Third-Party Web Tracking

Franziska (Franzi) Roesner
franzi@cs.washington.edu

CSE 484 Guest Lecture
October 16, 2013
Who am I?

• 5\textsuperscript{th} (& final) year PhD student advised by Yoshi
• I’ve worked on:
  – Automobile security
  – Third-party web tracking
  – Permission granting in smartphones (etc.)
  – Securing embedded user interfaces
  – Security/privacy for augmented reality
Today

• Background on web security
• Understanding web tracking
• Measuring web tracking
• Building new web tracking defenses
Same-Origin Policy

Website origin = (scheme, domain, port)

<table>
<thead>
<tr>
<th>Compared URL</th>
<th>Outcome</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.example.com/dir/page.html">http://www.example.com/dir/page.html</a></td>
<td>Success</td>
<td>Same protocol and host</td>
</tr>
<tr>
<td><a href="http://www.example.com/dir2/other.html">http://www.example.com/dir2/other.html</a></td>
<td>Success</td>
<td>Same protocol and host</td>
</tr>
<tr>
<td><a href="http://www.example.com:81/dir/other.html">http://www.example.com:81/dir/other.html</a></td>
<td>Failure</td>
<td>Same protocol and host but different port</td>
</tr>
<tr>
<td><a href="https://www.example.com/dir/other.html">https://www.example.com/dir/other.html</a></td>
<td>Failure</td>
<td>Different protocol</td>
</tr>
<tr>
<td><a href="http://en.example.com/dir/other.html">http://en.example.com/dir/other.html</a></td>
<td>Failure</td>
<td>Different host</td>
</tr>
<tr>
<td><a href="http://example.com/dir/other.html">http://example.com/dir/other.html</a></td>
<td>Failure</td>
<td>Different host (exact match required)</td>
</tr>
<tr>
<td><a href="http://v2.www.example.com/dir/other.html">http://v2.www.example.com/dir/other.html</a></td>
<td>Failure</td>
<td>Different host (exact match required)</td>
</tr>
</tbody>
</table>

[Example thanks to Wikipedia.]
Same-Origin Policy (DOM)

- Only code from same origin can **access HTML elements** on another site (or in an iframe).

  - **www.example.com** can access HTML elements in the iframe (and vice versa).
  - **www.evill.com** (the parent) **cannot** access HTML elements in the iframe (and vice versa).
Same-Origin Policy (Cookies)

• **For cookies:** Only code from same origin can read/write cookies associated with an origin.
  – Can be set via Javascript (`document.cookie=...`) or via `Set-Cookie` header in HTTP response.
  – Can narrow to subdomain/path (e.g., [http://example.com](http://example.com) can set cookie scoped to [http://account.example.com/login](http://account.example.com/login).)
Same-Origin Policy (Cookies)

- Browsers **automatically include cookies** with HTTP requests.
- **First-party cookie**: belongs to top-level domain.
- **Third-party cookie**: belongs to domain of embedded content.
Same-Origin Policy (Scripts)

• When a website **includes a script**, that script runs in the context of the embedding website.

```html
<head>
  <script src="http://otherdomain.com/library.js"></script>
</head>
```

The code from **http://otherdomain.com** can access HTML elements and cookies on **www.example.com**.

• If code in the script sets a cookie, under what origin will it be set?
Bigger browsing profiles
= increased value for trackers
= reduced privacy for users

(Hypothetical tracking relationships only.)
Tracking is Complicated

- Much discussion of tracking, but limited understanding of how it actually works.
- Our goals:
  - Understand the tracking ecosystem.
    - How is tracking actually done in the wild?
    - What kinds of browsing profiles do trackers compile?
    - How effective are defenses available to users?
  - Address gaps with new defense (ShareMeNot).
Mechanisms Required By Trackers

- **Ability to store user identity** in the browser
  - Browser cookies
  - HTML5 LocalStorage and Flash cookies (LSOs)
  - Not considering more exotic storage mechanisms or approximate fingerprinting

- **Ability to communicate** visited page and user identity **back to tracker**
  - Identity: Cookies attached to requests
  - Visited page: HTTP referrers
  - Both: scripts that embed information in URLs
Tracking: The Simple Version

- **Within-Site**: First-party cookies are used to track repeat visits to a site.
- **Cross-Site**: Third-party cookies are used by trackers included in other sites to create profiles.

Diagram:

- **Cookie Database**: tracker.com: id=789
- **http://site2.com**
  - `<iframe src=tracker.com/ad.html>`
  - ad
  - cookie: id=789
- **processing engine**
  - cookie: id=789
  - 9:30am: user 789 visited site1.com
  - 9:31am: user 789 visited site2.com
- **logs**:
## Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td><strong>Vanilla</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td><strong>Forced</strong></td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>

**Evolution: Embedding analytics libraries**

**Evolution: Third-party cookie blocking**
Quirks of Third-Party Cookie Blocking

• Option blocks the **setting** of third-party cookies: all browsers

• Option blocks the **sending** of third-party cookies: **only Firefox**

• Result: Once a third-party cookie is somehow set, **it can be used** (in most browsers).
Forced Tracking

High-level point:
On most browsers, **if a tracker can ever set a cookie, third-party cookie blocking is rendered ineffective.**
## Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Evolution: Embedding analytics libraries</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td><strong>Evolution: Third-party cookie blocking</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td>Forced</td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td><strong>Evolution: Complex ad networks</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>
Referred Tracking

High-level point:
One tracker with client-side state can enable tracking by partners without client-side state.
## Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Type (Name)</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Evolution: Embedding analytics libraries</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td><strong>Evolution: Third-party cookie blocking</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vanilla</td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td><strong>Evolution: Complex ad networks</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forced</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td><strong>Evolution: Social networks</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>
Personal Tracking

- Just loading these buttons (not clicking on them) enables tracking.
- Users visit these sites directly.
- This tracking is often not anonymous (linked to accounts).
## Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td><strong>Vanilla</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td><strong>Forced</strong></td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td><strong>Referred</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td><strong>Personal</strong></td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>

### Evolution:
- **Embedding analytics libraries**
- **Third-party cookie blocking**
- **Complex ad networks**
- **Social networks**
Measurement Study

• Tool: TrackingTracker Firefox add-on that crawls the web and automatically categorizes trackers.

• 3 data sets
  – Alexa Top 500
    • 5 pages per domain: main page and up to 4 links
  – Alexa Non-Top 500
    • Sites ranked #501, #601, #701, etc.
    • 5 pages per domain: main page and up to 4 links
  – AOL search logs
    • 300 unique queries for 35 random users
Tracking Prevalence (Top 500)

- 524 unique trackers on 500 domains

- 457 domains (91%) embed at least one tracker.
  (97% of those include at least one cross-site tracker.)

- 50% of domains embed between 4 and 5 trackers.

- One domain includes 43 trackers.
Top 20 Trackers on Top 500 Domains

- **Within-Site**
- **Cross-Site (Personal)**
- **Cross-Site (Anonymous)**

Tracker Prevalence (# Domains)

- google-analytics.com: 297
- doubleclick.net: 189
- facebook.com: 154
- google.com: 149
- quantserve.com: 109
- twitter.com: 105
- atdmt.com: 93
- revsci.net: 81
- advertising.com: 60
- addthis.com: 45
- adnxs.com: 44
- serving-sys.com: 40
- youtube.com: 34
- addthiscdn.com: 33
- bluekai.com: 32
- mediaplex.com: 30
- 29
- 27
- 26
Each line represents one user.

AOL Users' Profile Sizes by Top 20 Cross-Site Trackers

- **Doubleclick:**
  - Avg 39% (Max 66%)

- **Facebook:**
  - Avg 23% (Max 45%)

- **Google:**
  - Avg 21% (Max 61%)
LocalStorage and Flash Cookies

• Surprisingly little use of these mechanisms!

• Of 524 trackers on Alexa Top 500:
  – Only 5 set unique identifiers in LocalStorage
  – 35 set unique identifiers in Flash cookies

• Respawnning:
  – LS → Cookie: 1 case; Cookie → LS: 3 cases
  – Flash → Cookie: 6 cases; Cookie → Flash: 7 cases
Building New Systems

1. **ShareMeNot**: defense vs. personal tracking
   - Still allows social media widgets to be used.

   ![Not Allowed](http://sharemenot.cs.washington.edu)

2. **TrackingObserver**: a platform for web tracking detection, measurement, prevention
   - Dynamic detection improves on state-of-the-art blacklist methods.
www.franziroesner.com
franzi@cs.washington.edu