Statically Detecting JavaScript Obfuscation and Minification Techniques in the Wild

Marvin Moog[†][*], Markus Demmel[†], Michael Backes[*], and **Aurore Fass[*]**

[†] Saarland University

[*] CISPA Helmholtz Center for Information Security

DSN 2021

Motivation - JavaScript

Popular client-side programming language

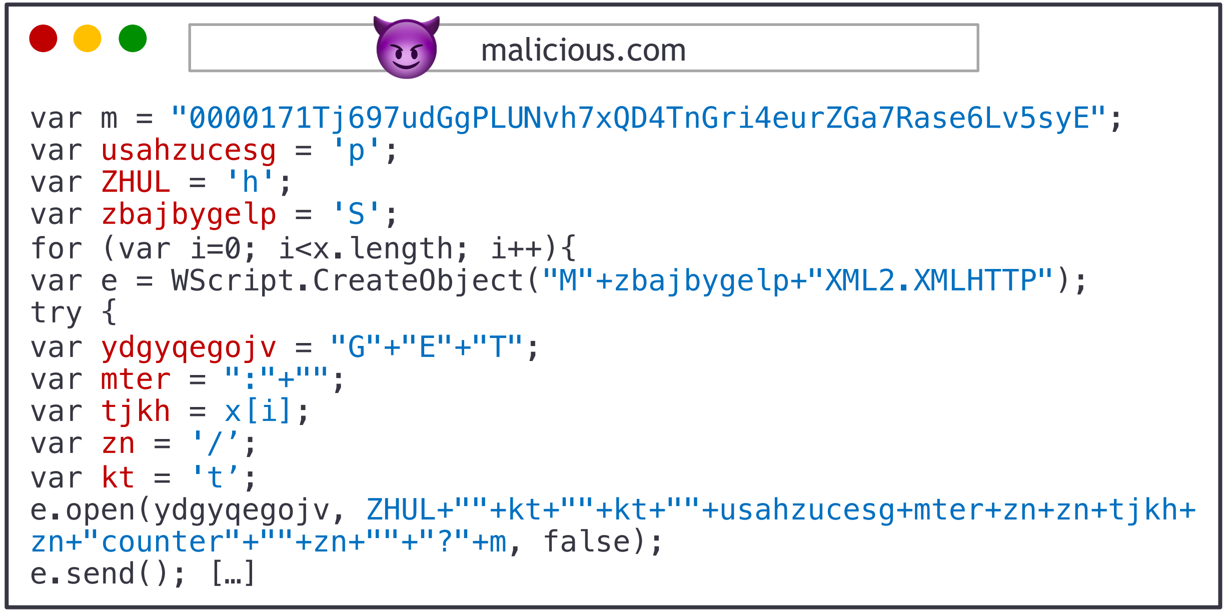
- JavaScript usage ≥ 97% websites [1]

Attack vector

- Aims at harming victims, e.g., exploiting vulnerabilities, stealing sensitive user data
- Code transformations:
  - hide the maliciousness of the code
  - impede its detection

[1] https://w3techs.com

Malicious JavaScript – Code Transformations



```
var m = "0000171Tj697udGgPLUNvh7xQD4TnGri4eurZGa7Rase6Lv5syE";
var usahzucesg = 'p';
var ZHUL = 'h';
var zbajbygelp = 'S';
for (var i=0; i<x.length; i++){
var e = WScript.CreateObject("M"+zbajbygelp+"XML2.XMLHTTP");
try {
var ydgyqegojv = "G"+"E"+"T";
var mter = ":"+"";
var tjkh = x[i];
var zn = '/’;
var kt = 't’;
e.open(ydgyqegojv, ZHUL+""+kt+""+kt+""+usahzucesg+mter+zn+zn+tjkh+
zn+"counter"+""+zn+""+"?"+m, false);
e.send(); […]
```

Randomization obfuscation

Data obfuscation

[2] Kaplan et al., MSR-TR 2011

[3] Skolka et al., WWW 2019

Motivation - JavaScript

Popular client-side programming language

- JavaScript usage ≥ 97% websites [1]
- Code transformations:
  - optimize website performance (e.g., save bandwidth / reduce loading times)
  - protect code privacy and intellectual property

Attack vector

- Aims at harming victims, e.g., exploiting vulnerabilities, stealing sensitive user data
- Code transformations:
  - hide the maliciousness of the code
  - impede its detection

[1] https://w3techs.com

JavaScript Code Transformations

Minification

- Aim: reducing code size
  - minification simple (e.g., shortening variable names, deleting whitespaces)
  - minification advanced (e.g., function inlining, conditional operator)

[4] https://javascript-minifier.com
[5] https://developers.google.com/closure/compiler

Obfuscation

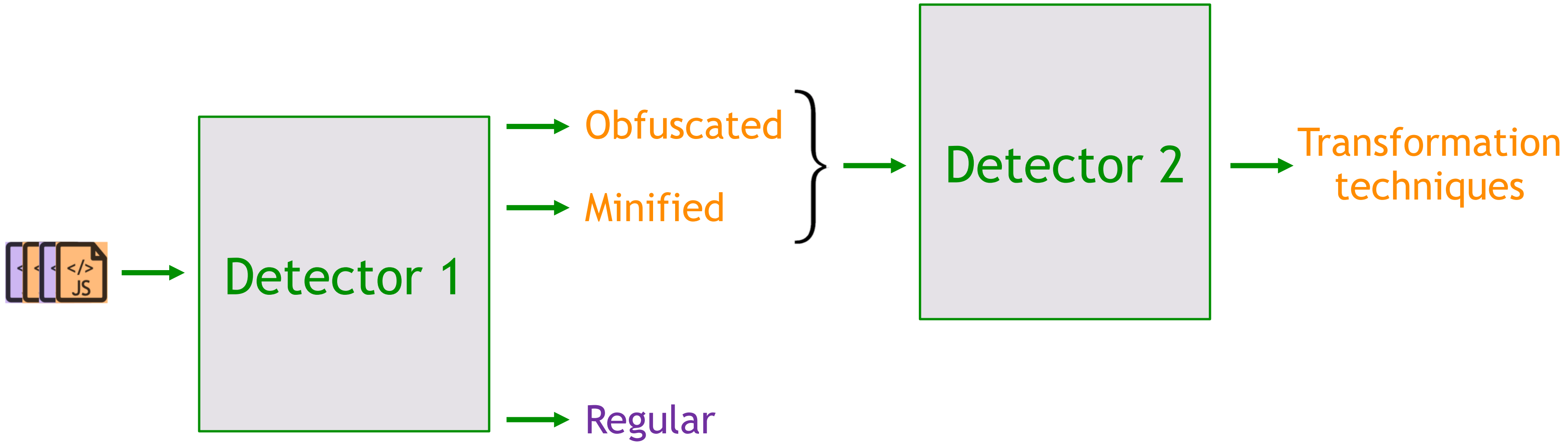
- Aim: hindering code analysis
  - identifier obfuscation
  - string obfuscation
  - global array
  - no alphanumeric
  - dead-code injection
  - control-flow flattening
  - self-defending
  - debug protection

[6] https://obfuscator.io
[7] https://github.com/aemkei/jsfuck
[8] https://github.com/anseki/gnirts

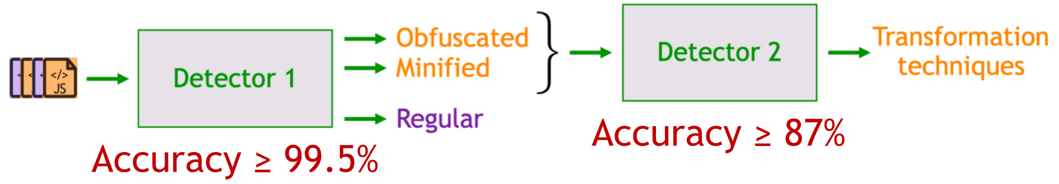
Contribution

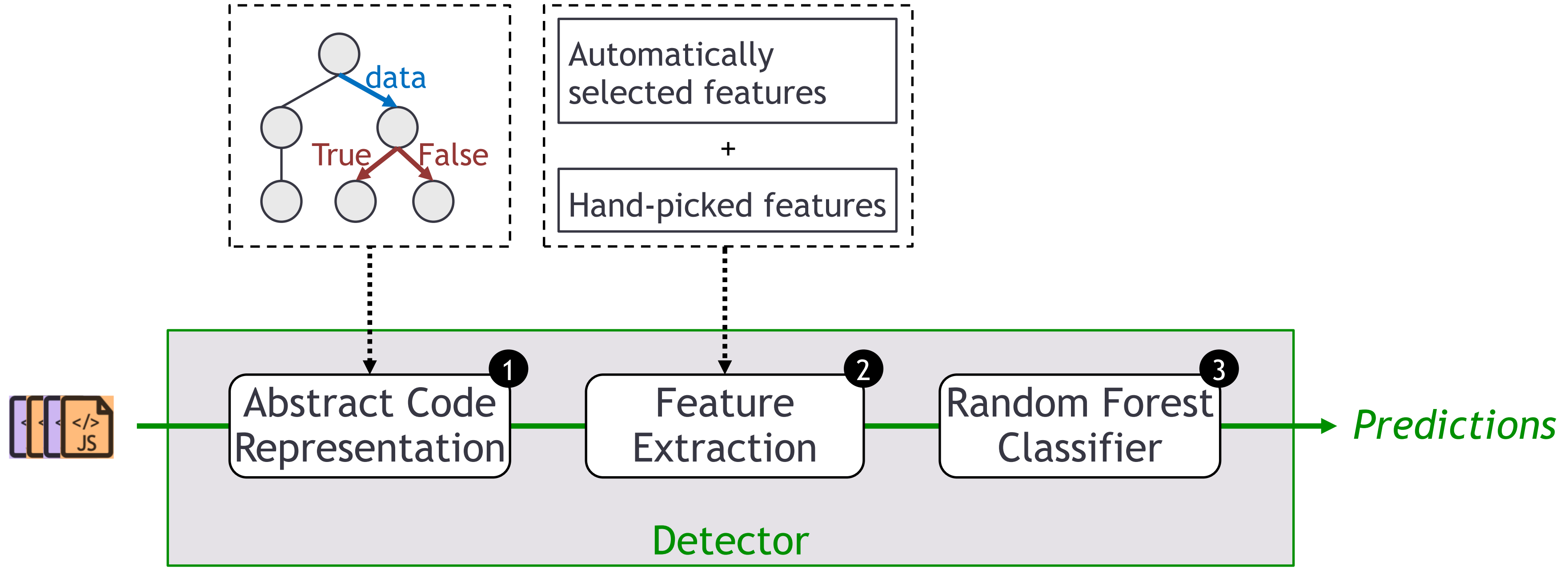
- Empirical study of JavaScript code transformations
  - benign vs. malicious code transformation techniques
  - evolution over time

Approach Overview



Approach Overview





[9] Fass et al., ACSAC 2019: https://github.com/Aurore54F/JStap

Code Transformations in the Wild

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Alexa Top 10k websites

- 89.40% of the websites contain ≥ 1 transformed script
- 68.60% of the scripts are transformed

Manual analysis:

- 83/100 regular
- 96/100 minified
- 99/100 obfuscated
- 100/100 transformed

Code Transformations in the Wild

Alexa Top 10k websites

- 89.40% of the websites contain ≥ 1 transformed script
- 68.60% of the scripts are transformed

↳ Most prevalent transformation techniques:

- minification simple
- minification advanced

➢ Minification used to reduce loading times, i.e., improve website performance

npm Top 10k packages

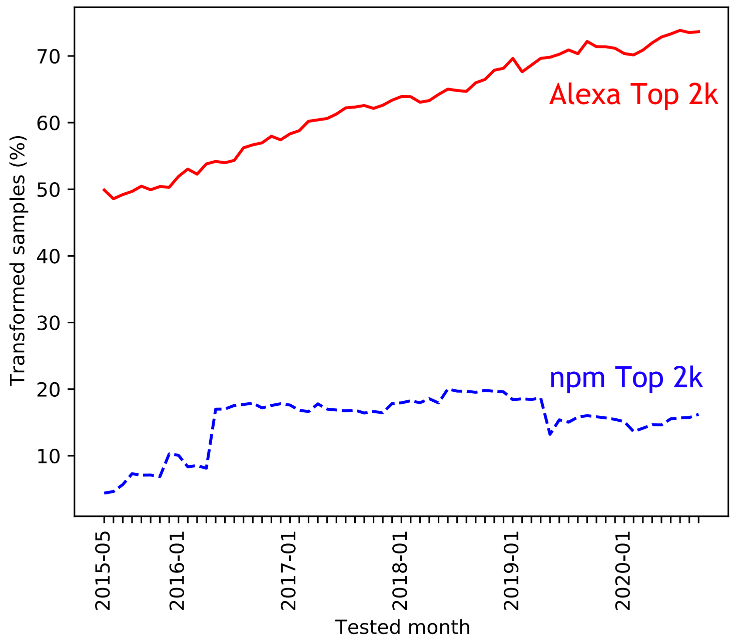
- 15.14% of the packages contain ≥ 1 transformed script
- 8.70% of the scripts are transformed

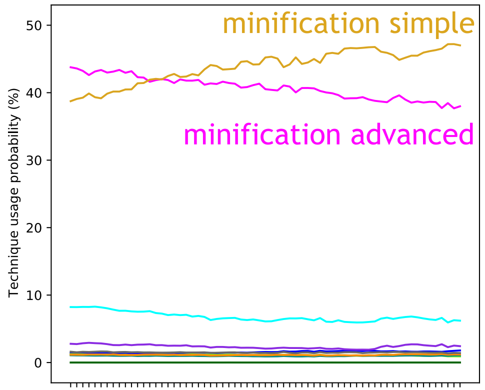
↳ Most prevalent transformation techniques:

- minification simple
- minification advanced

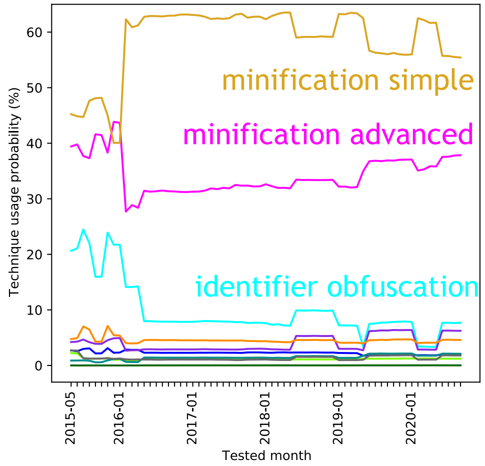
➢ Transformation/minification not popular

Evolution of Code Transformations over Time





Alexa Top 2k



npm Top 2k

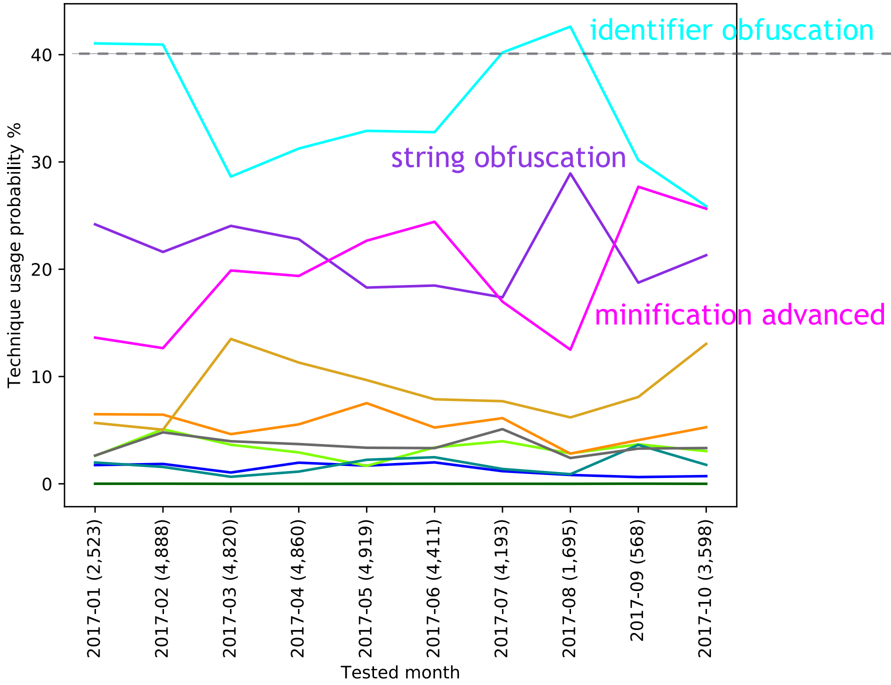
Code Transformations in Malicious JavaScript

Source	Collected	#JS	Transformed?
DNC	2015-2017	4,514	65.94%
Hynek	2015-2017	29,484	73.07%
BSI	2017	36,475	28.93%

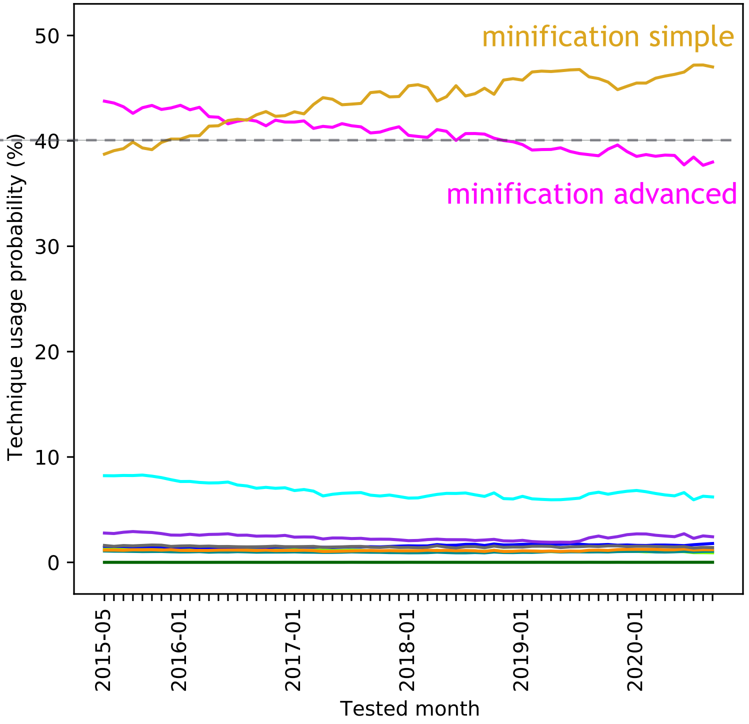
Malicious vs. Benign Code Transformations

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Malicious JS (BSI)



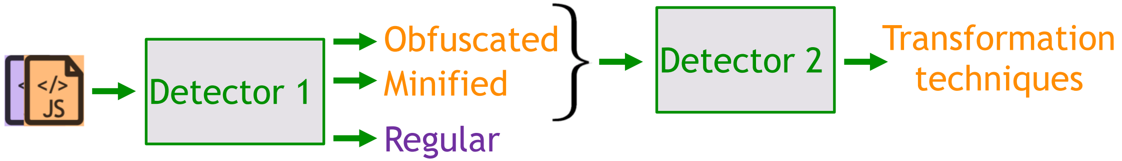
Benign JS (Alexa Top 2k)



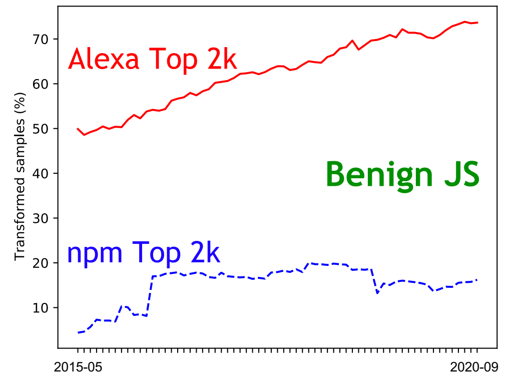
Conclusion

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

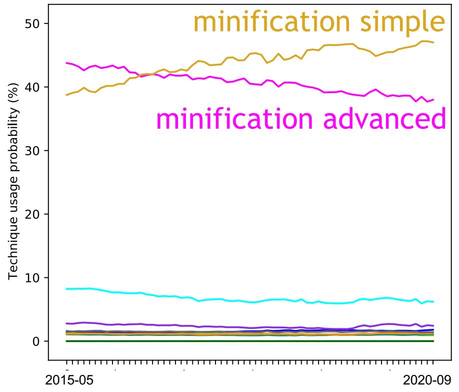
➢ Studied the prevalence of JavaScript code transformations



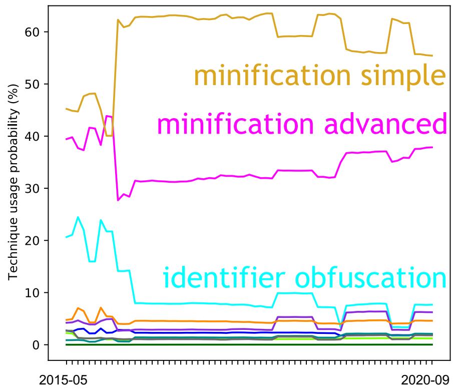
MarM15/js-transformations



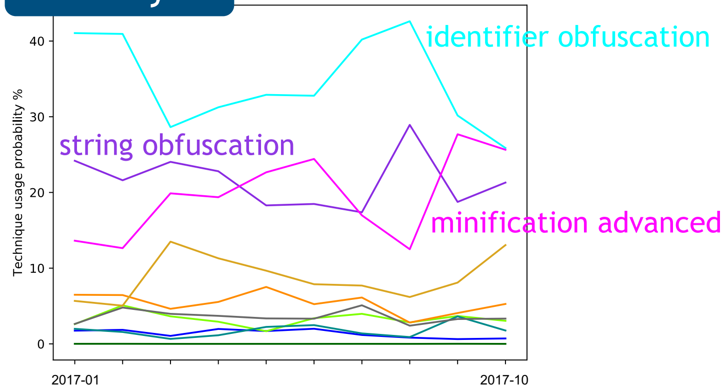
Thank you



Benign JS: Alexa Top 2k



Benign JS: npm Top 2k



Malicious JS: BSI