

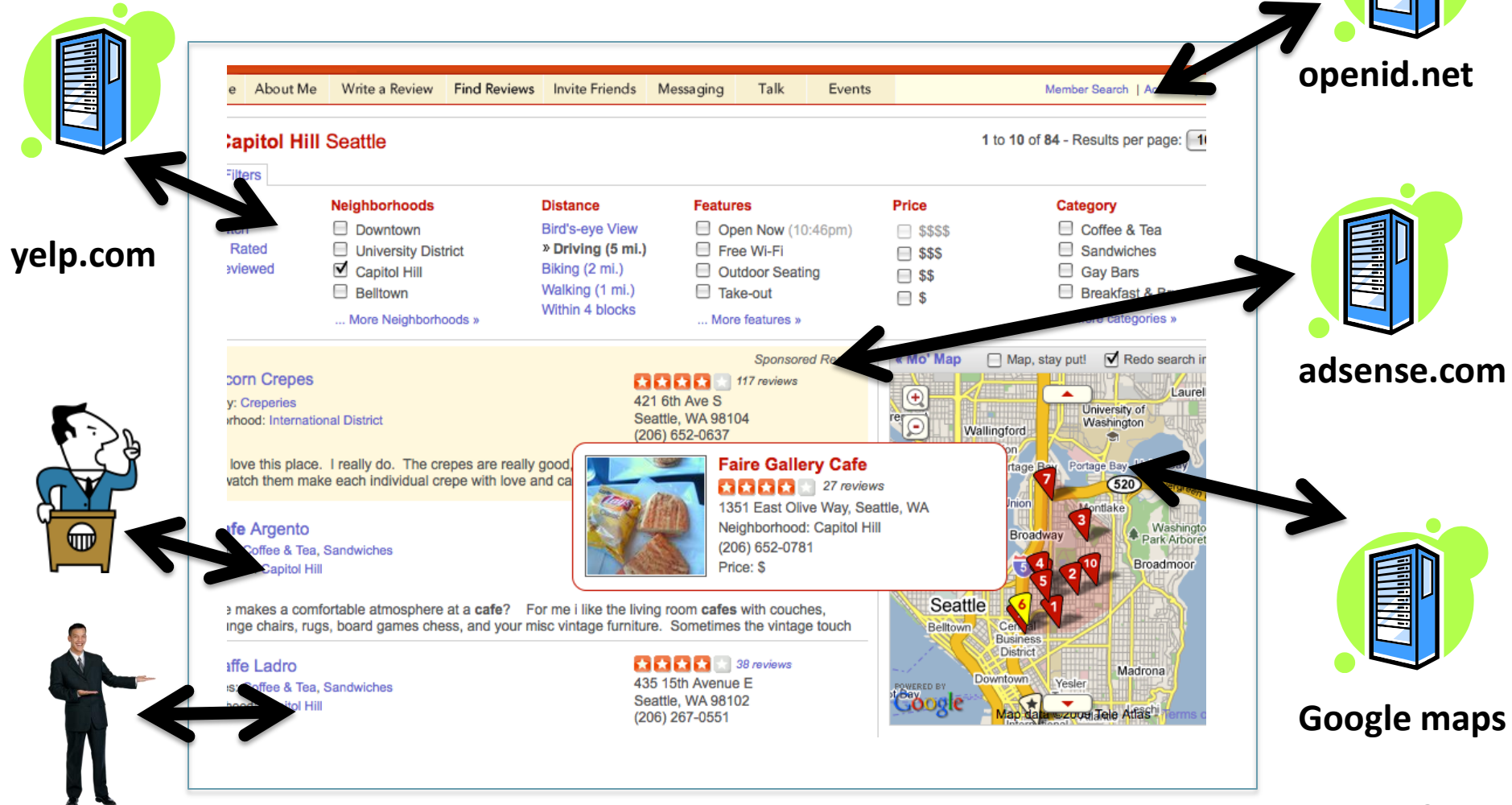
ConScript

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

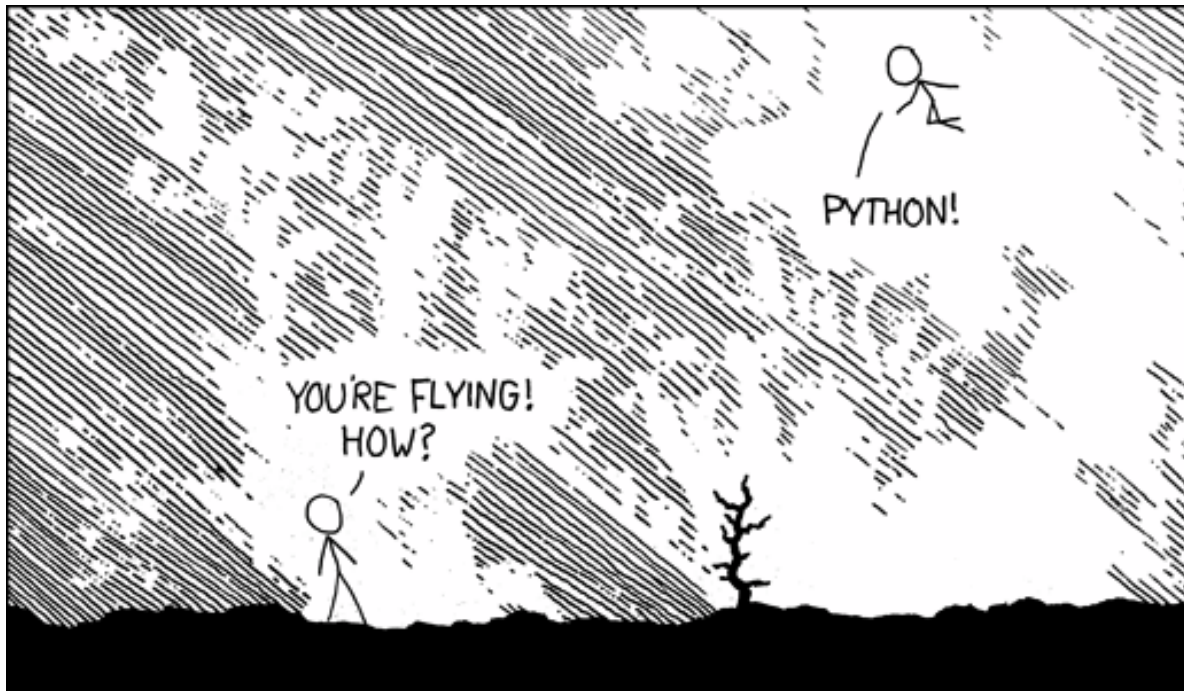
Leo Meyerovich
UC Berkeley

Benjamin Livshits
Microsoft Research

Web Programmability Platform



Rich Internet Applications are Dynamic



Yelp.com:

main.js

... jQuery.js

... adSense.js

... GoogleMaps.js

... OpenID_API.js

flexible runtime composition

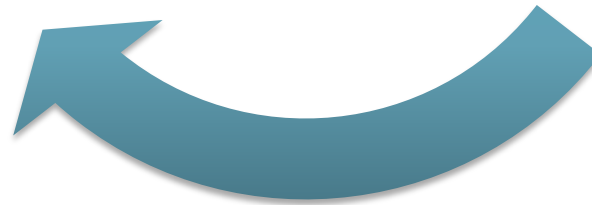
... but little control.

Towards Safe Programmability for the Web



Mash-ups

Can't trust other
people's code



Goals and Contributions

control loading
and use of scripts

- protect benign users
- by giving control to hosting site
- ConScript approach: aspects for security

express many
policies *safely*

- 17 hand-written policies
- correct policies are hard to write
- proposed type system to catch common attacks
- implemented 2 policy generators

browser support

- built into IE 8 JavaScript interpreter
- runtime and space overheads under 1% (vs. 30-550%)
- smaller trusted computing base (TCB)

approach

protect benign users by giving control
to the hosting site
: aspects for security

ConScript

- Approach
 - protect benign Web users
 - give control to the hosting site
- How
 - Browser-supported aspects for security

Contributions of ConScript

A case for aspects in browser

- protect benign users by giving control to hosting site
- ConScript approach: aspects for security
- built into IE 8 JavaScript interpreter

Correctness checking

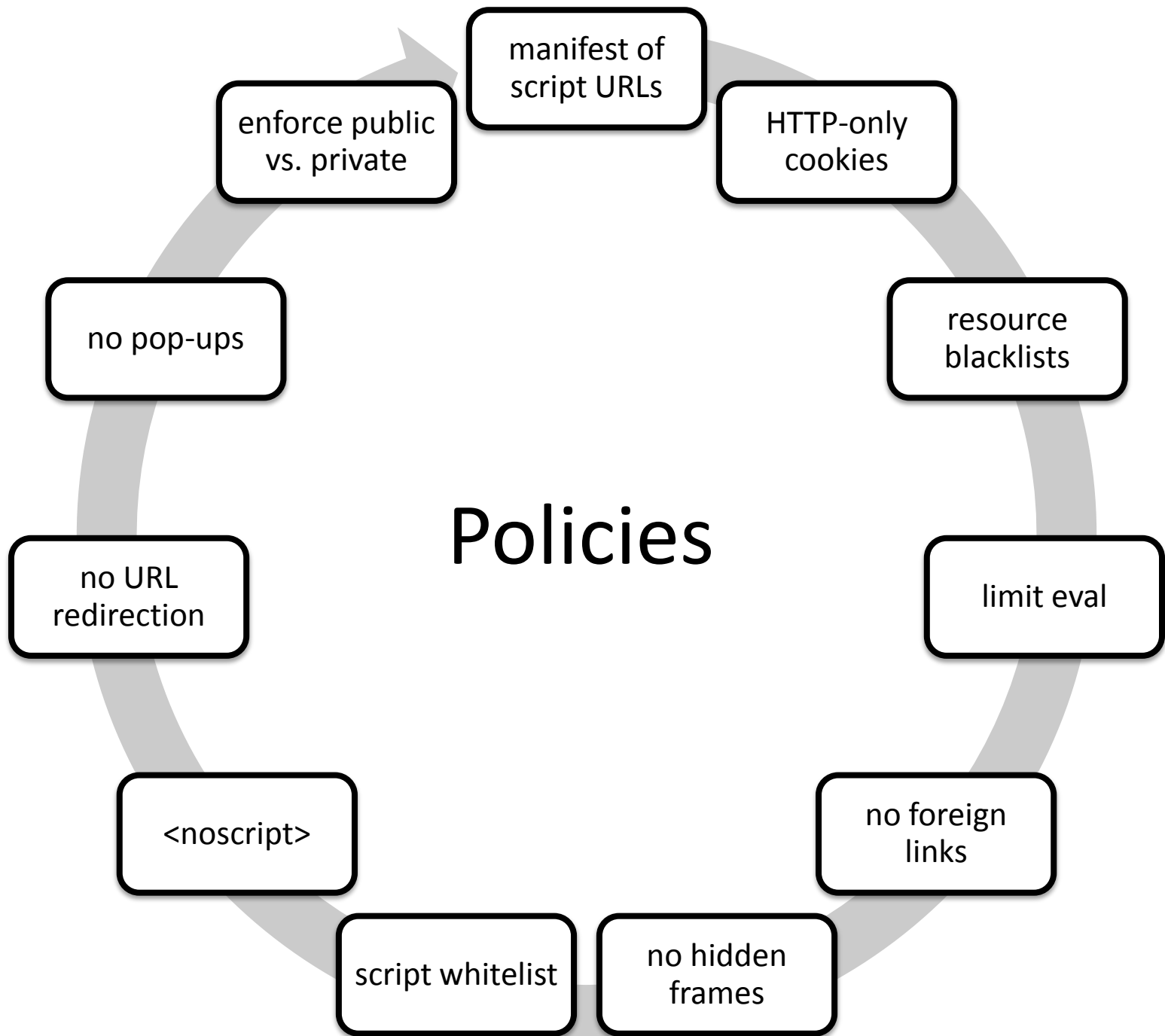
- Policies are easy to get wrong
- Type system to ensure policy correctness

Expressiveness

- 17 hand-written policies
- Comprehensive catalog of policies from literature and practice
- implemented 2 policy generators

Evaluation

- Tested on real apps: Google Maps, Live Desktop, etc.
- runtime and space overheads under 1% (vs. 30-550%)
- smaller trusted computing base (TCB)



CONSCRIPT aspects

implementing aspects in IE8

checking CONSCRIPT policies

generating CONSCRIPT policies

performance

No `postMessage`: A Simple Policy?

Wrapping: [[Caja, DoCoMo, AOJS, lightweightjs, Web Sandbox, ...]]

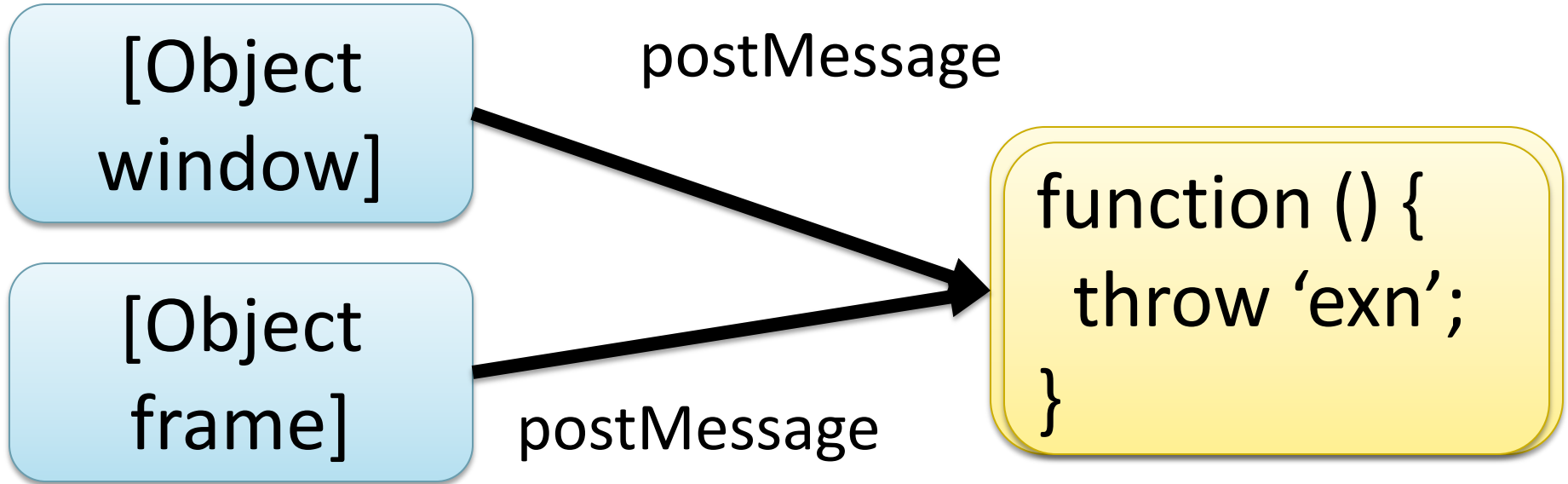
```
window.postMessage = function () {};  
frame1.postMessage("msg", "evil.com")
```

Aspects: [[AspectJ]]

```
void around(String msg, String uri) :  
call DOM.postMessage(String m, String u)  
{ /* do nothing instead of call */ }
```

... no classes in JavaScript / DOM ...

Specifying Calls using References



```
around(window.postMessage,  
        function () { throw 'exn'; }));
```

ConScript Interface

1. Functions

DOM: `aroundExt(postMessage, function (pm2, m, uri) { ... });`

JS: `aroundNat(eval, function (eval, str) { ... });`

User-defined: `aroundFnc(foo, function (foo2, arg1) { ... });`

2. Script introduction

`<script>`: `aroundScr(function (src) { return src + ';' + pol;});`

`inline`: `aroundInl(function (src) { return src + ';' + pol;});`

CONSCRIPT aspects

implementing aspects in IE8

checking CONSCRIPT policies

generating CONSCRIPT policies

performance

Problem: Implementation?

Source Rewriting [[aojs, docomo, caja, sandbox, fbjs]]

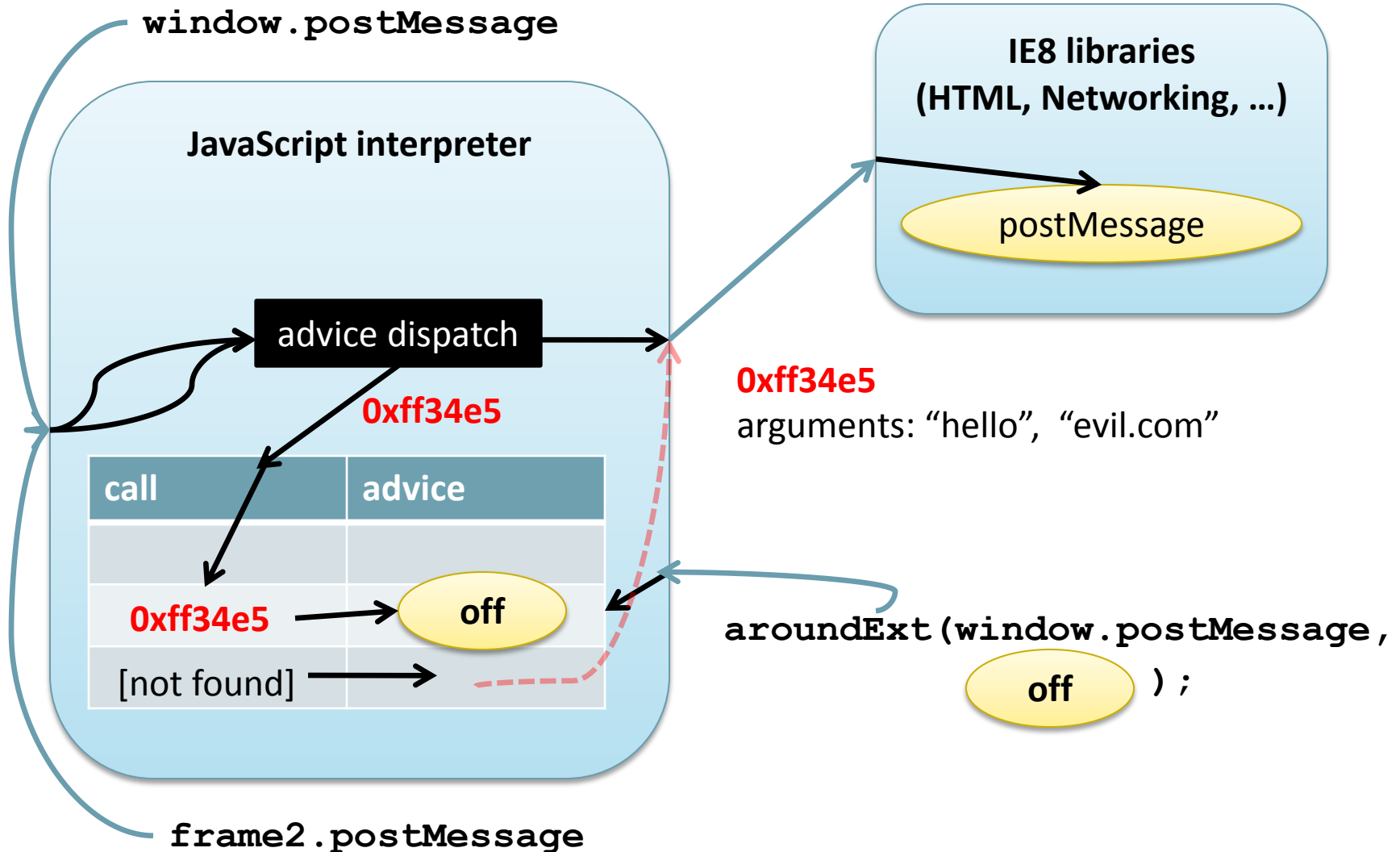
```
function f () { ... }
```



```
function f () {<before> ... <after>}
```

- ☹️ 50%-450% more to transfer, 30-70% slowdown
- ☹️ limited: native (DOM) functions, dynamic code?
- ☹️ big assumptions: adds parser to TCB, ...

Mediating DOM Functions



Resuming Calls

```
function foo () { }
```

```
function advice1 (foo2) {  
  if (ok()) {  
  
    foo2();  
  } else throw 'exn'; }  
}
```

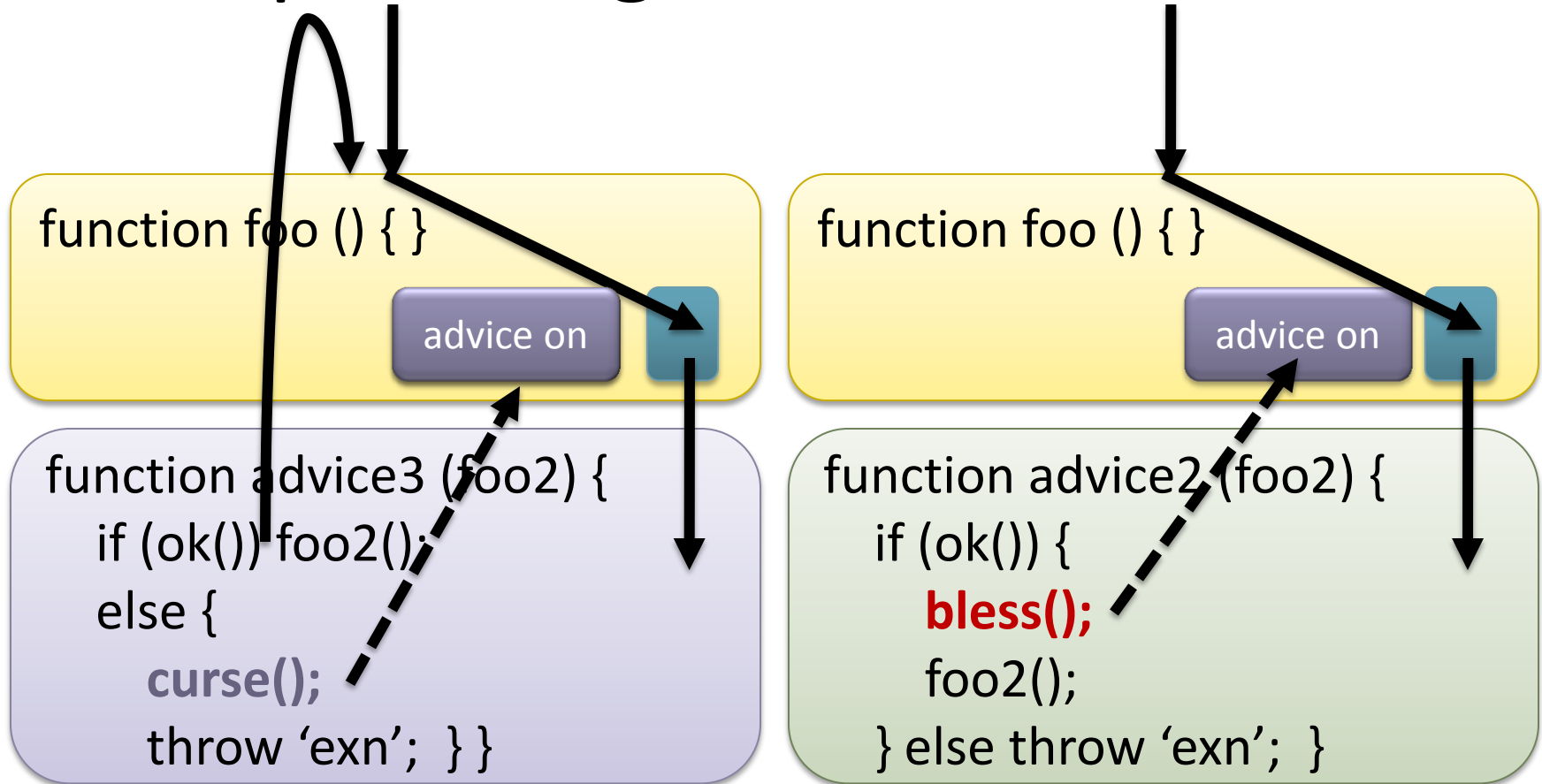
```
function foo () { }
```

advice off

```
function advice2 (foo2) {  
  if (ok()) {  
    bless();  
    foo2();  
  } else throw 'exn'; }  
}
```

bless() temporarily disables advice for next call

Optimizing the Critical Path



- calling advice turns advice off for next call
- **curse()** enables advice for next call

CONSCRIPT aspects

implementing aspects in IE8

checking CONSCRIPT policies

generating CONSCRIPT policies

performance

Basic Usage

script
whitelist

SURGEON GENERAL'S WARNING

Policies are written in a small JavaScript subset.

Applications only lose a few dangerous features.

```
<script src="main.js" policy="noEval()" />
```

Policy Integrity

Objects defined with policy constructors do not flow out

Old Policy

```
around(postMessage, function (m, url) {  
  w = {"msn.com": true};  
  ...  
})
```

Policy Integrity

Objects defined with policy constructors do not flow out

Old Policy

```
around(postMessage, function (m, url) {  
  w = {"msn.com": true};  
  ...  
})
```

policy object: must protect

unknown: do not pass privileged objects!

Policy Integrity

Objects defined with policy constructors do not flow out

Old Policy

```
around(postMessage, function (m, url) {  
  w = {"msn.com": true};  
  ...  
})
```

User Exploit

```
postMessage("", "msn.com");  
w["evil.com"] = 1;  
postMessage("", "evil.com");
```


Policy Integrity

Objects defined with policy constructors do not flow out

New Policy

```
around(postMessage, function (m, url) {  
  var w = {"msn.com": true};  
  ...  
})
```

User Exploit

```
postMessage("", "msn.com");  
w["evil.com"] = 1;  
postMessage("", "evil.com");
```

Policy Integrity

Objects defined with policy constructors do not flow out

New Policy

```
around(postMessage, function (m, url) {  
  var w = {"msn.com": true};  
  ...  
})
```

policy object: must protect

unknown: do not pass privileged objects!

Maintaining Integrity

1. Policy objects do not leak out of policies
2. Access path integrity of calls (no prototype hijacking)
 - ML-style type inference
 - ☹ basic
 - 😊 program unmodified
 - 😐 only manually tested on policies
 - JavaScript interpreter support
 - `call(ctx, fnc, arg1, ...)`, `hasOwnProperty(obj, "fld")`
 - `caller`

Transparency

- If running with policies throws no errors
 - ... for same input, running without should be safe
 - **empty advice should not be functionally detectable**
- Difficult with wrapping or rewriting
 - `Function.prototype.apply, exn.stacktrace, myFunction.callee, arguments.caller, myFunction.toString, Function.prototype.call`
 - correctness vs. compatibility vs. performance ...
- Simpler at interpreter level
 - rest up to developer
 - no proof

CONSCRIPT aspects

implementing aspects in IE8

checking CONSCRIPT policies

generating CONSCRIPT policies

performance

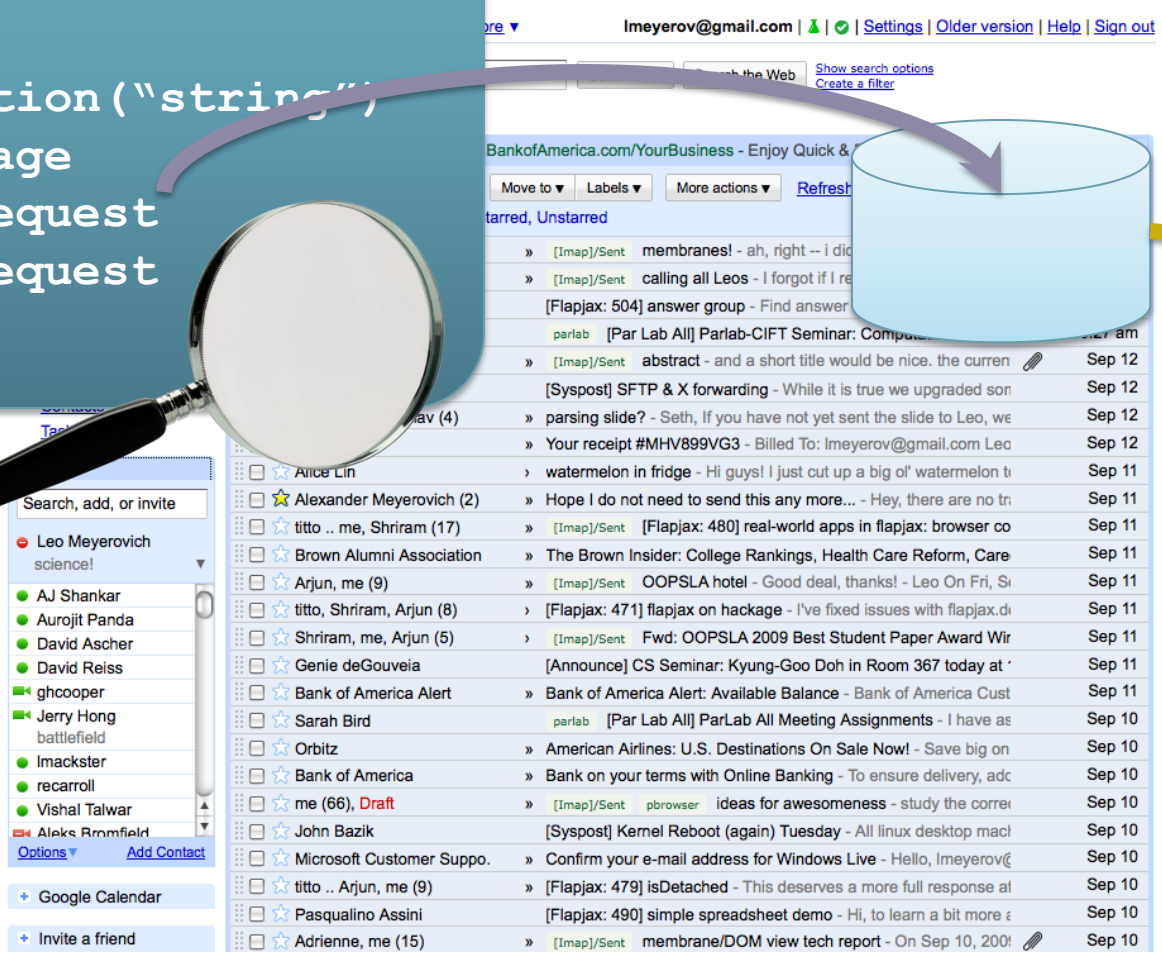
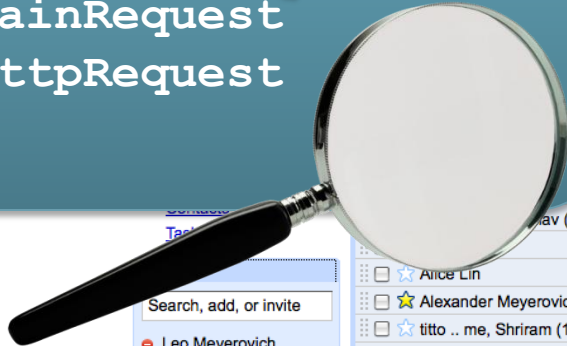
Automatically Generating Policies

- Intrusion detection
 - can we infer and disable unneeded DOM functions?
- C# access modifiers
 - can we enforce access modifiers like *private*?
- ASP policies
 - can we guarantee no scripts get run in `<% echo %>`?

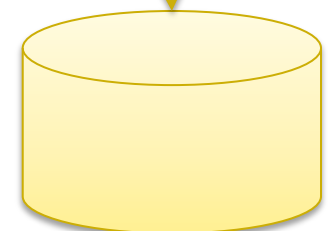
Intrusion Detection 1: Learn Blacklist

log

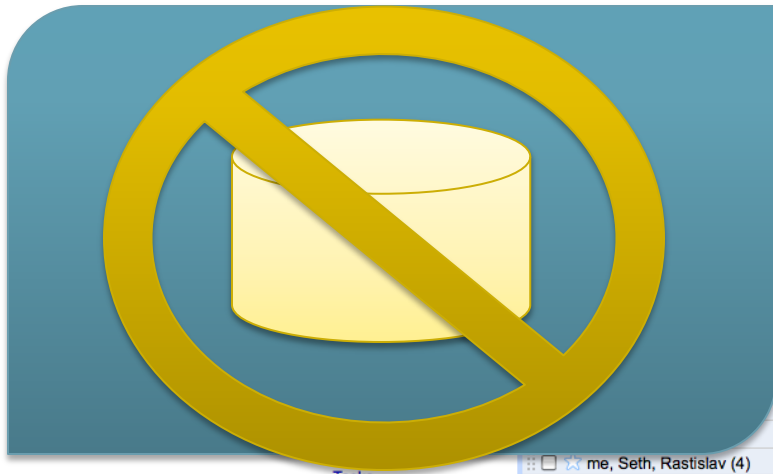
```
eval  
new Function("string")  
postMessage  
XMLHttpRequest  
XMLHttpRequest  
...
```



audit



Intrusion Detection 2: Enforce Blacklist



Imeyerov@gmail.com | Settings | Older version | Help | Sign out

Search Mail Search the Web Show search options Create a filter

BankofAmerica.com/YourBusiness - Enjoy Quick & Easy Online Bankii Sponsored Link

Move to Labels More actions Refresh 1 - 50 of 14193 Older Oldest

starred, Unstarred

» [imap]/Sent	membranes! - ah, right -- i didn't yet because the pn	12:52 pm
» [imap]/Sent	calling all Leos - I forgot if I replied -- but yes on all i	12:16 pm
[Flapjax: 504]	answer group - Find answer for any question here I	11:00 am
parlab	[Par Lab All] Parlab-CIFT Seminar: Computational Finan	10:27 am
» [imap]/Sent	abstract - and a short title would be nice. the curren	Sep 12
[Syspost]	SFTP & X forwarding - While it is true we upgraded son	Sep 12
»	parsing slide? - Seth, If you have not yet sent the slide to Leo, we	Sep 12
»	Your receipt #MHV899VG3 - Billed To: Imeyerov@gmail.com Leo	Sep 12
»	watermelon in fridge - Hi guys! I just cut up a big ol' watermelon ti	Sep 11
»	Hope I do not need to send this any more... - Hey, there are no tr	Sep 11
» [imap]/Sent	[Flapjax: 480] real-world apps in flapjax: browser co	Sep 11
»	The Brown Insider: College Rankings, Health Care Reform, Care	Sep 11
» [imap]/Sent	OOPSLA hotel - Good deal, thanks! - Leo On Fri, Si	Sep 11
» [Flapjax: 471]	flapjax on hackage - I've fixed issues with flapjax.di	Sep 11
» [imap]/Sent	Fwd: OOPSLA 2009 Best Student Paper Award Wir	Sep 11
» [Announce]	CS Seminar: Kyung-Goo Doh in Room 367 today at '	Sep 11
»	Bank of America Alert: Available Balance - Bank of America Cust	Sep 11
» parlab	[Par Lab All] ParLab All Meeting Assignments - I have as	Sep 10
»	American Airlines: U.S. Destinations On Sale Now! - Save big on	Sep 10
»	Bank on your terms with Online Banking - To ensure delivery, adc	Sep 10
» [imap]/Sent	pbrowser ideas for awesomeness - study the corre	Sep 10
» [Syspost]	Kernel Reboot (again) Tuesday - All linux desktop macl	Sep 10
»	Confirm your e-mail address for Windows Live - Hello, Imeyerov@	Sep 10
» [Flapjax: 479]	isDetached - This deserves a more full response al	Sep 10
» [Flapjax: 490]	simple spreadsheet demo - Hi, to learn a bit more e	Sep 10
» [imap]/Sent	membrane/DOM view tech report - On Sep 10, 200!	Sep 10

Tasks

Chat

Search, add, or invite

- Leo Meyerovich science!
- AJ Shankar
- Aurojit Panda
- David Ascher
- David Reiss
- ghcooper
- Jerry Hong battlefield
- Imackster
- recarroll
- Vishal Talwar
- Aleks Bromfield

Options Add Contact

Google Calendar

Invite a friend

Enforcing C# Access Modifiers

```
class File {  
  public File () { ... }  
  private open () { ... }  
  ...  
}
```

Script#
compiler

```
function File () { ... }  
File.construct = ...  
File.open = ...  
...
```

C#

policy
generator

JavaScript

```
around(File, pubEntryPoint);  
around(File.construct, pubEntryPoint);  
around(File.open, privCall);
```

ConScript

CONSCRIPT aspects
implementing aspects in IE8
checking CONSCRIPT policies
generating CONSCRIPT policies
performance

Performance

Microbenchmarks: 1.2x (vs. 3.4x)

Initialization time: 0-1%

Runtime: 0-7% (vs. 30+%)

File size blowup: < 1% (vs. 50+%)

Microbenchmark: Mediation Overhead

```
var raw = obj.f;  
obj.f = function () { raw();}
```

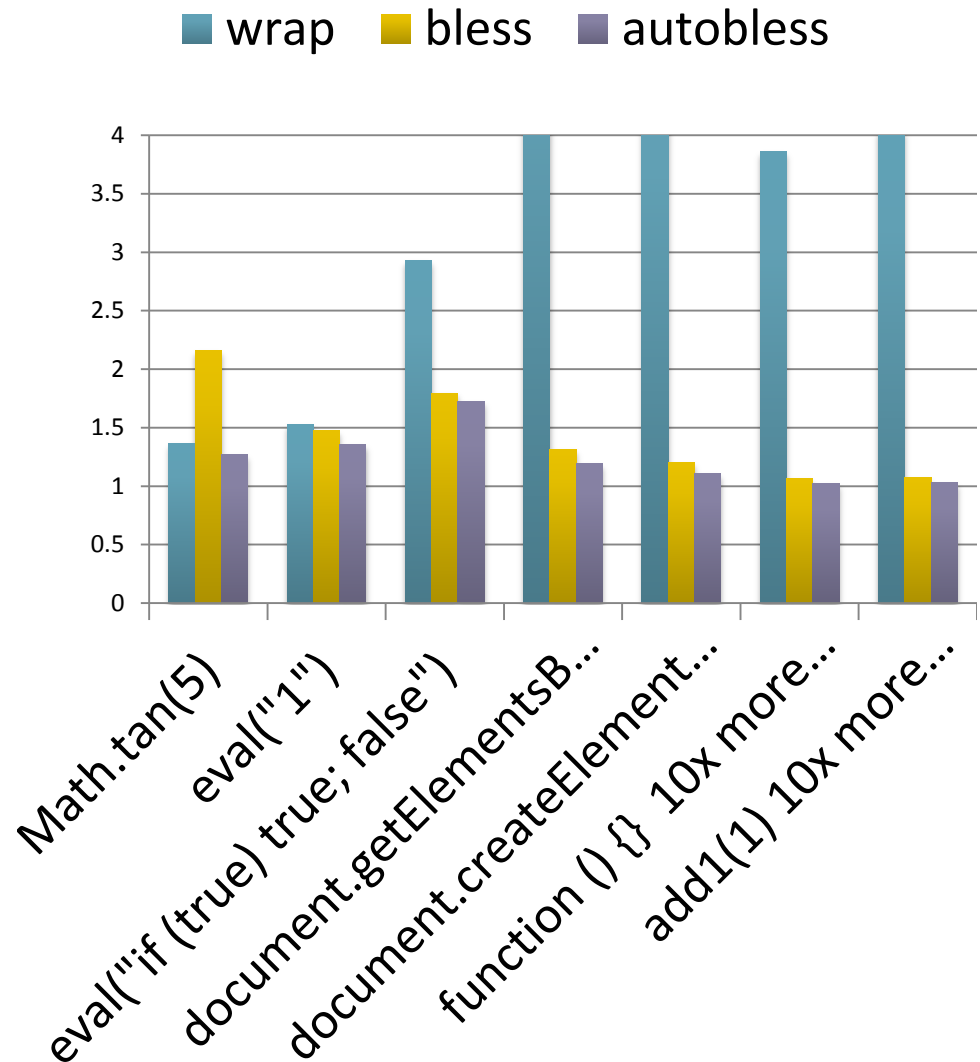
3.42x

```
function advice3  
bless();  
foo2();  
}
```

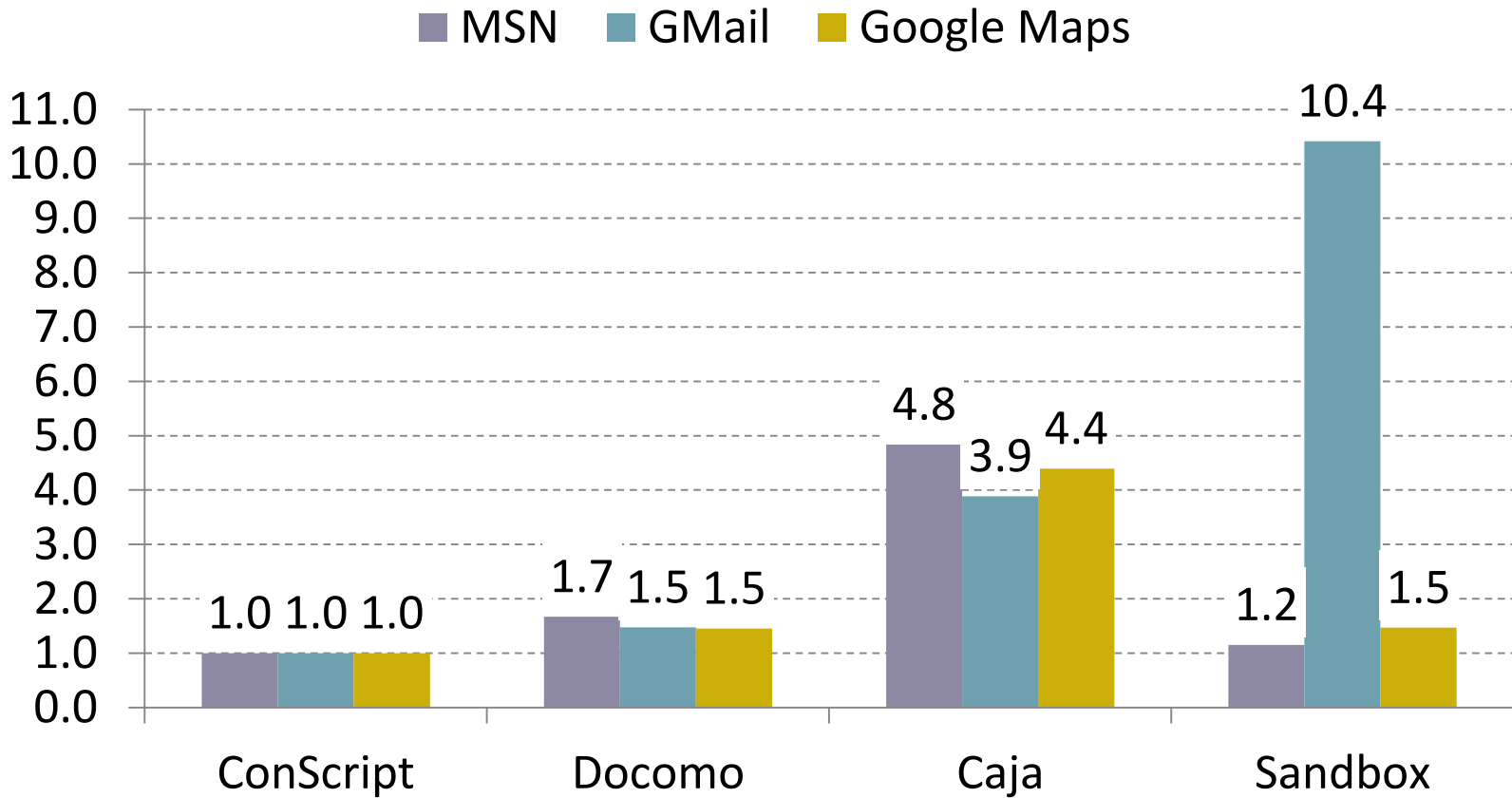
1.44x

```
function advice3  
foo2();  
}
```

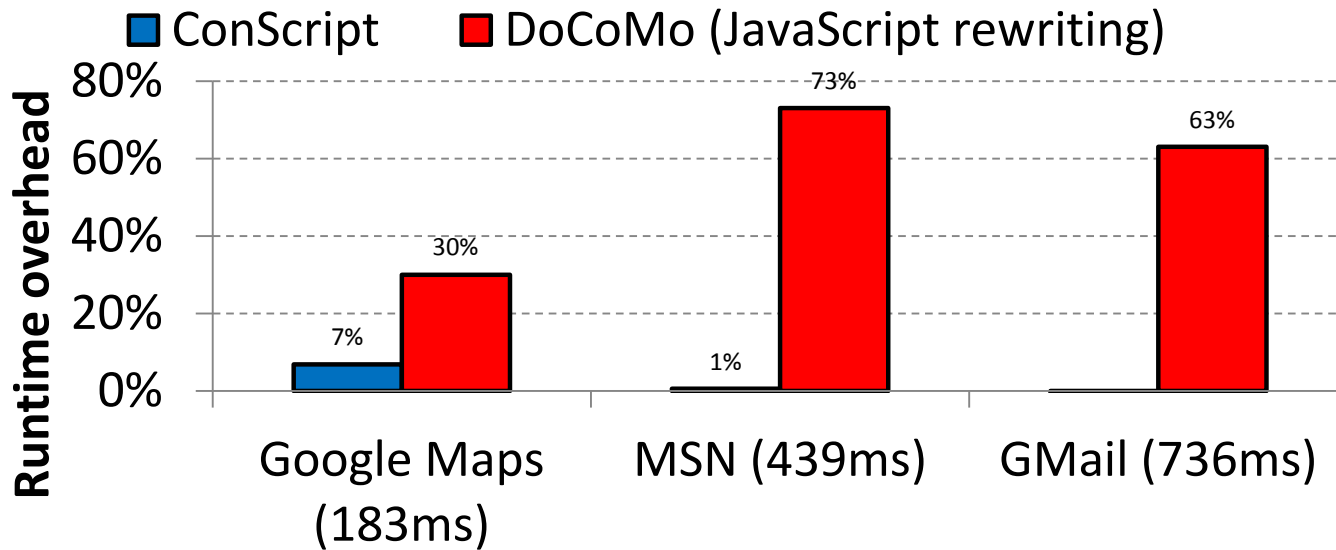
1.24x



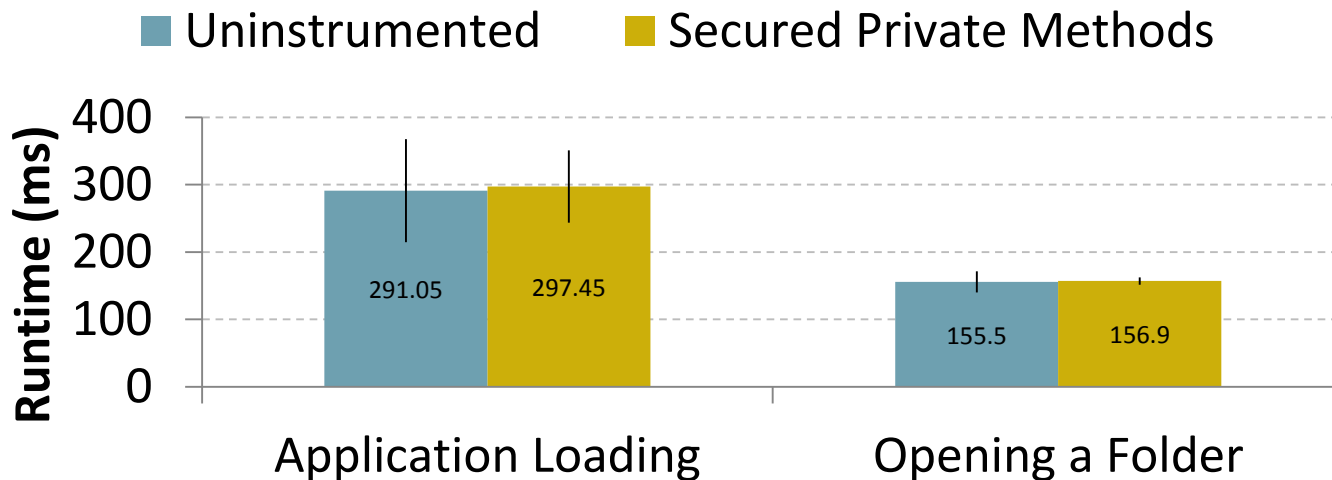
File Size Increase (IDS)



Runtime Overhead



Intrusion
Detection
System



Access
Modifier
Enforcement

Goals and Contributions

control loading
and use of scripts

- protect benign users
- by giving control to hosting site
- ConScript approach: aspects for security

express many
policies *safely*

- 16 hand-written policies
- correct policies are hard to write
 - proposed type system to catch common attacks
 - implemented 2 policy generators

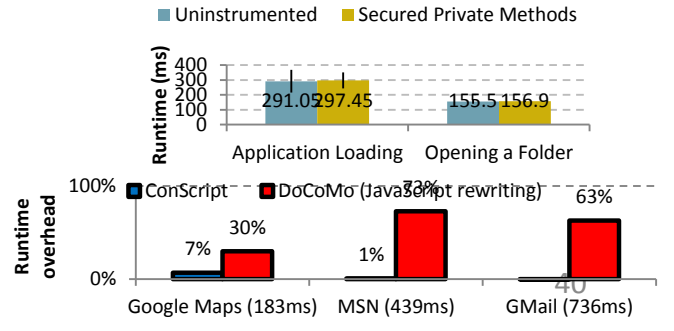
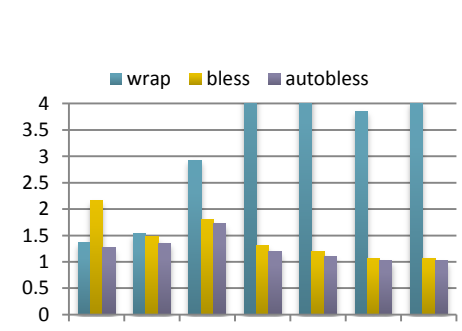
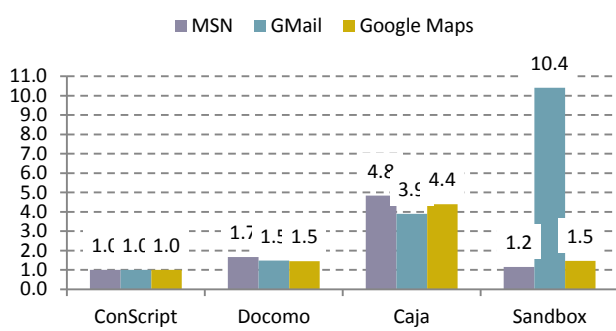
browser support

- built into IE 8 JavaScript interpreter
- runtime and space overheads under 1% (vs. 30-550%)
- smaller trusted computing base (TCB)

Questions?



$\frac{}{\Gamma \vdash i : K \text{ where } i \in \mathbb{R} \cup \text{STRING} \cup \{\text{null}, \text{undefined}\}} \text{ (prim)}$
 $\frac{\Gamma \vdash e : T}{\Gamma \vdash \text{typeof } e : K} \text{ (typeof)}$
 $\frac{\Gamma \vdash e : T}{\Gamma \vdash \text{toPrimitive}(e) : K} \text{ (toPrim)}$
 $\frac{x : T \notin \Gamma}{\Gamma \vdash x : U} \text{ (unsafe env var)}$
 $\frac{\Gamma \vdash o : T \quad \Gamma \vdash v : T}{\Gamma \vdash (o = v) : T} \text{ (asgn)}$
 $\frac{\Gamma \vdash o : T_1 \leq K \quad \Gamma \vdash i : T_2}{\Gamma \vdash o.f : U} \text{ (u stat get)}$
 $\frac{\Gamma \vdash o : T_1 \leq K \quad \Gamma \vdash i : T_2}{\Gamma \vdash o[i] : U} \text{ (u dyn get)}$
 $\frac{\Gamma \vdash o : T_1 \leq K \quad \Gamma \vdash v : T_2 \leq K}{\Gamma \vdash (o.f = v) : U} \text{ (u stat set)}$
 $\frac{\Gamma \vdash o : T_1 \leq K \quad \Gamma \vdash i : T \quad \Gamma \vdash v : T_2 \leq K}{\Gamma \vdash (o[i] = v) : U} \text{ (u dyn set)}$
 $\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \phi \in \{\&\&, ||, ==\}}{\Gamma \vdash t_1 \phi t_2 : K} \text{ (binop)}$
 $\frac{\Gamma \vdash t_1 : T_1 \leq K \quad \Gamma \vdash t_2 : T_2 \leq K}{\Gamma \vdash t_1 + t_2 : K} \text{ (add/concat)}$
 $\frac{\Gamma \vdash t : T \quad \phi \in \{!, +\}}{\Gamma \vdash \phi t : K} \text{ (unop)}$
 $\frac{\Gamma \vdash s_1 \quad \Gamma, e : U \vdash s_2}{\Gamma \vdash \text{try } \{s_1\} \text{ catch } (e) \{s_2\}} \text{ (try)}$
 $\frac{\Gamma \vdash e : T \leq K}{\Gamma \vdash \text{throw } e} \text{ (throw)}$
 $\frac{\Gamma \vdash o : K \quad \Gamma \vdash \vec{a} : \vec{T} \leq \mathbf{K}}{\Gamma \vdash [\text{new}] o(\vec{a}) : U} \text{ (u f app)}$
 $\frac{\Gamma \vdash p : T_1 \leq K}{\Gamma \vdash \text{around}(p, f) : K} \text{ (around)}$
 $\frac{x : T \notin \Gamma}{\Gamma \vdash x : K} \text{ (unsafe env var)}$
 $\frac{\Gamma \vdash o : K}{\Gamma \vdash o.f : K} \text{ (u stat get)}$
 $\frac{\Gamma \vdash o : K \quad \Gamma \vdash i : T}{\Gamma \vdash o[i] : K} \text{ (u dyn get)}$
 $\frac{\Gamma \vdash o : K \quad \Gamma \vdash \vec{a} : \vec{T} \leq \mathbf{K}}{\Gamma \vdash [\text{new}] o.f(\vec{a}) : U} \text{ (u m app)}$
 $\frac{\Gamma \vdash o : K \quad \Gamma \vdash i : T \quad \Gamma \vdash \vec{a} : \vec{T} \leq \mathbf{K}}{\Gamma \vdash [\text{new}] o[i](\vec{a}) : U} \text{ (u d m app)}$
 $\frac{\Gamma \vdash \vec{v} : \vec{T} \quad T_f = \{ \dots, f_{|v|} : T_{|v|} \}}{\Gamma \vdash \{f : \vec{v}\} : T_f} \text{ (k obj lit)}$
 $\frac{\Gamma \vdash \vec{v} : \vec{T} \quad T_f = \{\text{length} : K\} \cup \{ \dots, |v| : T_{|v|} \}}{\Gamma \vdash [\vec{v}] : T_f} \text{ (k arr lit)}$
 $\frac{\Gamma \vdash o : T_1 \in \text{RECORD} \quad f : T \in T_1}{\Gamma \vdash o.f : T} \text{ (k stat get)}$
 $\frac{\Gamma \vdash o : T \in \text{RECORD} \quad i : T \leq K}{\Gamma \vdash \text{hasProp}(o, i) : K} \text{ (k hasProp)}$
 $\frac{\Gamma \vdash o : T_1 \quad f : T \in T_1}{\Gamma \vdash (o.f = v) : T} \text{ (k stat set)}$
 $\frac{\Gamma, \vec{a} : \vec{T} \vdash \text{ret}(s) : T_2}{\Gamma \vdash \text{function}(\vec{a}) \{s\} : \vec{T} \rightarrow T_2} \text{ (k abstr)}$
 $\frac{\Gamma, \text{this} : U, \text{arguments} : \{ \} \vdash f : \vec{T} \rightarrow T_2}{\Gamma \vdash f(\vec{a}) : T_2} \text{ (k f app)}$
 $\frac{\Gamma \vdash \vec{a} : \vec{T} \quad (f : \vec{T} \rightarrow T_2) \in R}{\Gamma, \text{this} : R, \text{arguments} : \{ \} \vdash o : R} \text{ (k m app)}$
 $\frac{\Gamma \vdash o.m(\vec{a}) : T_2}{\Gamma \vdash o.m(\vec{a}) : T_2} \text{ (k m app)}$



END.