

# Fingerprinting the Fingerprinters

An Analysis of Fingerprinting-Scripts and their Actors on the Internet

Sebastian Neef @ CodeTalks 2022

TU Berlin

2022-09-15



## Sebastian Neef

- CodeTalks 2018: Bugbounties
- Master thesis on Browser FP<sup>[1]</sup> ;-)
- PhD candidate @ TU Berlin
- IT-Sec Freelancer, CTF-Player, Bughunter, etc.
- @gehaxelt

# What is on our agenda?

- ① What is browser fingerprinting (used for)?
- ② How can we detect and analyze fingerprinting (scripts)?
- ③ What fingerprinting scripts and actors are active on the internet?
- ④ What can be done about it?

# An example

- Have you ever wondered how this works:

It looks like you tried to sign in from a different location, device, or browser:

Date: 2022-07-29 10:15:33 UTC

Account: [user](#) @ [.de](#)

Location: DE

IP Address: 84.162.174.227

Operating system: Linux x86\_64

Browser: Firefox 102.0

Enter this 6 digit code on the sign in page to confirm your identity:

501246

# An example

- Have you ever wondered how this works:

It looks like you tried to sign in from a different location, device, or browser:

Date: 2022-07-29 10:15:33 UTC

Account: [user](#) @ [.de](#)

Location: DE

IP Address: 84.162.174.227

Operating system: Linux x86\_64

Browser: Firefox 102.0

Enter this 6 digit code on the sign in page to confirm your identity:

501246

- IP address changes often and geolocation isn't always perfect
  - User-Agent header change with software updates
- ⇒ So how is a different device detected?

# What is browser fingerprinting?

- ⇒ A technique to uniquely **(re-)identify browsers or devices** based on their properties:
- A browser's fonts, plugins, quirks, HTTP headers, language, ...
  - A device's RAM, CPU, GPU, battery, screen resolution, sound card, ...

# What is browser fingerprinting?

- ⇒ A technique to uniquely **(re-)identify browsers or devices** based on their properties:
- A browser's fonts, plugins, quirks, HTTP headers, language, ...
  - A device's RAM, CPU, GPU, battery, screen resolution, sound card, ...

- **Live-Demo:**  
[amiunique.org](https://amiunique.org)

Are you unique ?

Yes! You are unique among the 4716965 fingerprints in our entire dataset.

The following informations reveal your OS, browser, browser version as well as your timezone and preferred language. Moreover, we show the proportion of users sharing the



10.52%



37.66%



0.24%

Search:

JavaScript attributes

Attribute	Similarity ratio	All time	Value
User agent ⓘ	0.02%		Mozilla/5.0 (X11; Linux x86_64; rv:93.0) Gecko/20100101 Firefox/93.0
Platform ⓘ	16.24%		Linux x86_64

## Security++

- Mitigate unauthorized access  
(think: 2FA)
- Identify malicious activity  
(think: CAPTCHAs)
- Bot protection
- Potential actors:
  - Financial institutions
  - (e)Commerce
  - High-security sites
  - ...

# Dual-Use-Dilemma: Security vs. Privacy

## Security++

- Mitigate unauthorized access (think: 2FA)
- Identify malicious activity (think: CAPTCHAs)
- Bot protection
- Potential actors:
  - Financial institutions
  - (e)Commerce
  - High-security sites
  - ...

## Privacy--

- Track users without cookies
- Identify users across sites, browsers or incognito tabs
- Happens without consent
- Hard to block or circumvent
- Potential actors:
  - Ad-Tech
  - (e)Commerce
  - ...

# Dual-Use-Dilemma: Security vs. Privacy

## Security++

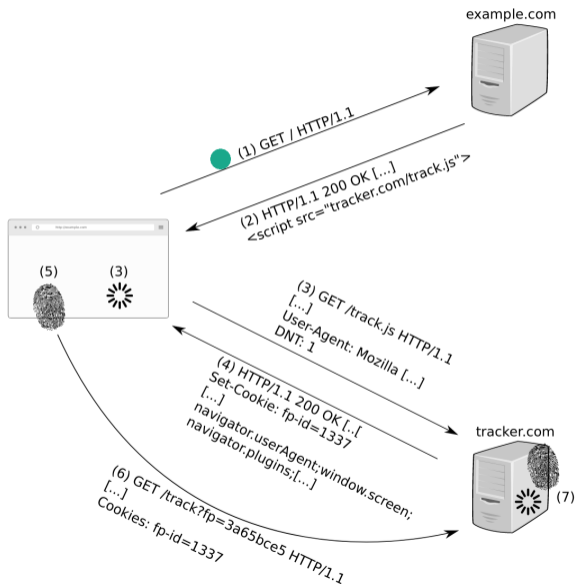
- Mitigate unauthorized access (think: 2FA)
- Identify malicious activity (think: CAPTCHAs)
- Bot protection
- Potential actors:
  - Financial institutions
  - (e)Commerce
  - High-security sites
  - ...

## Privacy--

- Track users without cookies
- Identify users across sites, browsers or incognito tabs
- Happens without consent
- Hard to block or circumvent
- Potential actors:
  - Ad-Tech
  - (e)Commerce
  - ...

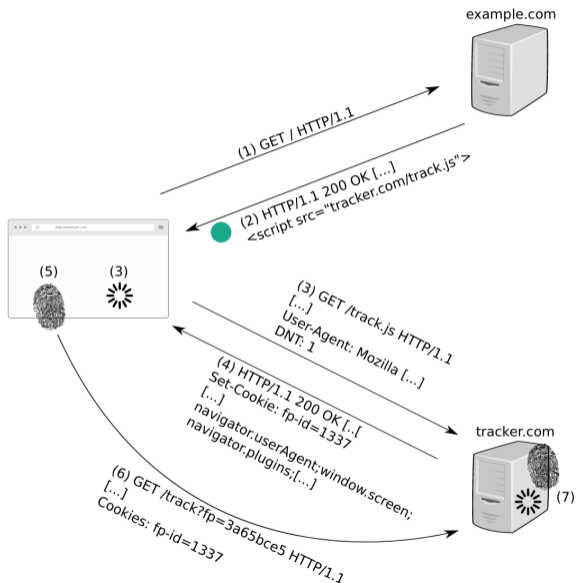
⇒ A user does not know what purpose it serves.  
They might not even know it is happening...

# How does it work?



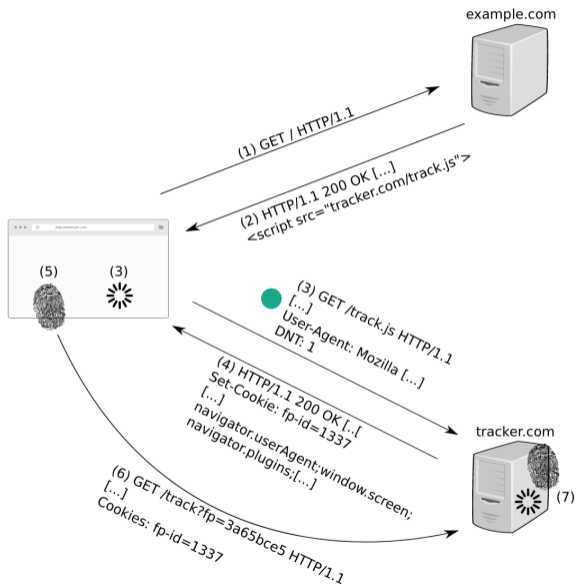
- Browser sends request to `example.com`

# How does it work?



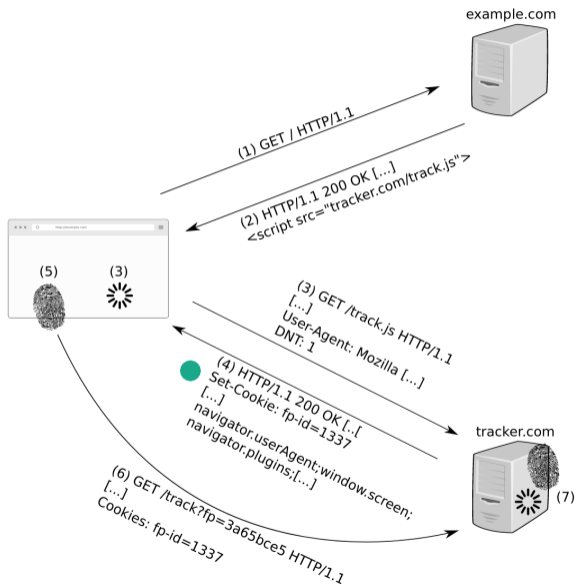
- Example.com responds with a site and a reference to tracker.com/track.js

# How does it work?



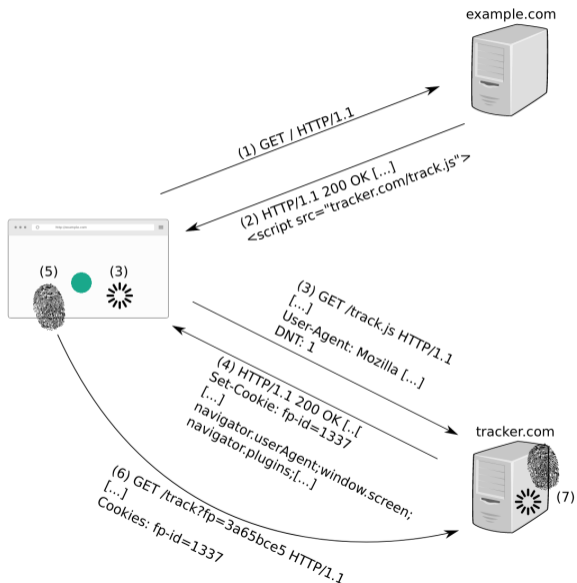
- Browser processes the document and requests track.js.
- Request contains HTTP headers (passive fingerprinting)

# How does it work?



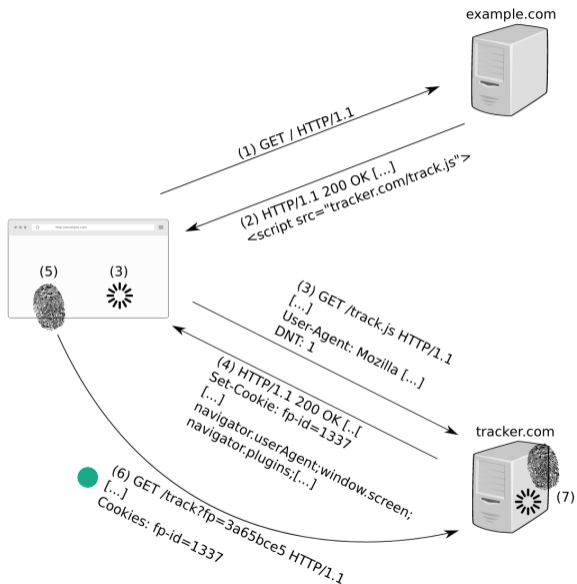
- tracker.com response with cookies and fingerprinting code (track.js)

# How does it work?



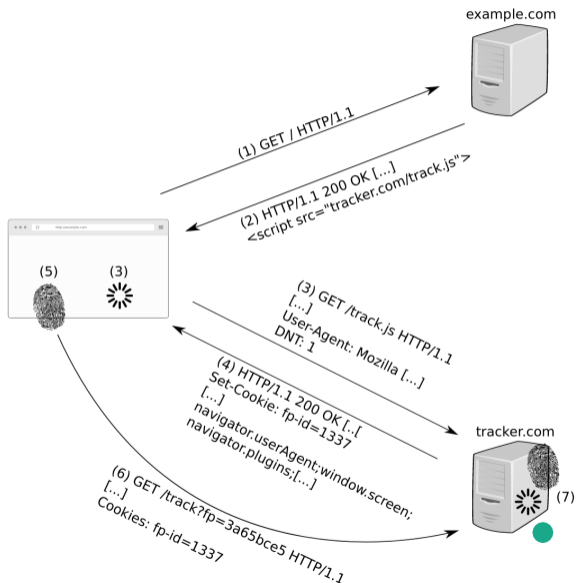
- Browser processes the track.js file and properties are collected

# How does it work?



- Track.js sends the fingerprint or the FP data to tracker.com (active fingerprinting)

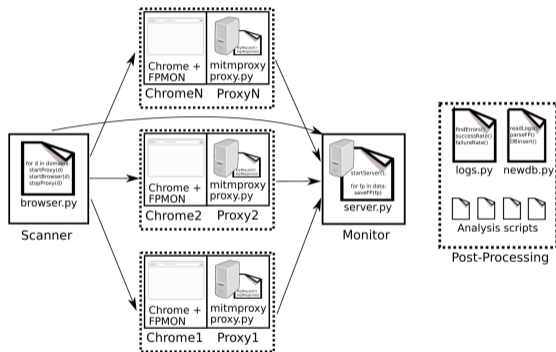
# How does it work?



- Tracker.com calculates a hash  
→ fingerprint
  - Correlates it to a user
- ⇒ **It's effective**

# How can we detect and measure fingerprinting?

- We developed **FPNET**<sup>[1]</sup> and **FPMON**<sup>[2]</sup> to analyze websites



⇒ Scan the Alexa Top 10k and monitor access to 115 JavaScript properties mapped to 40 *Fingerprinting Features*

Domain	www.metacafe.com
JS Attributes Tracked	89% (102/115)
Fingerprinting Features	95% (38/40)
Aggressive Features	95% (17/18)
<b>Sensitive</b>	<b>Aggressive</b>
<ul style="list-style-type: none"><li>Platform</li><li>User agent</li><li>Timezone</li><li>Mobile</li><li>Screen window</li><li>Content language</li><li>Storage</li><li>Browser vendor</li><li>Online status</li><li>DoNotTrack</li><li>Product</li><li>Vendor</li><li>Vendor sub</li><li>Build ID</li><li>CPU class</li><li>App code name</li><li>Cookies enabled</li><li>Java enabled</li><li>Browser language</li><li>System language</li><li>Drag and drop</li></ul>	<ul style="list-style-type: none"><li>App version</li><li>List of plugins</li><li>Webdriver</li><li>Permissions</li><li>JS fonts</li><li>Canvas</li><li>Geolocation</li><li>WebGL</li><li>Product sub</li><li>Operating system</li><li>CPU concurrency</li><li>Media devices</li><li>Battery status</li><li>Device memory</li><li>Connection</li><li>Audio</li><li>Audio and video formats</li></ul>
<b>Highest Scoring Scripts</b>	
38	<a href="http://js-ad-score.com:score.min.js">js-ad-score.com:score.min.js</a>
7	<a href="http://securepubads.g.doubleclick.net:puba...">securepubads.g.doubleclick.net:puba...</a>
4	<a href="http://secure.quantserve.com:quant.js">secure.quantserve.com:quant.js</a>

# Fingerprinting the Fingerprinters

- 1 FPMON wraps document, window, JS-API properties and functions before page-load.

```
const fp_metric_userAgent = ["navigator.userAgent"];
const fp_metric_platform = ["navigator.platform"];
const fp_metric_enabled_cookies = ["navigator.cookieEnabled"];
const fp_metric_timezone = ["getTimezoneOffset()", "window.Intl"];
const fp_metric_content_language = ["navigator.languages", "navigator.userLanguage", "navigator.language"];
const fp_metric_canvas = ['getImageData()', 'getLineDash()', 'measureText()', 'isPointInPath()'];
```

```
... detectFingerprinting = function() {
...   function overrideFunction(item) {
...   }
... }
... const attributesToMonitor = {
... };
... function printAccess(prop, subProp) {
... }
...
... // initialization
... window.navigator.monitorFingerprinting = {};
... const originalValues = {};
... var fpmon_log_var = {};
... var fpmon_log_var_ret = {};
... var fpmon_log_fun = {};
... var fpmon_log_fun_ret = {};
```

```
function callOrigin() {
  let originalFunc = Error.prepareStackTrace;
  let callerfile;
  var ret_dict = {}
  try {
    Error.prepareStackTrace = function (err, stack) {
      return stack;
    };
    let currentfile;
    const err = new Error();
    currentfile = err.stack[0].getFileName();
    for(i = 0; i < err.stack.length; i++) {
      if(err.stack[i].getFileName() === undefined || err.stack[i].getFileName() === null) {
        continue
      }
      if (currentfile !== err.stack[i].getFileName()) {
        ret_dict['file'] = err.stack[i].getFileName()
        break;
      }
    }
  } catch (e) {}
  Error.prepareStackTrace = originalFunc;
  //console.log("=====")
  return ret_dict;
}
```

## 2 Monitor a script's behavior on the function-level

gID	sID	oID	Property or function	Script origin	Feature group
0	0	0	navigator.userAgent	tracker.com/fp.js	UserAgent
1	0	1	screen.height	example.com/main.js	Screen_window
2	1	0	getImageData()	tracker.com/fp.js	Canvas
3	1	1	screen.width	example.com/main.js	Screen_window
4	2	0	drawArrays()	tracker.com/fp.js	WebGL
5	3	0	fillText()	tracker.com/fp.js	JS_fonts
...	...	...	...	...	...

TABLE 4: Example property and function observations including the associated feature group and script origin. The global-ID (gID), script-ID (sID) and origin-ID (oID) are essential to preserve the observation order.

# Fingerprinting the Fingerprinters

- 3 Compute script signatures and script scores

example.com/main.js: `Screen_window;Screen_window;...`(1)

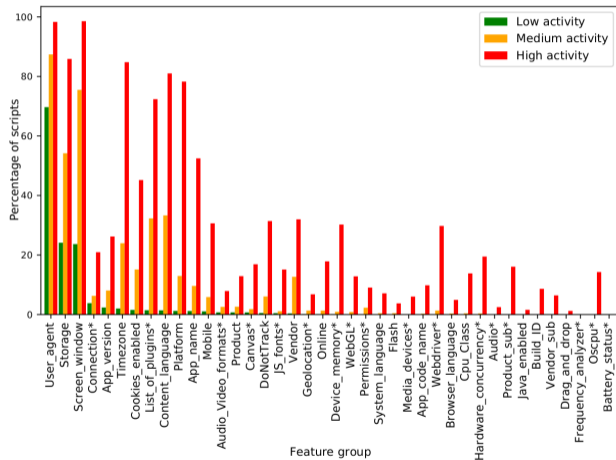
tracker.com/fp.js: `UserAgent;Canvas;WebGL;JS_fonts;...` (4)

- 4 Classify the activity levels based on the script score

Activity	Score	Interpretation
Low	$\text{score} < 3$	The script is likely benign.
Medium	$3 \leq \text{score} \leq 6$	The script exhibits limited fingerprinting activity.
High	$\text{score} > 6$	The script is considered to deliberately fingerprint the user.

TABLE 6: Categorization of scripts into low, medium, high fingerprinting activity based on their score.

# How active are the scripts?



- Script activity distribution:
  - 59% low
  - 32% medium
  - ~9% high

FIGURE 9: Distribution of feature groups for fingerprinting scripts with low, medium and high activity. \* denotes aggressive feature groups.

# Who is behind these scripts?

- *high activity* scripts grouped by their behavior  $\rightarrow \geq 350$  networks
  - Larger networks often have a lower score  $\rightarrow$  Tracking?
  - Smaller networks often have a higher score  $\rightarrow$  Security?

Score	Size	$Sig_{len}$	Files	Typ. Names	Script Domain(s)	Page Domain(s)
8	1,343	30	371	pubads[...].js, ...	doubleclick.net	blackdoctor.org, ebay.com, ...
26	232	97	230	1ad6cd50, 5e4f5e70, ...	dhl.com, dnb.com, ...	dhl.com, dnb.com, ...
36	5	269	1	device.js	maxmind.com	mediafire.com, ...
7	62	25	40	E-v1.js, embed_shepherd- v1.js, ...	wistia.com, wis- tia.net	paychex.com, rochester.edu, privy.com, ...
...	...	...	...	...	...	...

TABLE 7: Example fingerprinting networks with their properties.

# Who are behind these scripts?

Actor	Category	Networks	Score (Aggr.)	Pages
Google DoubleClick	Web Advertisements	19	10 (2)	1,583
Google AdSense	Search Engines	11	8 (1)	544
Yandex Metrica	Search Engines	52	14 (3)	367
Akamai	Computers	2	28 (10)	292
FingerprintJS	No Category	9	20 (10)	133

TABLE 8: Top 5 actors ranked by the amount of pages they can fingerprint.

# Who are behind these scripts?

Actor	Category	Networks	Score (Aggr.)	Pages
Maxmind	Computers	1	36 (14)	5
Moat	Web Advertisements	5	34 (12)	114
Adsko.re	Web Advertisements	2	29 (10)	19
Akamai	Computers	2	28 (10)	292
ShieldSquare (Perfdrive)	Vehicles	1	23 (6)	11

TABLE 9: Top 5 actors covering at least 5 pages ranked by the score.

# Who are behind these scripts?

## Device Tracking

MAXMIND

Capture more data and catch more fraud with device tracking.

Oracle Moat Measurement

ORACLE

Oracle Moat is an ad measurement and marketing analytics suite designed to help advertisers, publishers, and platforms measure media performance across the breadth of their digital and TV advertising campaigns.

## Real-Time Bot Protection For All Web, Mobile, and API endpoints

radware

Safeguard your online revenue, reduce the risk of data breaches and improve operational efficiency.

## All-Round Web Analytics

From traffic trends to mouse movements – get a comprehensive understanding of your online audience and drive business growth.

Get started

Try live demo

Vandix Metrics

The highest accuracy device identification for mobile and web

Fingerprint

Stop fraud, spam, and account takeover with 99.5% accurate device fingerprinting as a service.



Advanced Protection For Your Ads Margins.

hCaptcha

Stop more bots. Start protecting user privacy.

The #1 privacy-first CAPTCHA for web, mobile, and more.

sift

Online fraud stops here.

Sift's Digital Trust & Safety Suite is every fraudster's nightmare. We secure your business at scale and support explosive growth. Proactively stop account takeover, payment fraud, scam content, and chargebacks from destroying your brand's integrity. So: what's your goal?



PROTECT

DIGITAL TRUST & SAFETY

GROW



## State-of-the-Art Protection Against Scraping

DATA DOME

Online fraud & bot management for mobile apps, websites, & APIs.

BRIGHTEDGE INNOVATIONS

## SEO at the Speed of Search

Industry's first and only SEO solution to give marketers real-time research, recommendations, and rankings – everything an SEO needs all in a unified platform.

GROWTH MARKETING FOR DIGITAL MARKETERS

We grow the market leaders of tomorrow



<sup>1</sup>Screenshots taken from [8-18]

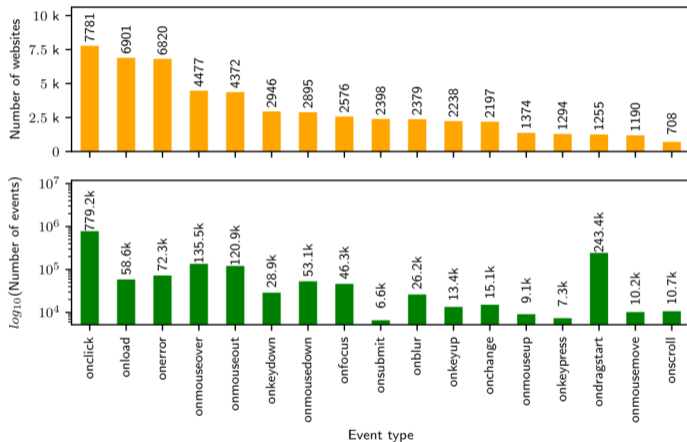
# What can we do about it?

- Browser fingerprinting happens in the background without the user noticing or consenting to it. (Or did you?)
- Do we need more regulation, i.e. fingerprinting-consent popups?

# What can we do about it?

- Browser fingerprinting happens in the background without the user noticing or consenting to it. (Or did you?)
- Do we need more regulation, i.e. fingerprinting-consent popups?
- Possible defenses against fingerprinting:
  - Disabling JavaScript ?
  - Installing privacy plugins ?
  - Blocking or detecting fingerprinting scripts ?
  - Randomizing properties ?
  - Homogeneous properties ?

# Disabling JavaScript?



- $\geq 87\%$  of the Alexa top 10k pages use event `on-*` handlers.
  - Reduced functionality or usability without JavaScript
- ⇒ Not suitable for most internet users :-/

FIGURE 8: Top 15 event handlers by occurrences and website distribution from an analysis of the Alexa Top 10,000.

# Using Privacy Plugins?

- Some plugins help, others not much according to Kybranz<sup>[4]</sup>
- Most use deny-lists → Some scripts can be blocked, but certainly not all.

Domain	Content Topic	Score Chrome	Ad-block	Duck-duckgo	Privacy Badger
metacafe.com	Video Sharing	95%	95%	95%	20%
easyjet.com	Flight Service	73%	73%	73%	73%
nasdaq.com	Stock Market	70%	70%	70%	68%
bankofamerica.com	Finance	65%	58%	65%	65%
savethechildren.org	Non-profit	63%	45%	23%	43%
nytimes.com	News	60%	58%	18%	60%
coinbase.com	Privacy Service	58%	58%	58%	58%
fingerprntjs.com	Fingerprinter	53%	53%	53%	53%
sba.gov	Government	40%	25%	10%	18%
theguardian.com	News	30%	23%	15%	18%

TABLE 5.3: Privacy enhancing browser extensions have a positive impact on limiting fingerprint activity, in contrast to no extensions used. Measured with FPMON on a Chrome browser with one extension loaded at a time.

# Blocking or detecting fingerprinting scripts ?

- Some scripts implement evasion techniques :-/

# Blocking or detecting fingerprinting scripts ?

- Some scripts implement evasion techniques :-/
- URL changes (~9.8k URLs):
  - ~90% filename changes (e.g. /710de559 → /710de4e3)
  - 12 domain changes (e.g. xqheb9yszyrd.com → vk771nizckm6.com)
  - ~800 filename and domain changes  
(e.g. .com\_mssgddsdst1.js → .co.uk\_jywraijszxpbtq.js)

# Blocking or detecting fingerprinting scripts ?

- Some scripts implement evasion techniques :-/
- URL changes (~9.8k URLs):
  - ~90% filename changes (e.g. /710de559 → /710de4e3)
  - 12 domain changes (e.g. xqheb9yszyrd.com → vk771nizckm6.com)
  - ~800 filename and domain changes  
(e.g. .com\_mssgddsdst1.js → .co.uk\_jywraijzxsptbytq.js)
- Behavioral changes ( ~70k URLs):
  - > 86% with identical script signature
  - ~92% of the 14% have identical scores → Changed behavior

# Blocking or detecting fingerprinting scripts ?

- Some scripts implement evasion techniques :-/
- URL changes (~9.8k URLs):
  - ~90% filename changes (e.g. /710de559 → /710de4e3)
  - 12 domain changes (e.g. xqheb9yszyrd.com → vk771nizckm6.com)
  - ~800 filename and domain changes  
(e.g. .com\_mssgddsdst1.js → .co.uk\_jywraijzxsxptbytq.js)
- Behavioral changes ( ~70k URLs):
  - > 86% with identical script signature
  - ~92% of the 14% have identical scores → Changed behavior

⇒ It's not easy, but implementing a behavior-based extension would be awesome.  
Anyone up for this?

# Randomized properties?

- Randomize the return values (slightly)
- Might break functionality depending on the property and randomness.
- Firefox has `privacy.resistFingerprinting` <sup>[5]</sup> which disguises some information and disables some features.

# Homogeneous properties?

- Make all browsers appear the same
- Only works if a lot of browsers and users do it
- Prominent example: TOR browser <sup>[6]</sup>
- Apple's Safari does this, too <sup>[7]</sup>

# So what can we do?

- More research and education on that topic!

# So what can we do?

- More research and education on that topic!
- Choose companies, products and services wisely!

# So what can we do?

- More research and education on that topic!
- Choose companies, products and services wisely!
- Think of more anti-FP measures.

Thanks for listening!

# Questions?

Feel free to reach out: [neef@tu-berlin.de](mailto:neef@tu-berlin.de)

# References I

- ① FPNET tool: <https://github.com/gehaxelt/MasterThesis-FPNET>
- ② FPMON extension: <https://fpmon.github.io/fingerprinting-monitor/>
- ③ FPMON paper: <https://dl.acm.org/doi/abs/10.1145/3463676.3485599>
- ④ Kybranz' thesis <https://github.com/KybranzF/Master-Thesis>
- ⑤ Firefox's anti FP: <https://support.mozilla.org/en-US/kb/firefox-protection-against-fingerprinting>
- ⑥ Tor-Browser anti FP: <https://blog.torproject.org/browser-fingerprinting-introduction-and-challenges-ahead/>
- ⑦ Safari anti FP: <https://support.apple.com/guide/safari/prevent-cross-site-tracking-sfri40732/mac>

## References II

- 8 <https://www.maxmind.com/en/solutions/minfraud-services/device-tracking>
- 9 <https://metrca.yandex.com/about?>
- 10 <https://fingerprint.com/>
- 11 <https://www.oracle.com/cx/advertising/measurement/>
- 12 <https://www.adscore.com>
- 13 <https://www.radware.com/products/bot-manager/>
- 14 <https://sift.com/>
- 15 <https://www.brightedge.com>
- 16 <https://www.hcaptcha.com>
- 17 <https://datadome.co>
- 18 <https://www.dcmn.com>