Trackers Bounce Back: Measuring Evasion of Partitioned Storage in the Wild

Audrey Randall
University of California, San Diego
United States
aurandall@eng.ucsd.edu

Peter Snyder
Brave Software
United States
pes@brave.com

Alisha Ukani
University of California, San Diego
United States
aukani@ucsd.edu

Alex Snoeren
University of California, San Diego
United States
snoeren@cs.ucsd.edu

Geoff Voelker
University of California, San Diego
United States
voelker@cs.ucsd.edu

Stefan Savage
University of California, San Diego
United States
savage@cs.ucsd.edu

Aaron Schulman
University of California, San Diego
United States
schulman@cs.ucsd.edu

ABSTRACT

This work presents a systematic study of navigational tracking, the latest development in the cat-and-mouse game between browsers and online trackers. Navigational tracking allows trackers to aggregate users’ activities and behaviors across sites by modifying their navigation requests. This technique is particularly important because it circumvents the increasing efforts by browsers to partition or block third-party storage, which was previously necessary for most cross-website tracking. While previous work has studied specific navigational tracking techniques (i.e. “bounce tracking”), our work is the first effort to systematically study and measure the entire category of navigational tracking techniques. We describe and measure the frequency of two different navigational tracking techniques on the Web, and find that navigational tracking is present on slightly more than ten percent of all navigations that we made. Our contributions include identifying 214 domains belonging to at least 104 organizations tracking users across sites through link decoration techniques using direct or indirect navigation flows. We identify a further 23 domains belonging to at least 16 organizations tracking users through bounce tracking (i.e. bouncing users through unrelated third parties to generate user profiles). We also improve on prior techniques for differentiating user identifiers from non-sensitive information, which is necessary to detect one class of navigational tracking. We discuss how our findings can be used to protect users from navigational tracking, and commit to releasing both our complete dataset and our measurement pipeline.

ACM Reference Format:
Audrey Randall, Peter Snyder, Alisha Ukani, Alex Snoeren, Geoff Voelker, Stefan Savage, and Aaron Schulman. 2022. Trackers Bounce Back: Measuring Evasion of Partitioned Storage in the Wild. In Proceedings of ACM Conference (Conference’17). ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Over the past few years, tensions have deepened between those collecting detailed user behavior data for advertising purposes and privacy-conscious users who wish to be unmonitored. While there are some efforts to find a compromise between these positions (e.g., allowing the collection of aggregated, anonymized data [12, 28]), none have yet managed to satisfy advertisers or privacy advocates [17, 45, 52]. In the absence of such a solution, privacy-focused browsers (i.e., browsers for which privacy is seen as a competitive advantage) have rolled out changes that block one of the core mechanisms used by trackers to collect and associate information about a user across different websites, thereby building a profile of that user.

Previously, trackers could aggregate information on users across websites by storing information about a user on one site that was visible to that tracker on all other sites that included its scripts. Trackers commonly used third-party cookies for this purpose, although any type of storage could be used, including local storage, IndexedDB, and the cache [4]. Trackers could use this shared storage to build shared state for each user across every website that contained the tracker’s scripts. However, several browsers are now employing an anti-tracking defense called “partitioned storage,” which breaks this sharing assumption. By partitioning all storage by the domain of the top level website, a tracker should not be able to use this state to associate information they glean about a user on one site, with the information they collect about the user on any other site.

However, trackers have responded by implementing a new class of tracking technique: navigational tracking. Navigational tracking is the process of tracking a user across websites by modifying their...
navigation requests. The tracker accomplishes this style of tracking by either adding identifying information, which will then be shared across first party boundaries, or by momentarily redirecting the user to its own domain, where it can record information as a first party itself, or both. In each case, navigational tracking allows trackers to link information about users across sites, circumventing the browser’s attempt to partition such information.

Navigational tracking builds on, and differs from, previously well-known tracking techniques such as “bounce tracking” and “cookie syncing”. Navigational tracking builds on bounce tracking by allowing trackers to aggregate information about users’ behavior on websites, in addition to the websites’ URLs that bounce tracking already collects. Navigational tracking also exceeds the capabilities of cookie syncing: cookie syncing allows trackers to pass information to each other on a single site, but does not allow them to share information across websites when storage is partitioned. Navigational tracking represents the next step in the cat-and-mouse game between browsers and trackers. It restores trackers’ ability to aggregate user data across websites, evading the most recent anti-tracking protections from browsers.

This work presents a systematic measurement of navigational tracking. In so doing, we make the following contributions to understanding online tracking and improving Web privacy:

1. We perform the first systematic measurement of navigational tracking in the wild.
2. We construct a multi-stage analysis pipeline, nicknamed CrumbCruncher, to crawl the Web and measure how frequently each navigational tracking technique occurs.
3. We categorize the behaviors of trackers, including which categories of sites are more likely to engage in navigational tracking.
4. We improve on prior techniques for differentiating user identifiers from other values.
5. We contribute to countermeasures against navigational tracking, by sharing a dataset of sites that participate in each form of it.

The remainder of our work is organized as follows. Section 2 covers the background of navigational tracking and related work. Section 3 describes the design of our crawler, CrumbCruncher, and its capabilities and limitations. Section 4 presents our findings, including the most common participants in navigational tracking and a summary of their behaviors and categories. Section 5 describes the limitations of our work. Section 6 details our contribution to the countermeasures that various entities have taken against navigational tracking. Section 7 details related work, and Section 8 concludes.

2 BACKGROUND

Navigational tracking is an approach advertisers use to sidestep privacy measures in web browsers that block cross-site tracking. In this section, we describe what marketing capabilities advertisers lose when browsers block cross-site tracking, how browsers block cross-site tracking, and how advertisers regain that capability with navigational tracking. We also describe how navigational tracking differs from other tracking methods, such as bounce tracking and cookie syncing.

2.1 Why advertisers track users across websites

Advertisers benefit from being able to track a user’s activity across visits to different websites. Two of the primary marketing functions that depend on cross-site tracking are affiliate marketing and identity resolution.

Affiliate marketing is a marketing model in which one website drives traffic to another website and then receives a commission for each visitor who makes a purchase on the other website. This creates a countermeasure between script contexts and the location of their cookies in the hierarchical namespace. First-party cookies are stored in the top level of the namespace, but third-party cookies are stored in “folders” beneath them. Consequently, if a tracking script is loaded on two different websites, it no longer stores and reads cookies in the same location in the namespace.

2.2 Third-party cookies made cross-site tracking easy

For over a decade, browsers allowed advertisers to perform cross-site tracking functions with third-party cookies. Advertisers that showed up on another website could simply leave a cookie on a browser to record that user’s identity. Then, that identity would be shared across all websites that contained the advertiser’s scripts. Third-party cookies present a threat to privacy, because they make it so easy to track a user across different websites. Therefore, they are slowly being phased out. In particular, several popular browsers have implemented partitioned storage to isolate third-party cookies so they cannot be used for cross-site tracking. Several browsers implement this isolation by default, including Firefox, Safari, and Brave [30, 35, 53], other browsers provide the option to isolate third-party cookies.

Third-party cookie isolation is implemented as follows. Instead of storing cookies in a “flat” namespace, the browser uses a hierarchical one, where the hierarchy is based on the source of the script or tag that stored the cookie. This is similar to storing files in a directory structure instead of a single folder. Figure 1 shows the relationship between script contexts and the location of their cookies in the hierarchical namespace. First-party cookies are stored in the top level of the namespace, but third-party cookies are stored in “folders” beneath them. Consequently, if a tracking script is loaded on two different websites, it no longer stores and reads cookies in the same location in the namespace.
Hierarchical Cookie Storage

2.3 Sidestepping Partitioned Storage with Navigational Tracking

Advertisers have been able to sidestep browsers’ partitioned storage protections to perform cross-site tracking using a technique called navigational tracking. Advertisers and marketers also refer to navigational tracking as “server-to-server” or “cookie-less” tracking.

Navigational tracking works by modifying the user’s navigation requests, either by redirecting the user to a third-party tracker before redirecting to the intended destination, or by adding information to the navigation URLs in the form of query parameters, or both. Figure 2 shows this process in detail.

In navigational tracking, the user is sent through a navigation path. This path begins at the originator website, where the user clicks a link. It may then pass through zero or more redirectors, which are invisible to the user but have the right to store first-party cookies. Finally, the user is sent to the site the link originally pointed to, the destination. These redirects make it possible to track a user’s identity across websites, even when the originator, redirector(s), and destination are all different first-parties (i.e., they may come from different domain names).

The way that navigational trackers achieve this is by sharing user identities across websites with a technique called link decoration. An example of link decoration is shown in Figure 2. Link decoration is when the originator website, or a third-party script on that site, appends a user’s identity to a link to an external site, specifically in the form of a query parameter. Any site on the navigation path that follows from clicking on that link, whether redirector or destination, can see the information in the query parameter and store it as a first-party cookie under its own domain name. These cookies are shown in Figure 2 highlighted in yellow. Navigational tracking can serve as a replacement for some use cases for third-party cookies if a website passes its user identifier (UID) as a query parameter to the redirected website.

UID-sharing is undesirable because it evades the browser’s partitioning of information between first parties. It allows identifiers to be linked across partitioned cookie storage. Consider the case in Figure 2, where a tracking script exists on both the originator and the destination websites. Assume that tracking script was loaded from the same source on both websites. If the browser uses partitioned storage to block third-party cookies, the tracking script cannot share the UID with the originator and the destination using a cookie. However, if the script shares the UID with the destination by passing it as a query parameter, the tracker can track the user’s identity across both sites. Any redirector in the navigation path can store the UID from the query parameter as a first-party cookie, as Figure 2 shows highlighted in yellow.

Note that a redirector is not always needed to share identities between sites with link decoration; an originator site and destination site can decide to directly share user identifiers with each other. This can be used for affiliate marketing because it allow an advertiser to link the user’s identity across cookies stored for two different domain names. Redirectors provide the benefit of being able to track user behavior in aggregate across many sites. Indeed, redirectors make it possible to perform Identity Resolution without third-party cookies by linking identities of users across different websites. Also, if the tracker is not confident that the destination website contains one of its tracking scripts, an advertiser may use a redirector to store the originator’s UID. In addition, the tracker might use redirectors as fallback mechanisms, in case the destination blocks its tracking script in some cases redirectors may not be necessary to track users across sites, but they do so anyway, providing an inefficient implementation of navigational tracking.

A weaker form of navigational tracking is known as “Bounce Tracking”. In Bounce Tracking, trackers do not set or read identifiers from the URL (and so are not able to link identifiers across first-party cookies). Instead, the tracker only collects information about the user’s browsing behavior the tracker is able to build.

2.3.1 We do not know the popularity of navigational tracking. Navigational tracking functionality is advertised on the websites of several ad tech companies, such as Tinuiti, CJ, Rakuten Advertising, Twilio Segment, and others [1, 6, 14, 42, 44]. However, it is unclear how widely these new techniques have been adopted. Navigational tracking methods are reportedly more difficult to implement than third-party cookie-based methods, and require extra work on the advertiser’s side [29]. According to one advertiser blog, most advertisers still use cookie-based tracking pixels rather than navigational tracking pixels [5]. On the other hand, a different affiliate advertising newsletter stated that navigational tracking is now a “common framework” implemented across “over 80% of advertisers” [50].

Koope et al. [36] studied the prevalence of bounce tracking. However, the prevalence of Navigational tracking is yet to be studied. Therefore, we perform the first study of the prevalence of navigational tracking in the wild.
2.4 Differences from previous tracking techniques

Navigational tracking is most closely related to two previously studied tracking techniques, bounce tracking and cookie syncing. However, navigational tracking builds on, and differs from, these techniques in several ways.

2.4.1 Navigational tracking builds on bounce tracking. Navigational tracking is a superset of bounce tracking, and some forms of navigational tracking allow trackers to build much more privacy-invasive profiles of users than bounce tracking alone. A bounce tracker can record which originator and destination websites a user has visited, and link this list to its UID for the user, which it can store as a first party cookie. However, a bounce tracker cannot link any information about a user’s behavior on websites to this UID when the browser uses partitioned storage. A link decorator tracker can not only view which websites a user visits and tie that list to its UID, but can also link information it gleans about the user’s behavior from the originator and destination.

Consider the following example: Say a link decorator tracker modifies navigation paths to insert itself as a redirector, and also has scripts on the originator and destination. The link decorator tracker’s scripts can collect information on the originator, such as the links the user clicks, purchases the user makes, and so forth, and send it back to its server along with the UID from the originator. When the user clicks a link, the tracking script decorates it with the user identifier from the originator and inserts a redirect to its own domain. Now, the tracker can associate its own first-party cookie for this user with the UID its script collected on the originator site, yielding far more information about the user than the simple knowledge of which sites the user visited.

2.4.2 Navigational tracking is not cookie syncing. Cookie syncing is when multiple third parties on a single first-party site share user identifiers with each other. However, third parties cannot share information across first party websites using cookie syncing when partitioned storage is in place. For example, if a group of trackers appears on the originator, those trackers can share the user identifier from the originator between themselves. However, assume that the user then navigates to the destination, which contains the same group of trackers. Those trackers no longer have access to their stored information (such as cookies) about what the user did on the originator, because the storage available to them on the destination is partitioned away from the storage on the originator. The trackers may share the destination’s UID between themselves as well, but they cannot link that identifier to the one from the originator.

3 METHODOLOGY

We implemented CrumbCruncher using both Puppeteer, to automate site visits and record cookies and local storage, and a custom Chrome extension, to record web requests. 1 CrumbCruncher begins by visiting a “seeder domain”, which we draw from the Tranco ranking of the globally most-popular 10,000 domains [37]. It then performs a ten-step random walk starting at that seeder domain, using the following methodology.

3.1 Performing the Random Walk

To make the first step of the random walk, CrumbCruncher first allows the website of the seeder domain to load completely. Some seeder domains taken from the Tranco rankings do not actually host user-facing websites: in fact, many are high in the ranking because they are commonly visited as third party advertisers. These websites return various errors, including HTTP 400 or 500 response codes or “connection refused” errors, when CrumbCruncher attempts to visit them. If CrumbCruncher cannot load a seeder domain’s website, it moves on to the next.

Next, CrumbCruncher records all cookies and local storage values under the top-level frame. (We refer to these collectively as “storage values” in the remainder of the paper.) CrumbCruncher then chooses either a frame (<iframe>) or an anchor (<a>) element to click on. It prioritizes choosing an element that will navigate to a site it has not visited yet. Failing that, CrumbCruncher next prioritizes visiting a site with a different eTLD+1 (i.e. a site with a

---

1 While Puppeteer is capable of recording most web requests, it cannot guarantee that it can attach request handlers before any requests on a page have been sent [10, 11]. We found during initial testing that this led to a significant number of missed requests, and realized CrumbCruncher needed to record requests using a browser extension instead.
different origin) than the current site. This requirement is important because navigational tracking only occurs when information is transferred across first-party contexts, and thus across sites with different tTLDs. If CrumbCruncher cannot find an element that will lead to a site with a different origin, it selects an element to click at random.

Once it has selected an element, CrumbCruncher clicks it. The click either causes the current tab to load a new page, a new tab to be opened, or no change to occur. The latter case might occur for several reasons. For one example, the clicked element may not have been intended to cause a navigation: it might instead cause a dialog box to appear or an animation to play. For another example, the navigation may not have caused a new page to load. Hash-only navigation (i.e., navigating from “https://example.com” to “https://example.com#foo”) exhibits this behavior: it does not cause a new document request to be sent, which means a new page does not load. If the clicked element causes a navigation to occur, CrumbCruncher waits for ten seconds to collect storage values and then exits. If the clicked element did not trigger a navigation, CrumbCruncher chooses a new element to click.

For each click, the browser extension collects all web requests and sends them to a local “recording server,” a simple HTTP server that writes requests to disk. A separate recording server is necessary because browser extensions generally cannot perform file I/O by themselves.

This process constitutes one step in the random walk. CrumbCruncher repeats this process, starting at each new page loaded by the click in the previous step, nine more times. It then selects a new seeder domain to start the next random walk.

3.2 Multiple parallel crawlers
The primary purpose of this work is to measure the prevalence of link decoration that is used to transfer user identifiers. While a single crawler is capable of detecting whether link decoration has occurred, it cannot tell definitively if the information that decorated the URL is a user identifier or not. Prior work made this determination in two ways: by using heuristics based on the potential identifier’s format ([23, 24]) and by running two crawlers in parallel with different user profiles ([25, 36]). If a token is the same across two different browser profiles, it is unlikely to be a user identifier. However, we note that session identifiers and other transient properties can also be different across different profiles, even if their purpose is not to identify a user. CrumbCruncher improves on prior work by running four crawlers for each step of the random walk. Each crawler visits the same starting website, clicks on the same link, and arrives at the same destination site (except in the case of errors, discussed further in Section 3.3 below). The importance of using four crawlers to accurately identify tracking is explained below.

3.2.1 The four crawlers. The first two crawlers, which we call “Safari-1” and “Safari-2”, have different user profiles and use a Safari User-Agent string. If a token is identical in both crawls, it cannot distinguish different users and we discard it from our analysis. The third crawler, “Safari-1R”, runs after Safari-1 and Safari-2 have finished one step of the random walk: it repeats the same crawl step using the same user profile as Safari-1. If a token is different across Safari-1 and Safari-2, but is the same for Safari-1 and Safari-1R, it is likely to be a user-identifying token. If the token is different across all three crawlers, it is likely to be a non-user-identifying token, such as a session identifier.

The fourth crawler, “Chrome-1,” also has a separate user profile but uses a Chrome User-Agent string. Its purpose is not to distinguish user identifiers, but to check whether navigational tracking occurs at the same frequency across browsers. This strategy did not work out as well as we had hoped. We discovered that even when crawlers are kept synchronized as much as possible, an instance of navigational tracking that appears on one might not appear on another, due to the dynamic nature of sites on the Web. For more details, see section 4.4.

However, we were able to use Chrome-1 to distinguish user identifiers in cases where a navigational tracking instance did not occur on the three Safari crawlers, but did occur on Chrome-1. Chrome-1 runs in parallel with the first two crawlers, Safari-1 and Safari-2.

3.2.2 Coordinating the crawlers. CrumbCruncher uses a central coordinator to synchronize the execution of Safari-1, Safari-2, and Chrome-1. The central coordinator is another local HTTP server. In addition to coordinating crawler execution, it also chooses the element that the three crawlers should click under the constraint that the same element appears in the content loaded by all three crawlers. This choice is not always straightforward. Even when accessed simultaneously, websites often load dynamic content: elements that appear on one crawler’s website might not appear on the others.

Upon loading a site, the three parallel crawlers send a list of all anchor and iframe elements to the central coordinator. These lists contain the elements’ properties, location, bounding boxes, and path. The coordinator compares the three lists to find elements that are the same across all three instances of the site. We consider elements to be the same if any of three heuristics are met:

1. They are anchors and their href values are the same (not including query parameters).
2. They have the same property names (the values may differ) and similar bounding boxes (the y-coordinate may differ, to allow for elements that render at different heights on the page).
3. They have the same property names and the same path.

These heuristics are imperfect: they may incorrectly label elements as the same when they are not, or incorrectly discard elements. To mitigate these possibilities, CrumbCruncher compares the fully qualified domain name (FQDN) of the site each crawler has landed on at the end of every crawl step. If all three FQDNs are not the same, CrumbCruncher terminates that random walk. We still include this step in our analyses, because it often occurs when CrumbCruncher has clicked on different advertisements that each exhibit a separate instance of navigational tracking.

3.3 Crawler Limitations
Figure 3 shows how many individual crawl steps failed and the reasons for the failures. CrumbCruncher retries each failed crawl step once; if the failure occurs a second time, it halts that random walk and moves to the next seeder domain. The majority of
failures occurred because CrumbCruncher was unable to find an element that was the same across all three synchronized crawls, or because elements were erroneously determined to be the same. Another source of error was network errors such as “Connection Refused.” These errors usually occurred on seeder domains that did not point to user-facing websites (for example, domains that are popular because they are used in advertisement networks). A less common error occurred when Puppeteer failed to attach the `domcontentloaded` handler before the `domcontentloaded` event fired: this happens because Puppeteer cannot guarantee that its own code runs before any page events have occurred [10, 11].

3.4 Collecting navigational tracking instances

We run CrumbCruncher on twelve “t2.large” Amazon EC2 instances. Each EC2 instance has a different set of 834 seeder domains. The full crawl of 10,000 seeder domains takes approximately three days to complete.

Once CrumbCruncher has finished its random walk of the full set of seeder domains, we search for any user-identifying tokens that are transferred across first-party contexts. We define “different first-party contexts” as the case when the originator (the site the token was originally found on) has a different eTLD+1 than any of the redirectors or the destination site (the sites the token was passed to via a query parameter). We use the term “token” to refer to the value of a name-value pair, whether that pair is a cookie, a local storage object, or a query parameter. For example, in Figure 2, the token’s name would be “uid” and the token itself would be “123.” We do not look for tokens in the names of name-value pairs because Fouad et al. found that storing user identifiers as names rather than values was a very uncommon practice [25].

We extract tokens from query parameters by recursively attempting to parse each one as JSON or URL-encoded values until we have extracted every individual token. For example, if a query parameter contains a JSON string that itself contains several URL-encoded tokens, we extract each URL-encoded token individually.

For each crawl step, we extract all tokens that CrumbCruncher has encountered. We then remove all of the tokens that were not passed across first-party contexts. Next, we attempt to distinguish the user identifiers.

3.4.1 Finding user identifiers. We compare the values of each token across the three crawlers Safari-1, Safari-2, and Safari-1R. If the token’s value is the same in Safari-1’s and Safari-1R’s crawls, and different in Safari-2’s crawl, we consider it a user identifier. If not, we discard it.

However, we found that the majority of potential navigational tracking instances did not occur on all four crawlers: in fact, many occured on only one. For example, if each crawler loaded the same originator website and clicked the same iframe element, but the iframes contained different advertisements, each advertisement might present a different navigation path and therefore a different instance of navigational tracking. If a token is present in two crawls with different profiles, we can still discard it if its value is the same across crawls. Similarly, we can discard tokens that have multiple values in crawls with the same user profile. However, even after discarding all the tokens we can, a large number remain that might not be user identifiers: both tokens that were present in two crawls, and tokens that were only present in a single crawl. We therefore employ two additional methods to filter out user identifiers in these cases: programmatic and manual heuristics.

Our programmatic heuristics are similar to those of [23, 24, 36]. We remove tokens that appear to be dates or timestamps, tokens that appear to be URLs, and tokens that are less than seven characters long. We do not impose any restrictions based on cookie expirations, as in some previous work [23, 24]. However, even after applying these filters, manual inspection of the remaining tokens revealed a high number of obvious false positives. These

| Profiles                                      | # Tokens |
|----------------------------------------------|----------|
| Two identical profiles, one or more different profiles | 325      |
| Two or more different profiles only          | 171      |
| Two identical profiles only                  | 20       |
| One profile only                             | 445      |

Table 1: Combinations of profiles that user identifiers appeared on.
included natural language strings separated by delimiters (such as "Dental_internal_whitepaper_topic," "share_button"), concatenated words with no delimiter ("sweetmagnolias," "trustpilot"), semi-abbreviated words ("navimail"), acronyms ("en-US"), and more. Filtering most of these out programmatically presented a significant challenge. We therefore concluded that programmatic heuristics would be insufficient to distinguish user identifiers from other tokens, and resorted to removing obvious false positives by hand. Our final, conservative, strategy was to remove tokens that were composed of any combination of natural language words, coordinates, domains, or obvious acronyms like "en-US." Table 1 shows how many of the final set of user identifiers were present on which combinations of profiles.

In the end, we identified 577 out of 1581 tokens that we had to manually remove because the programmatic filters failed to recognize them. This number is significantly higher than we expected, and underscores the value of using multiple crawlers to distinguish user identifiers compared with manual heuristics.

### 4 RESULTS

In total, we observed 10,814 unique navigational paths using CrumbCruncher. We define a unique navigation path as the unique list of URLs visited during the navigation from originator to destination, including the originator and destination themselves. We consider unique navigation paths rather than total navigational paths since CrumbCruncher visited some websites multiple times, and clicked the same link multiple times if it could find no alternative that allowed the crawl to move forward. This resulted in many duplicate cases of navigational tracking.

Among those paths, we discovered a surprising number of instances of navigational tracking with link decoration. Specifically, in the paths that CrumbCruncher crawled, link decoration (i.e., sharing UIDs across sites) appeared on 8.11% of the paths, and bounce tracking on 2.74% of the paths.

It is particularly surprising that such a significant percentage of advertisers have implemented navigational tracking, especially given that Chrome—the most widely used web browsers—still permits tracking with third-party cookies by default. Additionally, it is surprising that link decoration is more common than bounce tracking. Although link decoration can yield much more information about a user than bounce tracking can alone, link decoration has a higher implementation and operational cost than bounce tracking (e.g., scripts to share identifiers across sites). We speculate that the affiliate advertising market may be driving the adoption of navigational tracking: affiliate programs have reportedly been failing to attribute conversions because of browsers’ third party cookie blocking [29], and link decoration allows conversions to be attributed correctly.

In the rest of this section we break down the navigational tracking we discovered in detail to understand who is implementing it, how they implement it, and why they implement it.

#### 4.1 Link decoration tracking

##### 4.1.1 Link decoration redirectors

We first evaluate what domain names we observed acting as redirectors in link decoration-based tracking. Table 2 shows the most commonly-occurring redirectors we found in the navigation paths we observed. For this figure, instead of discussing unique URL chains, we instead focus on unique chains of fully qualified domain names (FQDNs). This allows us to capture how widely a redirector is spread across the web, without over-counting instances many repeated instances of navigational tracking that originate on a single website. Indeed, we observed the same navigational path appeared on many links on the same website, so our metric of unique FQDN chains does not give more weight to redirectors appear on multiple identical navigational paths on the same website.

We further classify the redirectors listed in Table 2 into two groups: "trackers" and "potential trackers." (marked with an asterisk). Trackers are redirectors that we observe to have observed to be exhibiting clear behavior that is indicative that they are a tracker used by multiple websites. We develop a conservative heuristic that allows us to identify redirectors that appear to have no other purpose in the navigation paths besides tracking (i.e., collecting and storing UIDs across multiple websites).

| Redirector                                      | Count | % of FQDN Chains |
|-------------------------------------------------|-------|------------------|
| adclick.g.doubleclick.net                       | 36    | 11.2             |
| googleads.g.doubleclick.net                     | 20    | 6.2              |
| advance.lexis.com                              | 10    | 3.1              |
| d.agkn.com                                      | 9     | 2.8              |
| blds.zog.link                                  | 9     | 2.8              |
| ad.doubleclick.net                              | 8     | 2.5              |
| gm.demdex.net                                   | 8     | 2.5              |
| www.kinopoisk.ru*                               | 7     | 2.2              |
| secure.jbs elsevierhealth.com                   | 6     | 1.9              |
| t.myvisualiq.net                               | 6     | 1.9              |
| 11173410.searchiqnet.com                       | 6     | 1.9              |
| optout.hearstmags.com*                         | 6     | 1.9              |
| signin.lexisnexis.com*                         | 6     | 1.9              |
| trc.taboola.com                                 | 5     | 1.6              |
| l.instagram.com*                                | 5     | 1.6              |
| ads.adfox.ru*                                   | 5     | 1.6              |
| www.facebook.com*                               | 5     | 1.6              |
| reseau.umontreal.ca*                            | 5     | 1.6              |
| l.facebook.com                                 | 4     | 1.2              |
| rtb-use-mfadsrvr.com                            | 4     | 1.2              |
| www.campaignmonitor.com*                       | 4     | 1.2              |
| 6102.xg4ken.com*                                | 4     | 1.2              |
| swallowcrockerybless.com*                       | 4     | 1.2              |
| montreal.imodules.com*                         | 4     | 1.2              |
| www.getfeedback.com*                            | 4     | 1.2              |
| kuwosm.world.tmall.com*                        | 4     | 1.2              |
| www.awin1.com*                                  | 3     | 0.9              |
| www.zenaps.com*                                 | 3     | 0.9              |
| pr.ybp.yahoo.com*                              | 3     | 0.9              |
| go.dgdp.net                                     | 3     | 0.9              |

Table 2: Counts of the most common link decorator trackers observed in unique FQDN chains. *Potential tracker
We also use this categorization in Figures 7 and 8.

Note that this heuristic has limitations. The less often CrumbCruncher sees a redirector, the less likely it is to observe multiple originators and destinations for that redirector, which means it does not classify that redirector as a tracker. Consequently, our “potential tracker” category contains several known trackers as well. We also use this categorization in Figures 7 and 8.

We find that many of the top redirectors that we observe in many unique paths are also classified as trackers using this heuristic. This means that there are several popular advertisers that use navigational tracking, likely as an alternative to third-party cookies. We also found there are many less-common redirectors that appear in multiple navigation paths but are not classified as trackers by this heuristic. This may indicate that there are specific FQDNs used for implementing navigational tracking across all of the websites within an organization.

In Table 2, we see that the most commonly used navigational trackers are DoubleClick, which is an advertising company owned by Google, and agkn.com, which is part of the NeuStar AdAdvisor service. The third most common domain we observe, advance.lesxis.com, redirects to a signin page for a company called LexisNexis [2]. We speculate that advance.lesxis.com may be an example of a redirector whose purpose is to direct unauthenticated users to a signin page, rather than to track them. Otherwise, there is a long tail of redirector domain names used in a small number of navigational paths.

4.1.2 Link decoration originators and destinations. Next, we investigate which organizations acted as as originators or destinations during navigational tracking most frequently. An organization is defined as the owner of an FQDN, which we discover using the Disconnect entity list [20] when possible. An entity is counted once per unique FQDN chain. We were able to attribute 104 out of 214 FQDNs to organizations in the Disconnect entity list. Where we were not able to attribute an FQDN to its owning organization, we left the FQDN as a domain.

Figure 4 shows that the most common originator in our measurements is Sports Reference LLC, an organization that maintains several websites with statistics for popular American sports. This company owns several sports-themed domains whose websites link frequently to each other, such as stathead.com, hockey-reference.com, baseball-reference.com, and others [3]. CrumbCruncher spent several random walks in this ecosystem of websites, and we observed that most of the links between Sports Reference LLC’s sites performed link decoration and shared user identifiers. Sports Reference LLC may be attempting to get around the domain-based partitioning that browsers employ, in order to share information through all of its affiliated websites. Even though these websites have different domains, they are owned by the same organization. Other common originators and destinations, such as Facebook and Google, are perhaps not surprising since Facebook and Google both participate in navigational tracking as redirectors as well, and also were often visited by our crawler because of their popularity.

We further break down the originators and destinations by placing them into categories by their purpose. We use the categorization defined by the IAB Tech Lab Content Taxonomy [31] as provided by Webshrinker [55] in 2019. The dataset contains 404 domain categories [54]. Out of 339 unique second level domains, we were able to categorize 307. Figure 5 shows which categories of websites that participate in navigational tracking are most common in our dataset. The counts of websites per category reflect the number of unique FQDNs in that category, so that each FQDN is represented once even if CrumbCruncher encountered it multiple times. For example, even though Facebook is a common originator as seen in Figure 4, in Figure 5 it only shows up twice as an originator; once for facebook.com and once for instagram.com, both in the “Social Networking” category.

Interestingly, “News/Weather/Information” is the second most common category overall, and the most common category for originators. This result is consistent with previous studies that found news websites have an above-average amount of more traditional tracking mechanisms, such as fingerprinting and tracking pixels [23, 32]. We speculate, based on the contents of a few of these originators, that news websites may have an above-average number of advertisements in iframes that perform navigational tracking when clicked.

4.1.3 Third parties. After a UID has been transferred through the entire navigation path, it may not have finished its journey: third parties on the destination site may also send it back to their own servers. Figure 6 shows the 20 most common eTLD+1s of the targets of web requests that included UIDs sent from destination sites. In some cases, the web request to the third party might only contain the UID because the request contains the entire URL of the destination site, which still has the UID as a query parameter. However, since the third party server still gained access to the UID, we include all requests here, even if they did not separate out the UID and send it deliberately.

The websites listed in this figure include many common trackers, such as Google Analytics and DoubleClick, as well as some less well-known ones like Polar and Yotpo.

4.2 Link Redirector Behavior

Figure 8 shows the paths that user identifiers take as they are transferred between top level frame domains. The second bar, “Originator to Destination,” shows the number of identifiers that were transferred through direct frame decoration (i.e. directly between domains without any redirects) and thus no redirectors existed in that navigation path. We observe that the majority of tokens are transferred across the entire path from originator to destination. This is in part because redirectors in the middle of a chain often pass on the entire URL of the intended destination, and if that URL was decorated with the identifier by the originator, all redirectors see the decorated URL and therefore can record the UID if they choose.

We also see cases where UIDs are transferred through part of the navigation path, but not all of it. We hypothesize that these represent cases where the entities involved have chosen to share
Trackers Bounce Back: Measuring Evasion of Partitioned Storage in the Wild

Conference '17, July 2017, Washington, DC, USA

Figure 4: Most common entities involved in navigational tracking as originators or destinations.

Figure 5: Categories of websites that participate in navigational tracking as originators or destinations.

Figure 6: Most common domains involved in requests from the destination site.

directly from the originator to the destination without passing through any redirectors in between. When there is at least one tracker in a redirection chain, it is most often the case that there is only one tracker; but it cases of two or three trackers appear as well. The most common trackers that appear in navigation path lengths of exactly 3 (i.e. only one redirector) are googleads.g.doubleclick.net and adclick.g.doubleclick.net. We also see that the longer chains with at least one known tracker tend to involve multiple trackers — not just one tracker with multiple non-tracking redirectors. The most common chain involving at least two trackers is from awin1.com to zenaps.com (both of which are owned by AWIN),
though no sequence of redirectors dominates here. We note that the probability of at least one known tracker (as defined in Section 4.1.1) being involved in a navigation path increases as the number of redirectors does. This suggests that long chains of redirectors are likely used so that multiple trackers can share information. For example, we observed that one navigation path starting at a coupon-collecting website (couponfollow.com) passed through a partner site to couponfollow.com owned by the same person, then passed through four different trackers before arriving at the final destination (a retailer). Each of these trackers had the ability to record information about the ad the user had clicked and their apparent interest in the retailer’s products.

4.3 Bounce tracking

In addition to our study of link decoration, we collected the redirectors in navigation paths that did not transfer any UIDs: the potential bounce trackers. We filtered these redirectors to find the probable trackers using the same method described in Section 4.1.1.

The remainder are shown in Table 3. The “Instances” column signifies the number of times that each tracker, or combination of trackers, appeared as a redirector in each unique FQDN chain. A right arrow (→) indicates that the leftmost tracker always appeared immediately before the rightmost tracker in the navigation path. A double-headed arrow (↔) indicates that trackers in that group appeared in several different orders within the navigation chain. We separate groups of trackers from individual trackers for two reasons: first, because trackers would otherwise be double-counted in Table 3, and second, to show the trackers that apparently have partnerships with each other. We group multiple domains owned by the most common tracker, DoubleClick, for brevity. If the bounce tracking domain was also seen participating in link decoration at any point, we mark it with a “Y” in the “Link decorator?” column. For groups of trackers, the “Link decorator?” column applies to every domain in the group: we did not find any groups for which some entities sometimes performed link decorators but other entities never participated in link decoration.

We note that about half of the domains that we observed performing bounce tracking also participated in link decoration at some point. Prior work by Koop et al. also enumerated a list of entities that perform bounce tracking [36]. We note that while our list is less extensive, it does not differ significantly from that of Koop et al. where it does overlap.

4.4 Navigational tracking on Chrome vs. Safari

We originally hypothesized that navigational tracking might be more common on Safari than Chrome, because trackers might detect that the browser was Safari and use navigational tracking to evade Safari’s third party cookie blocking. Since Chrome does not yet block third party cookies by default, trackers might not
use navigational tracking on this browser since it is not necessary. However, we did not find overwhelming evidence that navigational tracking is more common on Safari than Chrome.

On the other hand, we cannot rule out the possibility, because our methodology has some limitations with regards to this question. CrumbCruncher cannot tell with perfect accuracy when all of its parallel crawlers have clicked the same element, even after the fact. Without this information, we cannot be certain whether a particular link or iframe employed navigational tracking on one browser but not on another. CrumbCruncher might have mistakenly clicked an element that does not use navigational tracking, even though the correct element, which does use navigational tracking, existed on the page.

Because of this limitation, we would only be confident that more navigational tracking is used on Safari than Chrome if the difference in number of tracking instances was overwhelming.

5 LIMITATIONS

CrumbCruncher has several limitations. First, we only look for UIDs that are transferred in the query parameters of URLs, and not by other methods. For example, trackers reportedly sometimes use link decoration on the document.referrer header, instead of the link of the destination page [56]. Our initial reasoning was that there are a wide variety of ways to transfer user identifiers, so we could simply check once a crawl was complete for UIDs that had mysteriously appeared in different contexts without being passed through a URL. In practice, this turned out to be difficult: instances of navigational tracking often did not show up on at least three of the synchronized crawls, so UID detection had to be done via heuristics. These heuristics gave large numbers of false positives: it turned out that when the same value appeared on two different websites, the most common explanation was that the value was not a UID and had simply happened to be generated on both sites. To reduce our false positive rate, we chose to consider only values that we had observed get transferred across at least two first party contexts.

Second, if a website is setting UIDs based on IP address, our methodology will discard them, because all four crawlers executing a single crawl step come from the same IP.

Third, instances of navigational tracking frequently occurred on fewer crawlers than the three necessary for us to use our UID identification technique. The UID we identified using heuristics, especially the UIDs that relied only on programmatic and manual heuristics because they only appeared on one crawler, not two, may have false positives despite our manual efforts to remove them.

6 COUNTERMEASURES

6.1 Existing Mitigations

Defending against navigational tracking is not straightforward. The most effective but least practical countermeasure would be to block unexpected redirection completely; this approach would, of course, break a wide variety of benign functionality. SS0 redirects, link shorteners, and redirections from expired domains are just a few examples of functionality that would break if redirections were disallowed. Given the difficulty of designing defenses that do not degrade user experience, most defenders (whether browsers or browser extensions) have so far opted for blacklist-based approaches.

Many browsers are already trying to defend against navigational tracking. Safari provides users with the option to prevent cross-site tracking; when selected, Safari will delete cookies and website data from third-party trackers unless the user interacts with the site as a first-party website [9]. Safari labels a site as performing navigational tracking if 1) it automatically redirects the user to another site, and 2) it did not receive a user activation [48]. Safari also classifies a site as a navigational tracker if it participates in a redirect chain to another known navigational tracker.

Firefox also defends against navigational trackers, as defined by Disconnect’s Tracker Protection List [21, 39]. It will clear all storage from tracking sites after 24 hours unless the user has loaded the site as a first-party in the previous 45 days [48].

 Brave has multiple approaches for preventing navigational tracking. First, if the browser is navigating to a link with a query parameter for another destination URL, Brave will simply redirect to the URL in the query parameter [47]. If the browser cannot detect the final destination of the navigation, it allows the navigation to proceed, but inserts an interstitial warning users that they will be tracked if they continue. Brave also maintains a list of navigation tracking URLs created from crowd-sourced and existing open-source information, as well as a blacklist of query parameter names that are commonly used for navigational tracking. Finally, Brave clears the storage areas associated with any sites it classifies as navigational trackers as soon as the user closes the tab that loaded them.

While Chrome is in the process of deprecating third-party cookies [46], it does not appear to implement any features to defend against navigational tracking.

Some browser extensions have begun to implement protections against navigational tracking as well. For example, Privacy Badger [26] – a browser extension by the Electronic Frontier Foundation that blocks cross site tracking – identifies when a tracker wraps links in a redirector, and extracts the destination link from the query parameter in the redirection link [16]. Another extension, uBlock Origin, implements an interstitial-based approach similar to Brave’s [40].

6.2 Proposed Mitigations

Because existing defenses against navigational tracking are primarily built on blocklists, the data collected by CrumbCruncher can be valuable for enforcing the current prevention mechanisms of browsers and privacy-focused browser extensions. We commit to publishing our list of token names and trackers in a publicly available GitHub repo, although we will refrain from recording the link here at this time in order to preserve the double-blind nature of the review process. We provide two contributions: first, we will make available the list of query parameter names that were used to transfer user identifiers across first-party contexts. This list can be added to blocklists of parameter names, which are currently usually crowd sourced rather than based on a systematic study of navigational tracking behavior in the wild. Second, we will publish the list of entities that participate in navigational tracking as redirectors in the navigation path. This list will enforce blocklists
of entity names and domains that browsers should block if they appear in the middle of a navigation path.

7 RELATED WORK

The work that is most closely related to our own is Koop et al.’s study of bounce tracking [36]. Koop et al. crawled the Alexa Top 50,000 websites, clicked on an average of ten links per page, and recorded the navigation paths that included HTTP redirection. They then identified the redirectors on those navigation paths and recorded whether the redirectors stored a cookie. Koop et al. did not measure whether storage areas were linked or whether user identifiers were passed across domains. While they did keep track of whether a middle domain was also present as a third party tracker on the source site, and whether that third party tracker stored a third party cookie before the redirect chain began, they do not consider whether storage areas were linked between different trackers, or any different entities. Koop et al. also did not click on iframes. Finally, Koop et al. differentiated user identifiers from other cookies using a set of heuristics, which we improve on using our three-crawler method.

Urban et al. collected user identifiers that were sent as query parameters, in order to graph the relationships between the companies that owned the URLs’ domains [51]. Their study focused on companies’ compliance with the GDPR, and did not discuss the methods by which identifiers were shared in detail.

Englehardt et al. looked for trackers on the homepages on the Alexa Top 1M sites, and measured cookie syncing between trackers on a single website [23]. This study did not look for cookies getting shared across different websites, and their crawler did not interact with the websites studied.

Trackers may circumvent partitioned storage protections using techniques that do not rely on redirection chains, such as CNAME cloaking [18, 19]. CNAME cloaking is the procedure of mapping a website subdomain to a third party domain using a DNS CNAME record. This technique allows trackers to share their first party cookies, because the browser is fooled into attaching cookies from the original website’s subdomain rather than the third party domain it redirects to [19]. Trackers can access session cookies, even those belonging to financial institutions, this way [8, 43]. In this work, we focus on redirection-based circumvention only.

Redirection chains can be used for multiple malicious purposes besides tracking users by circumventing partitioned storage. For example, redirection chains can download malware or perform click fraud [7, 38].

Multiple groups have attempted to differentiate between identifiers that are capable of tracking users and identifiers that are not. Most determine which tokens are used to track an individual by performing multiple crawls [4, 23–25, 51]. If a token is identical (or nearly identical) across two crawls with different user profiles, its purpose is probably not to track individual users. Many works compare identifiers between crawls using the Ratcliff-Obersharm algorithm [13] for computing string similarity [4, 23, 24, 51]. Other signals commonly used to differentiate user-tracking identifiers include their expiration dates and length in characters.

In response to third party cookie deprecation, some websites are also fingerprinting user’s browsers in order to persistently track users across websites. It is well known that a small number of device features can uniquely fingerprint someone online [22].

A 2013 study crawled 20 pages for each of the Alexa top 10,000 sites and found 40 performed browser fingerprinting [41]. They found Skype was the most popular site that performed fingerprinting, and the categories of sites that performed the most fingerprinting were pornography and personals/dating.

A more recent study improved detection of fingerprinting code by using machine learning [32]. They then measured the Alexa top 100,000 sites and found that 10 percent of them perform fingerprinting. They find fingerprinting is more common with popular sites, as almost 25 percent of the Alexa top 10,000 sites perform fingerprinting.

Jang et al. have also measured how many websites invoke user privacy [34]. One privacy-invading technique they study is “history sniffing”, i.e. the website detecting the user’s browsing history by checking whether links are blue (unvisited) or purple (visited) [33]. They find that 46 of the Alexa top 50,000 sites perform history sniffing, and that most used one of two third-party applications to do so.

8 CONCLUSION

In this work, we present the first systematic study of navigational tracking, a technique that allows trackers to evade browsers’ protections against cross-website tracking. We find that link decoration, the more powerful variant of navigational tracking, is present across 8.1% of the navigations paths we observed, and bounce tracking, the less powerful variant, is present across 2.7% of paths. We publish a list of the entities that participate in navigational tracking, and classify these entities according to their behavior. Our findings can be used by browsers to improve protections against navigational tracking.

Understanding the scope of navigational tracking, and the techniques by which it is conducted, is important to continue to improve privacy on the Web. Browsers are increasingly (though not yet universally) trying to protect their users from being tracked. Understanding how trackers are circumventing new browser privacy protections is important, to make sure privacy improvements aren’t lost as quickly as they’re gained.

REFERENCES

[1] [n.d.]. Collecting Data on the Client or Server- Segment Documentation. https://segment.com/docs/guides/how-to-guides/collect-on-client-or-server/
[2] [n.d.]. Lexis® - Sign In | LexisNexis. https://signin.lexisnexis.com/inaccess/app/signin/back=https%3A%2F%2Fadvance.lexis.com%3A443%2Faci=1a
[3] [n.d.]. Sports Reference | Sports Stats, fast, easy, and up-to-date. https://www.sports-reference.com/
[4] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The web never forgets: Persistent tracking mechanisms in the wild. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 674–689.
[5] AdQuadrant. 2020. What is a Tracking Pixel? https://www.adquadrant.com/blog/what-is-a-tracking-pixel
[6] Rakuten Advertising. 2020. Navigating Tracking Prevention Technologies. https://blog.rakutenadvertising.com/news/tracking-prevention-technologies-2/
[7] Mitsuaki Akiyama, Takeshi Yagi, Takeshi Yada, Tatsuya Mori, and Youki Kadobayashi. 2017. Analyzing the ecosystem of malicious URL redirection through longitudinal observation from honeypots. computers & security 69 (2017), 155–173.
[8] Assel Aliyeva and Manuel Egele. 2021. Oversharing Is Not Caring: How CNAME Cloaking Can Expose Your Session Cookies. In Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security. 123–134.
