Rethinking Web Platform Extensibility

Mohan Dhawan
Rutgers University
Gateway to the World Wide Web
Gateway to the World Wide Web

March 13, 2013
Mohan Dhawan
Google will not load.
Google will eat itself.
Google will rule the world.
Google will i.am
Google will you be my Valentine?
Google will disappear.
Google will beat Apple.
Google will take over the world.

Yahoo! News

Most Popular

Coffee Might Trigger First Heart Attack in Some

Most Viewed

MULTIPLE HIGHLIGHTS

Most Recommended

Search

Yahoo! My Yahoo! Mail

Search

Internet Home

Home | U.S. | Business | World | Entertainment | Sports | Tech | Politics | Science | Health | Travel

Search Results

Most Popular

Coffee Might Trigger First Heart Attack in Some

TUESDAY, Aug. 15 (HealthDay News) — An occasional cup of coffee might trigger first heart attacks in some people, a new study suggests.

Most Viewed

Coffee Might Trigger First Heart Attack in Some

TUESDAY, Aug. 15 (HealthDay News) — An occasional cup of coffee might trigger first heart attacks in some people, a new study suggests.

Most Recommended

Coffee Might Trigger First Heart Attack in Some

TUESDAY, Aug. 15 (HealthDay News) — An occasional cup of coffee might trigger first heart attacks in some people, a new study suggests.

Most Emailed News

1. Home sales decline in 28 states, D.C.

WASHING ron - The slowdown in the once-sizzling housing market is spreading, with 28 states and the District of Columbia reporting spring sales declines led by big drops in former hotter areas of Arizona.

Folgers Simply Smooth

Enjoy a Second Cup

New!
March 13, 2013

Mohan Dhawan
Web platform extensibility

Client

Internet

Web Server
Web platform extensibility
Web platform extensibility
Web platform extensibility
Web browser and Web application extensibility are complementary
Web platform extensibility

Client-hosted Web applications can enhance Web browser functionality

Browser extensions can enhance Web application functionality

Web browser and Web application extensibility are complementary
An operating systems analogy

User application

Operating system

- Device drivers
- Memory manager
- Scheduler, ...
Problems with OS extensibility (I)

Problem: Buggy & vulnerable drivers

User application

Operating system

Device drivers
Memory manager
Scheduler, ...
Problems with OS extensibility (I)
Problems with OS extensibility (I)
Problems with OS extensibility (I)

Some solutions

1) SPIN [SOSP’95]
Key idea: Language-based safety

2) VINO [OSDI’96]
Key idea: SFI + Lightweight transactions

3) Nooks [SOSP’03]
Key idea: Lightweight protection domains
Problems with OS extensibility (II)

Problem: OS bloat, rigid interfaces
Problems with OS extensibility (II)

Solution

Exokernel [SOSP’95]

Key idea: Applications manage complexity
Exokernel operating system

User application

Device drivers
Memory manager
Scheduler, ...

User application

Device drivers
Memory manager
Scheduler, ...

Exokernel

March 13, 2013
Mohan Dhawan
Exokernel operating system

User application manages complexity
My thesis

We can leverage operating system principles to secure and enhance the extensibility of the Web platform.
Contributions

• Characterizing extensibility issues in web browsers
  – Analyzed the nature of information flow in JavaScript-based web browser extensions [ACSAC’09, ECOOP’12a]

• Extensibility as a language and system primitive
  – Designed and implemented a language runtime system to secure web application extensibility [PLAS’10, ECOOP’12b]
  – Built a novel browser architecture that enables applications to leverage extensibility for security and robustness [SOSP’11]
Part I: Securing Web platform extensions with Sabre and Transcript
Analogy: OS extensions

User application

Web application

Operating system

Web browser

Device driver 1

Device driver 2

Extension 1

Extension 2
Analogy: OS extensions

Operating system

User application

Device driver 1

Device driver 2

Web browser

Web application

Extension 1

Extension 2
Analogy: OS extensions

User application

Operating system

Device driver 1

Device driver 2

Web application

Web browser

Extension 1

Extension 2
Extensibility with vulnerability?


CVE-2011-2425, CVE-2011-2424, CVE-2011-2417, CVE-2011-2416, CVE-2011-2415, CVE-2011-2414, ...

CVE-2012-2990, CVE-2010-4597, CVE-2010-4588, CVE-2010-3973, CVE-2010-2888, CVE-2009-3737, ...

CVE-2012-2877, CVE-2012-0958, CVE-2011-3647, CVE-2011-3001, CVE-2011-1812, CVE-2011-2785, ...
Extensibility with vulnerability?

CVE-2013-0422, CVE-2011-0873, CVE-2011-0871, CVE-2011-3965,
CVE-2011-2425, CVE-2011-2424, CVE-2011-2417, CVE-2011-2416,
CVE-2011-2420, CVE-2011-2421, CVE-2012-2990, CVE-2012-2877,
CVE-2012-0958, CVE-2012-2888, CVE-2010-4597, CVE-2010-3973,
CVE-2010-4588, CVE-2010-2888, CVE-2012-2990, CVE-2011-3647,
CVE-2011-3001, CVE-2011-1812, CVE-2011-2785,
CVE-2009-3737, ...
JavaScript-based extensions

Extensions run with privileges of the hosting principal
JavaScript-based extensions

Web browser extensions

Cookies
Browsing History
Preferences

Web application extensions

Passwords
Network
File System
Operating System
JavaScript-based extensions

Web browser extensions

Web application extensions

Cookies
Local Storage
Application Data

Keyboard and Mouse Events
AJAX
Web browser extension example

Victim

Vulnerable Firefox and Greasemonkey

Other extension

Grease-monkey

www.evil.com
Web browser extension example

Victim accidently visits www.evil.com

Victim accidently visits www.evil.com

Vulnerable Firefox and Greasemonkey

Grease-monkey

Other extension
Web browser extension example

Victim

Vulnerable Firefox and Greasemonkey

www.evil.com responds with a Web page containing malicious JavaScript code

www.evil.com
Web browser extension example

Malicious script exploits bugs in browser and extension to get access to Greasemonkey APIs

Vulnerable Firefox and Greasemonkey

Victim

Other extension

Greasemonkey

www.evil.com
Web browser extension example

3. Malicious script exploits bugs in browser and extension to get access to Greasemonkey APIs

Victim

Vulnerable Firefox and Greasemonkey

Other extension

Access to privileged Greasemonkey APIs

www.evil.com
Web browser extension example

Malicious script uses privileged extension APIs to read sensitive files on the file system.

Victim

Vulnerable Firefox and Greasemonkey

Other extension

File System

www.evil.com

March 13, 2013
Mohan Dhawan
Web browser extension example

Malicious script uses extension APIs to send sensitive files to the remote attacker

Vulnerable Firefox and Greasemonkey

Other extension

Victim

www.evil.com
Web application extension example

Some NYTimes.com readers have seen a pop-up box warning them about a virus and directing them to a site that claims to offer antivirus software. We believe this was generated by an unauthorized advertisement and are working to prevent the problem from recurring. If you see such a warning, we suggest that you not click on it. Instead, quit and restart your Web browser. Questions and comments can be sent to adtraffic@nytimes.com.
Web application extension example

• Rogue third party advertisement
  – Displayed image of fake virus scan
Problems with extensions

Untrusted extensions put client security and privacy at risk
Security for extensions

- **Sabre [ACSAC’09] (Best Student Paper Award) secure Web browser extensions**
  - Analyze information flow in JavaScript-based extensions
  - Sabre found several information flow violations in legacy extensions

- **Transcript [PLAS’10, ECOOP’12b] secures Web application and Web browser extensions**
  - Implements *isolation* as a first class primitive for JavaScript
Sabre: Analyzing information flow in JavaScript-based extensions
Sabre: Key idea

Sabre enhanced Firefox with extensions
Sabre: Key idea

Sabre enhanced Firefox with extensions
Sabre: Key idea

Sabre enhanced Firefox with extensions

1. Appropriately mark data as sensitive
Sabre: Key idea

Sabre enhanced Firefox with extensions

2. Track the propagation of security label in the browser
Sabre: Key idea

Sabre enhanced Firefox with extensions

Take action when sensitive data is externalized
Sabre: Greasemonkey attack

Sabre enhanced Firefox with extensions
Sabre: Greasemonkey attack

Sabre enhanced Firefox with extensions
Sabre: Greasemonkey attack
# Results: Categorizing benign extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>HTML forms</th>
<th>HTTP channel</th>
<th>File system</th>
<th>Load URLs</th>
<th>JS events</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ad Block</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All-in-one Sidebar</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Cool Previews</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Download Statusbar</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Fast Video Download</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Forecastfox</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Foxmarks Synchronizer</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ghostery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Google Previews</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Greasemonkey (v0.8.1)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NoScript</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDF Download</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PwdHash</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SpeedDial</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>StumbleUpon</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stylish</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Tab Mix Plus</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>User Agent Switcher</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Video DownloadHelper</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web-of-Trust</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

March 13, 2013  Mohan Dhawan
## Results: Categorizing benign extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>HTML forms</th>
<th>HTTP channel</th>
<th>File system</th>
<th>Load URLs</th>
<th>JS events</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Ad Block</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 All-in-one Sidebar</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>3 Cool Previews</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 Download Statusbar</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 Fast Video Download</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6 Forecastfox</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>7 Foxmarks Synchronizer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>8 Ghostery</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>9 Google Previews</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>10 Greasemonkey (v0.8.1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11 NoScript</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>12 PDF Download</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>13 PwdHash</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>14 SpeedDial</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>15 StumbleUpon</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>16 Stylish</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>17 Tab Mix Plus</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 User Agent Switcher</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>19 Video DownloadHelper</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 Web-of-Trust</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

**Whitelisting / declassification of trusted extensions is essential**

March 13, 2013

Mohan Dhawan
Results: Accuracy

• Vulnerable and malicious extensions
  – GreaseMonkey v0.3.3
  – Firebug v1.01
  – FFsniff
  – BrowserSPY

• Result
  – Precisely identified all flow violations
  – No false positives during normal web browsing
Transcript: Language-based security for Web platform extensions
Example: Online text editor

Hello All,

I spent like forever writing the best email I’ve ever written with all of my good travel tips. When I hit send I noticed my login expired and I lost everything. I guess that is how it goes in the wild west.

**Revise all of**

Redundant expressions needlessly repeat ideas when fewer words clearly communicate the meaning. They add no value to your documents. *(How To/Tools: Quick reference tips)*

**Try a simpler word for that is**

Simple words help you express your message clearly. Replacing complex words with simpler words whenever possible lets your readers concentrate on your ideas and information. *(How To/Tools: Complex and Abstract Words by Nick Wright)*

Replace that is with

- (omit) when possible

I'm due to PCS soon, so I'm putting the word out now to ensure it gets out, here are my top European travel tips for shift workers stationed at Ramstein AB:
Example: Online text editor

Editor’s DOM modified by third party Web application extension
UI redressing attack

This is what you see

This is what you click on

z-index: -1

opacity: 0.0
z-index: 0
State of the art: Access control
State of the art: Access control

<iframe> is rigid and hampers functionality

Web Application

<iframe> sandboxed third party content
State of the art: Access control

<i>iframe</i> is rigid and hampers functionality

Need a fine-grained JavaScript sandbox
Access control sufficient?

Reference monitors use access control policies to sandbox untrusted third party JavaScript content.
Access control policies may allow seemingly innocuous, but undesirable JavaScript heap and DOM changes.
Access control is not sufficient.
Access control not sufficient

(i) Lose all unsaved work
or
(ii) Continue unsafe work
Access control is not sufficient

Undo all effects of untrusted code on any policy violation.
Access control \( \land \) sufficient

Undo all effects of untrusted code on any policy violation

Speculative execution provides isolation
• Enhance JavaScript language with **speculation**
  – Execute untrusted content speculatively

  – Commit changes after policy enforcement
Transcript goals (I)

eval this with
Transcript goals (I)

```javascript
> eval(`var foo = 42;`);
> foo
42
```
> eval(<N/W or User Input>);
> ??
Transcript goals (I)

Handle arbitrary third party code including dynamic constructs
Transcript goals (II)

- Cookie stealing
- UI Redressing
- Drive-by downloads
- Annoying popups
- Undesired navigation
- X-domain communication

Enforce powerful security policies on 3rd party behavior
Features of Transcript

• **JavaScript speculation**
  – Speculative execution of *unmodified* third party JavaScript code

• **Suspend/resume speculative execution**
  – Web application can mediate external actions like DOM and AJAX operations (*akin to a system call*)

• **Speculative DOM updates**
Transcript in action

```javascript
var sp = speculate {
    ... body.appendChild(overlay);
    ...
};

do {
    ...
    sp = sp.resume();
    ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system
Transcript in action

```javascript
var sp = speculate {
    ...
    body.appendChild(overlay);
    ...
};

iBlock

do {
    ...
    sp = sp.resume();
    ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system

Clone

DOM_{orig}

DOM_{SP}
Transcript in action

```
var sp = `speculate {
  ...
  body.appendChild(overlay);
  ...
};

do {
  ...
  sp = sp.resume();
  ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system

- 3rd-party
- call stack
- Web app
- ....
Transcript in action

```javascript
var sp = speculate {
    ... body.appendChild(overlay);
    ...
};

do {
    ...
    sp = sp.resume();
    ...
} while(sp.isSuspended());
sp.commit();
```

Transcript runtime system

Speculation object sp

3rd party call stack R/W sets

DOM_{sp}

Web app
Transcript in action

```javascript
var sp = speculate {
  ...
  body.appendChild(overlay);
  ...
};

do {
  ...
  sp = sp.resume();
  ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system

DOM<sub>SP</sub>
var sp = \texttt{speculate} \{ 
  ...
  body.appendChild(overlay);
  ...
\};

do {
  ...
  sp = sp.resume();
  ...
} while(sp.isSuspended());

sp.commit();
var sp = \texttt{speculate} \{ 
    ... 
    body.appendChild(overlay); 
    ... 
\};

do { 
    ... 
    sp = sp.resume(); 
    ... 
} while(sp.isSuspended());

sp.commit();
Transcript in action

```javascript
var sp = speculate {
    ...
    body.appendChild(overlay);
    ...
};

iBlock

do {
    ...
    sp = sp.resume();
    ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system
Transcript in action

```javascript
var sp = speculate {
  ...
  body.appendChild(overlay);
  ...
};

doi{
  ...
  sp = sp.resume();
  ...
} while(sp.isSuspended());

sp.commit();
```

Transcript runtime system

- sp’s write set
- Heap\(_{orig}\)
- DOM’\(_{SP}\)
- Heap\(_{new}\)
- DOM\(_{new}\)
Implementation

Firefox 3.7a4

SpiderMonkey
Implementation

Firefox 3.7a4

SpiderMonkey

<html>
<head>
<title>
<body>
<div>
<div>
<title>
Implementation

Firefox 3.7a4

SpiderMonkey

Speculate construct
Speculation object
Suspend/Resume
Read/Write logs

{JS}
Applicability of Transcript

**JS Menu:** No network or cookie access

**Picture Puzzle:** Disallow attaching key event handlers

**Spell Checker:** No **Ajax** if cookies were read

**GreyBox:** `<iframe>`s to whitelisted URLs only

**Color Picker:** No **innerHTML** in host’s context
Applicability of Transcript

Observed no change in the behavior of third party code

GreyBox: <iframe>s to whitelisted URLs only

Color Picker: No innerHTML in host’s context
Application benchmarks

- JS Menu: Original 0.102, Transcript (JS only) 0.179, Transcript (full) 0.252
- Picture Puzzle: Original 0.118, Transcript (JS only) 0.156, Transcript (full) 0.170
- Spell Checker: Original 0.156, Transcript (JS only) 0.196, Transcript (full) 0.216
- GreyBox: Original 0.118, Transcript (JS only) 0.144, Transcript (full) 0.155
- Color Picker: Original 0.147, Transcript (JS only) 0.521, Transcript (full) 0.652
- Average: Original 0.128, Transcript (JS only) 0.239, Transcript (full) 0.289
Application benchmarks

Overhead = 0.16s
Application benchmarks

- **JS Menu**: Original 0.102, Transcript (JS only) 0.179, Transcript (full) 0.252
- **Picture Puzzle**: Original 0.118, Transcript (JS only) 0.156, Transcript (full) 0.170
- **Spell Checker**: Original 0.156, Transcript (JS only) 0.196, Transcript (full) 0.216
- **GreyBox**: Original 0.118, Transcript (JS only) 0.144, Transcript (full) 0.155
- **Color Picker**: Original 0.147, Transcript (JS only) 0.521, Transcript (full) 0.652
- **Average**: Original 0.128, Transcript (JS only) 0.239, Transcript (full) 0.289

*Time in seconds*
Summary: Part I

• Sabre uses information flow tracking across browser subsystems to prevent security and privacy violations
  – Exploited browser extensions can cause loss of sensitive information

• Transcript implements speculative execution for JavaScript to provide isolation & recovery
  – Enforcement of powerful security policies
  – No restriction or changes to third party code
Part II: Enhancing Web platform extensibility with Atlantis
Analogy: OS and Web browser

- User application
  - Device drivers
  - Memory manager
  - Scheduler, ...

- Web application
  - Rendering engine
  - DOM
  - HTML/CSS parser
  - JavaScript runtime

- Operating system
- Web browser
The Web protocol

HTTP
The Web protocol
The Web protocol

- JavaScript DOM
- Bindings
- HTTP
- CSS
- HTML
The Web protocol

JavaScript DOM
Bindings

Java

HTTP

JavaScript

Silverlight

CSS

HTML

PDF

Flash

Quicktime
The Web protocol

- HTTP
- HTML
- CSS
- JavaScript
- JavaScript DOM
- Bindings
- Silverlight
- Java
- Flash
- Quicktime
- WebGL
- WebGL
- Web workers
- Web sockets
- <video>
- <canvas>
- PDF
- CSS
The Web protocol

- Geolocation
- JSON
- Web sockets
- Web workers
- JavaScript
- JavaScript DOM Bindings
- WebGL
- HTTP
- CSS
- PDF
- <video>
- <canvas>
- HTML
- DOM
- Storage
- Java
- Silverlight
- Flash
- Quicktime
- Geolocation
- JavaScript DOM Bindings
- WebGL
- HTTP
- CSS
- PDF
- <video>
- <canvas>
- HTML
- DOM
- Storage
- Java
- Silverlight
- Flash
- Quicktime
The **complex** Web protocol

“Now, this is just a simulation of what the blocks will look like once they’re assembled.”
The **complex** Web protocol

Complex but standardized browser APIs

“Now, this is just a simulation of what the blocks will look like once they’re assembled.”
The complex Web protocol

Complex but standardized browser APIs

Brittle API implementations

"Now, this is just a simulation of what the blocks will look like once they're assembled."
The complex Web protocol

Complex but standardized browser APIs

Brittle API implementations

Applications fail differently on different browsers
The *complex* Web protocol

**Hard to write secure and robust Web applications**

Complex but standardized browser APIs

Applications fail differently on different browsers
Example: IE’s Event interface bug

Security vulnerability in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized
Example: IE’s Event interface bug

Security vulnerability in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized

Virtual keyboards and keypads are no longer safe
Example: IE’s Event interface bug

Skype

IE running exploit code
Example: IE’s Event interface bug

Security vulnerability in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized

An attacker can access all mouse movements simply by displaying ads on Web pages
Example: IE’s Event interface bug

Security vulnerability in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized

- YouTube and NYTimes potential attack vectors
- Bug already exploited by two ad analytics firms
Example: IE’s **Event** interface bug

**Security vulnerability** in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized

- IE’s DOM implementation populates the **Event** object with mouse events & attributes
- JavaScript can poll for mouse coordinates, **but allowed only for pages in focus with cursor in it**
Example: IE’s Event interface bug

Security vulnerability in IE v6-10, allows mouse cursor to be tracked anywhere on the screen even if IE is minimized

- IE’s DOM implementation populates the Event object with mouse events & attributes
- JavaScript can poll for mouse coordinates, but allowed only for focused pages with cursor in it

Hard to write secure and robust Web applications
Monolithic v/s Exokernel

User application

Exokernel

Device drivers
Memory manager
Scheduler, ...

Device drivers

Scheduler, ...

Memory manager
Monolithic v/s Exokernel

User application manages complexity
Our solution: Atlantis

Extensibility enhances security & robustness

Web application

- Rendering engine
- DOM
- HTML/CSS parser
- JavaScript runtime

Web application

- Rendering engine
- DOM
- Markup/CSS parser
- Scripting runtime

Atlantis

March 13, 2013
Mohan Dhawan
Atlantis architecture

Scripting Runtime  | Layout and Rendering  | Markup Parser  | DOM Tree
---|---|---|---
Syphon Interpreter

UI  | Network

Principal Instance Creation  | Cross-PI Messaging  | Storage  | Device Server
Atlantis architecture

Web page with 3 isolation domains

Master Kernel

Scripting Runtime  Layout and Rendering  Markup Parser  DOM Tree

Syphon Interpreter

UI  Network

Principal Instance Creation  Cross-PI Messaging

Storage  Device Server
Atlantis architecture

- Defined by the Web page
  - Scripting Runtime
  - Layout and Rendering
  - Markup Parser
  - DOM Tree

- Per-instance Kernel
  - UI
  - Network

- Master Kernel
  - Principal Instance Creation
  - Cross-PI Messaging
  - Storage
  - Device Server

Web page with 3 isolation domains
Defining the Web stack

<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
  </environment>

Syphon Interpreter
Defining the Web stack

<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
</environment>
Defining the Web stack

<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
  </environment>

compiler.syp

Syphon Interpreter
Defining the Web stack

<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
</environment>
Defining the Web stack

```
<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
</environment>
```
Defining the Web stack

<environment>
   <compiler='http://foo/compiler.syp'>
   <markupParser='http://bar/mParser.js'>
   <runtime='http://baz/runtime.js'>
</environment>

Syphon Interpreter

compiler.syp

mParser.syp

Syphon Interpreter
Defining the Web stack

```xml
<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
  </environment>
```

![Diagram of Syphon Interpreter with compiler.syp and mParser.syp modules]
Defining the Web stack

<environment>
  <compiler='http://foo/compiler.syp'>
  <markupParser='http://bar/mParser.js'>
  <runtime='http://baz/runtime.js'>
</environment>

Syphon Interpreter

compiler.syp  mParser.syp  runtime.syp
Defining the Web stack

High-level application runtime

Layout and Rendering  DOM tree  AJAX library

compiler.syp  mParser.syp  runtime.syp

Syphon Interpreter
Defining the Web stack

High-level application runtime

Syphon Interpreter

Atlantis Kernel
Defining the Web stack

High-level application runtime

Atlantis Kernel APIs
✓ Bitmap rendering
✓ Frame creation and destruction
✓ Cross-frame messaging
✓ Low-level GUI events
✓ Blocking/non-blocking HTTP sockets
Default Web stack

<html>
<title>HTML</title>
<body>
This is HTML!
</body>
</html>

No <environment> tag present
Default Web stack

Syphon Interpreter

HTML

<html>
<title>HTML</title>
<body>
This is HTML!
</body>
</html>

JScompiler.syp  HTML+CSSparser.syp  DOM.syp

Syphon Interpreter
Default Web stack

Common case: Third party extensible Web stack

Syphon Interpreter

JScompiler.syp  HTML+CSSparser.syp  DOM.syp
Extensibility

```html
<div>
  innerHTML
</div>  

≡

"<div>Hello</div>"

```
Extensibility

Enables JavaScript code injection attacks or XSS

innerHTML

"<div> Hello </div>"
Extensibility

```javascript
var comment = document.getElementById("commentBox");
var contentParent = document.getElementById("parent");
contentParent.innerHTML = comment.value;
```
What if this is malicious JavaScript?
Extensibility

```javascript
var comment = document.getElementById("commentBox");
var contentParent = document.getElementById("parent");
contentParent.innerHTML = comment.value;
```

What if this is malicious JavaScript?

**Ability to shim innerHTML and automatically install a sanitizer**

What if this is malicious JavaScript?
Extensibility

var comment = document.getElementById("commentBox");
contentParent.innerHTML = comment.value;

What if this is malicious JavaScript?

Ability to shim `innerHTML` and automatically install a sanitizer

What if this is malicious JavaScript?
Extensibility

Developers own the DOM in Atlantis
Summary: Part II

• **The Web protocol is complex and huge**
  – No individual browser gets it all right

• **JavaScript frameworks, microkernel browsers**
  – Useful but cannot hide all browser quirks or introspect black-box components

• **Atlantis is an exokernel browser**
  – Kernel handles low-level networking, GUI events, bitmap rendering
  – Application defines higher-level abstractions
  – Strong security and powerful extensibility
Conclusion and Future directions
Conclusion

• Modern browsers provide limited extensibility with much less security [ACSAC’09, ECOOP’12a]
• Extensibility is important for developing novel browser-based user applications [IMC’12]
• Transcript [PLAS’10, ECOOP’12b] and Atlantis [SOSP’11] reason about browser extensibility from languages and systems perspectives to provide novel solutions
Future directions

• **Next-generation Web browser**
  – Abstractions for building novel user applications
  – Mobile, Internet-enabled AR/HUD and consumer devices

• **Security & Privacy**
  – Content-based security for web applications
  – Impact of new HTML5 and legacy browser APIs on end-user privacy
Acknowledgements

• **Dissertation advisors**
  – Vinod Ganapathy and Liviu Iftode

• **PhD committee members**
  – Uli Kremer and Kapil Singh

• **External collaborators**
  – Chung-chieh Shan (Indiana University)
  – Vern Paxson (UC Berkeley / ICSI Berkeley)
  – Renata Cruz Teixeira (LIP6 Paris)
  – Christian Kreibich, Mark Allman, Nicholas Weaver (ICSI Berkeley)
  – James Mickens (Microsoft Research Redmond)
  – Úlfar Erlingsson (Google)

• **Discolab members**
  – Aniruddha Bohra, Arati Baliga, Steve Smaldone, Pravin Shankar, Lu Han, Shakeel Butt, Rezwana Karim, Amruta Gokhale, Liu Yang, Nader Boushehrinejadmoradi, and several other past and present members
Thank you.

Contact information

http://paul.rutgers.edu/~mdhawan/
mdhawan@cs.rutgers.edu
Backup slides
Defense against UI redressing attack

var commitSp = true;
var sp = speculate {

... 

document.body.appendChild(overlay);
};

do{
    var obj = sp.getObject(), arg = sp.getArgs();
    switch(sp.getCause()) {
        case "appendChild":
            if (uiRedressing(arg[0])) commitSp = false;
            else obj.appendChild(arg[0]);
            break;
    } /* end switch */
    sp = sp.resume();
} while(sp.isSuspended());

If (commitSp) sp.commit();
Dynamic script loading

3rd party call stack

Execute `newcode.js` in parent context

Callee

Caller

SP delimiter

Execution context for `newcode.js`

document.write(''<script src="newcode.js"></script>'');
Example: Layout and rendering

```html
<html>
  <div width="49.5%">
  </div>
  <div width="50.5%">
  </div>
</html>
```
Normalized execution time v/s IE8

- LocalVarAccess
- GlobalVarAccess
- ObjPropDirect
- ObjPropOnProto
- ObjPropNested
- ObjAllocation
- Math
- FCalAppToNative
- FCalAppToApp

Microbenchmarks

March 13, 2013  
Mohan Dhawan

60