Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors

Umar Iqbal
University of Iowa

Steven Englehardt
Mozilla Corporation

Zubair Shafiq
University of California-Davis

Overview

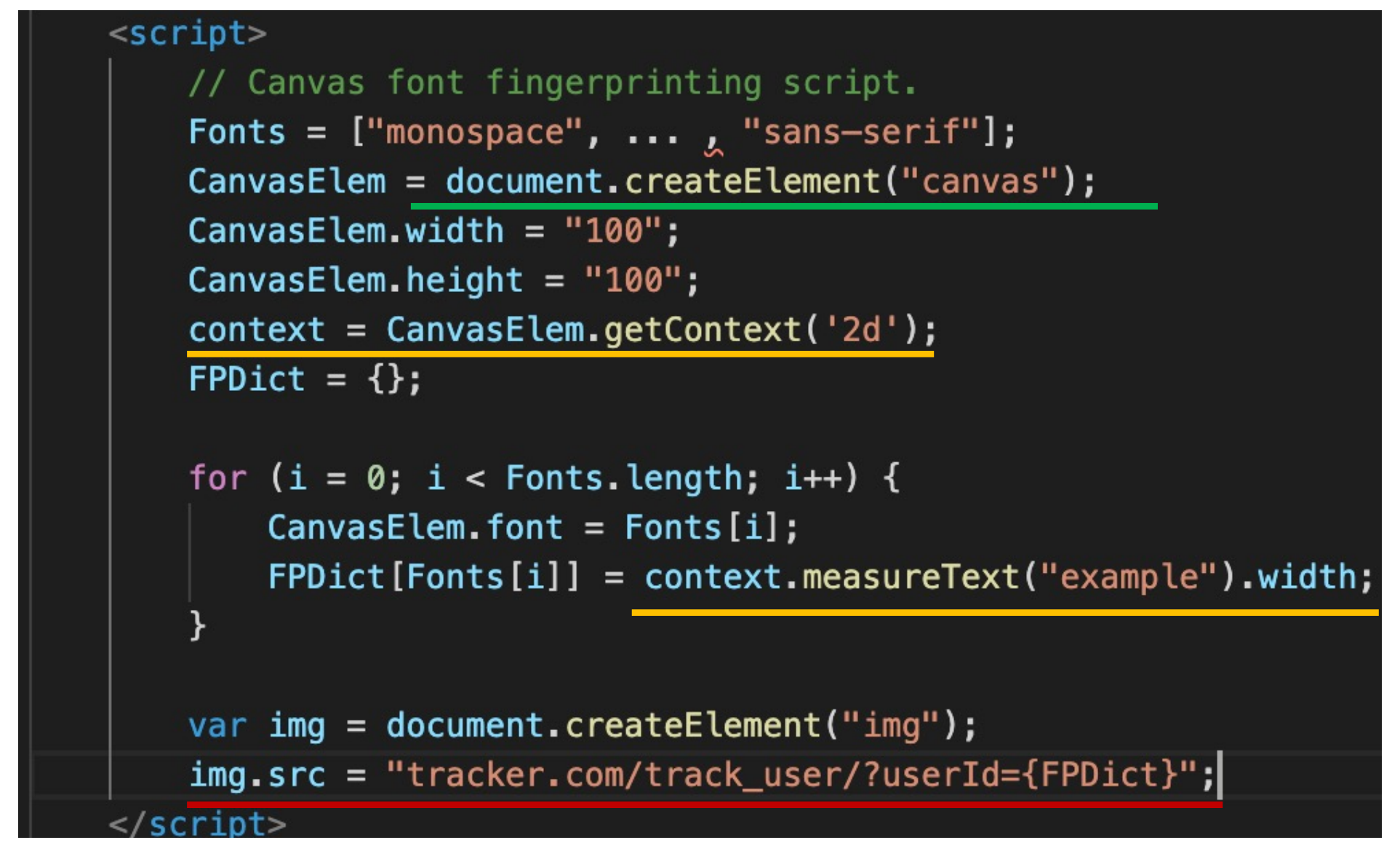
- ❑ Mainstream browsers are implementing countermeasures against third-party cookie-based cross-site tracking
- ❑ Trackers are expected to migrate to browser fingerprinting, which does not require cookies, to track users
- ❑ Existing countermeasures against fingerprinting limit website functionality, cause website breakage, and are not scalable
- ❑ We propose FP-Inspector, a syntactic-semantic machine learning approach that detects browser fingerprinting
- ❑ FP-Inspector detects 26% more scripts than the state-of-the-practice with an accuracy of 99%

Countering fingerprinting is hard!

- ❑ Detection requires sophisticated JavaScript analysis
  - ❑ Difficult and time consuming
  - ❑ Cannot be effectively scaled
- ❑ Existing protection mechanisms instead put blanket restrictions on APIs
  - ❑ Remove - Normalize - Randomize APIs
  - ❑ Goal is to break the uniqueness of APIs
- ❑ These restrictions interfere with the expected functionality of APIs
  - ❑ Limit and break the functionality of websites when APIs are used for benign purposes
- ❑ Prior research has proposed to detect browser fingerprinting scripts automatically with heuristics
  - ❑ Manually crafted and require presence of certain APIs with specific parameters
- ❑ Heuristics have two key issues:
  - ❑ Narrowly defined
  - ❑ Only work on execution traces
- ❑ Heuristics have accuracy and coverage issues

Fingerprinting Behavior

- ❑ Script's internal and external context
- ❑ Perspective about the inner workings of the script
  - ❑ Does the script mostly contain APIs that have fingerprinting potential?
- ❑ Perspective about the script's collaboration with external entities
  - ❑ Does the script interact with external entities?



```
<script>
    // Canvas font fingerprinting script.
    Fonts = ["monospace", ... , "sans-serif"];
    CanvasElem = document.createElement("canvas");
    CanvasElem.width = "100";
    CanvasElem.height = "100";
    context = CanvasElem.getContext('2d');
    FPDict = {};

    for (i = 0; i < Fonts.length; i++) {
        CanvasElem.font = Fonts[i];
        FPDict[Fonts[i]] = context.measureText("example").width;
    }

    var img = document.createElement("img");
    img.src = "tracker.com/track_user/?userId={FPDict}";
</script>
```

HTML canvas element

APIs with fingerprinting potential

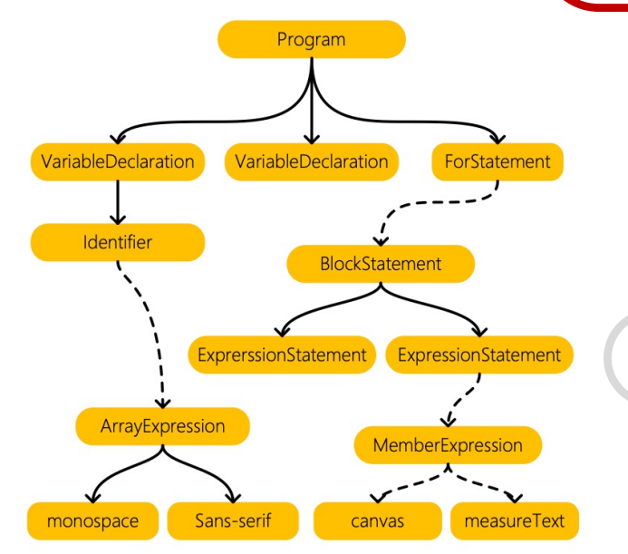
Network request with fingerprint

Syntactic & Semantic

Syntactic Representation

Source as a signal of functionality

People write code differently



ForStatement:var | MemberExpression:getContext

MemberExpression:measureText | CallExpression:canvas

Semantic Representation

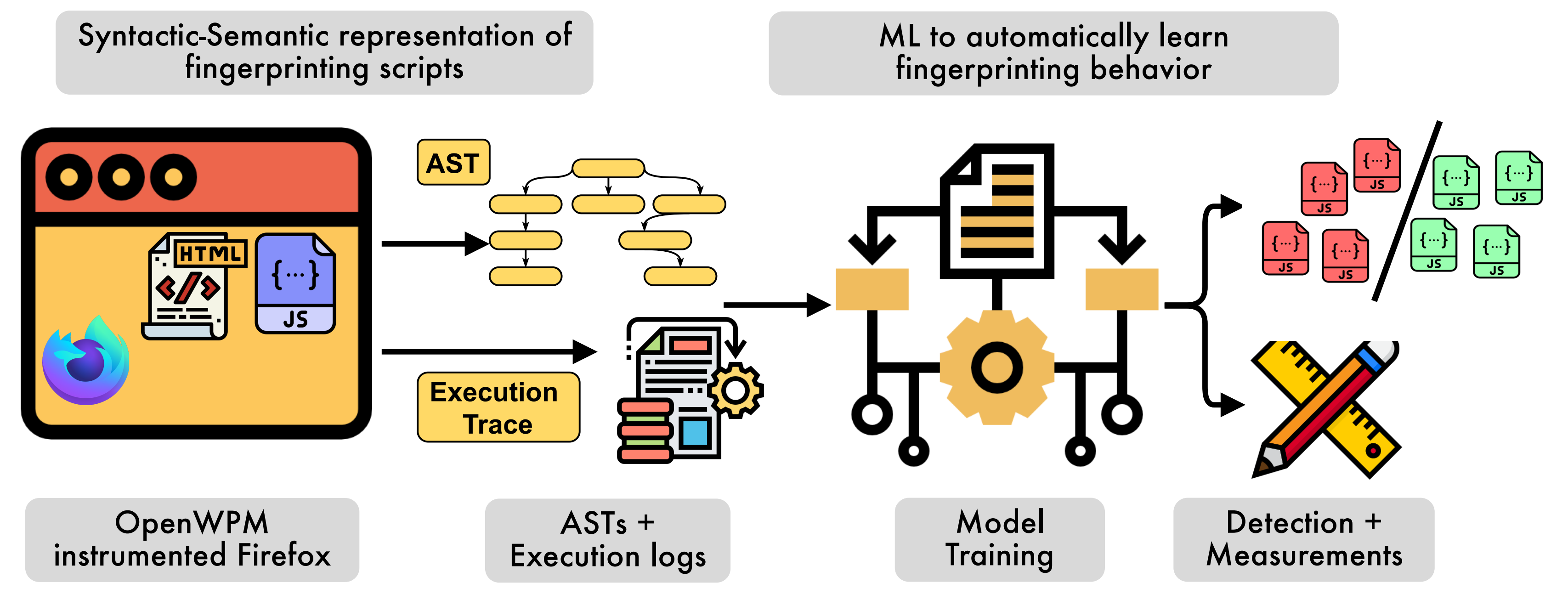
Execution as a signal of functionality

Frequency + Higher-level semantic of APIs

HTMLCanvasElement.width	100
HTMLCanvasElement.height	100
CanvasRenderingContext2D.measureText	50 (no. of calls)
CanvasRenderingContext2D.measureText	7 (no. of chars)

FP-Inspector

- ❑ A syntactic-semantic machine learning approach that detects browser fingerprinting
- ❑ Syntactic-semantic representation to model script's behavior
  - ❑ Syntactic representation is created through Abstract Syntax Trees (ASTs)
  - ❑ Semantic representation is created through script's execution
- ❑ Machine learning to learn fingerprinting patterns
  - ❑ Combination of APIs commonly used for fingerprinting
  - ❑ Limited interaction with the webpage
  - ❑ Communication with external entities


Syntactic-Semantic representation of fingerprinting scripts
ML to automatically learn fingerprinting behavior
HTML
JS
AST
Execution Trace
OpenWPM instrumented Firefox
ASTs + Execution logs
Model Training
Detection + Measurements

Evaluation

No ground truth

Heuristics as ground truth[1]

Decision tree-based ensemble

Iterative improvement

26% more scripts than heuristics

99.9% accuracy

10% net gain with syntactic + semantic

One-third of popular websites

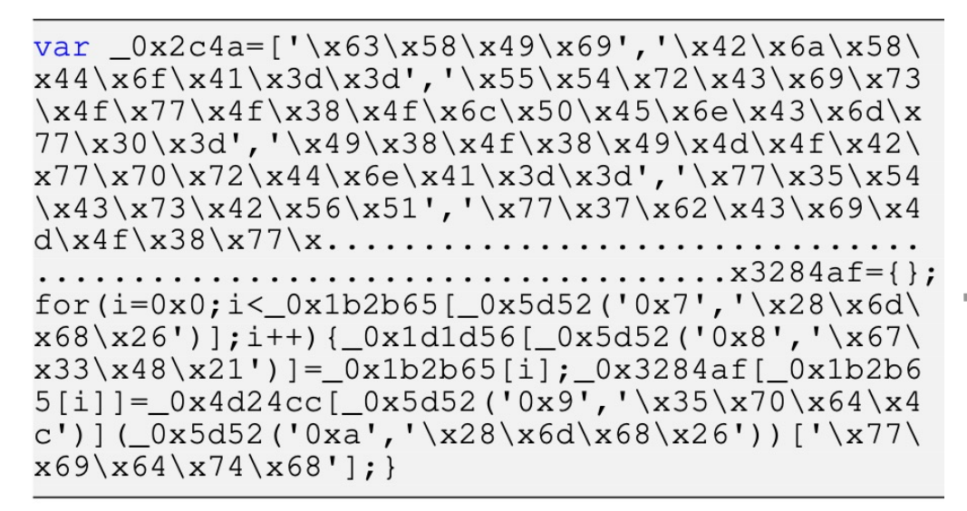
519 / Million '16[1]

2,349 / 100K '19

Websites that use ads for monetization

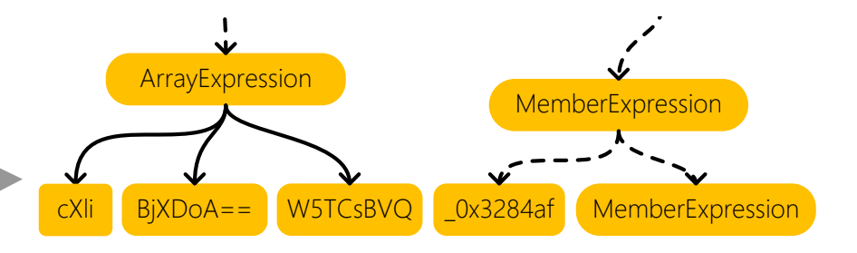
Popular trackers & Ad fraud services

Syntactic vs. Semantic

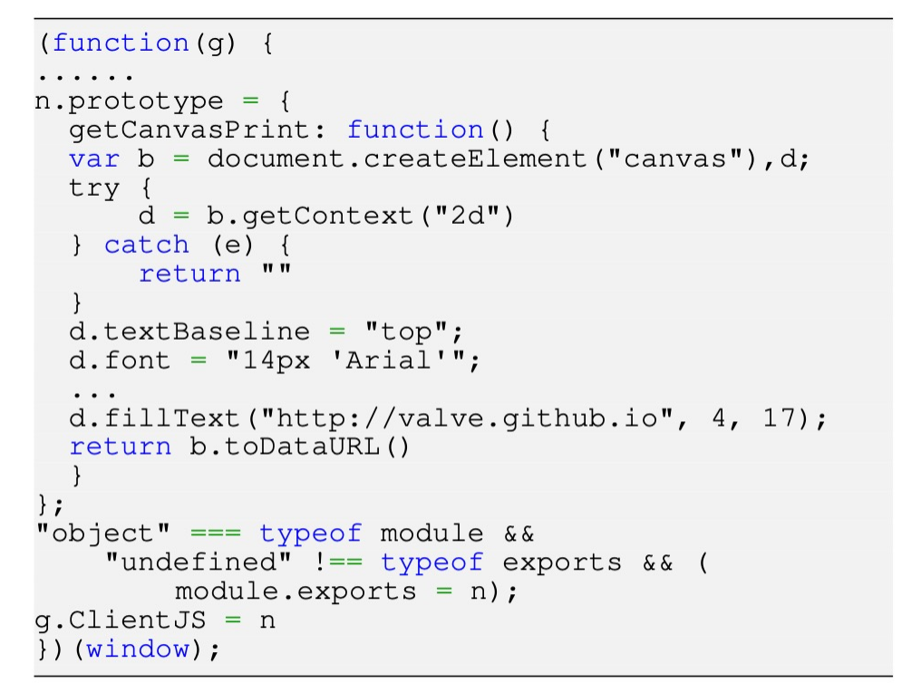


```
var _0x2c4a=['\x63\x58\x49\x69','\x42\x6a\x58\
x44\x6f\x41\x3d\x3d','\x55\x54\x72\x43\x69\x73
\x4f\x77\x4f\x38\x4f\x6c\x50\x45\x6e\x43\x6d\x
77\x30\x3d','\x49\x38\x4f\x38\x49\x4d\x4f\x42\
x77\x70\x72\x44\x6e\x41\x3d\x3d','\x77\x35\x54
\x43\x73\x42\x56\x51','\x77\x37\x62\x43\x69\x4
d\x4f\x38\x77\x.............................
.......................................x3284af={};
for(i=0x0;i<_0x1b2b65[_0x5d52('0x7','\x28\x6d\
x68\x26')];i++){_0x1d1d56[_0x5d52('0x8','\x67\
x33\x48\x21')]=_0x1b2b65[i];_0x3284af[_0x1b2b6
5[i]]=_0x4d24cc[_0x5d52('0x9','\x35\x70\x64\x4
c')](_0x5d52('0xa','\x28\x6d\x68\x26'))['\x77\
x69\x64\x74\x68'];}
```

Obfuscated



Syntactic representation



```
(function(g) {
......
n.prototype = {
  getCanvasPrint: function() {
  var b = document.createElement("canvas"),d;
  try {
      d = b.getContext("2d")
  } catch (e) {
      return ""
  }
  d.textBaseline = "top";
  d.font = "14px 'Arial'";
  ...
  d.fillText("http://valve.github.io", 4, 17);
  return b.toDataURL()
  }
};
"object" === typeof module &&
    "undefined" !== typeof exports && (
        module.exports = n);
g.ClientJS = n
})(window);
```

Dormant

NULL | NULL

Semantic Representation

Key Takeaways

- ❑ FP-Inspector improves the state-of-the-art in browser fingerprinting detection by incorporating syntactic-semantic representation
- ❑ Fingerprinting adoption is on the rise with more than one-third of top 1K popular websites using fingerprinting
- ❑ We open source our implementation and detections so that the community can benefit from them
- ❑ FP-Inspector's detections are incorporated by popular filter lists, such as EasyPrivacy, Disconnect, and DuckDuckGo
- ❑ More analysis like fingerprinting APIs discovery in the paper!

Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors
Umar Iqbal, Steven Englehardt, and Zubair Shafiq
IEEE S&P 2021

Paper

Source & Data

Umar Iqbal
@umaarr6
umariqbal.com

Contact details