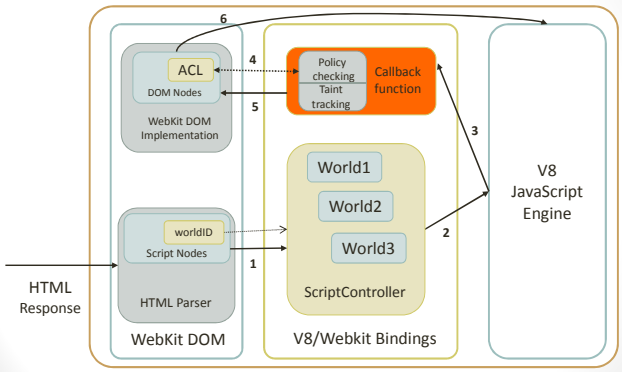


Enforcement Overview



Chromium Architecture

Enforcement Overview



Chromium Architecture

Node ACL Enforcement

Subject	Policy	Semantic
DOM node		Worlds that may access this
DOM node		Worlds that may modify this

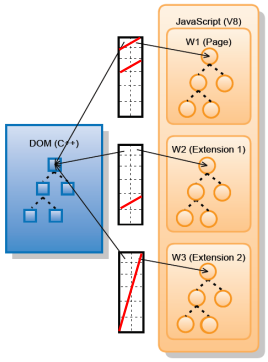
RACL enforcement:

- Hide handle of node;

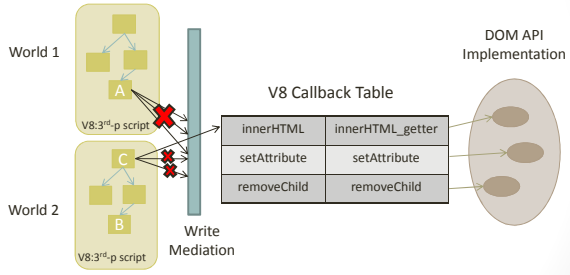
WACL enforcement:

- Add mediation to corresponding DOM APIs.

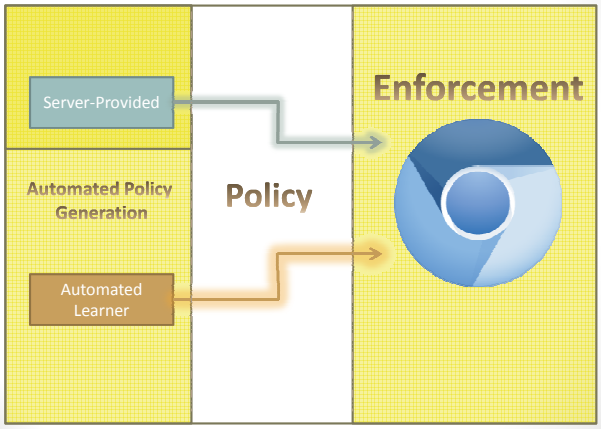
Hiding parts of DOM



DOM Element ACL policy



Overview



Annotated Page Example

```
<html>
<body>
<div id='public'>
Hello, world!
</div>
<div id='secret'>
This is a secret
</div>
<script src='third-party.js'>
</script>
</body>
</html>
```

Original HTML

```
<html>
<body>
<div id='public' RACL='3rd-p' WACL='3rd-p'>
Hello, world!
</div>
<div id='secret'>
This is a secret
</div>
<script src='third-party.js' worldID='3rd-p'>
</script>
</body>
</html>
```

Annotated HTML

Server-Provided Policy

Developers Manual effort:
- Requires significant effort
- Easy to forget
- Almost impossible for high-profile/dynamic sites

Web Framework Assisted:
- Declare policy once, automate the rest

GuardRails Integration

GuardRails is an extension for Ruby on Rails framework that makes it easy for developers to define security policies by writing annotations.

GuardRails provides a character-level precision taint tracking system to trace sensitive data flows.

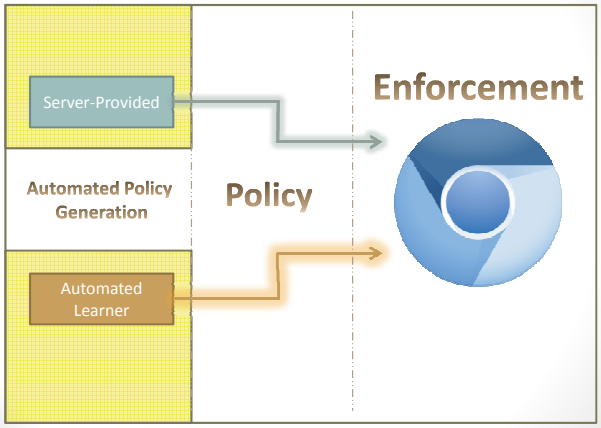
```
# @ :read_worlds, :name, ["World1"]
class Cart...
```

GUARDRAILS

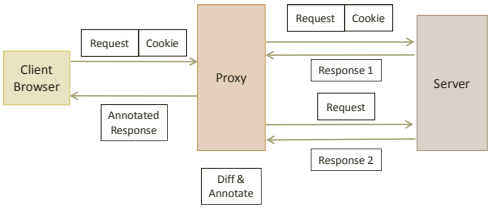
```
Name: <span RACL="World1">SomeProduct</span><br/>
Description: <span RACL="World1, World2" WACL="World1,
World2">Accessories for <b RACL="World1">Some Other
Product</b></span>
```

Jonathan Burket, Patrick Mutchler, Michael Weaver, Muzzammil Zaveri, and David Evans. June 2011. GuardRails: A Data-Centric Web Application Security Framework. In *2nd USENIX Conference on Web Application Development* (WebApps'11).

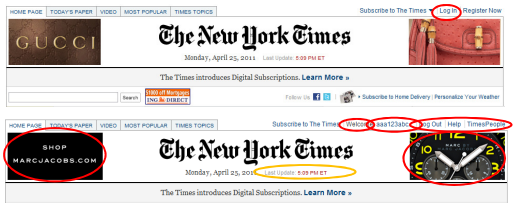
Overview



Policy learner workflow



Limitations of Automated Learning

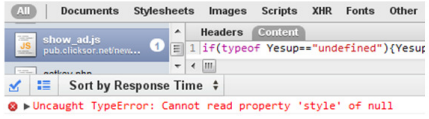


Side effect of sending two requests:
- Double traffic, significant higher latency.

- Extra requests may cause undesired server state changes.

Experiments

Security

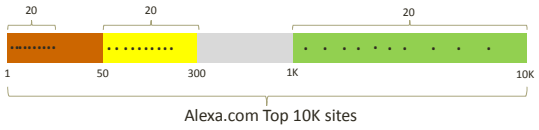


Compatibility

Policy inference

Compatibility Experiments

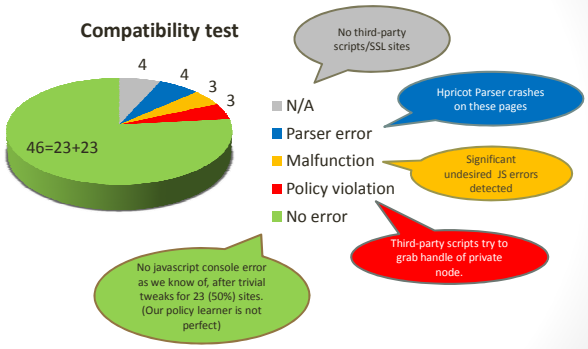
- Isolating the execution context of third-party scripts could possibly cause problems in real-world websites.
- Tried the modified browser on 60 sites.



Alexa.com Top 10K sites

- We use our automatic policy learner to derive the policies for each site.
  - We manually corrected third-party script identification errors generated by policy learner.

Compatibility Results



Policy Learner Result

Alexa Ranking	1-20	50-300	1000+
Sample size	13	11	18
% of public nodes before login	71.6%	95.7%	98%
% of public nodes after login	52.6%	78.4%	83%
% of nodes switched from public to private after login	26.6%	18%	15.3%
# of third-party scripts embedded	0.84	2.61	2.2

Conclusion

Current web technologies don't match site trust models.

We have made one step toward easier deployment of web security mechanism.

- Automatic policy learning is hard without server side assistance.

- We hope the idea of integrating server framework extensions with client-side protection could be widely adopted.

UNIVERSITY of VIRGINIA

Thank you!
Questions?

UNIVERSITY of VIRGINIA

Backup slides

One-way object access

- Host is entirely separated with third-party scripts.

```
<script type="text/javascript">

 var _gaq = _gaq || [];
 _gaq.push(['_setAccount', 'UA-XXXXX-X']);
 _gaq.push(['_trackPageview']);
 _gaq.push(['_addTrans',
  '1234',        // order ID - required
  'Acme Clothing',  // affiliation or store name
  '11.99',        // total - required
  '1.29',         // tax
  '5',           // shipping
  'San Jose',      // city
  'California',    // state or province
  'USA'           // country
 ]);
</script>
```

One-way object access

- In Javascript, the window object is the super object of all other objects.
- Two new attribute for script tags:

$$< script\ sharedLibId =' string' >$$

$$< script\ useLibId =' string' >$$

- The window object of the scripts with *sharedLibId* is injected into main world as a custom object.
- Third-party scripts may use other party's script by adding *useLibId* string.

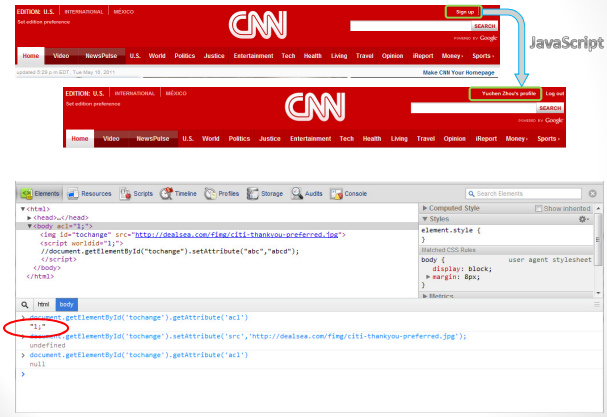
One-way object access

- Host is entirely separated with third-party scripts.

```
<script src="GA.js" worldID="Analytics"
sharedLibId="GA"></script>
<script type="text/javascript">

 var _gaq = GA._gaq || [];
 GA._gaq.push(['_setAccount', 'UA-XXXXX-X']);
 GA._gaq.push(['_trackPageview']);
 GA._gaq.push(['_addTrans',
  '1234',        // order ID - required
  'Acme Clothing',  // affiliation or store name
  '11.99',        // total - required
  '1.29',         // tax
  '5',           // shipping
  'San Jose',      // city
  'California',    // state or province
  'USA'           // country
 ]);
</script>
```

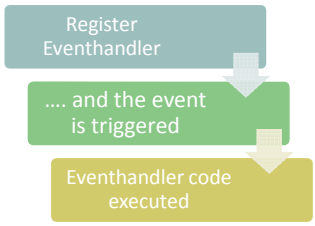
Node tainting



Immutable policy attributes

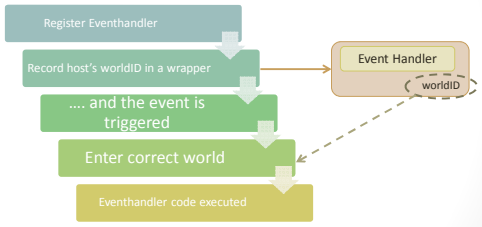
- All abovementioned policy attributes are made immutable to prevent malicious scripts from changing them.

Eventhandler



Normal event handling process

Eventhandler



Modified event handling process

Experiment Result - Security

- We constructed test-cases according to W3C standard for each defense mechanism we implemented, example test cases include:

Attack Type	Examples
Directly calling DOM API to get node handlers	document.getElementById(), nextSibling(), window.nodeID
Directly calling DOM API to modify nodes	nodeHandler.setAttribute(), innerHTML=, style=, nodeHandler.removeChild()
Probing host context for private variables/functions	referring to host variables, calling host functions, explicitly calling event handlers
Accessing special properties	document.cookie, open(), document.location

Third-party scripts identification

Host: Engadget.com

```
<script type="text/javascript" src="http://o.aolcdn.com/omniunih.js"></script>
<script type="text/javascript" src="http://o.aolcdn.com/js/mg2.js"></script>
<script async src="http://engadget2.disqus.com/count.js"></script>
<script type="text/javascript">…</script>
<script type="text/javascript">…</script>
<script src="http://www.engadget.com/traffic/?t=js&bv=&os=&tz=&lg=&rv=&rsv=&pw=%2F&cb=1403145037"
```

Definition: Any scripts that come from an external domain. Inline scripts are considered as trusted.

Policy Learner Result

- Identifying third-party scripts
  - False positives
    - Content Delivery Networks (CDN), mostly seen in top websites;
    - JavaScript libraries (jQuery, e.g.).
  - False negatives
    - Code snippets that assist a bigger script (Google Analytics, e.g.);
    - Copy third-party scripts to local server (rare cases).

  Added Heuristics:
    - Add whitelist for specific website's CDNs and common JS libraries;
    - Search for specific patterns in code snippet and mark them as third-party script.
- Private node identification

Policy Violations

- Washingtonpost.com (fb)
- Imtalk.org (addthis)
- Mysql.com(some script, grab the 'logout' button)

Example Results – Sites Ranked 50-100

Site	$Public_{Nologin}$		$Public_{Login}$		3rd-p scripts	Compatibility Issues	Trusted Domain
Twitpic	87/ 109	83%	150/ 193	77%	Crowdscience Scorecardresearch Quantserve Fmpub gstatic	Guest variable inline access	Googleapis.com twitter
washington post	1721/ 1722	99%	1783/ 1975	90%	Facebook	Guest variable inline access Policy violation	
Digg	934/ 967	97%	652/ 1000	65%	Diggstatic.com scorecardresearch		Facebook
Expedia	748/ 814	92%	746/ 814	92%	Intentmedia		
Vimeo	400/ 413	97%	202/ 431	47%	Google Analytics Quantserve		Vimeocdn.com
Statcounter	457/ 457	100%	137/ 190	72%	Doubleverify		
Bit.ly	102/ 105	97%	86/ 121	71%	Twitter Google Analytics	Guest variable inline access	
Indeed.com	126/ 128	98%	120/ 129	93%	Jobsearch Google Analytics scorescardresearch	Policy violation	
Yelp.com	782/ 794	98%	733/ 848	86%	Google Analytics		Yelpcdn.com

References

- [1] Google Analytics market share. http://metricmail.tumblr.com/post/904126172/google-analytics-market-share
- [2] What they know. http://blogs.wsj.com/wtk/
- [3] Adam Barth, Adrienne Porter Felt, Prateek Saxena, and Aaron Boodman. Protecting Browsers from Extension Vulnerabilities. In 17th Network and Distributed System Security Symposium, 2010.