A Study of JavaScript Constructs used in Top Alexa Sites

Dolière Francis SOME
Advisors:
Tamara Rezk
Nataliia Bielova

Assumptions

JS

http://slideshare.net

Motivations

- **Study of real world JavaScript programs**
  - Statistics of constructs
  - Propose a subset
  - Popular librairies
  - Security

Overview

- Collection of real world programs
- Constructs and subset
- Popular libraries
- Security
- Related Work
- Conclusion

Overview

- **Collection of real world programs**
- **Constructs and subset**
- **Popular libraries**
- **Security**
- **Related Work**
- **Conclusion**

Data collection

JS

- **Set of Pages**

```
<!-- Related domain pages and links -->
<a href="A.com/.../page.html" />
<a href="subdomain.A.com/.../page.html" />
<a href="B.com/.../page.html" /> : NOT TAKEN !!!
```

- **Scripts**

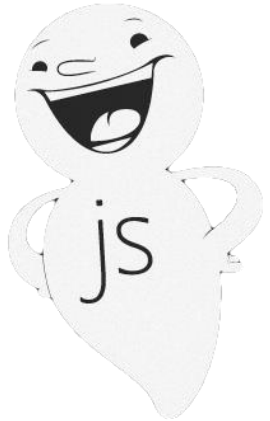
```
<!- Inline scripts -->
<script> var o = {a:"ee", b: 12, ...} ...</script>
<!- Remote scripts -->
<script type="text/javascript" src="C.com/jscript.js">
</script>
```

Tools

- **10,000 Top sites from Alexa**

- **PhantomJS**
  - Headless browser with scriptable Javascript API

- **CasperJS**
  - Navigation scripting for PhantomJS



- **Inria Sophia Cluster**
  - Calculations
  - Storage

Results

#Pages Visited	1,500,000
#Remote Inclusions	21,910,713
#Unique JavaScript librairies	2,352,826
#Unique domains	67,697
#Inline scripts	For each page visited
Remote inclusions size average	12.159 Ko
Biggest remote inclusion file size	11.458 Mo

Overview

- Collection of real world programs
- **Constructs and subset**
- Popular libraries
- Security
- Related Work
- Conclusion

Tools

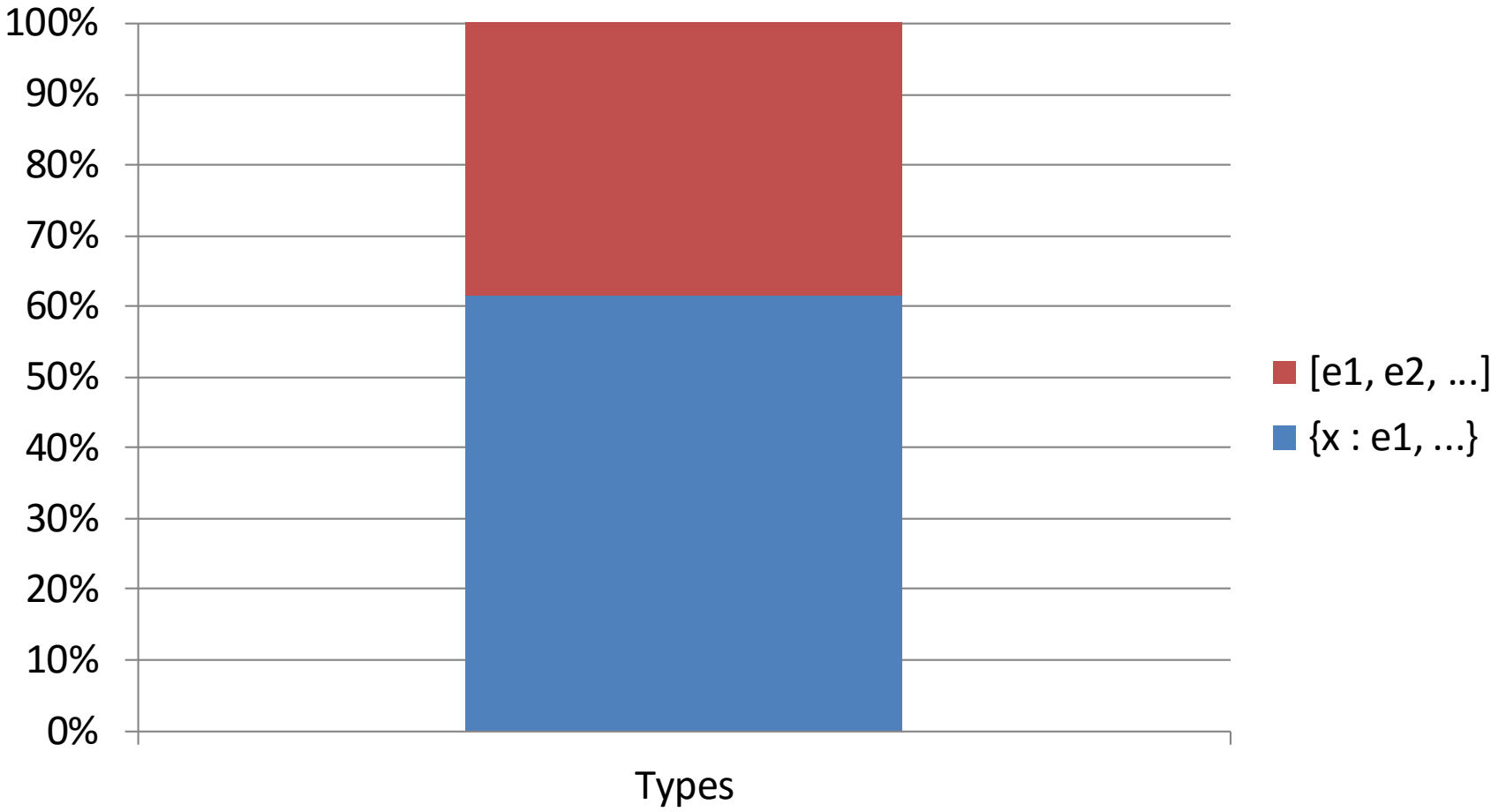
- **Esprima parser**
  - Open source
  - Full support for ECMAScript 6 standard
  - Modified with counters for constructs

http://esprima.org

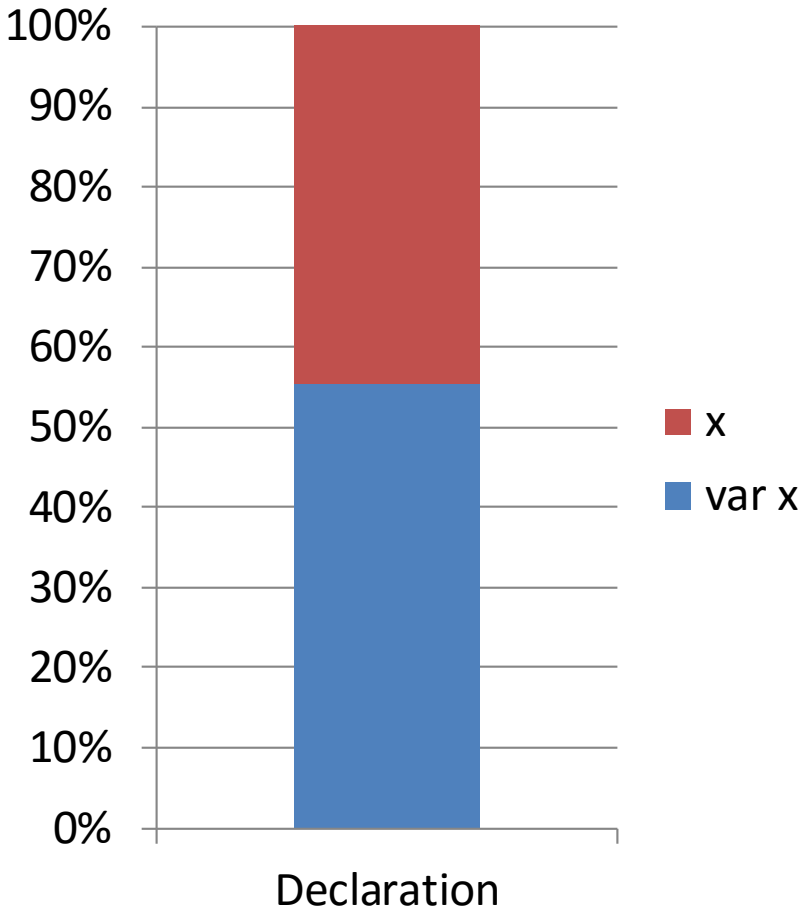
- **Cumulative counting of each construct throughout all JavaScript files**

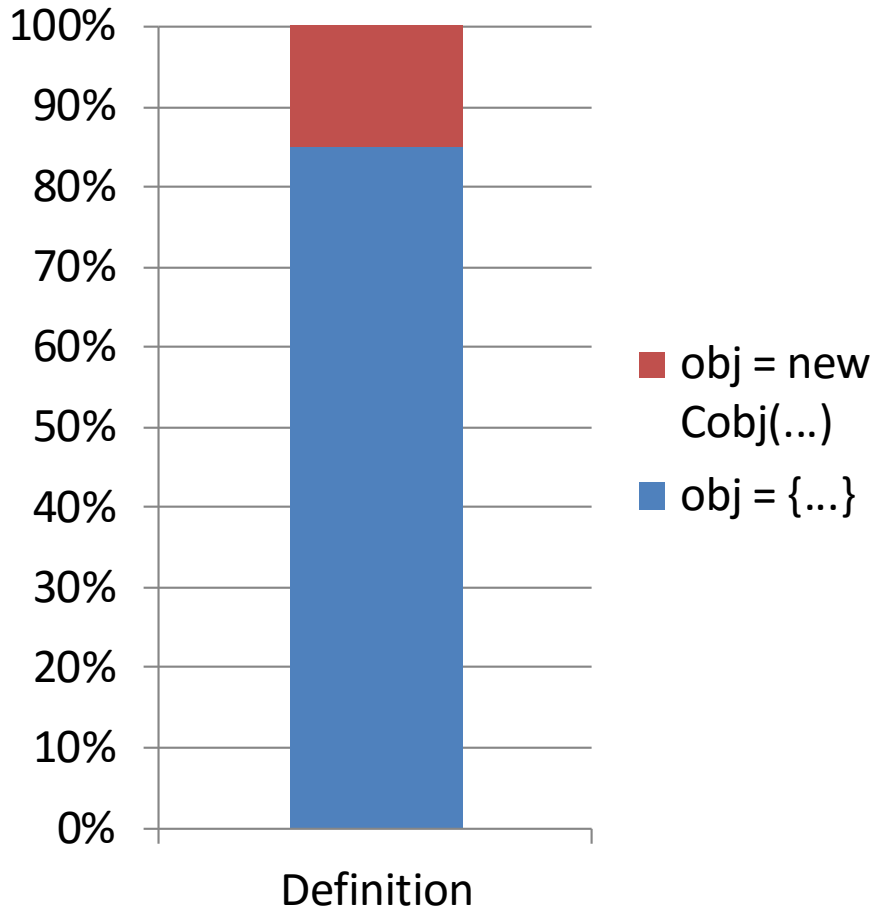
- **Average of occurrence of each construct per JavaScript file**

Objects : types

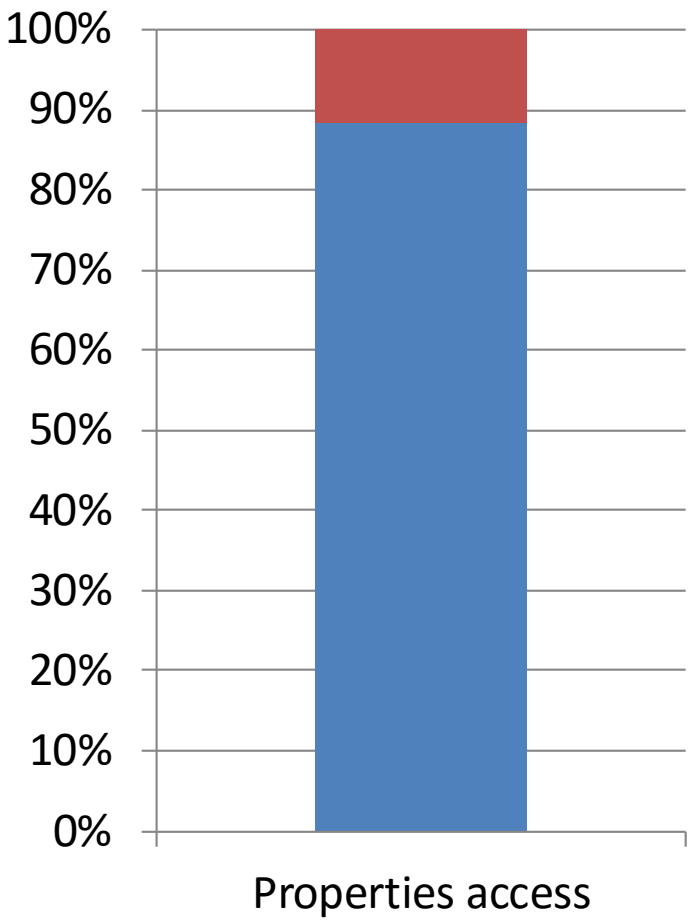


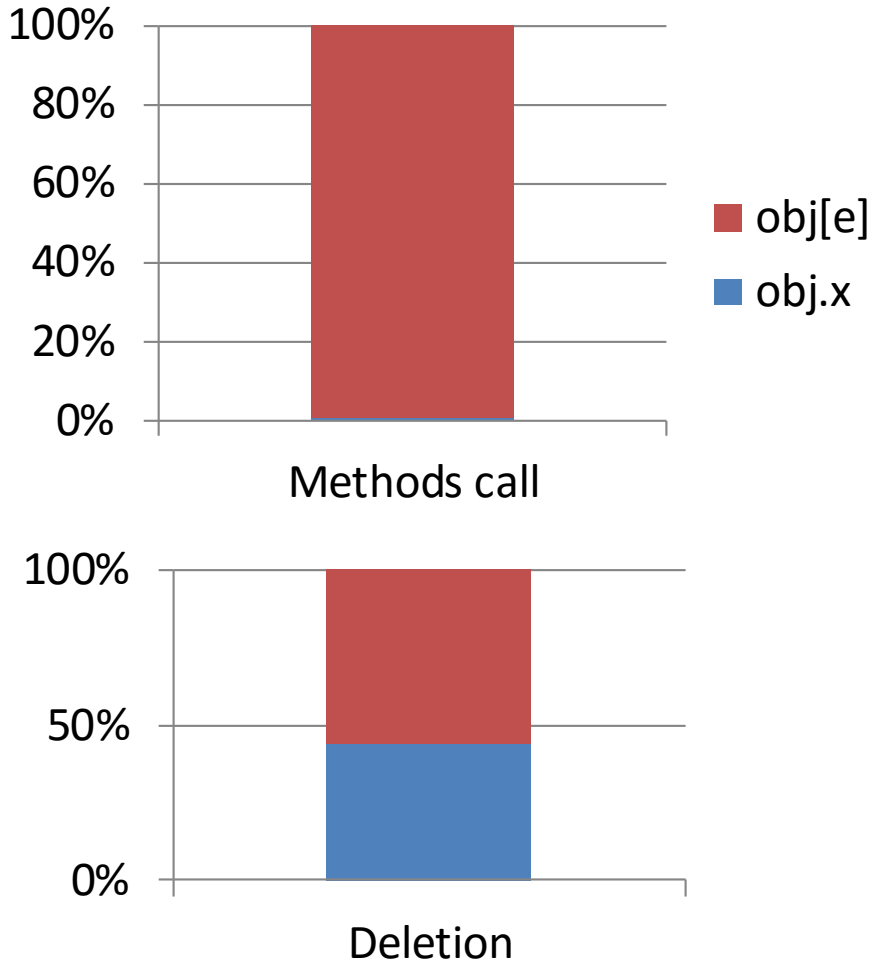
Objects : declaration



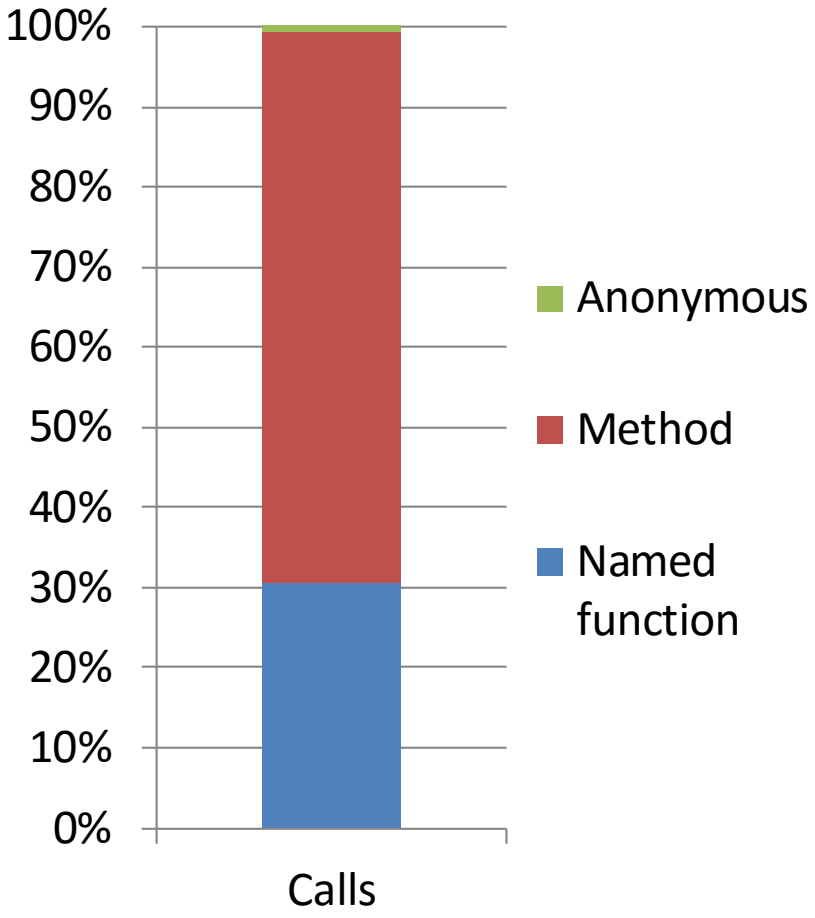


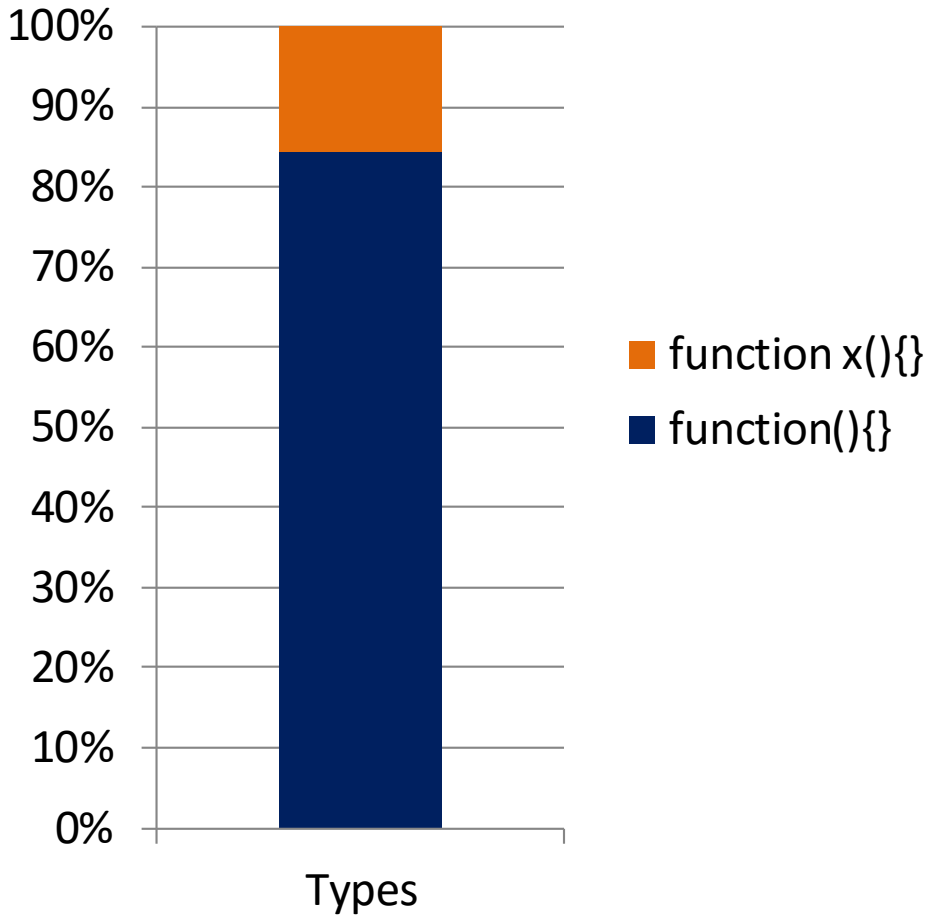
Objects: access



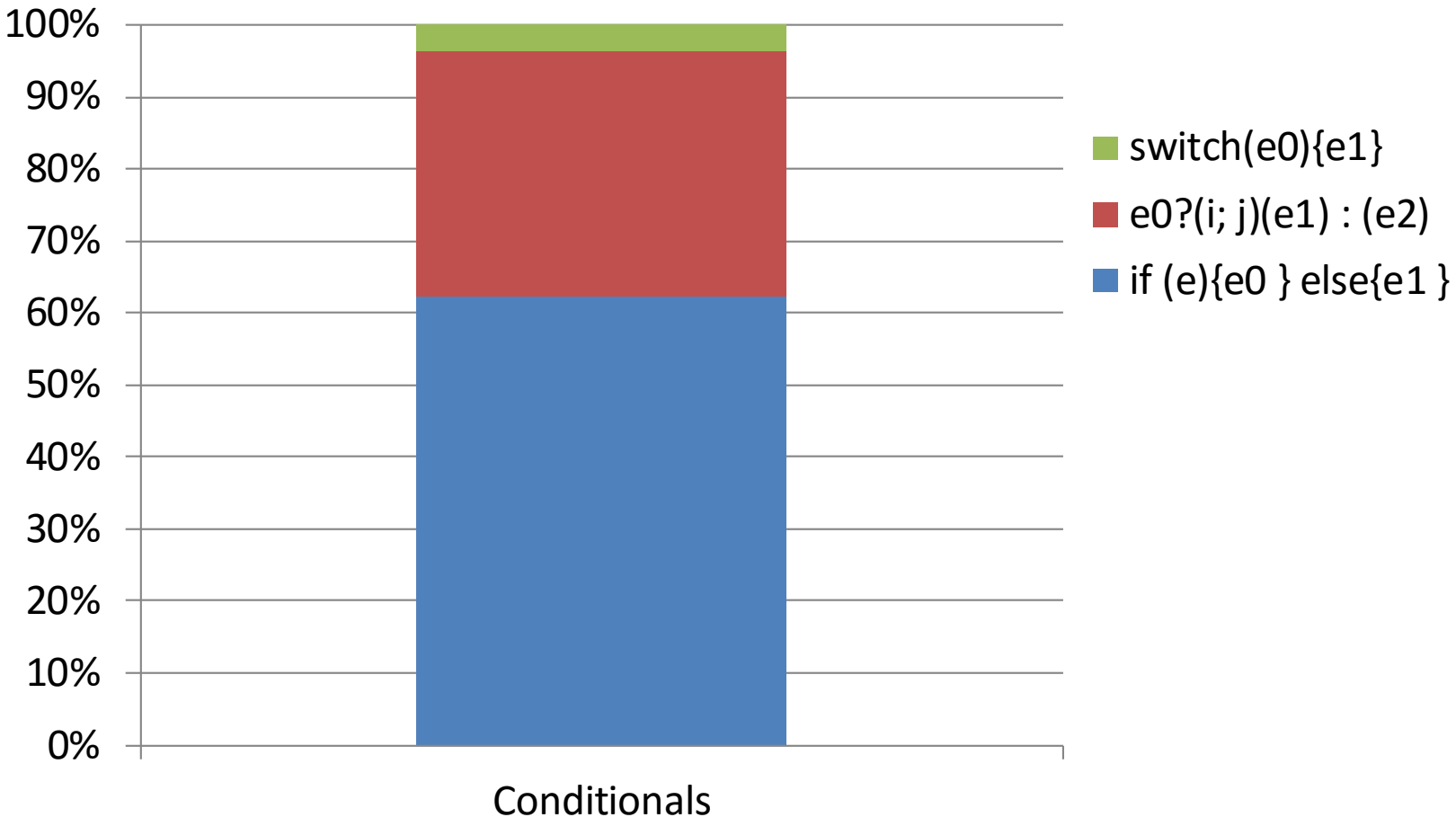


Functions

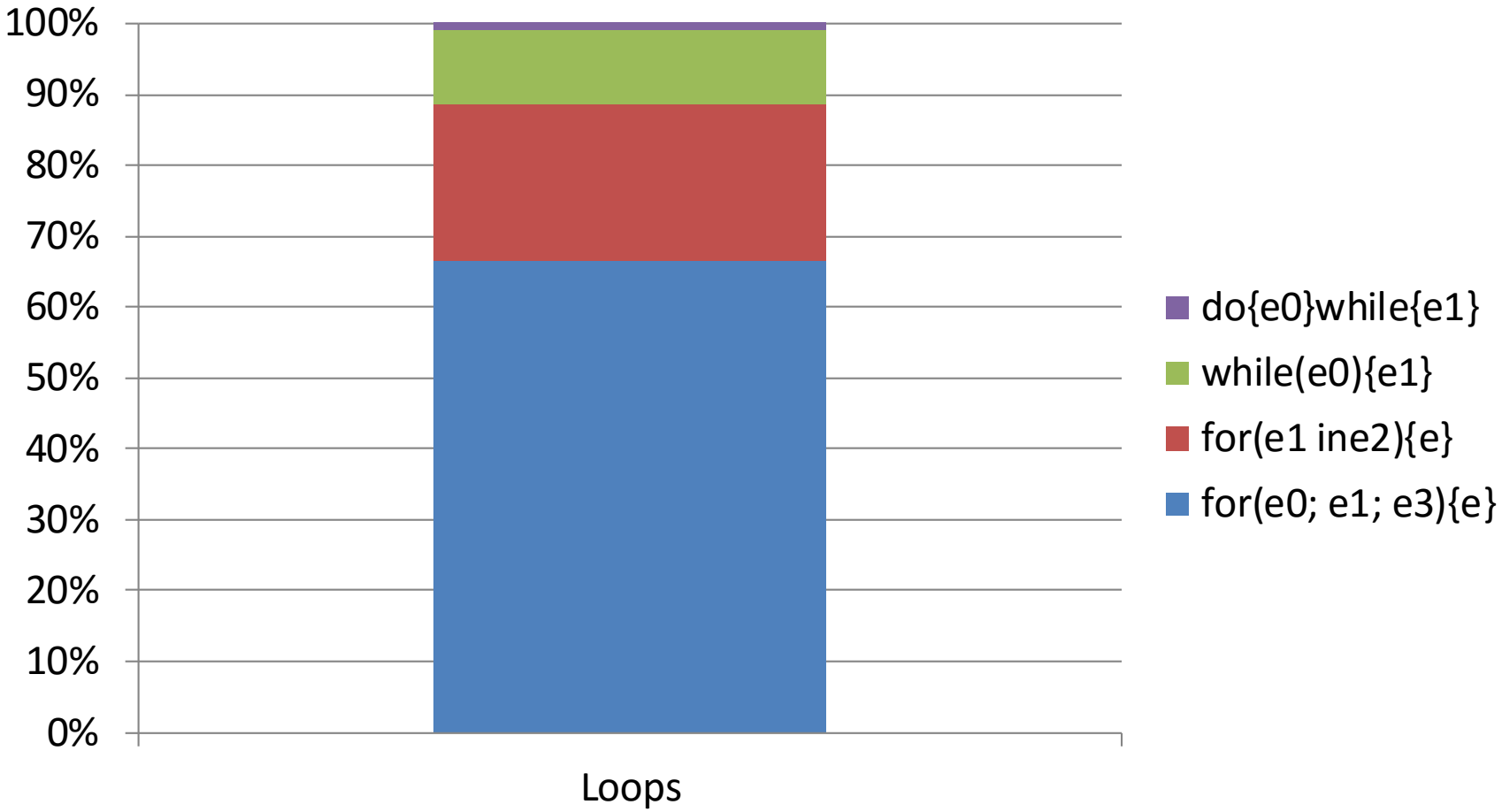

100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Calls
Anonymous
Method
Named function


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Types
function x(){}
function(){}

Conditionals



Loops


100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
Loops
do{e0}while{e1}
while(e0){e1}
for(e1 ine2){e}
for(e0; e1; e3){e}

Subset

JS

$e, e_0, e_1, e_2 \in$ **Expr** ::=

x		**delete** $e[e_1]$
v		e_0 **op** e_1
var $y_1, ..., y_n$		**op** e | e **op**
{}		$x = e$
[]		e_1, e_2
$e_0.e_1$		;
$e_0[e_1]$		**if**(e){e_0} **else**{e_1}
function(e){ e_0}		**while**(e){e_1}
function x(e){ e_0}		{**break**}
this e		**throw** e
return e		**try** {e}
$e_0(e_1)$		**catch** (e_0){e_1}
new e		**eval**(e)

Percentage of coverage

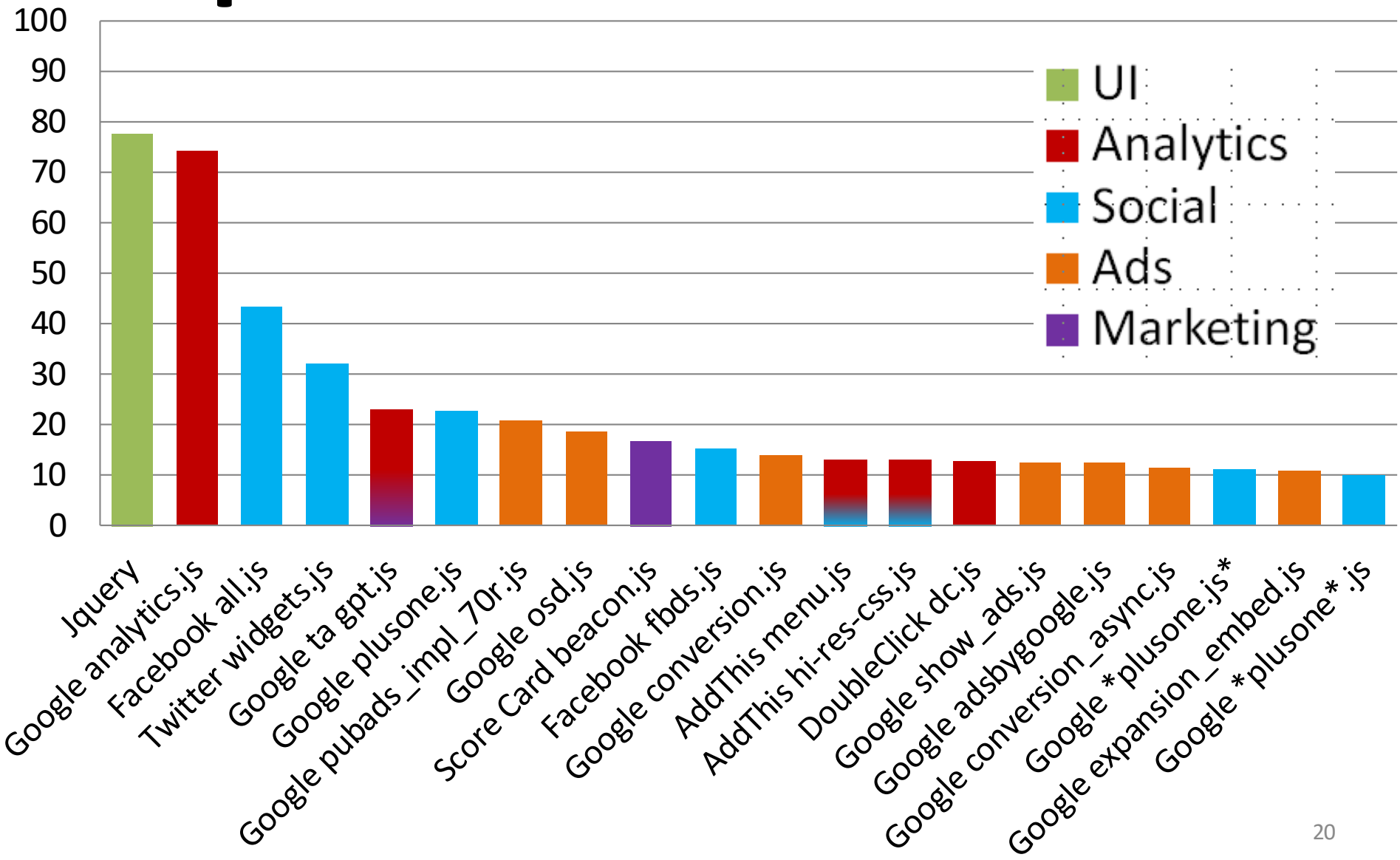
- **98.27% of JavaScript files covered**
- **Simulate some constructs with others**
- **Constructs removed**
  - With statement : 111,838
  - Debugger statement : 3,523
  - TemplateLiteral & TemplateElement : 7
  - Assignment pattern **:** 4
- **Constructs not occuring in our statistics**
  - **ECMAScript 6 (**Class, import, export, super, for of, arrow functions…)

2

Overview

- Collection of real world programs
- Constructs and subset
- **Popular libraries**
- Security
- Related Work
- Conclusion

Popular Libraries



Overview

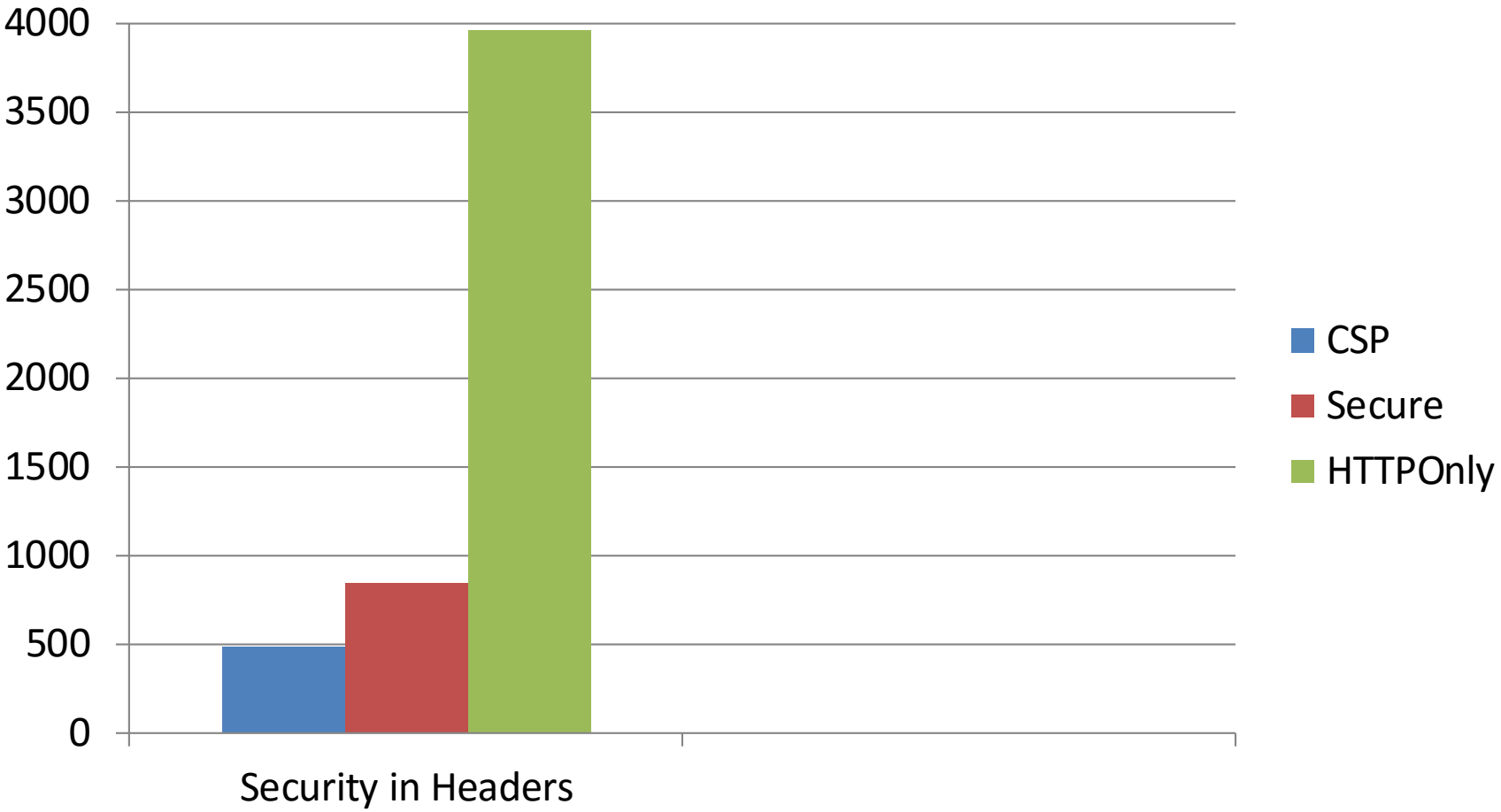
- Collection of real world programs
- Constructs and subset
- Popular libraries
- **Security**
- Related Work
- Conclusion

Security in headers

JS

- **Content Security Policy (CSP)**
  - Complementary to Same Origin Policy
  - Mitigate and report XSS attacks
    - ✓ Restrict domain from which to load contents
    - ✓ Specify allowed protocols
- **Cookies**
  - **Secure cookies**
    - ✓ **Should be sent using HTTPS connection**
  - **HTTPOnly**
    - ✓ **Cookies not accessible through JavaScript**
    - ✓ **Only exchanged through HTTP connection**
- **Combination**

CSP & Cookies


4000
3500
3000
2500
2000
1500
1000
500
0
Security in Headers
CSP
Secure
HTTPOnly

Overview

- Collection of real world programs
- Constructs and subset
- Popular libraries
- Security
- **Related Work**
- Conclusion

Related Work

Results	Nikiforakis et al. [1]	Ours	Variation
#pages	3,300,000	1,500,000	- 54.54%
#remote inclusions	8,439,799	13,803,919	63.557%
#unique remote inclusions	301,968	2,352,826	+7.79 (ratio)
#unique-addressed remote host	20,225	67,697	+3.34 (ratio)
#inclusions with IP addresses	23,063	2,802	- 87.87%
#inclusions with localhost	133	178	+ 33.83%
#Popular libraries	Top 10	Top 20	+50%

[1] Nick Nikiforakis, Luca Inverizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna You are What You Include: Large-Scale Evaluation of Remote JavaSCript Inclusions In Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012), Raleigh, NC, USA

Related Work (cont.)

JS

- [An analysis of the dynamic behavior of JavaScript programs, Richards et al., PLDI 2010]
  - Properties are often added after initialization time
  - Properties are often deleted
  - The use of eval is quite frequent
  - Program size is not modest
  - Execution time is dominated by hot loops

Related Work (cont.)

- [The Eval that Men Do: A Large-scale Study of the Use of Eval –Richards et al., ECOOP'11]
  - What is **eval** used for ?
    - ✓JSON – JSONP
    - ✓Loading libraries
    - ✓Read, assignment, and test type of variables
    - ✓Functions call
  - **Aliased for scoping purpose**

```
function normalEval(str){eval(str);console.log(x);}

function aliasEval(str){var alias = eval; alias(str); console.log(x);}>

var str = "var x = 2;";
```

- `normalEval(str);`
  `2`
- `console.log(x);`
  `Uncaught ReferenceError: x is not defined`
- `aliasEval(str);`
  `2`
- `console.log(x);`

Related Work (cont.)

JS

- [Remedying The Eval that Men Do, Jensen et al. , ECOOP'11]
  - **Alternatives to trivial the use of eval**
    - ✓87% runtime call sites and 75% static eval code arguments fall in
      - ➢JSON parsing
      - ➢Loading libraries
      - ➢Read, assignment, and test type of variables
      - ➢Functions/methods call
  - **Transforming non trivial eval arguments**
    - ✓28 non trivial programming patterns
    - ✓44 calls to eval
    - ✓33 calls eliminated

Related Work (cont.)

JS

- http://trends.builtwith.com
  - **Popular libraries**

	trends.builtwith.com	Our results
JQuery	72.79%	77.871%
Google Analytics	69.70%	74.209%
Facebook Connect	43.60%	43.487%

  - **Content Security Policy**
    - 71 of 10k
    - 421 of 100k
    - 6,252 of 949,457
  - **Results here are base only on sites' homepage**

Overview

- Collection of real world programs
- Constructs and subset
- Popular libraries
- Security
- Related Work
- **Conclusion**

Conclusion

- Large scale crawl of Top 10,000 sites
- Statistics of constructs
- Proposition of representative subset
- Security in HTTP Headers
- State of the art

Discussion

- How to group similar libraries ?
- How to crawl scripts available after login ?

References

- [1] Nick Nikiforakis, Luca Inverizzi, Alexandros Kapravelos, Steven Van Acker, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna You are What You Include: Large-Scale Evaluation of Remote JavaSCript Inclusions In Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS 2012), Raleigh, NC, USA
- [2] Gregor Richards, Sylvain Lebresne, Brian Burg and Jan Vitek. An Analysis of the Dynamic Behavior of JavaScript Programs. In Proceedings of the 2010 ACM SIGPLAN conference on Programming language design and implementation, PLDI '10, pages 1{12, New York, NY, USA, 2010. ACM.
- [3] Gregor Richards, Christian Hammer, Brian Burg and Jan Vitek. The Eval that Men Do. A Large-scale Study of the Use of Eval in JavaScript Applications. Purdue University - University of Washington
- [4] Yuchen Zhou, and David Evans. Why Aren't HTTP-only Cookies More Widely Deployed University of Virginia
- [5] Simon Holm Jensen, Peter A. Jonsson and Anders Møller. Remedying the Eval that Men Do In Procceedings of the International Symposium on Software Testing and Analysis, 2012