Practical Blended Taint Analysis for JavaScript

PROLANGS@VT

VirginiaTech

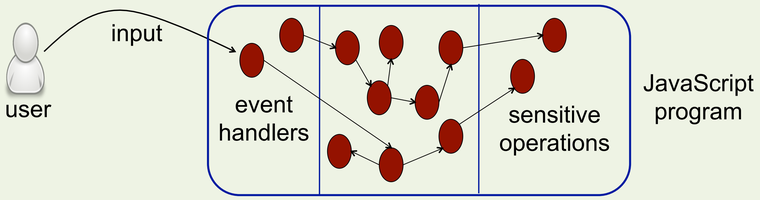
Shiyi Wei and Barbara G. Ryder
Department of Computer Science
Virginia Tech

Introduction

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

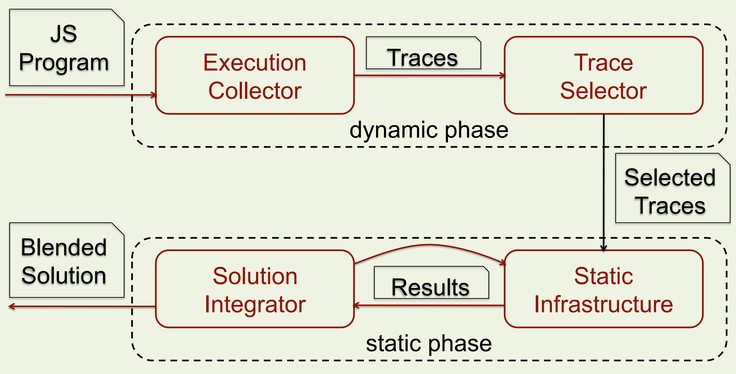
JavaScript Security

- 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

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

Figure 1: dynamic code generation (*eval*) example

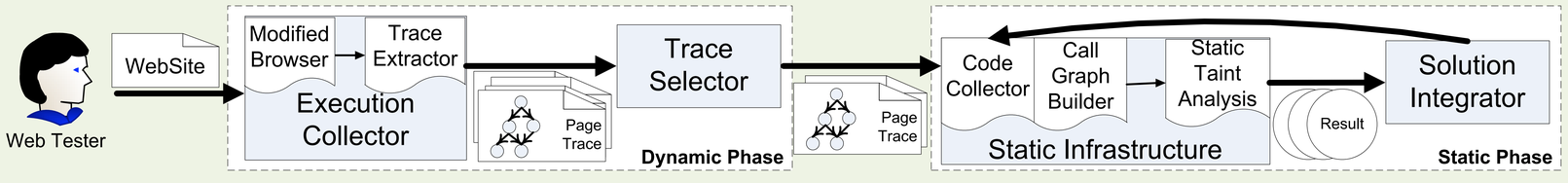
```
function(){
   if(arguments.length===1){
      evt=null;
      L=arguments[0]
   }else{
      evt=arguments[0];
      L=arguments[1]
   }
   J(evt,L)
}
```

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

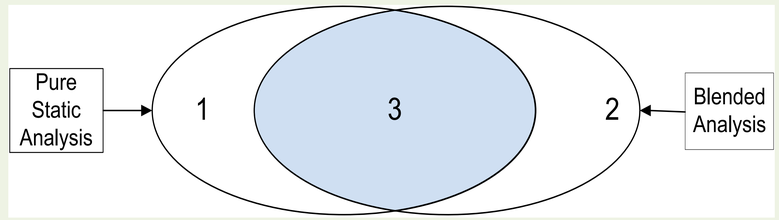
Benchmarks

Website	Page count	Trace count
facebook.com	27	62
google.com	22	55
youtube.com	15	30
yahoo.com	30	69
wikipedia.org	27	65
amazon.com	9	13
twitter.com	32	53
blogspot.com	9	17
linkedin.com	32	54
msn.com	13	21
ebay.com	40	72
bing.com	7	14
totals	***263***	***525***

Blended Taint Analysis for JavaScript Web Applications



Blended vs. Static Analysis



Experimental Results

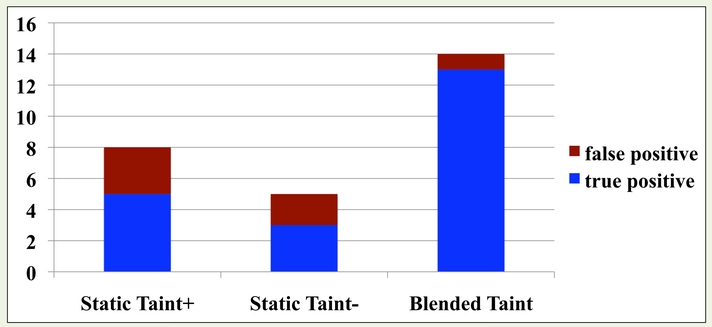


Figure 3: taint analysis results

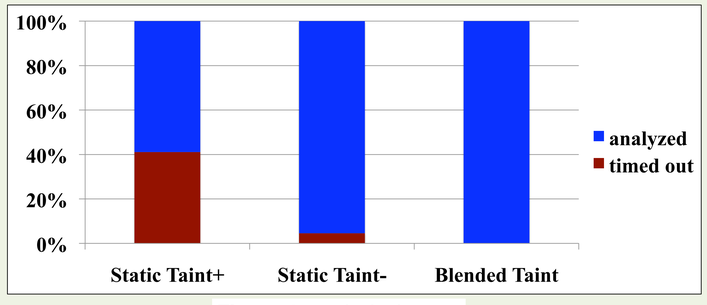


Figure 4: analysis time

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:
[1] Shiyi Wei and Barbara G. Ryder. Practical blended taint analysis for JavaScript. In proceedings of the 2013 International Symposium on Software Testing and Analysis (ISSTA), Pages 336-346.

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