Online Tracking

A 1-million-site Measurement and Analysis

Steven Englehardt
@s_englehardt

Arvind Narayananan
@random_walker

This research was supported by NSF award CNS 1526353, a grant from the Data Transparency Lab, and an Amazon AWS Credits Research Grant.
Visiting 2 websites results in 84 third parties contacted
Tracking with browser state
Tracking with browser state
Tracking with browser state
Tracking with fingerprinting

Browser

example.com

JS
Tracking with fingerprinting

Browser

Fonts
Screen Resolution
Plugins

example.com
Tracking with fingerprinting

Browser

Fonts
Screen Resolution
Plugins

example.com
Open Web Privacy Measurement (OpenWPM)

https://github.com/citp/OpenWPM
The Princeton Web Census

Monthly
1 Million Site Crawl

Collecting:
- Javascript Calls
- All javascript files
- HTTP Requests and Responses
- Storage (cookies, Flash, etc)
Measurement is effective because most actors are not malicious

1. Bulk of trackers respond to pressure from publishers, users, and regulators

2. Not trying to avoid detection

3. High risk for malicious actions
Research findings from the Princeton Web Census
The long tail of third-party tracking
A consolidated tracking ecosystem
Only 6 organizations are present on >10% of sites
Almost all top third parties cookie sync

45 of top 50 third parties sync cookies (85% chance any two share an ID)
New browser features used for fingerprinting

Canvas

WebRTC

Audio

Battery

https://webtransparency.cs.princeton.edu/webcensus/
Detecting Fingerprinting

// Measurement Code
instrumentObject(window.CanvasRenderingContext2D.prototype, ...);
instrumentObject(window.HTMLCanvasElement.prototype, ...);
Detecting Fingerprinting

// Measurement Code
instrumentObject(window.CanvasRenderingContext2D.prototype, ...);
instrumentObject(window.HTMLCanvasElement.prototype, ...);

// Canvas Fingerprinting Example
ctx = canvas.getContext("2d");
ctx.fillText("hello world", 2, 15);
ctx.fillStyle = "#f60";
ctx.fillRect(125, 1, 62, 20);
fp = canvas.toDataURL();
Detecting Fingerprinting

// Measurement Code
instrumentObject(window.CanvasRenderingContext2D.prototype, ...);
instrumentObject(window.HTMLCanvasElement.prototype, ...);

// Canvas Fingerprinting Example
ctx = canvas.getContext("2d");
ctx.fillText("hello world", 2, 15);
ctx.fillStyle = "#f60";
ctx.fillRect(125, 1, 62, 20);
fp = canvas.toDataURL();

Measurement Logs
(SCRIPT_URL, "getContext", "2d")
(SCRIPT_URL, "fillText", "hello world", 2, 15)
(SCRIPT_URL, "fillStyle", "#f60")
(SCRIPT_URL, "fillRect", 125, 1, 62, 20)
(SCRIPT_URL, "toDataURL", "data: ...")
Detecting Fingerprinting

// Measurement Code
instrumentObject(window.CanvasRenderingContext2D.prototype, ...);
instrumentObject(window.HTMLCanvasElement.prototype, ...);

// Canvas Fingerprinting Example
ctx = canvas.getContext("2d");
ctx.fillText("hello world", 2, 15);
ctx.fillStyle = "#f60";
ctx.fillRect(125, 1, 62, 20);
fp = canvas.toDataURL();

Measurement Logs
(SCRIPT_URL, "getContext", "2d")
(SCRIPT_URL, "fillText", "hello world", 2, 15)
(SCRIPT_URL, "fillStyle", "#f60")
(SCRIPT_URL, "fillRect", 125, 1, 62, 20)
(SCRIPT_URL, "toDataURL", "data: ...")

Post-measurement Analysis
1. Examine API use for fingerprinting
2. Check for tampering / instrumentation inspection
Detecting Fingerprinting

1. Observe a sequence of API calls
2. Techniques clustered together
3. Results of calls combined and sent to server
4. Limited API use beyond that for fingerprinting
Abusing WebRTC candidate generation for tracking

Source: http://www.html5rocks.com/en/tutorials/webrtc/basics/
Without user intervention, a tracking script can:

1. Reveal the user’s real IP address when behind a VPN

2. Reveal the user’s local IP address for each local interface.
WebRTC `dataChannel` requires no permissions

Without user intervention, a tracking script can:

1. Reveal the user’s real IP address when behind a VPN

2. Reveal the user’s local IP address for each local interface.

More identifying for corporate and university users.
Measuring the use of WebRTC for tracking

Measurement Code:

```javascript
// Access to webRTC
instrumentObject(
    window.RTCPeerConnection.prototype,
    "RTCPeerConnection", true
);
```
Measuring the use of WebRTC for tracking

Measurement Code:

```javascript
// Access to webRTC
instrumentObject(
    window.RTCPeerConnection.prototype,
    "RTCPeerConnection", true
);

~90% of unsolicited dataChannel use on homepages is for tracking

57 scripts on 625 sites.
Using AudioContext for fingerprinting

Used by:

cdn-net.com script

![Diagram]

\[ \text{SHA1}([-121.36, -121.19, ...]) \rightarrow \text{eb8a30ad7...} \]
Using AudioContext for fingerprinting

Used by:

cdn-net.com script

Used by:

pxi.pub and ad-score.com scripts
Implications for Tor Browser

WebAudio allows you to write data to AudioBuffers and perform effects/manipulation/spectral analysis on them, and extract their contents.

If the underlying routines are OS supported, they may be fingerprintable. 
https://developer.mozilla.org/en-US/docs/Web_Audio_API
Do Privacy Tools Help?
Privacy tools effectively block stateful tracking

- Third-party cookie blocking
  - 32 out of 50,000 sites work around blocking by redirecting the top-level domain
  - Average number of third-parties per site reduced from ~18 to ~13

- Ghostery
  - Average number of third-parties per site reduced from ~18 to ~3
  - Very few third-party cookies are set
Crowdsourced lists miss fingerprinters

<table>
<thead>
<tr>
<th>Technique</th>
<th>Percentage of Scripts</th>
<th>Percentage of Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>EasyList + EasyPrivacy</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Crowdsourced lists miss fingerprinters

<table>
<thead>
<tr>
<th>Technique</th>
<th>Percentage of Scripts</th>
<th>Percentage of Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas</td>
<td>25%</td>
<td>88%</td>
</tr>
</tbody>
</table>

**EasyList + EasyPrivacy**
Crowdsourced lists miss fingerprinters

<table>
<thead>
<tr>
<th>Technique</th>
<th>Percentage of Scripts</th>
<th>Percentage of Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas</td>
<td>25%</td>
<td>88%</td>
</tr>
<tr>
<td>Canvas Font</td>
<td>10%</td>
<td>91%</td>
</tr>
</tbody>
</table>
Crowdsourced lists miss fingerprinters

<table>
<thead>
<tr>
<th>Technique</th>
<th>Percentage of Scripts</th>
<th>Percentage of Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas</td>
<td>25%</td>
<td>88%</td>
</tr>
<tr>
<td>Canvas Font</td>
<td>10%</td>
<td>91%</td>
</tr>
<tr>
<td>WebRTC</td>
<td>5%</td>
<td>6%</td>
</tr>
</tbody>
</table>
## Crowdsourced lists miss fingerprinters

<table>
<thead>
<tr>
<th>Technique</th>
<th>Percentage of Scripts</th>
<th>Percentage of Sites</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canvas</td>
<td>25%</td>
<td>88%</td>
</tr>
<tr>
<td>Canvas Font</td>
<td>10%</td>
<td>91%</td>
</tr>
<tr>
<td>WebRTC</td>
<td>5%</td>
<td>6%</td>
</tr>
<tr>
<td>AudioContext</td>
<td>6%</td>
<td>2%</td>
</tr>
</tbody>
</table>
Takeaways

1. Trackers are employing an increasingly diverse set of techniques
2. Measurement heavily influences and controls the adoption of new techniques and tracking norms.
3. Crowdsourced tracking protection misses less popular trackers/techniques
4. Frequent measurement and automated detection provide a path forward

Email: ste@cs.princeton.edu   Twitter: @s_englehardt   Web: senglehardt.com
Canvas fingerprinting returns in the absence of measurement

**May 2014:** 5% of sites  *The Web Never Forgets (Acar, et al.)*

**Aug 2014:** ~0.1% of sites  *(Approximate)*

**Jan 2016:** 2.6% of sites

Percentage of the Alexa top 100k sites
Using Battery Status to Track

The Leaking Battery, Olejnik et. al. (2015)
Using Battery Status to Track

Battery Status:
level: 0.11
dischargeTime: 12867

The Leaking Battery, Olejnik et. al. (2015)
Using Battery Status to Track

Battery Status:
level: 0.11
dischargeTime: 12867

The Leaking Battery, Olejnik et. al. (2015)
Using Battery Status to Track

Battery Status:
level: 0.11
dischargeTime: 12867

The Leaking Battery, Olejnik et. al. (2015)
Using Battery Status to Track

Battery Status:
level: 0.11
dischargeTime: 12867

The Leaking Battery, Olejnik et. al. (2015)
Using Battery Status to Track

Discovered manually in 2 scripts on about 22 sites
(full measurement is future work)

Battery Status:
level: 0.11
dischargeTime: 12867

The Leaking Battery, Olejnik et. al. (2015)
Using AudioContext for fingerprinting

Live test page: https://audiofingerprint.openwpm.com/
Browsers remove BatteryStatus API citing privacy
Google pays $17m to settle Safari cookie privacy-bypass charge

Settlement ends a two-year investigation into Google's cookie practic

By Liam Tung | November 19, 2013 -- 10:03 GMT (02:03 PST) | Topic: Google

Google will pay $17m to settle claims by dozens of US states that it bypassed privacy settings in Apple's Safari browser designed to block third-party ad cookies.

The deal with 37 states and the District of Columbia prevents Google from installing
Multiple settlements for subverting cookie clearing