Introduction

- JavaScript is the lingua franca of web applications
- JavaScript is a flexible programming language because of its dynamism
  - dynamic code generation
  - function variability
  - constructor polymorphism
  - dynamic typing and prototyping

JavaScript Security

- The dynamism is a double-edged sword
  - security exploits
- Detecting security vulnerabilities using program analysis techniques
  - integrity: taint analysis

JavaScript Blended Analysis Framework

Design goal: a practical general-purpose combination of dynamic and static analysis capable of capturing the effects of the dynamic features of JavaScript.

Dynamic Features in JavaScript

```javascript
for (n = 1; n < 20; n++) {
    xe = "s.prop" + n + "=myUe(s.prop" + n + ");
    ex = "s.eVar" + n + "=myCp(s.prop" + n + "," + "eC" + n + ");
    to = "typeof(s.prop" + n + ")";
    if (eval(to) != "undefined") {
        eval(xe);
        eval(ex);
    }
}
```

Figure 1: dynamic code generation (eval) example

```
function(){
    if (arguments.length==2){
        evt=null;
        L=arguments[0];
        }else{
        evt=arguments[0];
        L=arguments[1];
        }
        J(evt,L)
}
```

Figure 2: variadic function example from www.linkedin.com

Benchmarks

<table>
<thead>
<tr>
<th>Website</th>
<th>Page count</th>
<th>Trace count</th>
</tr>
</thead>
<tbody>
<tr>
<td>facebook.com</td>
<td>27</td>
<td>62</td>
</tr>
<tr>
<td>google.com</td>
<td>22</td>
<td>55</td>
</tr>
<tr>
<td>youtube.com</td>
<td>15</td>
<td>30</td>
</tr>
<tr>
<td>yahoo.com</td>
<td>30</td>
<td>69</td>
</tr>
<tr>
<td>wikipedia.org</td>
<td>27</td>
<td>65</td>
</tr>
<tr>
<td>amazon.com</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>twitter.com</td>
<td>32</td>
<td>53</td>
</tr>
<tr>
<td>blogspot.com</td>
<td>9</td>
<td>17</td>
</tr>
<tr>
<td>linkedin.com</td>
<td>32</td>
<td>54</td>
</tr>
<tr>
<td>msn.com</td>
<td>13</td>
<td>21</td>
</tr>
<tr>
<td>ebay.com</td>
<td>40</td>
<td>72</td>
</tr>
<tr>
<td>bing.com</td>
<td>7</td>
<td>14</td>
</tr>
<tr>
<td><strong>totals</strong></td>
<td><strong>263</strong></td>
<td><strong>525</strong></td>
</tr>
</tbody>
</table>

Blended vs. Static Analysis

Experimental Results

Future Work

- Handle other JavaScript dynamic features
  - dynamic typing and prototyping
- Explore more client problems
  - program understanding
- Apply blended analysis to other dynamic programming languages

References:

This work is funded by IBM Open Collaborative Research Program and National Science Foundation CCF-0811518.