

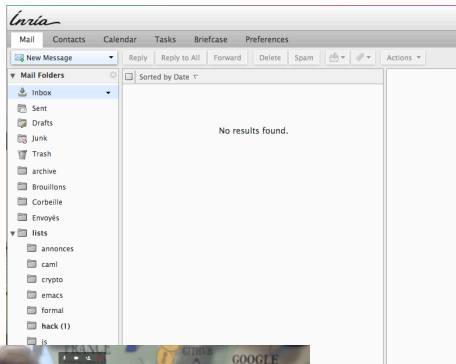


SecCloud

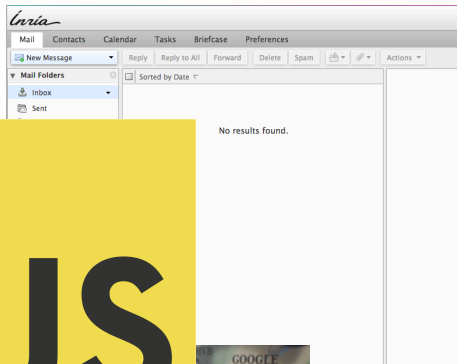
Alan Schmitt

May 28, 2018

Web Applications are like Magic



Web Applications are like Magic



JS



Our Data



The Browser

Programs from the Web



direct flow

```
var public = secret
```

indirect flow

```
if (secret) {  
  public = true  
} else {  
  public = false  
}
```

Non interference is a hyperproperty

```
/* Source */  
var x = true  
var y = true  
if (secret) {  
  x = false  
}  
if (x) {  
  y = false  
}  
public = y
```

```
/* assume secret is true */  
var x = true  
var y = true  
if (secret) {  
  x = false  
}  
  
public = y
```

```
/* assume secret is false */  
var x = true  
var y = true  
  
if (x) {  
  y = false  
}  
public = y
```

A comprehensive language-based approach to the definition, analysis, and implementation of secure applications developed using JavaScript.

1. formal semantics of JavaScript
2. static and dynamic analyses
3. preventive information flow control

JSCert

Correctness

JSRef



Coq world

“real” world



Ocaml extraction

Parser

BISECT

ECMAScript Language test262

Tests To run: 2782 | Total tests on: 2782 | Pass: 2757 | Fail: 25 | Failed to load: 0

Chapter --ch02 (201 tests)	Pass
Chapter --ch03 (206 tests)	Pass
Chapter --ch04 (94 tests)	Pass
Chapter --ch05 (3058 tests)	Pass

Lessons from JSCert



Hard to keep pace with the standardisation

JSCert inductive definition is too big

A implementation close to the spec is very useful

ECMA2017
(English prose)



JSCert
(Coq, inductive)



New JSRef
(OCaml, recursive)



JSExplain
(JS, recursive)



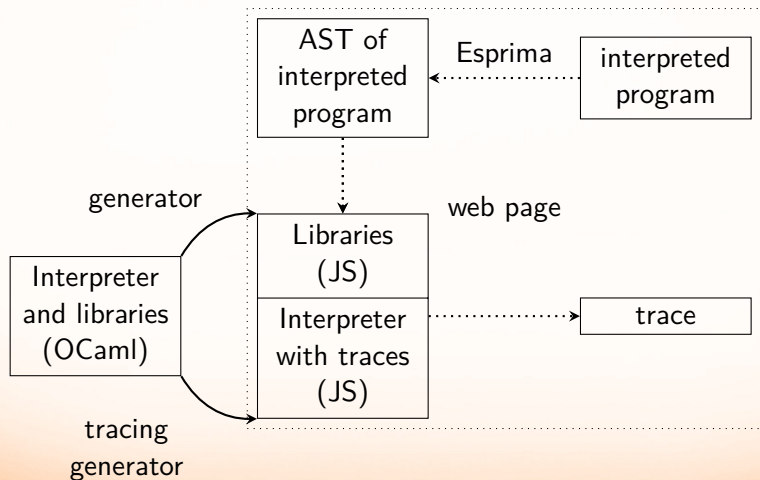
Proofs

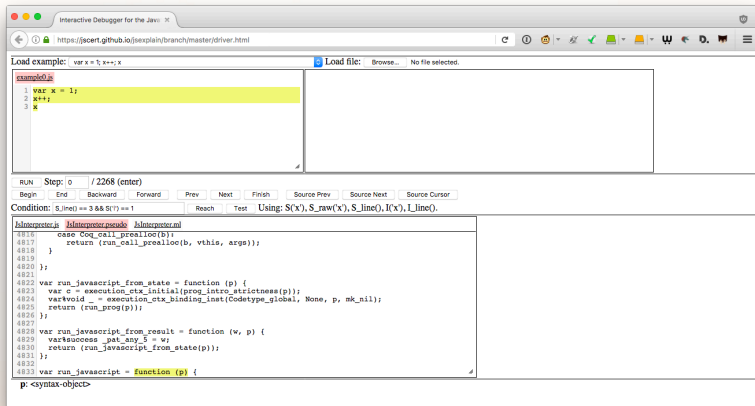


Testing



Debugging





Interactive Debugger for the Java VM

https://jscert.github.io/jsexplain/branch/master/driver.html

Load example: var x = 1; x++; x

Load file: Browse... No file selected.

```
example0.js
1 var x = 1;
2 x++;
3 #
```

RUN Steps: 0 / 2268 (enter)

Begin End Backward Forward Prev Next Finish Source Prev Source Next Source Cursor

Condition: S[line] == 3 && S[?] == 1 Reesh Test Using: S(x'), S_raw(x'), S_line(), l(x'), L_line().

```
js/interpreter.js js/interpreter.pseudo js/interpreter.ml
4816 case Coq_call_prealloc(b);
4817   return (run_call_prealloc(b, vthis, args));
4818 }
4819 };
4820 };
4821 };
4822 var run_javascript_from_state = function (p) {
4823   var c = execution_ctx_initial(prog_intro_strictness(p));
4824   var void _ = execution_ctx_binding_inst(Codetype_global, None, p, mk_nil);
4825   return (run_prog(p));
4826 };
4827 };
4828 var run_javascript_from_result = function (w, p) {
4829   var success_pat_any_5 = w;
4830   return (run_javascript_from_state(p));
4831 };
4832 };
4833 var run_javascript = function (p) {
p: <syntax-object>
```

<https://jscert.github.io/jsexplain/branch/master/driver.html>

Static analyses: before running the program

- pro** considers the whole program

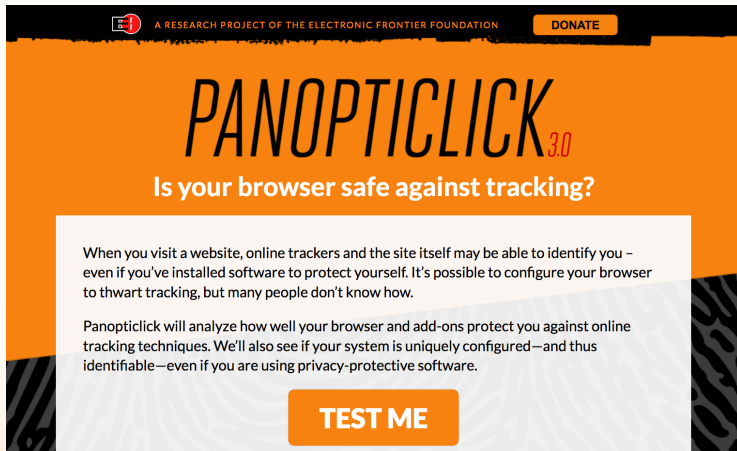
- con** may be less precise


Dynamic analyses: as the program runs

- pro** sees only code that runs, access to exact values

- con** does not capture every information flow

Hybrid analyses: combine both



 A RESEARCH PROJECT OF THE ELECTRONIC FRONTIER FOUNDATION [DONATE](#)

PANOPTICCLICK^{3.0}

Is your browser safe against tracking?

When you visit a website, online trackers and the site itself may be able to identify you – even if you've installed software to protect yourself. It's possible to configure your browser to thwart tracking, but many people don't know how.

Panopticlick will analyze how well your browser and add-ons protect you against online tracking techniques. We'll also see if your system is uniquely configured—and thus identifiable—even if you are using privacy-protective software.

[TEST ME](#)



A RESEARCH PROJECT OF THE ELECTRONIC FRONTIER FOUNDATION

DONATE

Hybrid Information Flow Monitoring Against Web Tracking

Frédéric Besson, Nataliia Bielova, and Thomas Jensen

*Inria
Rennes, France*

When you visit a website, you are tracked, even if you've installed a privacy extension to thwart tracking.

Panopticklick will analyze the tracking technique used to identify you— even if you use a privacy extension.

Browser Randomisation against Fingerprinting: a Quantitative Information Flow Approach

Frédéric Besson, Nataliia Bielova, and Thomas Jensen*

Inria, France

Hybrid Monitoring of Attacker Knowledge

Frédéric Besson, Nataliia Bielova and Thomas Jensen

Inria, France

Given a huge formal semantics, how to prove non-interference?
Solution transform a hyperproperty (of the semantics) into a simple property (of the multi-semantics)

Theorem

If a program is interferent, then there exists a derivation in the annotated multi-semantics that witnesses it.



facets: values with several values (e.g. private and public)
faceted evaluation:

function(x)	x: a true false
y = true	y: true
z = true	z: true
if (x)	PC a
y = false	y: a false true
if (y)	PC \bar{a}
z = false	z: a true false
return z	

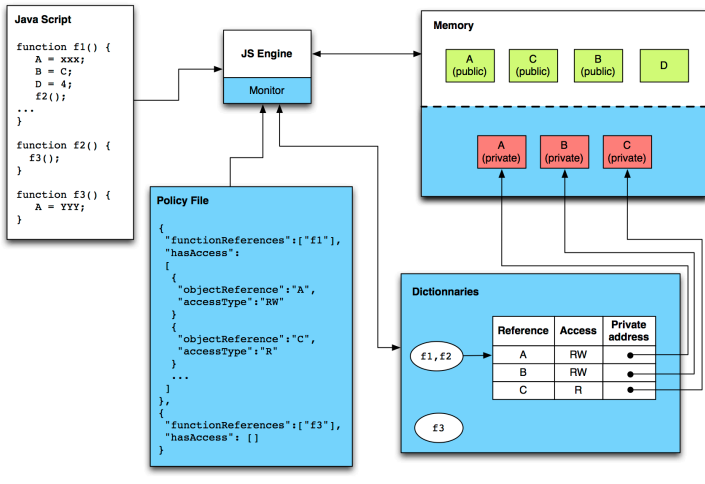
Credits: Florent Marchand de Kerchove

experimentation by extending Narcissus

Split addresses



Change the address of references depending on the execution stack



Implementation in the Chromium V8 engine

Highlights

- ▶ formalization of the full JavaScript language
- ▶ analyses proven in Coq
- ▶ practical tools

Future

- ▶ transfer to TC39
- ▶ usable formalization of JavaScript
- ▶ extension to other languages (Hop.js)