

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

JavaScript Analysis Challenges

Dynamic characteristics

- dynamic code generation
- function variadicity
- constructor polymorphism
- dynamic typing
- prototype-based inheritance

Software engineering challenges

- program understanding
- security
- optimization

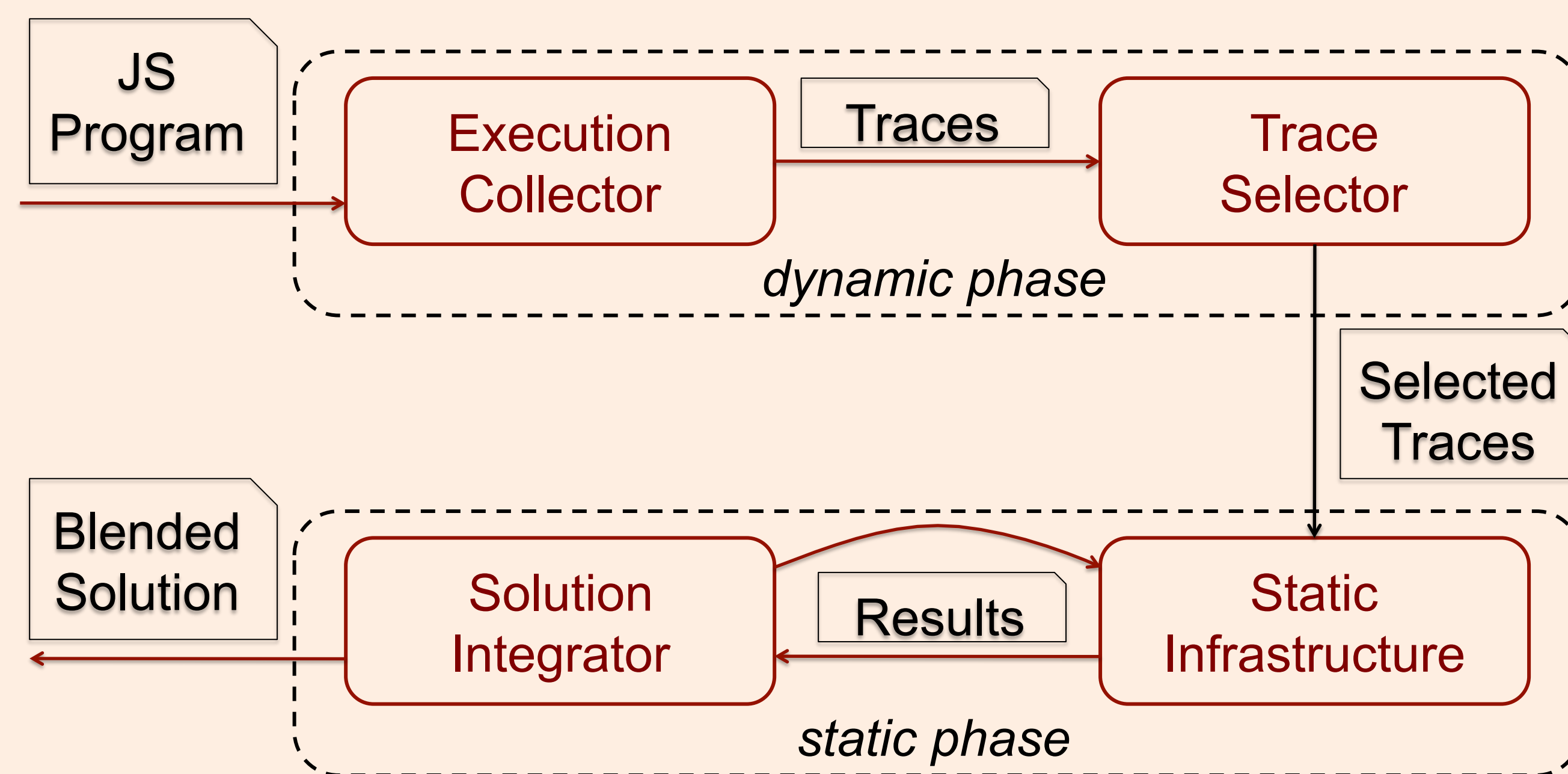
Examples of key features

- reflective mechanisms:
 - ♦`for(n=1; n<20; n++) {`
 - `xe= "s.prop" + n + "=myUe(s.prop" + n + ")";`
 - `eval(xe);}`
 - ♦other forms of eval: `Function`, `setTimeout`, `Write`, etc.
 - ♦`setTimeout('sendRequest(' + action + ', ' + validate + ')', 1000);`
- dynamic object behavior
 - ♦a JavaScript object == {local property | inherited property}
 - ♦`v.p = new A(); → delete v['p']; //v.p is undefined → v.p = new C();`

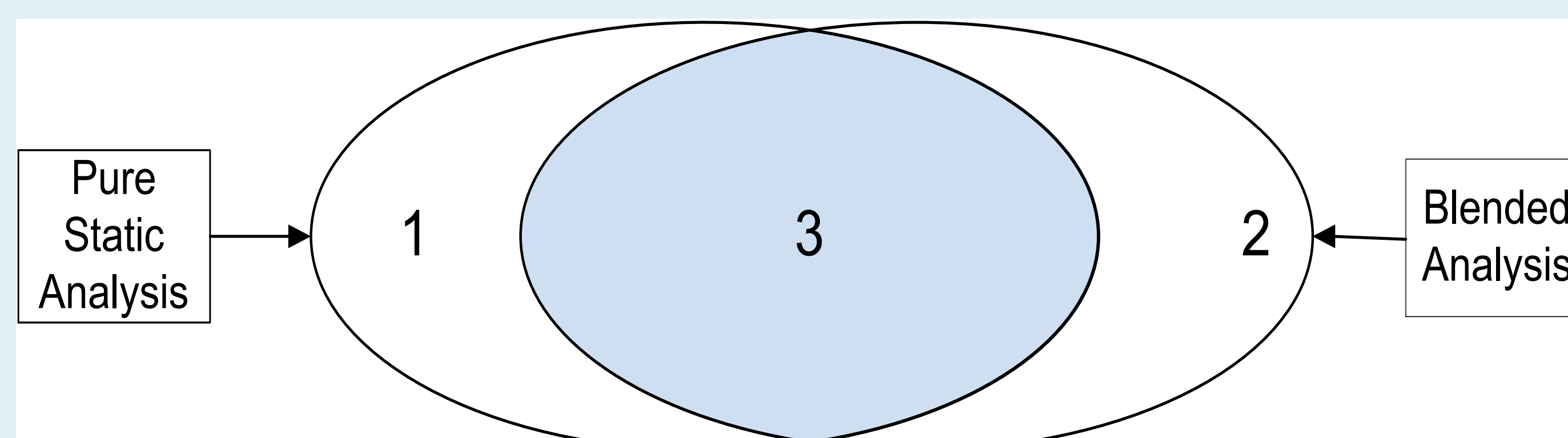
Our Solution for eval:

JavaScript Blended Analysis Framework^[1]

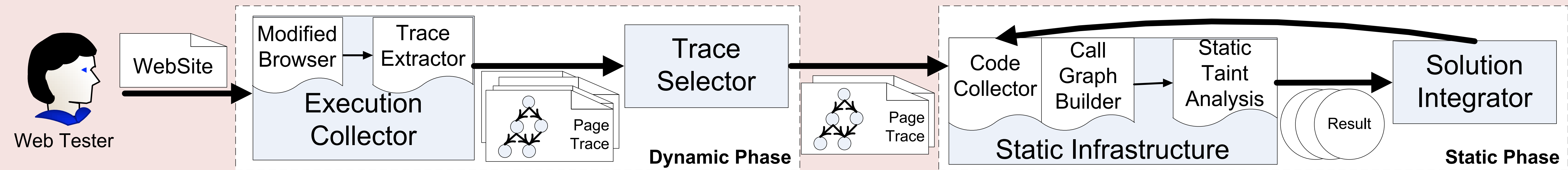
Design goal: a practical general-purpose **combination of dynamic and static analysis** capable of capturing the effects of the dynamic features of JavaScript.



Blended vs. Static Analysis

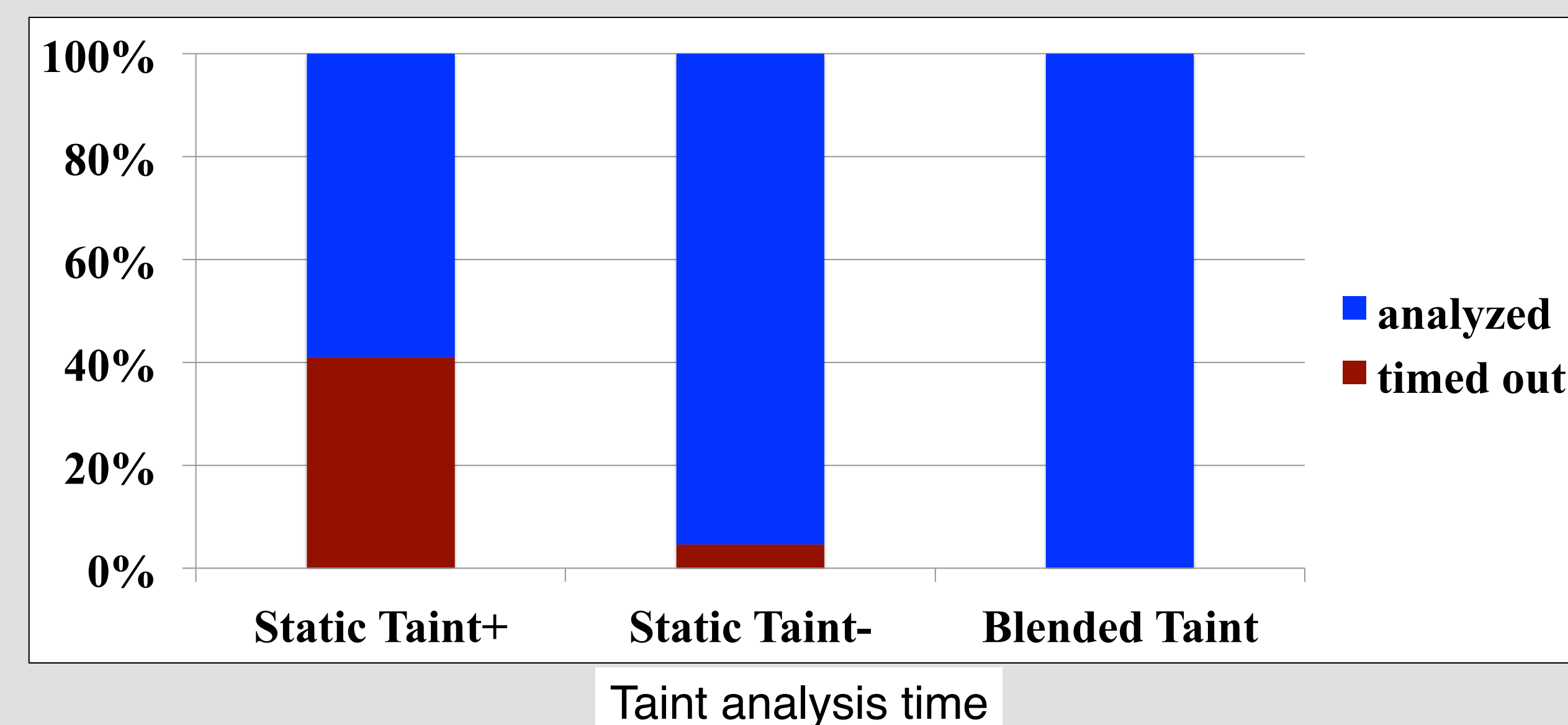
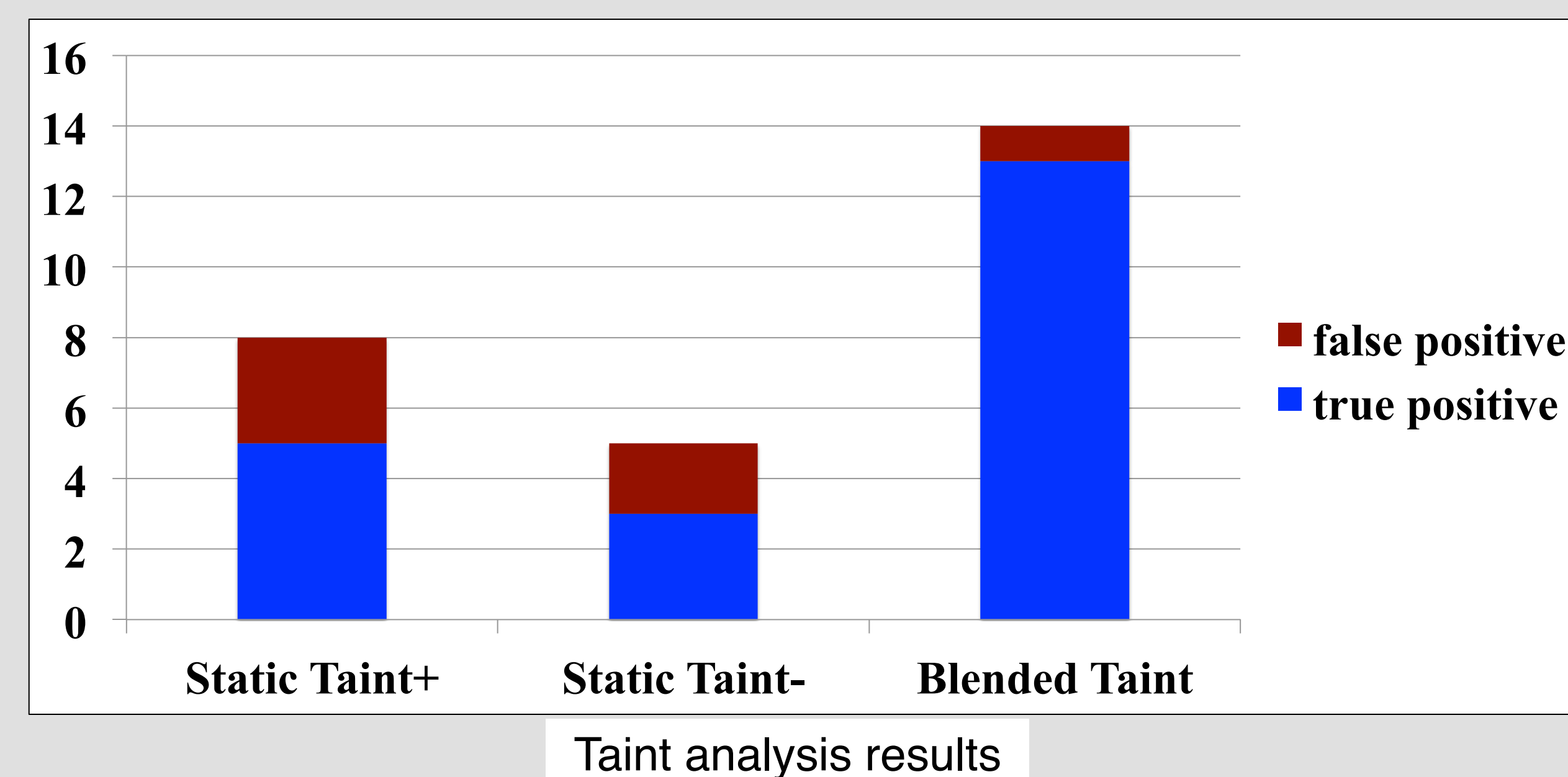


Security Application: Blended Taint Analysis for JavaScript Websites



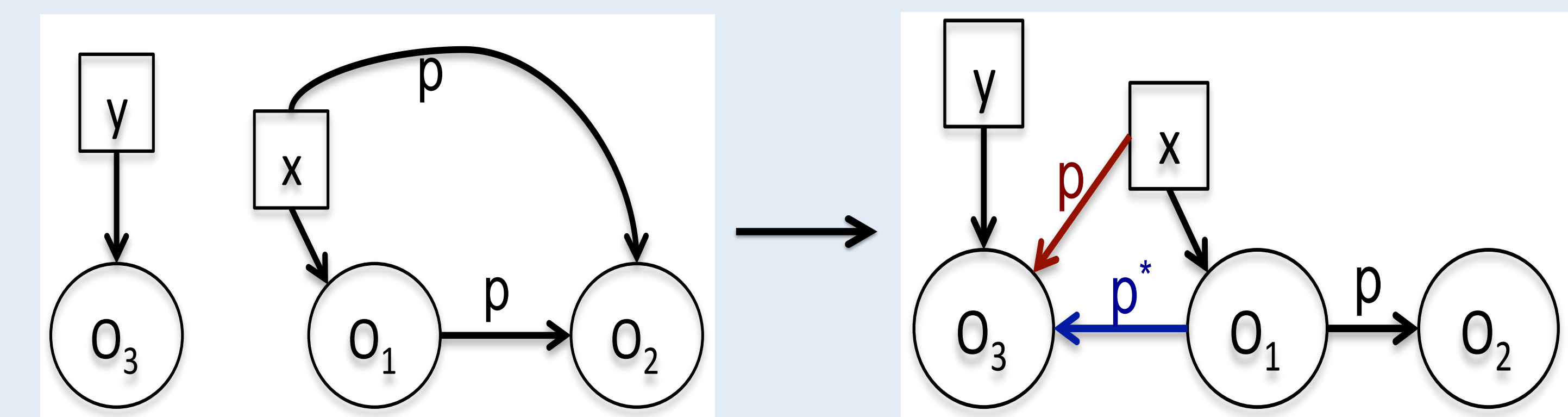
Blended Taint Analysis Results

- Static Taint+: JavaScript library + application code
- Static Taint-: application code only



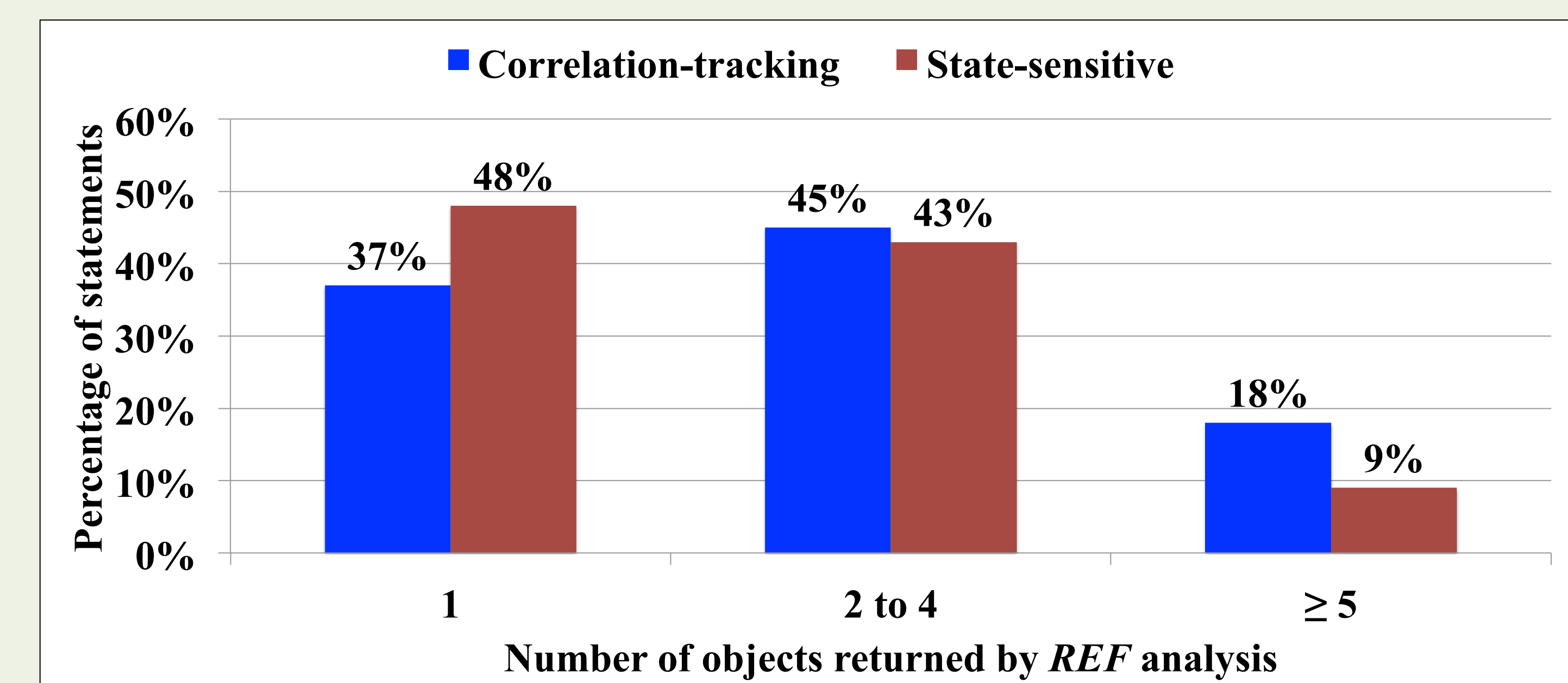
Our Solution for Dynamic Object Behavior: State-sensitive Points-to Analysis^[2]

- Partially flow-sensitive analysis via State-Preserving Block Graph (SPBG)
- Calling context: an approximation of the object state of the receiver object
- Extended points-to graph with annotations
- Points-to analysis transfer functions
 - ♦e.g., $x.p = y$



Program Understanding Application: REF Analysis Results

- REF analysis calculates the set of objects returned by property lookup at a property read statement (i.e., $x = y.p$) or call statement (i.e., $x = y.p(...)$).
- The results of REF analysis can be used for building software IDEs supporting smart code completion.
- Comparison: correlation-tracking^[3] vs. state-sensitive analysis



Benchmark Websites

- | | | |
|----------------|---------------|---------------|
| •facebook.com | •google.com | •ebay.com |
| •youtube.com | •yahoo.com | •bing.com |
| •wikipedia.org | •amazon.com | •linkedin.com |
| •twitter.com | •blogspot.com | •msn.com |

References:

- [1] Shiyi Wei and Barbara G. Ryder. Practical blended taint analysis for JavaScript. *ISSTA* 2013.
- [2] Shiyi Wei and Barbara G. Ryder. State-sensitive points-to analysis for the dynamic behavior of JavaScript objects. *ECOOP* 2014.
- [3] Manu Sridharan, Julian Dolby, Satish Chandra, Max Schafer, Frank Tip. Correlation tracking for points-to analysis of JavaScript. *ECOOP* 2012.