CookieGraph: Understanding and Detecting First-Party Tracking Cookies

Shaoor Munir
smunir@ucdavis.edu
UC Davis
USA

Sandra Siby
sandra.siby@epfl.ch
EPFL
Switzerland

Umar Iqbal
umar@cs.washington.edu
University of Washington
USA

Steven Englehardt
se@senglehardt.com
Independent Researcher
USA

Zubair Shafiq
zubair@ucdavis.edu
UC Davis
USA

Carmela Troncoso
carmela.troncoso@epfl.ch
EPFL
Switzerland

ABSTRACT

As third-party cookie blocking is becoming the norm in mainstream web browsers, advertisers and trackers have started to use first-party cookies for tracking. To understand how first-party cookies are being used with respect to third-party cookies, we conduct a differential measurement study on 10K websites with third-party cookies allowed and blocked. We find that first-party cookies are used to store and exfiltrate identifiers to known trackers even when third-party cookies are blocked.

As opposed to third-party cookie blocking, first-party cookie blocking is not practical because it would result in major breakage of website functionality. We propose CookieGraph, a machine learning-based approach that can accurately and robustly detect first-party tracking cookies. CookieGraph detects first-party tracking cookies with 90.20% accuracy, outperforming the state-of-the-art CookieBlock approach by 17.75%. We show that CookieGraph is fully robust against cookie name manipulation while CookieBlock's accuracy drops by 15.68%. While blocking all first-party cookies results in major breakage on 32% of the sites with SSO logins, and CookieBlock reduces it to 10%, we show that CookieGraph does not cause any major breakage on these sites.

Our deployment of CookieGraph shows that first-party tracking cookies are used on 93.43% of the 10K tested websites. We find that 98.39% of these first-party tracking cookies are in fact set by third-party scripts that are embedded in the first-party context. We also find evidence of first-party tracking cookies being set by fingerprinting scripts. The most prevalent first-party tracking cookies are set by major advertising entities such as Google, Facebook, and TikTok.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; **Web application security**; • **Computing methodologies** → Classification and regression trees.

KEYWORDS

measurements, machine learning, web security, privacy, cookies

ACM Reference Format:
Shaoor Munir, Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. 2023. CookieGraph: Understanding and Detecting First-Party Tracking Cookies. In *Proceedings of Conference on Computer and Communications Security (CCS'23).* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Major browser vendors such as Safari, Firefox, and Google Chrome have either blocked or are in the process of blocking *third-party cookies* – cookies set on domains that differ from the domain of the site visited by a user [22, 80, 89]. Because third-party cookies are accessible to the same domain that sets them across different sites that a user visits, they are widely used for cross-site tracking (i.e., linking a user's browsing activity across sites). Due to their ubiquitous use in tracking, the question arises as to how trackers will respond to third-party cookie blocking. *First-party cookies* – cookies that are set on the same domain as that being visited by a user – are of particular interest to advertisers and trackers because they will still be available in the face of third-party cookie blocking. However, since first-party cookies are only accessible from the setting domain, it remains to be seen how they can be used in lieu of third-party cookies for cross-site tracking.

Prior literature has shown that first-party cookies set by third-party scripts can be exfiltrated to tracking endpoints [41, 53, 75]. Prior work has also shown that trackers use browser fingerprinting to re-spawn first-party cookies [54]. Yet, there is no work studying the full spectrum of tracking possible through first-party cookies; and crucially, no countermeasures exist to specifically detect and block first-party tracking cookies. To fill this gap, we first investigate the use of first-party cookies by known trackers and then use our findings to develop a machine-learning based approach, CookieGraph, to detect and block first-party tracking cookies.

We first perform a differential measurement study comparing the use of first- and third-party cookies on 10K websites across multiple

CCS'23, November 26–30, Copenhagen, Denmark
© 2023 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

parallel crawls, both with third-party cookies enabled and blocked. We show that third-party cookie blocking does not significantly impact the sharing of identifiers to known tracking endpoints because major trackers are already using first-party cookies. Our analysis reveals that these trackers store identifiers in first-party cookies based on probabilistic and deterministic information.

Unlike third-party cookies, blocking all first-party cookies is not practical as many of these cookies are required for legitimate website functionality. An alternative could be the use of privacy-enhancing request blocking tools [61, 77, 78] that would also block the cookies set by the requested resources. Unfortunately, our evaluation shows that these tools also cause breakage because tracking cookies are often set by domains that also set functional cookies. Researchers have recently started to develop approaches to detect and block (both first and third-party) tracking cookies [39, 58]. However, these approaches rely on content-based features such as cookie names and values, which can lead to a high number of false positives (and consequently major website breakage) while also being susceptible to evasion [77].

To address these limitations, we design and implement CookieGraph, a machine-learning approach to detect first-party tracking cookies. Instead of using content-based features, CookieGraph captures fundamental tracking behaviors exhibited by first-party cookies that we discover in our differential measurement study. CookieGraph is able to detect first-party tracking cookies with 90.20% accuracy, outperforming the state-of-the-art CookieBlock [39] approach by 17.75%. We also show that blocking all first-party cookies results in major breakage on 32% of the sites with SSO Login, which is improved to 10% by CookieBlock. In contrast, CookieGraph does not cause any major breakage on these sites. Moreover, CookieGraph is robust to evasion through cookie name manipulation while CookieBlock's accuracy degrades by 15.68%.

We deploy CookieGraph on 10K websites to find 46,237 first-party tracking cookies on 93.43% of the 10K websites. The most prevalent first-party tracking cookies are set by major advertising entities, such as Google, Facebook, and TikTok, and then exfiltrated to a large number of other advertising and tracking endpoints. We find that 98.39% of the first-party tracking cookies are in fact set by third-party scripts, 77 of which also conduct fingerprinting, that are served from a total of 1,588 unique third-party domains.

In summary, our key contributions are as follows:

(1) We conduct a **large-scale differential measurement study** to understand the usage of first-party cookies by trackers when third-party cookies are blocked. Our analysis shows that blocking third-party cookies does not reduce the number of tracking requests containing identifiers and provides evidence that trackers use first-party cookies in lieu of third-party cookies for tracking.
(2) We introduce CookieGraph, **a machine-learning based countermeasure** to detect and block first-party tracking cookies. CookieGraph captures fundamental tracking behaviors of first-party cookies that. CookieGraphoutperforms the state-of-the-art in terms of accuracy, robustness, and breakage minimization.
(3) We **deploy CookieGraph on 10K out of the top-100K websites** to measure the prevalence of first-party tracking cookies. We detect a total of 1,588 distinct domains that set first-party tracking cookies, including major advertising entities such as Google, and show that fingerprinting scripts set first-party cookies on 984 sites.

Paper Organization: The rest of this paper is organized as follows: Section 2 provides an overview of the recent developments and related work on third-party and first-party cookies. Section 3 describes the threat model of first-party cookies. Section 4 presents our differential measurement study to evaluate the impact of third-party cookie blocking on the use of first-party cookies by trackers. Section 5 describes the design and evaluation of CookieGraph. We discuss the limitations of CookieGraph in Section 6 and conclude in Section 7.

2 BACKGROUND & RELATED WORK

2.1 Adoption of third-party cookies for tracking

Cookies were originally designed to recognize returning users, *e.g.*, to maintain virtual shopping carts [69]. Soon, they were adopted by third-parties to track users across websites and serve targeted ads [7]. Early standardization efforts mostly focused on limiting unintended cookie sharing across domains [44] and, despite well-known privacy concerns [1], largely ignored the intentional misuse of cookies by third-parties for cross-site tracking. Over the years, the use of third-party cookies for cross-site tracking has become prevalent [40, 45, 74, 75]. Prior research has found that the vast majority of third-party cookies are set by advertising and tracking services [45] and that the third-party cookies outnumber first-party cookies by a factor of two [40] – and up to four when they contain identifiers [75].

2.2 Countermeasures against third-party cookies

2.2.1 Safari. Since its inception in 2003, Safari has blocked third-party cookies from domains that have not been visited by the user as full-fledged websites [82]. In 2017, Safari introduced Intelligent Tracking Prevention (ITP). ITP used machine learning to automatically detect third-party trackers. It revoked storage access from classified domains if users did not interact with them on a daily basis [83]. Since 2017, ITP went through several iterations, i.e., ITP 1.1 [84], ITP 2.0 [85], ITP 2.1 [86], ITP 2.2 [87] and ITP 2.3 [88], eventually leading to full third-party cookie blocking [89].

2.2.2 Firefox. Firefox experimented with third-party cookie blocking in 2013 [49, 50], but did not ship default-on third-party cookie blocking until the release of Enhanced Tracking Protection (ETP) in 2018 [70]. ETP blocks third-party cookies based on a blocklist of trackers provided by Disconnect [6]. As of 2022, Firefox has launched Total Cookie Protection (TPC) which partitions all third-party cookie access [22]. Partitioning ensures that cookies set by a third-party on one site are distinct from those set by the same third-party on other websites, eliminating the third-party's ability to track users across those websites.

2.2.3 Internet Explorer and Microsoft Edge.

Amongst the mainstream browsers that have deployed countermeasures against third-party cookies, Internet Explorer (IE) and Microsoft Edge have the most permissive protections. IE blocked third-party cookies from domains that did not specify their cookie usage policy with the P3P response header [2]. However, website owners often misrepresented their own cookie usage policies, which rendered P3P ineffective [67]. Since 2019, Microsoft Edge blocks access to cookies and storage in a third-party context from some trackers, based on Disconnect's tracking protection list [6, 14, 80].

2.2.4 Chrome.

Google Chrome is the only mainstream browser that does not restrict third-party cookies in any way in its default mode. In 2020, Google announced plans to phase out third-party cookies in Chrome by 2022 [76]. However, the plan has been postponed several times and the latest timeline suggests the phasing out of cookies by late 2024 [55].

2.3 Adoption of first-party cookies for tracking

While third-party cookies are widely considered as the main mechanism for cross-site tracking, trackers have also relied on first-party cookies for various forms of tracking, as described below.

Same-site tracking. As early as 2012, Roesner *et al.* [74], noted that third-party tracking scripts, embedded on the main webpage (*i.e.*, in first-party context), set first-party cookies. First-party cookies enable *same-site tracking*, where trackers can determine whether a user is revisiting a website or internal pages of a site. While not as invasive as tracking users across different sites, significant information about a user can be gleaned from tracking their activity on the sites they frequent (*e.g.*, a social media or news site).

Cross-domain same-site tracking. First-party cookies can also be used for *cross-domain same-site tracking*, where a website's cookies are shared by trackers to other domains. In 2020, Fouad *et al.* [53] found that trackers sync first-party cookies to several third-parties on as many as 67.96% of the websites they tested. In 2021, Chen *et al.* [41] found that more than 90% of the websites contain at least one first-party cookie that is set by a third-party script. Similar to Fouad *et al.*, they found that at least one first-party cookie is exfiltrated to a third-party domain on more than half of the tested websites, raising concerns that these cookies might be used for tracking. Sanchez *et al.* [75] echoed these concerns, uncovering several instances where different third-parties interacted with the same first-party cookies. They conclude, through a large-scale measurement study of top websites and a number of case studies, that even after blocking third-party cookies, users are still at risk of tracking through first-party cookies.

Cross-domain sharing of first-party cookies presents a bigger privacy issue for users than same-site tracking. While same-site tracking is only restricted to domains that are able to set first-party cookies, cross-domain sharing of first-party cookies allows other trackers, which are not collaborating with the first-party domains, to receive information about user activity. This simplifies operations for trackers as instead of collaborating with each different publisher to set first-party cookies, they can instead leverage tracking cookies set by another tracker to monitor user activity. With this practice, not only the domains which are setting first-party cookies can track users' activities on the site, but tracking is also extended to other domains which receive these first-party cookies.

Cross-site tracking. While third-party cookies have been used extensively in *cross-site tracking*, *i.e.*, where a tracker links a user's activity across sites, the mechanisms by which first-party cookies are used in cross-site tracking have not been studied so far. Oh *et al.* [71] perform experiments to determine sharing of first-party data with trackers in lieu of third-party cookie blockage, determining that identifiers such as email addresses were also being shared to popular trackers. Their experiments show that trackers make use of identifiers like email addresses to link user activity across different sites. They make use of this knowledge to perform identity entanglement, where an attacker can make use of an email address or other identifiers to influence the advertisements shown to a victim. This sharing of additional information when third-party cookies are blocked allows trackers to track users across different sites.

Previous research has also shown that it is non-trivial to generate first-party identifiers that are accessible across websites. Prior research has found that trackers often leverage browser fingerprinting to generate first-party tracking cookies [54]. Browser fingerprinting provides unique identifiers that are accessible across websites but drift over time [64]. However, identifiers generated through browser fingerprinting can be stored in cookies that persist even after fingerprints change. In addition to browser fingerprinting, several advertising and tracking services, such as Google Ad Manager [16] and ID5 [26], specify in their documentation that they also use publisher-provided identifiers (PPIDs), such as email addresses, to set first-party cookies.

We note that techniques such as CNAME cloaking also allow advertisers or trackers to use first-party cookies. In this paper, we do not focus on CNAME cloaking because first-party cookie leaks due to CNAME cloaking have already been extensively studied by prior work [46, 47].

2.4 Countermeasures against first-party cookies

2.4.1 Deployed countermeasures.

Safari is the only mainstream browser that has deployed protections against first-party tracking cookies. Safari's ITP expires first-party cookies and storage set by scripts in 7 days if users do not interact with the website [82]. This limit is lowered to 24 hours if ITP detects link decoration being used for tracking [82]. However, first-party cookie tracking does not require link decoration to be effective. In cases where link decoration is not used, trackers can still track users within the 7-day window and beyond if users interact with the website within the 7-day window.

2.4.2 Countermeasures proposed by prior research.

There exist two machine-learning-based approaches to detect first-party and third-party tracking cookies. Hu et al. [58]'s approach uses sub-strings in cookie names (*e.g.*, track, GDPR) as features to detect first-party and third-party tracking cookies. Bollinger et al. [39] proposed CookieBlock. CookieBlock uses several cookie attributes such as the domain name of the setter, cookie name, path, value, expiration, *etc.* as features to detect first-party and third-party tracking cookies.

These approaches rely on hard-coded content features which makes them susceptible to adversarial evasions (as we show later in Section 5.6.3). Moreover, these approaches mainly rely on self-disclosed cookie labels as ground truth which are known to be unreliable [81].

2.4.3 Request blocking approaches. Request blocking through browser extensions, such as Adblock Plus [3], and machine-learning-based tracker detection approaches proposed by prior research, *e.g.*, [77], can potentially block first-party tracking cookies set by requests. However, request blocking is prone to cause breakage because it blocks access to content or cookies that might be essential for website functionality. We confirm this is the case in Section 5.6.3)

Unique focus of this paper. Prior work has only incidentally measured the usage of first-party tracking cookies, and existing approaches to detect first-party tracking cookies are lacking. In this paper, we fill this void by conducting a large-scale study to measure the prevalence of first-party tracking cookies and develop an accurate and robust machine-learning approach, CookieGraph, aimed at detecting first-party cookies.

3 THREAT MODEL

In this section, we describe the threat model of tracking via first-party cookies.[1]

First- vs third-party cookies. Before describing the threat model, we define what we mean by first- and third-party cookies. Cookies can either be set by the Set-Cookie HTTP response header or by using document.cookie in JavaScript. Cookies set via response header from the *same* domain as the first-party are *first-party* cookies. Similarly, cookies set via response header from the a *different* domain than the first-party are *third-party* cookies. When cookies are set by a script, their classification depends on whether the script is embedded in a first- or third-party execution context. The cookies set by third-party scripts *running in the first-party context* are *first-party* cookies. The cookies set by third-party scripts *running in a third-party context* (e.g., third-party iframes) are *third-party* cookies.

There are three main entities in this threat model: users (the victim), trackers (the adversary), and publishers.

We assume that the user:

- visits different websites using one or more desktop/mobile devices that have distinct *fingerprints* [63]
- is not averse to logging in to those websites and providing PII (personally identifiable information) such as email addresses
- has third-party cookies disabled and first-party cookies enabled

We assume that the publisher:

- controls the content on the site being visited by the user
- embeds the tracker in the first-party context, allowing the tracker to set first-party cookies
- shares email and other deterministic identifiers (e.g., username, phone number) with the tracker, if provided by the user

We assume that the tracker:

- is present in a first-party context on the publisher's site
- can set and read first-party cookies using document.cookie
- can collect information such as IP addresses, screen resolution *etc*., which can be used to construct a device fingerprint

Trackers can use the information shared by the publisher, and the fingerprints collected by their own scripts to perform same-site, cross-domain same-site, and cross-site tracking. We describe them below.

Same-site tracking. A user visits the same publisher's site multiple times. During the first visit of the user, the tracker A sets a first-party cookie on the user's device. Upon subsequent visits by the user, tracker A can read the first-party cookie set and know that it is the same user who is revisiting the site. When performing same-site tracking, tracker A is able to gather information about the user across the pages maintained by the same publisher.

Same-site cross-domain tracking. After setting a first-party cookie on a user's device, tracker A also shares the first-party cookie with a different tracker B that is not itself present in the first-party context (and thus unable to set a first-party cookie of its own). On each subsequent visit of the user, tracker A shares the first-party cookie and the pages visited by the user with tracker B. Thus, without setting its own first-party cookie and directly colluding with the publisher, tracker B is also able to track the user's activity on the same site.

Cross-site tracking. Consider a scenario in which a user visits three different sites (publishers 1, 2, 3) where tracker A is embedded in the first-party context. The user visits sites 1 and 2 on one device, and site 3 on a different device. Publishers 2 and 3 ask the user for a deterministic identifier (*e.g.*, email address) which we

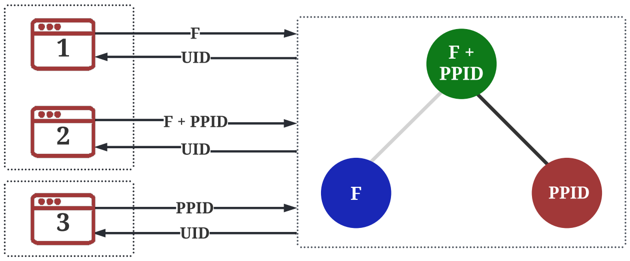

Figure 1: Cross-site tracking. Flow of information and identifiers through an identity graph for cross-site tracking. Initially, the user visits publishers 1 and 2 from one device. Tracker A, on publishers 1 and 2, collects and sends fingerprint F to the identity graph. The identity graph returns a UID for all the publisher visits, by matching fingerprints sent on each respective publisher. A publisher-provided ID, $PPID$, is also sent when visiting publisher 2. The user visits publisher 3 on a different device, thus tracker A is unable to construct a fingerprint which matches F. Publisher 3 sends a publisher-provided ID that matches $PPID$ provided on publisher 2. As a result, the identity graph matches and returns the same UID for publisher 3. This ID is stored in first-party cookies on the user's device for each respective publisher.

[1] This threat model is informed by prior literature [41, 53, 54, 71, 75] and our case studies of popular tracking services described in Appendix A.1.

denote as *PPID* (Publisher-Provided ID). Tracker A also constructs fingerprints on sites 1 and 2, denoted by F_i, where i denotes the publisher visited.

When the user visits sites 1 and 2, tracker A collects fingerprints F_1 and F_2, which are the same (*i.e.*, $F_1 = F_2 = F$) since they are all constructed for the same device. This allows tracker A to infer that the same user/device is visiting both sites. Tracker A links the deterministic identifiers and fingerprints belonging to the same user/device by constructing an *identity graph* (refer to Appendix A.1 for examples). The gray edge in Figure 1 shows the link in the identity graph constructed by tracker A for the fingerprints on sites 1 and 2.

The user then visits site 3 from a different device where tracker A is not able to construct the same fingerprint F. Publisher 3 asks the user for deterministic identifier (*e.g.*, email address), which is the same as the *PPID* provided by the user to publisher 2. Based on this additional information, tracker A can add a black edge to the identity graph.

Tracker A is finally able to connect all nodes in the identity graph to the user. Tracker A then assigns all connected nodes in the identity graph the same ID *UID*, which it can store in a first-party cookie on each of the sites. On each subsequent visit by the user to any of the sites, tracker A can now simply read the first-party cookie containing *UID*. Because *UID* is same across sites 1, 2 and 3, this allows tracker A to track the user across different sites.

4 MEASUREMENTS

In this section, we conduct a preliminary measurement study to investigate the usage of first-party cookies by advertising and tracking services (ATS) when third-party cookies are blocked.

4.1 Data Collection

Data collection. We use OpenWPM [51] to crawl a sample of 10K out of the top-100K websites [17]. To ensure that our crawls cover websites of variable popularity, we crawl the top 1K sites – ensuring coverage of the most popular websites– and uniformly sample another 9K sites from the sites ranked 1K-100K. To capture behaviors that may be different in the landing and internal pages of a website [37], we perform an interactive crawl that covers both kinds of pages. Specifically, for each site, we crawl its landing page and then select up to 20 internal pages to visit at random. We conduct four parallel crawls: two with third-party cookies enabled (`3P-Allowed`) and two with third-party cookies blocked (`3P-Blocked`). Parallelizing the crawls minimizes temporal variations across crawls and mitigates the effect of the dynamic behavior of websites. We repeat failed crawls up to four times. This enables us to successfully conduct the four parallel crawls for 99.38% of the 10K websites.

Labeling tracking activity. To label tracking, we use EasyList [8] and EasyPrivacy [9]. Specifically, we use them to label requests as tracking (ATS) or not tracking (Non-ATS). We label a request as tracking (ATS) if its URL matches the rules in either one of the lists. Otherwise, we label it as not tracking (Non-ATS).

Since the basic premise of tracking is to identify users, we are particularly interested in sharing of identifiers in these tracking requests. In line with prior work [52, 62], we define identifiers as a string that is longer than 8 characters and matches the regex $[a-zA-Z0-9\_=-]$. Using this definition, we look for identifiers in URL query parameters [73] and cookie values [40, 41, 48, 75].

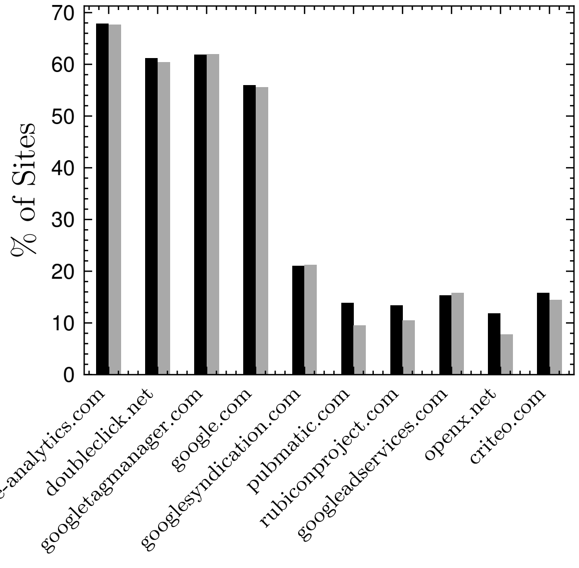

Figure 2: Presence of top-10 tracking domains. The plot shows the percentage of sites where at least one request containing an identifier is sent to a tracking domain.
(▬) `3P-Allowed`: Third-party cookies allowed
(▬) `3P-Blocked`: Third-party cookies blocked

4.2 Tracking under Third-Party Cookies Blocking

We first study whether blocking third-party cookies effectively eliminates ATS requests. We compare the number of requests containing identifiers with and without third-party cookies.

Table 1 shows the average number of requests for two parallel crawls conducted with third-party cookies allowed and blocked. Table 1 shows that there is only a modest reduction in the overall number of ATS requests when third-party cookies are blocked. The difference in the number of ATS requests containing identifiers is just 6.41%. This is surprising because cookie syncing, which is widely used for same-site-cross-domain and cross-site tracking [53, 72], entails sharing third-party identifier cookies in query parameters [40, 41, 48]. With third-party cookies blocked, cookie syncing between third-parties cannot occur and we would expect to see a larger drop in identifiers shared in ATS requests. ***We conclude that third-party cookie blocking does not effectively limit the exfiltration of identifiers to trackers.***

Next, we analyze whether third-party cookie blocking disparately impacts different ATS domains (eTLD+1). Figure 2 plots the percentage of sites with at least one ATS request with identifiers. Six of the top-10 ATS domains, all owned by Google, show only a negligible reduction in the number ATS requests with identifiers when third-party cookies are blocked. In contrast, three other ATS domains, owned by Pubmatic, Rubicon, and OpenX, show a significant reduction.

Table 1: Average number of requests per site in `3P-Allowed` and `3P-Blocked` configurations

Request Count	`3P-Allowed`	`3P-Blocked`	Change
Total	848.02	841.69	-0.75%
Tracking	358.97	349.56	-2.62%
Non-Tracking	489.05	492.13	0.63%
Tracking with ID	143.36	134.16	-6.41%
Tracking without ID	215.61	215.40	-0.10%

Table 2: Average number of first-party cookies per site in `3P-Allowed` and `3P-Blocked` configurations

1P Cookie Count	`3P-Allowed`	`3P-Blocked`	Change
Total	150.57	153.91	-2.22%
Set by Trackers	118.72	121.12	-2.02%
Set by Non-Trackers	31.85	32.79	-2.94%
Set by Trackers with ID	73.41	74.27	-1.17%
Set by Trackers without ID	45.30	46.85	-3.41%

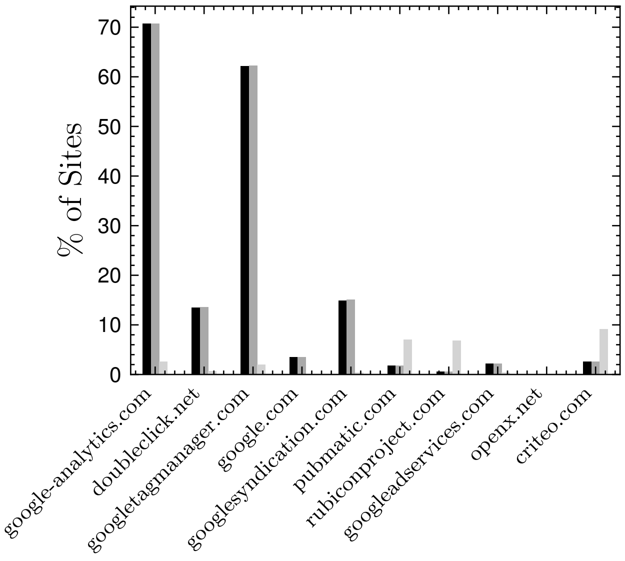

Figure 3: Comparison of percentage of sites on which first-party and third-party identifier cookies are set by ATS domains.
(▬) first-party identifier cookies set when third-party cookies are allowed
(▬) first-party identifier cookies set when third-party cookies are blocked
(▭) third-party identifier cookies set when third-party cookies are allowed

4.3 Tracking through First-Party Cookies

Table 1 shows that even after blocking third-party cookies, there is only a small decrease in ATS requests containing identifiers (6.41%). The identifiers in these ATS requests are likely originating from some storage mechanism other than third-party cookies. Since recent prior work has shown that ATS are increasingly using first-party cookies [41, 75], we next investigate whether first-party cookies are being used in lieu of third-party cookies to circumvent third-party cookie blockage.

We first compare the average number of first-party cookies in `3P-Allowed` and `3P-Blocked` crawls in Table 2. We observe only a minor difference in the average number of first-party cookies set with third-party cookies allowed/blocked. It is also noteworthy that 78.69% of the first-party cookies are set by ATS scripts. A further 61.31% of them are identifier cookies. We conclude that the vast majority of first-party cookies are in fact set by ATS and that they are not significantly impacted by third-party cookie blocking.

Next, we compare the setting of first- and third-party identifier cookies by ATS domains (eTLD+1 of the setting script URL) to understand if first-party cookie usage is equally prevalent across different ATSes. Figure 3 plots the percentage of sites where at least one first-party and/or third-party identifier cookie is set by a top-10 ATS domain.

For the six Google-owned ATS domains, which showed a negligible difference in requests containing identifiers after blocking third-party cookies, there is also little to no change in the use of first-party identifier cookies across both crawls. These domains do not set a large number of third-party identifier cookies, even when those are allowed, which likely explains why they were not impacted by third-party cookie blocking.

On the contrary, the other set of ATS domains for which we observe a reduction of identifiers (i.e., Pubmatic, Rubicon, and OpenX) do use more third-party identifier cookies than first-party identifier cookies when third-party cookies are authorized. This observation also explains the drastic drop in the number of requests containing identifiers to these other ATS domains after blocking third-party cookies in Figure 2. ***We conclude that trackers which are not affected by third-party cookie blocking are using first-party cookies as a replacement.***

4.4 Takeaway

Our differential measurement study reveals that third-party cookie blocking does not effectively prevent tracking. There is only a negligible reduction in the exfiltration of identifiers to trackers when third-party cookies are blocked. We find that this is because ATSes use first-party cookies in lieu of third-party cookies.

We also find that the impact of third-party cookie blocking is not uniform across different trackers. Some ATS domains show more reduction in the exfiltration of identifiers than others. This disparity exists because some trackers only use first-party cookies regardless of the availability of third-party cookies; while others are using both first-party and third-party cookies to store identifiers.

5 COOKIEGRAPH: DETECTING FIRST-PARTY TRACKING COOKIES

In this section, we describe CookieGraph, a graph-based machine learning approach to detect first-party ATS cookies. CookieGraph creates a graph representation of a webpage's execution based on HTML, network, JavaScript, and storage information collected by an

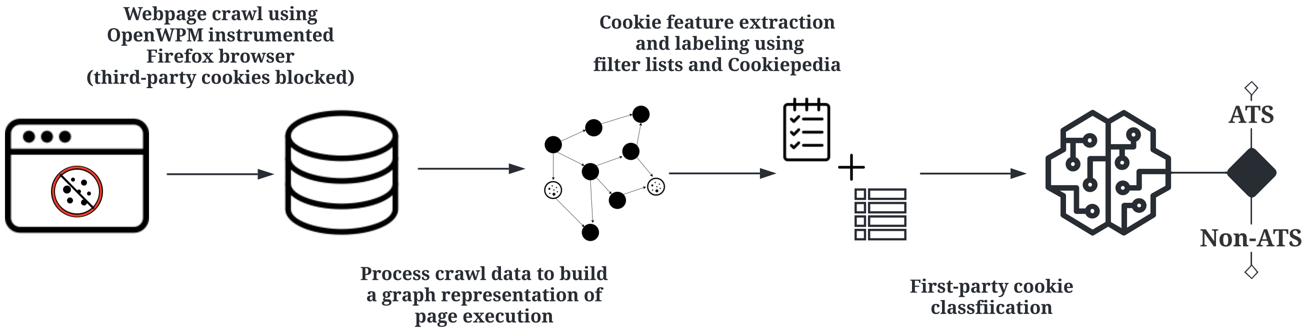
Webpage crawl using
OpenWPM instrumented
Firefox browser
(third-party cookies blocked)
Cookie feature extraction
and labeling using
filter lists and Cookiepedia
ATS
Non-ATS
Process crawl data to build
a graph representation of
page execution
First-party cookie
classfiication

Figure 4: Overview of CookieGraph pipeline: (1) Webpage crawl using an instrumented browser; (2) Construction of a graph representation to represent the instrumented webpage execution information; (3) Feature extraction for graph nodes that represent first-party cookies; and (4) Classifier training to detect first-party ATS cookies.

instrumented browser, in which first-party cookies are represented as storage nodes. CookieGraph extracts distinguishing features of these cookies and uses a random forest classifier to detect first-party ATS cookies. Figure 4 provides an overview of CookieGraph's pipeline.

5.1 Design and Implementation

Browser instrumentation. CookieGraph relies on our extended version of OpenWPM [51] to capture webpage execution information across HTML, network, JavaScript, and the storage layers of a webpage. Our measurements in Section 4 found significant use of localStorage in addition to cookies. Thus, we use the term "storage" to refer to both cookies and localStorage. In most cases, the description for cookies is also applicable to localStorage and vice versa.

CookieGraph captures HTML elements created by scripts, network requests sent by HTML elements (as they are parsed) and scripts, responses received by the browser, exfiltration/infiltration of identifiers in network requests/responses, and read/write operations on the browser's storage mechanisms.

Graph construction. The nodes in CookieGraph's graph represent HTML elements, network requests, scripts, and storage elements. When localStorage and first-party cookie nodes share the exact same name, CookieGraph considers them as one storage node. CookieGraph's edges represent a wide range of interactions among different types of nodes *e.g.*, scripts sending HTTP requests, scripts setting cookies *etc.* In addition to interactions considered by prior work [77], CookieGraph incorporates edges that model the actions associated to tracking using first-party cookies. We identify these actions from the result of our measurement study in Section 4, and the case studies described in Appendix A.1.Cookies are typically set with the values *infiltrated* with HTTP responses and are *exfiltrated* via URL parameters and request headers or bodies; CookieGraph captures infiltrations and exfiltrations by linking the script-read/write cookies in the first-party execution context to the requests of reader/writer script that contains those cookie values. In addition to plain text cookie values, CookieGraph also monitors Base64-, MD5-, SHA-1-, and SHA-256- encoded cookie values in URLs, headers, request, and response bodies. CookieGraph tracks the value of each cookie and associates the relevant interaction (exfiltration or infiltration) to the element which initiated the interaction. Because of our focus on identifiers, CookieGraph only captures cookie values that are at least 8 characters long (but it would be trivial to extend it to consider smaller cookie values).

Figure 5 illustrates how CookieGraph creates a graph representation. In this example, a third-party script from `tracker1.com` executes in a first-party context on the webpage, `example.com`. The script first reads `infoCookie` (1), which contains tracking information such as the publisher ID and a user signature. Then, the script sends the content of the cookie to `tracker1.com`'s sync endpoint via an HTTP POST request (2). The endpoint returns a user ID (UID) in the response body (3), which is stored in both a first-party cookie and localStorage named `IDStore` (4). At a later point, the script reads the value from `IDStore` (5) and exfiltrates the UID to two other tracking endpoints: to `tracker2.com` via a URL parameter (6) and to `tracker3.com` via an HTTP header (7).

Figure 6 shows the graph representation that CookieGraph generates for the execution of the example script. The nodes in the graph represent the script, the storage, and the network endpoints. The edge numbers show the actions performed in Figure 5. The dotted and dashed lines in the graph show the infiltration and exfiltration behaviors captured by CookieGraph. CookieGraph is not only able to capture the interactions of the script with the storage and the network endpoints but is also able to precisely *link exfiltration and infiltration of the first-party cookie* via an edge from the cookie node to the endpoint.

Feature extraction. We use CookieGraph's representation to extract two kinds of features.

Structural features represent relationships between nodes in the graph, such as ancestry information and connectivity. Structural

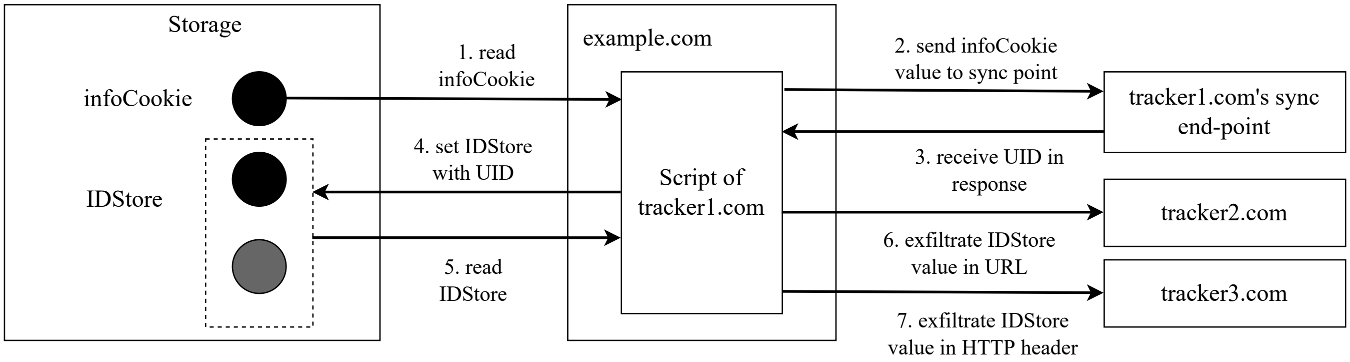

Figure 5: Example scenario to illustrate CookieGraph's graph construction (shown in Figure 5).

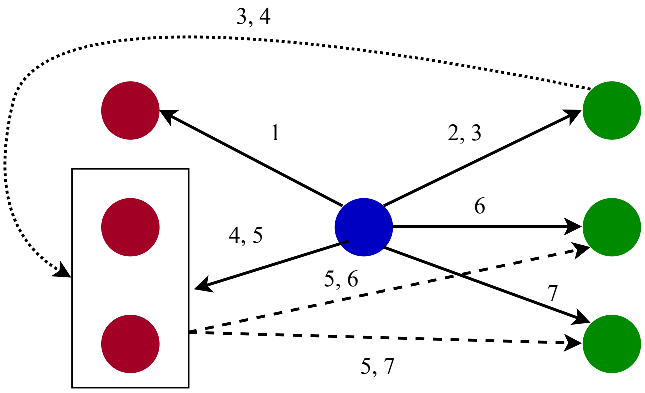

Figure 6: Graph representation of Figure 5 in CookieGraph. 🟢 network nodes, 🔵 script nodes, and 🔴 storage nodes. While the solid lines show the interactions of the script nodes with the storage and request nodes, the dashed (- - -) and dotted (. . .) lines represent the exfiltration and infiltration edges that are captured by CookieGraph.

features capture the relationships between the first-party cookie nodes and scripts on the page. For example, how many scripts interacted with a cookie or whether a script that interacted with a cookie also interacted with other cookies.

Flow features represent first-party ATS cookie behavior. We extract three types of flow features. First, we count the number of times a cookie was read or written. Second, we count the number of times a cookie was infiltrated via HTTP responses or exfiltrated via URL parameters, request headers, or request bodies. Third, features related to the setter of the cookie. Concretely, whether the setter's domain also acted as an end-point for other cookie exfiltrations, and whether the setter's domain was involved in redirect chains (since redirects are commonly used in tracking). The intuition behind the third category of features is that domains involved in setting first-party ATS cookies are also involved in sharing information with other ATSes.

CookieGraph does *not* use content features, e.g., based on cookie name, as they can be trivially used to evade the detector [62, 77].

5.2 Evaluation

Similar to previous work on graph-based webpage modelling [61, 77], we use a random forest classifier to distinguish between ATS and Non-ATS cookies. We first train and test the accuracy of this classifier on a carefully labeled dataset. Then, we deploy it on our 10K website dataset.

5.2.1 Ground truth labeling. We use two complementary approaches to construct our ground truth for first-party ATS cookies. We represent each first-party cookie as a cookie-domain pair since the same cookie name can occur on multiple sites.

Filter lists. We rely on filter lists [8, 9] as previous work has found them to be reasonably reliable in detecting ATS endpoints [61, 77]. Filter lists are designed to label resource URLs, rather than cookies. We adapt them to label cookies by assigning the label of a particular resource to all the cookies set by that resource. Since both ATS and Non-ATS cookies can be set by the same resource, this labeling procedure could result in a non-trivial number of false positives. To limit the number of false positives in our ground truth, we only label Non-ATS cookies based on filter lists: i.e., if a script that sets a cookie is not marked by *any* of the filter lists, we label these cookies as Non-ATS. Conservatively, if any one of the filter lists marks the cookie's setter as ATS, we label the cookie as Unknown.

Cookiepedia. Inspired by prior work [39], we use Cookiepedia [13] as an additional source of cookie labels. Cookiepedia is a database of cookies maintained by a well-known Consent Management Platform (CMP) called OneTrust [39, 57]. For each cookie/domain pair, Cookiepedia provides its purpose, defined primarily through the cookie integration with OneTrust. Each cookie is assigned one of four labels: strictly necessary, functional, analytics, and advertising/tracking. As Cookiepedia-reported purposes are self-declared, we adopt a conservative approach: we only label a cookie-domain pair as ATS if a cookie's purpose is declared as advertising/tracking or analytics in a particular domain. If the declared purpose is strictly necessary or functional, we label the cookie as Unknown, as the cookie might have been, mistakenly or intentionally, mislabeled.

We combine the results of the labeling approaches to obtain a final label for the cookies. If both approaches label a cookie as Unknown, its final label is Unknown. If only one of the approaches has a known label, this is the final label. If Cookiepedia marks a cookie as ATS and filter lists mark it as Non-ATS, we give precedence to

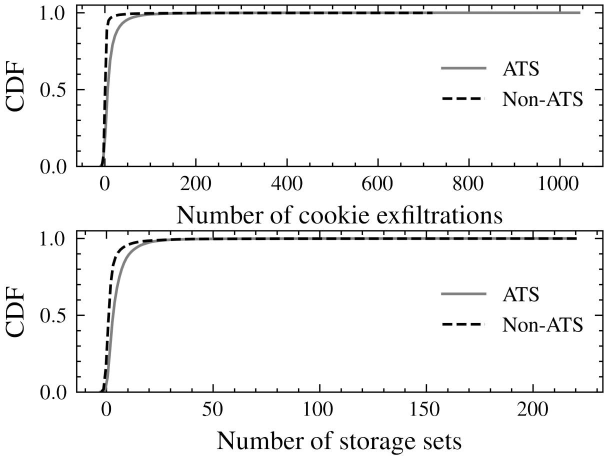

Figure 7: Feature distribution of cookie exfiltrations (top) and storage sets (bottom) for ATS and Non-ATS cookies. ATS cookies are exfiltrated and set more than Non-ATS cookies, resulting in flow features based on exfiltrations and sets being helpful for the classifier.

the Cookiepedia label and assign the final label as ATS because websites are unlikely to self-declare their Non-ATS cookies as ATS.

Using this labeling process, 20,927 out of 78,560 first-party cookies (26.64%) have a known (ATS or Non-ATS) label and the rest are labeled as Unknown. We observe that cookies set by the same script across two different sites are often labeled ATS in one instance and Unknown in another instance because Cookiepedia does not have data for the latter. As it is unlikely that an ATS script changes purpose across sites, we propagate the ATS label to all instances set by the same script. Using this label propagation, we label 51.76% of the data, with 21,875 (53.79%) ATS and 18,786 (46.20%) Non-ATS labels.

5.2.2 Classification. We train and test the classifier on the labeled dataset using standard 10-fold cross-validation. We ensure that there is no overlap in the websites used for training and testing in each fold. Similar to Section 5.1, we limit the classifier to cookies whose value is at least 8 characters long. The classifier has 89.84% precision and 92.52% recall, with an overall accuracy of 90.20%, indicating that the classifier is successful in detecting ATS cookies.

5.3 Feature Analysis

We conduct feature analysis to understand the most influential features in the classification of cookies. We find that the most influential features are the flow features, which capture cookie exfiltrations, set operations, and redirections by cookie setters. Figure 7 shows the distributions for the number of cookie exfiltrations (top) and the number of times a cookie is set (bottom), for ATS and Non-ATS cookies. ATS cookies are much more likely to be exfiltrated than Non-ATS cookies: ATS have a median number of 6 exfiltrations (mean/std is 11.11/15.95) as compared to a median of 0 for Non-ATS (mean/std is 0.62/5.29). Also, ATS cookies tend to be set much more frequently by scripts, with a median of 3 set operations (mean/std is 4.86/6.99) as compared to 1 for Non-ATS cookies (mean/std is 2.17/6.08). These findings confirm our conclusions in Section 4: first-party ATS cookies are used to store identifiers which are then exfiltrated to multiple endpoints.

Error analysis. We conduct a manual analysis of CookieGraph's false positives and false negatives to understand why the approach fails.

We find that the cookies that were most misclassified as ATS are those whose publicly available descriptions indicate they are used to track visitors on a page (e.g., `__attentive_id`, `messagesUtk`, `omnisendAnonymousID`) [4, 10, 12]. We also find a few instances of well-known Google Analytics cookies `_ga` and `_gid` that are labeled in ground truth as Non-ATS, but are classified by CookieGraph as ATS. Our manual inspection also shows that the false positives are not caused by misclassifications, but mostly that the tracking cookies flagged by CookieGraph were mislabeled as Non-ATS in the ground truth. In other words, CookieGraph has likely correctly classified these tracking cookies. We note that even after our procedures to improve ground truth labels, there may be cookies that did not have self-disclosed labels or were served from slightly different scripts (thereby missing our hash-based script matching) leading. This is a limitation of the ground-truth we use, as it relies on either the self-declaration of the cookie purpose or a match between the setting scripts to determine if a cookie is ATS. We leave the investigation of further methods of improving the ground truth labeling to future work.

Regarding false negatives, i.e., ATS cookies missed by CookieGraph, we mainly observe two cases. First, we have the case of finite coverage of encodings. A representative case is the `_pin_unauth` cookie. Its value is double-base64-encoded, which is not included in the list of potential encoding schemes used by CookieGraph to detect exfiltration. These false negatives can be averted by using a more comprehensive list of encoding schemes or by performing full-blown information flow tracking instead of approximating exfiltration flows; however, the latter would come at a performance cost as we discuss in Section 5.5.

Second, we have the case of lack of coverage of actions. Our crawl to create the graphs in CookieGraph may not capture all possible actions on a webpage. If CookieGraph does not capture sufficient activity during webpage execution, some cookies may not be triggered and therefore, the analysis will miss them. We further discuss these cases of false negatives in Section 6.1

5.4 Deployment

We deploy CookieGraph to classify all cookies, including Unknown cookies, in our crawl of 10K sites.

Prevalence of first-party ATS cookies. CookieGraph classifies 62.48% of the 74,003 first-party cookies in our dataset as ATS. We find that 93.43% of sites deploy at least one first-party ATS cookie. Of these sites, the average number of first-party ATS cookies per site is 6.29.

Who sets first-party ATS cookies? The vast majority (98.39%) of the first-party ATS cookies are set by third-party embedded scripts served from a total of 1,588 unique domains. This shows that first-party ATS cookies are in fact set and used by third-parties. These first-party cookies enable third-parties to perform ***same-site tracking*** as described in Section 3.

Who sends and receives first-party ATS cookies? Next, we analyze the most prevalent first-party cookies and the third-party entities that actually set them. Table 3 lists top-25 out of 5,019 first-party ATS cookies[2] based on their prevalence. Two major advertising entities (Google and Facebook) set first-party ATS cookies on approximately a third of all sites in our dataset. CookieGraph detects _gid and _ga cookies by Google Analytics as ATS on 77.11% and 68.88% of the sites. The public documentation acknowledges using these two first-party cookies to store user identifiers for tracking [24]. We also find evidence of widespread cross-domain first-party first-party ATS cookie sharing. For example, _gid and _ga cookies are respectively exfiltrated to 56 and 179 destination domains, more than 95% of which are non-Google domains.

CookieGraph detects _fbp cookie by Facebook as ATS on 33.22% of the sites. Their public documentation acknowledges that Facebook tracking pixel stores unique identifiers in the first-party _fbp cookie [21]. In fact, Facebook made a recent change to include first-party cookie support in its tracking pixel to avoid third-party cookie countermeasures [35]. It is again noteworthy that the _fbp cookie by Facebook is exfiltrated to 73 destination domains, more than 98% of which are non-Facebook domains.

This extensive sharing of first-party ATS cookies to other domains enables ***cross-domain same-site tracking***, through which a tracker who is unable to set first-party cookies is still able to track user activity on a site.

TikTok, a social media app that is known to aggressively harvest sensitive user information [27], also recently added support for setting first-party tracking cookies using TikTok Pixel [32, 34]. TikTok's first-party _ttp tracking cookie is present on 3.75% percent of sites, which is considerably lower than Facebook and Google but comparable to more specialized entities such as Criteo.

Criteo's cto_bundle cookie is amongst the most prevalent first-party ATS cookies. We observe that Criteo sets this first-party ATS cookie on 5.98% of the sites in our dataset.

Cross-site tracking. As discussed in Section 3, trackers can use deterministic (*e.g.*, email address) or probabilistic (fingerprinting) identifiers for cross-site tracking using first-party cookies.[3] We show that scripts that set first-party ATS cookies are also involved in fingerprinting.

First, we analyze the first-party cookies set by the scripts from entities that are known to engage in browser fingerprinting. We use Disconnect's sublist of fingerprinters [23, 30] from its tracking protection list [6]. We find that 45 (2.83%) distinct domains that set first-party cookies are also known fingerprinters. These domains are responsible for setting 41.45% of all first-party ATS cookies.

Second, we use FP-Inspector [59] to further determine whether first-party ATS cookies are set by fingerprinting scripts. Using FP-Inspector, we are able to find fingerprinting scripts on 1,264 out of 10k websites. We further find that fingerprinting scripts on 984 websites set first-party cookies. In total, 349 first-party cookies are set by fingerprinting scripts. 72 out of these 349 cookies, set by 77 different fingerprinting scripts, are classified by CookieGraph as ATS cookies. It is noteworthy that 70 of these 72 cookies (e.g., adtech_uid, tfstk, bafp, pxde) are not listed as tracking cookies on Cookiepedia. Our manual analysis of the remaining 268 Non-ATS cookies shows that they store non-identifiable information (*e.g.*, domain names, flags for cookie permissions).

5.5 Comparison with Existing Countermeasures

In this section, we compare CookieGraph with some of the existing countermeasures which are used to restrict the effect of first-party cookies.

Intelligent Tracking Prevention (ITP) is used by Safari as a broad countermeasure against online tracking activities. Under ITP, Safari limits the maximum expiry time of a first-party cookie set through JavaScript to seven days [82]. In addition, Safari limits this time to only 24 hours for known trackers.[4] While this can be a prudent countermeasure if the first-party tracking cookies were meant to be a storage for the identifier for the repeat visits of the user. However, as we have shown in the previous section, first-party tracking cookies are shared with a large number of other domains immediately after being set. This sharing of identifiers among different trackers is meant to enhance their ability to track users across different sites. Limiting the amount of time that a cookie is set for will not be able to stop this sharing of information, thus proving ineffective in protecting user privacy.

5.6 Comparison to classifier-based blocking

Next, we compare CookieGraph with state-of-the-art countermeasures against ATS, CookieBlock [39] and WebGraph [77], in terms of detection accuracy, website breakage, and robustness.

CookieBlock [39] is a state-of-the-art approach to classify cookies, including advertising/tracking and analytics. It makes use of both manually curated allow lists and a machine learning classifier, which mainly relies on features based on cookie attributes (cookie names and values).

WebGraph [77] is the state-of-the-art graph-based approach to classify ATS requests. Since WebGraph is not designed to directly classify cookies, we adapt it to this end by identifying ATS resources identified by WebGraph in 3P-Blocked and generating a block list of cookies for each domain set by those resources. This list is meant to mimic the effect of blocking these resources on first-party ATS cookies.

5.6.1 *Detection Accuracy.*

Table 4 compares the detection accuracy of CookieGraph with CookieBlock and WebGraph. CookieGraph outperforms both approaches in all metrics. The superiority in precision indicates that existing countermeasures result on many more false positives than CookieGraph. These additional false positives mean that previous approaches would block functional first-party cookies potentially affecting user experience.

5.6.2 *Website Breakage.*

We manually analyze the breakage caused by CookieGraph, CookieBlock and WebGraph's on 50 sites that are sampled from the 10K sites used in Section 4 (25 sites chosen randomly from the top 100 and other 25 from the rest). We list these sites in A.2.

[2] We report distinct tuples of the cookie name and the setter script's URL.

[3] While our automated crawls do not allow us to test the use of deterministic identifiers for cross-site tracking at scale, recent work [71] showed the use of email addresses and other deterministic identifiers by trackers such as Criteo.

[4] Firefox also limits the expiry time for cookies set by known trackers to 24 hours [68].

Table 3: List of top-25 ATS cookies detected by CookieGraph

Cookie	Script	Org.	Percentage	Destination	Top-3 Destination Domains		
Name	Domain		of Sites	Domains	# 1	# 2	# 3
_gid	google-analytics.com	Google	77.11%	56	google-analytics.com	doubleclick.net	mountain.com
_ga	google-analytics.com	Google	68.88%	179	google-analytics.com	doubleclick.net	google.com
_fbp	facebook.net	Facebook	33.22%	73	facebook.com	appier.net	google-analytics.com
_gcl_au	googletagmanager.com	Google	14.22%	21	google.com	doubleclick.net	tealiumiq.com
__gpi	googlesyndication.com	Google	14.02%	4	doubleclick.net	googleadservices.com	ezoic.net
_ga	googletagmanager.com	Google	12.79%	48	google-analytics.com	doubleclick.net	google.com
__gads	googlesyndication.com	Google	12.35%	2	doubleclick.net	googleadservices.com	
__gads	doubleclick.net	Google	11.68%	11	doubleclick.net	googleadservices.com	ezoic.net
_uetsid	bing.com	Microsoft	10.22%	15	bing.com	hotjar.com	tealiumiq.com
_uetvid	bing.com	Microsoft	10.22%	21	bing.com	hotjar.com	tealiumiq.com
__gpi	doubleclick.net	Google	10.11%	10	doubleclick.net	googleadservices.com	ezoic.net
_clck	clarity.ms	Microsoft	8.81%	9	tealiumiq.com	driftt.com	lmiutil.com
_hjTLDTest	hotjar.com	Hotjar	8.05%	1071	azercell.com	musinsa.com	google-analytics.com
_clsk	clarity.ms	Microsoft	7.88%	7	tealiumiq.com	driftt.com	clicktripz.com
cto_bundle	criteo.net	Criteo	5.98%	7	criteo.com	fullstory.com	ezoic.net
_ym_d	yandex.ru	Yandex	4.85%	178	google-analytics.com	yandex.ru	doubleclick.net
_ym_uid	yandex.ru	Yandex	4.85%	48	yandex.ru	adfox.ru	google-analytics.co
_pin_unauth	pinimg.com	Pinterest	4.57%	7	tealiumiq.com	fullstory.com	azure.com
__utma	google-analytics.com	Google	4.32%	3	google-analytics.com	fullstory.com	ringostat.net
__utmb	google-analytics.com	Google	4.32%	5	google-analytics.com	piwik.pro	intellimize.co
__utmz	google-analytics.com	Google	4.32%	2	google-analytics.com	ringostat.net	
__qca	quantserve.com	Quantcast	4.19%	29	rubiconproject.com	yahoo.com	openx.net
__utmc	google-analytics.com	Google	4.17%	5	fullstory.com	google.com	google-analytics.com
_ttp	tiktok.com	TikTok	3.75%	3	tealiumiq.com	m-pages.com	clicktripz.com
hubspotutk	hs-analytics.net	HubSpot	3.29%	34	hubspot.com	facebook.com	hsforms.com

Table 4: Classification accuracy of CookieGraph, WebGraph, and CookieBlock

Classifier	Accuracy	Precision	Recall
CookieGraph	90.20%	89.84%	92.52%
WebGraph	78.74%	71.59%	85.49%
CookieBlock	72.45%	69.95%	80.78%

We divide our breakage analysis into four categories of typical website usage: navigation (from one page to another), SSO (initiating and maintaining login state), appearance (visual consistency), and miscellaneous functionality (chats, search, shopping cart, etc.). We label breakage as major or minor for each category: major breakage – when it is not possible to use the functionality on the site included in either of the aforementioned categories, and minor breakage – when it is difficult, but not impossible, for the user to make use of the functionality. To assess website breakage, we compare a vanilla Chrome browser (with no countermeasures against first-party cookies) with browsers enhanced with an extension which blocks all first-party cookies classified as ATS by CookieGraph, enhanced with an extension which blocks all cookies set by resources labeled as ATS by WebGraph, and enhanced with the official CookieBlock extension [19]. We also include two additional configurations in this analysis, filter lists [8, 9], and a Google Chrome with all cookies blocked. We use two reviewers to perform the breakage analysis to mitigate the impact of biases or subjectivity. Any disagreements between the reviewers were resolved after careful discussion.

Out of the 50 sites, CookieGraph only had major breakage on one site where a cookie popup kept freezing up and preventing navigation around the website due to the deletion of a cookie that stores user preferences. In contrast, WebGraph, CookieBlock, and filter lists cause major breakage in one of the four categories on at least 6% of the sites. For example, WebGraph causes issues with cart functionality on etsy.com, complete homepage breakage on aliexpress.us, and SSO issues on other sites. Most of the breakage issues of CookieBlock relate to SSO logins and additional login-dependent functionality (*e.g.*, missing profile picture). Our results, that CookieBlock causes breakage on 10% of the sites with SSO logins, are similar to the 7-8% breakage reported by the authors [39]. Blocking all cookies results in major breakage on 32 percent of the sites tested, with SSO and cart functionality proving to be the most recurring issue.

We also find that WebGraph blocks some additional first-party cookies that are important for server-side functionality, but not directly related to user experience and therefore not immediately perceptible. For example, WebGraph blocks essential cookies such as `Bm_sz` cookie used by Akamai for bot detection, `XSRF-TOKEN` cookie used to prevent CSRF on different sites, and `AWSALB` cookies used by Amazon for load balancing. CookieGraph correctly classified these cookies at Non-ATS, and thus does not prevent these measures from being deployed.

5.6.3 Robustness. We compare the robustness to evasion of CookieGraph, CookieBlock, and WebGraph, *i.e.*, to intentional modifications of the cookies to cause the misclassification of ATS cookies as Non-ATS. Since ATS are known to engage in the arms race with

Table 5: Website breakage comparison of all three countermeasures.(🟩) signifies no breakage, (🟧) minor breakage, and (🟥) major breakage. Each cell represents the percentage of sites on which breakage was observed.

Classifier	**Navigation**		**SSO**		**Appearance**		**Miscellaneous**	
	Minor	Major	Minor	Major	Minor	Major	Minor	Major
CookieGraph	0%	2%	0%	0%	0%	0%	0%	0%
WebGraph	6%	2%	0%	2%	4%	2%	2%	2%
CookieBlock	2%	0%	0%	10%	0%	0%	2%	2%
Filter lists	4%	2%	0%	2%	2%	2%	2%	4%
No Cookies	8%	8%	0%	32%	6%	12%	2%	28%

privacy-enhancing tools [36, 60, 65], it is important to test whether the detection of first-party ATS cookies is brittle in the face of trivial manipulation attempts such as changing cookie names.

We evaluate robustness on a test set of 2,000 sites from our dataset which also have the required CMP needed by CookieBlock for data collection and training. This translates to a total set of 7,726 first-party cookies. We change the names of the cookies in our test set to randomly generated strings between 2 and 15 characters. Both CookieGraph and WebGraph are fully robust to manipulation of cookies names while CookieBlock's accuracy degrades by more than 15.68%, while precision and recall degrade by 15.08% and 16.54% respectively. CookieGraph and WebGraph are robust because they do not use any content features (features related to the cookie characteristics, such as cookie name or domain) since these can be somewhat easily manipulated by an adversary aiming to evade classification [77]. On the contrary, the most important feature of CookieBlock depends on the cookie name, i.e., whether the name belongs to the top-500 most common cookie names [38].

CookieGraph's implementation of flow features can be manipulated by an adversary by using a different encoding than it currently considers or by changing the domains of exfiltration endpoints. CookieGraph's robustness to these attacks can be improved by more comprehensive information flow tracking. However, full-blown information flow tracking would incur prohibitively high run-time overheads (up to 100X-1000X [56]) and implementation complexity in the browser [42, 43, 66, 79].

To assess the robustness of CookieGraph against manipulation of these flow features, we remove the features related to the flow of cookie information (exfiltration and infiltration of first-party ATS cookies) and then re-train/test the classifier. We find that CookieGraph's accuracy drops by only 2% when exfiltration and infiltration features are removed. Our feature analysis using information gain shows that, instead of focusing on exfiltration features, CookieGraph shifts focus to other features such as the number of local storage accesses by a script and redirections by cookie setters. While there is a slight performance degradation when these features are removed, CookieGraph is able to adapt and still outperforms existing countermeasures by more than 10% in terms of classification accuracy.

6 LIMITATIONS

6.1 Completeness

CookieGraph relies on a graph representation of interactions between different elements during webpage execution. The number of interactions captured depends on the intensity and variety of user activity on a webpage (*e.g.*, scrolling activity, number of internal pages clicked). Thus, it is possible that CookieGraph does not detect certain ATS cookies if user activity is insufficient as that would mean that its graph representation has not captured particular interactions between different elements in the webpage.

To study the impact of user activity, we recrawl sites performing two to three times more internal page clicks than in the original crawl. We specifically recrawl 238 sites where Criteo's `cto_bundle` cookie was originally classified as Non-ATS by CookieGraph. CookieGraph's deployment on the recrawled sites results in successful detection of Criteo's `cto_bundle` cookie as ATS on 121 of the 238 recrawled sites. We find that the average number of infiltrations (exfiltrations) increase from 1.54 to 2.95 (1.13 to 4.01) across the original and recrawled sites. We observed a similar trend for other prevalent first-party ATS cookies in our dataset.

We surmise that while there are cases where CookieGraph incorrectly classifies ATS as Non-ATS due to incompleteness of the graph representation, its decision reflects the behavior of the cookie at the time of classification. As more interaction is captured in the graph, CookieGraph is able to correctly switch the label to ATS. More importantly, CookieGraph never switches labels from ATS to Non-ATS due to increased interaction.

6.2 Deployment

CookieGraph's implementation is not suitable for runtime deployment due to the performance overheads associated with the browser instrumentation and machine learning pipeline. We envision CookieGraph to be used in an offline setting: First first-party ATS cookie-domain pairs are detected using CookieGraph and (2) the detected cookie-domain pairs are added to a cookie filter list such as those already supported in privacy-enhancing browser extensions (e.g., uBlock Origin [33]) for run-time blocking. We argue that a reasonably frequent (e.g., once a week) deployment of CookieGraph on a large scale would be sufficient in generating and keeping the filter list up-to-date. While advertisers and trackers can in theory change cookie names at a rate faster than CookieGraph's periodic deployment, updating cookie names frequently is challenging in practice because setting these first-party ATS cookies across many different sites requires tight coordination between different entities. To illustrate the practical issues associated with changing cookie names, consider the legacy `demdex` cookie set by Adobe's embedded script that is then exfiltrated to the demdex.net domain. Adobe's documentation explains that it is difficult to change the legacy name because "... it is entwined deeply with Audience Manager, the Adobe Experience Cloud ID Service, and our installed user base" [5, 15]. If advertisers or trackers are somehow able to overcome these practical challenges and change cookie names at a much faster pace, CookieGraph's online implementation for run-time cookie classification would be necessary. Further research is needed for efficient and effective online implementation of CookieGraph.

7 CONCLUSION

In this paper, we investigated the use of first-cookies for tracking. Through a large-scale differential measurement, we showed that trackers use first-party cookies to exfiltrate identifiers even when third-party cookies are blocked. We found that third-party cookie blocking is ineffective and blanket first-party cookie blocking is not practical because it results in major functionality breakage on almost one-third of sites. To detect and block first-party tracking cookies, we proposed CookieGraph, a machine-learning approach that captures fundamental tracking behaviors exhibited by first-party cookies. Our evaluation showed that CookieGraph outperformed the state-of-the-art in terms of detection accuracy, minimization of website breakage, and robustness to evasion attacks. Our deployment of CookieGraph on 10K websites provided evidence of widespread use of first-party tracking cookies on 93.43% of the tested sites. These first-party tracking cookies are set by third-party embedded scripts served from 1,588 domains that include major advertising entities such as Google, Facebook, and TikTok.

REFERENCES

[1] 1996. This bug in your PC is a smart cookie. https://archive.org/details/FinancialTimes1996UKEnglish.

[2] 2001. Internet Privacy with IE6 and P3P: A Summary of Findings. http://web.archive.org/web/20200731061208/http://www.spywarewarrior.com/uiuc/ie6-p3p.htm.

[3] 2022. AdBlock Plus. https://adblockplus.org/. https://adblockplus.org/

[4] 2022. Attentive cookie. https://docs.attentivemobile.com/pages/developer-guides/third-party-integrations/referral-marketing-platforms/talkable/. https://docs.attentivemobile.com/pages/developer-guides/third-party-integrations/referral-marketing-platforms/talkable/

[5] 2022. Cookies and the Experience Cloud Identity Service. https://experienceleague.adobe.com/docs/id-service/using/intro/cookies.html?lang=en. https://experienceleague.adobe.com/docs/id-service/using/intro/cookies.html?lang=en

[6] 2022. Disconnect tracking protection lists. https://disconnect.me/trackerprotection. https://disconnect.me/trackerprotection

[7] 2022. DoubleClick. https://web.archive.org/web/19970405225532/http://www.doubleclick.com/.

[8] 2022. EasyList. https://easylist.to/easylist/easylist.txt.

[9] 2022. EasyPrivacy. https://easylist.to/easylist/easyprivacy.txt.

[10] 2022. Hubspot cookie. https://knowledge.hubspot.com/reports/what-cookies-does-hubspot-set-in-a-visitor-s-browser. https://knowledge.hubspot.com/reports/what-cookies-does-hubspot-set-in-a-visitor-s-browser

[11] 2022. ID5 - First Party IDs and Identity Resolution Methods Explained. https://web.archive.org/web/20220408035339/https://id5.io/news/index.php/2022/03/24/first-party-ids-and-identity-resolution-methods-explained/.

[12] 2022. Omnisend cookie. https://support.omnisend.com/en/articles/1933402-explaining-and-managing-tracking-cookies. https://support.omnisend.com/en/articles/1933402-explaining-and-managing-tracking-cookies

[13] 2022. One Trust. Cookiepedia. https://cookiepedia.co.uk.

[14] 2022. Tracking Prevention in Microsoft Edge. https://docs.microsoft.com/en-us/microsoft-edge/web-platform/tracking-prevention.

[15] 2022. Understanding Calls to the Demdex Domain. https://experienceleague.adobe.com/docs/audience-manager/user-guide/reference/demdex-calls.html?lang=en. https://experienceleague.adobe.com/docs/audience-manager/user-guide/reference/demdex-calls.html?lang=en

[16] n.d.. About publisher provided identifiers. https://web.archive.org/web/20220614165742/https://support.google.com/admanager/answer/2880055?hl=en.

[17] n.d.. Alexa Site Ranking. https://www.alexa.com/.

[18] n.d.. Cartographer Identity Graph. https://web.archive.org/web/20220526085916/https://www.lotame.com/solutions/cartographer-identity-graph/.

[19] n.d.. CookieBlock. https://github.com/dibollinger/CookieBlock.

[20] n.d.. Criteo Online Identification). https://web.archive.org/web/20220819071808/https://filecache.investorroom.com/mr5ir_criteo/977/download/Criteo_Online_Identification_May2020.pdf/.

[21] n.d.. fbp and fbc Parameters. https://web.archive.org/web/20220722220344/https://developers.facebook.com/docs/marketing-api/conversions-api/parameters/fbp-and-fbc/.

[22] n.d.. Firefox rolls out Total Cookie Protection by default to all users worldwide. https://blog.mozilla.org/en/products/firefox/firefox-rolls-out-total-cookie-protection-by-default-to-all-users-worldwide/.

[23] n.d.. Firefox's protection against fingerprinting. https://support.mozilla.org/en-US/kb/firefox-protection-against-fingerprinting.

[24] n.d.. Google Analytics Cookie Usage on Websites). https://web.archive.org/web/20220812222800/https://developers.google.com/analytics/devguides/collection/gtagjs/cookie-usage.

[25] n.d.. ID5 Identity Cloud. https://web.archive.org/web/20220727094611/https://www.id5.io/identity-cloud/.

[26] n.d.. Identity Guide. https://web.archive.org/web/20220115155115/https://yieldbird.com/identity-guide/.

[27] n.d.. It's their word against their source code - TikTok Report. https://internet2-0.com/whitepaper/its-their-word-against-their-source-code-tiktok-report/.

[28] n.d.. Lotame – Data Collection Guide. https://web.archive.org/web/20210730071853/https://my.lotame.com/t/p8hxvnd/data-collection-guide.

[29] n.d.. Lotame Lightning Tag. https://web.archive.org/web/20220307010702/https://my.lotame.com/t/m1hxv7l/lotame-lightning-tag.

[30] n.d.. Our New Approach to Address the Rise of Fingerprinting. https://blog.disconnect.me/our-new-approach-to-address-the-rise-of-fingerprinting/.

[31] n.d.. Panorama ID. https://web.archive.org/web/20220327180718/https://www.lotame.com/panorama/id/.

[32] n.d.. TikTok Adds Third-Party Cookies To Its Pixel – And Tries To Eat Facebook's Lunch. https://web.archive.org/web/20220623232016/https://www.adexchanger.com/online-advertising/tiktok-adds-third-party-cookies-to-its-pixel-and-tries-to-eat-facebooks-lunch/.

[33] n.d.. uBlock Origin: Resources Library. https://github.com/gorhill/uBlock/wiki/Resources-Library#cookie-removerjs-.

[34] n.d.. Using Cookies with TikTok Pixel. https://web.archive.org/web/20220610074648/https://ads.tiktok.com/help/article?aid=10007540.

[35] n.d.. What Facebook's First-Party Cookie Means for AdTech. https://web.archive.org/web/20220729210450/https://clearcode.cc/blog/facebook-first-party-cookie-adtech/.

[36] Mshabab Alrizah, Sencun Zhu, Xinyu Xing, and Gang Wang. 2019. Errors, Misunderstandings, and Attacks: Analyzing the Crowdsourcing Process of Adblocking Systems. In *Proceedings of the 2019 Internet Measurement Conference (IMC)*.

[37] Waqar Aqeel, Balakrishnan Chandrasekaran, Anja Feldmann, and Bruce M Maggs. 2020. On landing and internal web pages: The strange case of jekyll and hyde in web performance measurement. In *Proceedings of the ACM Internet Measurement Conference*.

[38] Dino Bollinger. n.d.. Analyzing Cookies Compliance with the GDPR. https://www.research-collection.ethz.ch/handle/20.500.11850/477333. Thesis, ETH Zurich.

[39] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. 2022. Automating Cookie Consent and GDPR Violation Detection. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association.

[40] Aaron Cahn, Scott Alfeld, Paul Barford, and S. Muthukrishnan. 2016. An Empirical Study of Web Cookies. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 891–901.

[41] Quan Chen, Panagiotis Ilia, Michalis Polychronakis, and Alexandros Kapravelos. 2021. Cookie Swap Party: Abusing First-Party Cookies for Web Tracking. In *Proceedings of the Web Conference*.

[42] Quan Chen and Alexandros Kapravelos. 2018. Mystique: Uncovering information leakage from browser extensions. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1687–1700.

[43] Andrey Chudnov and David A Naumann. 2015. Inlined information flow monitoring for JavaScript. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 629–643.

[44] L. Montulli D. Kristol. 1997. HTTP State Management Mechanism. https://datatracker.ietf.org/doc/html/rfc2109.

[45] Savino Dambra, Iskander Sanchez-Rola, Leyla Bilge, and Davide Balzarotti. 2022. When Sally Met Trackers: Web Tracking From the Users' Perspective. In *USENIX Security Symposium*.

[46] Ha Dao, Johan Mazel, and Kensuke Fukuda. 2021. CNAME Cloaking-Based Tracking on the Web: Characterization, Detection, and Protection. *IEEE Transactions on Network and Service Management* (2021).

[47] Yana Dimova, Gunes Acar, Lukasz Olejnik, Wouter Joosen, and Tom Van Goethem. 2021. The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion. *PETS* (2021).

[48] Díaz-Morales and Roberto. 2015. Cross-Device Tracking: Matching Devices and Cookies. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*. 1699–1704.

[49] Brendan Eich. 2013. C is for Cookie. https://brendaneich.com/2013/05/c-is-for-cookie/.

[50] Brendan Eich. 2013. The Cookie Clearinghouse. https://brendaneich.com/2013/06/the-cookie-clearinghouse/.

[51] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of ACM CCS 2016*.

[52] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. 2015. Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *Proceedings of the 24th International Conference on World Wide Web*.

[53] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. 2020. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. *Proceedings on Privacy Enhancing Technologies* 2020 (04 2020),

499–518. https://doi.org/10.2478/popets-2020-0038

[54] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. 2022. My Cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting. In *Privacy Enhancing Technologies Symposium (PETS)*.

[55] Google. n.d.. The Privacy Sandbox. https://developer.chrome.com/docs/privacy-sandbox/.

[56] Daniel Hedin, Arnar Birgisson, Luciano Bello, and Andrei Sabelfeld. 2014. JSFlow: Tracking information flow in JavaScript and its APIs. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. 1663–1671.

[57] Maximilian Hils, Daniel W Woods, and Rainer Böhme. 2020. Measuring the emergence of consent management on the web. In *Proceedings of the ACM Internet Measurement Conference*.

[58] Xuehui Hu, Nishanth Sastry, and Mainack Mondal. 2021. CCCC: Corralling Cookies into Categories with CookieMonster. In *13th ACM Web Science Conference 2021*. Association for Computing Machinery, 234–242.

[59] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.

[60] Umar Iqbal, Zubair Shafiq, and Zhiyun Qian. 2017. The Ad Wars: Retrospective Measurement and Analysis of Anti-Adblock Filter Lists. In *IMC*.

[61] Umar Iqbal, Peter Snyder, Shitong Zhu, Benjamin Livshits, Zhiyun Qian, and Zubair Shafiq. 2020. AdGraph: A Graph-Based Approach to Ad and Tracker Blocking. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.

[62] Umar Iqbal, Charlie Wolfe, Charles Nguyen, Steven Englehardt, and Zubair Shafiq. 2022. Khaleesi: Breaker of Advertising and Tracking Request Chains. In *USENIX Security Symposium (USENIX)*.

[63] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)* 14, 2 (2020), 1–33.

[64] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. 2016. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *2016 IEEE Symposium on Security and Privacy (SP)*.

[65] Hieu Le, Athina Markopoulou, and Zubair Shafiq. 2021. CV-Inspector: Towards Automating Detection of Adblock Circumvention. In *Network and Distributed System Security Symposium (NDSS)*.

[66] Sebastian Lekies, Ben Stock, and Martin Johns. 2013. 25 million flows later: Large-scale detection of DOM-based XSS. In *Proceedings of the 2013 ACM SIGSAC conference on Computer and Communications Security*. 1193–1204.

[67] Pedro Giovanni Leon, Lorrie Faith Cranor, Aleecia M McDonald, and Robert McGuire. 2010. Token attempt: the misrepresentation of website privacy policies through the misuse of p3p compact policy tokens. In *Proceedings of the 9th Annual ACM Workshop on Privacy in the Electronic Society*.

[68] MDN. 2022. Redirect tracking protection. https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Privacy/Redirect_Tracking_Protection. https://developer.mozilla.org/en-US/docs/Mozilla/Firefox/Privacy/Redirect_Tracking_Protection

[69] Lou Montulli. 2013. The Reasoning Behind Web Cookies. http://montulli.blogspot.com/2013/05/the-reasoning-behind-web-cookies.html.

[70] Nick Nguyen. 2018. Latest Firefox Rolls Out Enhanced Tracking Protection. https://blog.mozilla.org/en/products/firefox/latest-firefox-rolls-out-enhanced-tracking-protection/.

[71] ChangSeok Oh, Chris Kanich, Damon McCoy, and Paul Pearce. 2022. Cart-Ology: Intercepting Targeted Advertising via Ad Network Identity Entanglement. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*.

[72] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2019. Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask. In *Proceedings of the World Wide Web (WWW) Conference*.

[73] 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.

[74] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and Defending Against Third-Party Tracking on the Web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)* (San Jose, CA). 155–168.

[75] Iskander Sanchez-Rola, Matteo Dell'Amico, , Davide Balzarotti, Pierre-Antoine Vervier, and Leyla Bilge. 2021. Journey to the center of the cookie ecosystem: Unraveling actors'; roles and relationships. In *S&P 2021, 42nd IEEE Symposium on Security & Privacy, 23-27 May 2021, San Francisco, CA, USA*.

[76] Justin Schuh. 2020. Building a more private web: A path towards making third party cookies obsolete. https://blog.chromium.org/2020/01/building-more-private-web-path-towards.html.

[77] Sandra Siby, Umar Iqbal, Steven Englehardt, Zubair Shafiq, and Carmela Troncoso. 2022. WebGraph: Capturing Advertising and Tracking Information Flows for Robust Blocking. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association.

[78] Alexander Sjösten, Peter Snyder, Antonio Pastor, Panagiotis Papadopoulos, and Benjamin Livshits. 2020. Filter List Generation for Underserved Regions. In *WWW*.

[79] Ben Stock, Sebastian Lekies, Tobias Mueller, Patrick Spiegel, and Martin Johns. 2014. Precise Client-side Protection against DOM-based Cross-Site Scripting. In *23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA, 655–670.

[80] Microsoft Edge Team. 2022. Introducing tracking prevention, now available in Microsoft Edge preview builds. https://blogs.windows.com/msedgedev/2019/06/27/tracking-prevention-microsoft-edge-preview/. https://blogs.windows.com/msedgedev/2019/06/27/tracking-prevention-microsoft-edge-preview/

[81] Alessandra Van Veen and AP de Vries. 2021. Cookie Compliance of Dutch Hospital Websites. (2021).

[82] WebKit. 2022. Tracking Prevention in WebKit. https://webkit.org/tracking-prevention/. https://webkit.org/tracking-prevention/

[83] John Wilander. 2017. Intelligent Tracking Prevention. https://webkit.org/blog/7675/intelligent-tracking-prevention/.

[84] John Wilander. 2018. Intelligent Tracking Prevention 1.1. https://webkit.org/blog/8142/intelligent-tracking-prevention-1-1//.

[85] John Wilander. 2018. Intelligent Tracking Prevention 2.0. https://webkit.org/blog/8311/intelligent-tracking-prevention-2-0/.

[86] John Wilander. 2019. Intelligent Tracking Prevention 2.1. https://webkit.org/blog/8613/intelligent-tracking-prevention-2-1/.

[87] John Wilander. 2019. Intelligent Tracking Prevention 2.2. https://webkit.org/blog/8828/intelligent-tracking-prevention-2-2/.

[88] John Wilander. 2019. Intelligent Tracking Prevention 2.3. https://webkit.org/blog/9521/intelligent-tracking-prevention-2-3/.

[89] John Wilander. 2020. Full Third-Party Cookie Blocking and More. https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more/.

A APPENDIX

A.1 Case Studies

In this section, we look at case studies of ATSes identified in Section 4.3 which are found to be extensively using first-party cookies for tracking purposes. We analyze the behavior of these ATS in our crawls, compare the observed behavior with their documentation, and create a generic model which all first-party-cookie-based ATSes follow in Section 3. We present case studies of three ATSes here: Lotame, ID5, and Criteo.

A.1.1 Lotame. Lotame is a data and identity management solution which claims to provide a single ID to users across multiple browsers, devices, and platforms. Lotame's Lightning Tag [29] packages the user visit data in a JSON object and sends it to its servers. Code 1 shows an example payload sent to Lotame. The payload includes IDs assigned by the website, third-party identifiers present on the site, certain user behaviors (configured through collaboration between the publisher and Lotame), and other custom rules defined per website [28]. Lotame processes the payload and matches the data with its *Cartographer Identity Graph* [18], and sends back an ID, called `panoramaID` [31], which is stored as a first-party cookie or in localStorage.

A.1.2 ID5 Universal ID. ID5 provides identity resolution for publishers and advertisers through its *Identity Cloud* [25]. ID5's script packages a payload that contains several deterministic identifiers, such as email, usernames, and phone numbers (if available) and as well as probabilistic identifiers include, such as IP address, user agent, and location of the user [11]. ID5 then processes the payload and matches the data with its Identity Cloud and send back an ID, called *universal_id*, which is stored as a first-party cookie and as well as in local storage. An example payload from ID5 is shown in Figure 2. We note that ID5 also provides *Partner Graph*, a service that enables information sharing among its partners [25]. Partner Graph allows different identify providers to exchange information with each other.

A.1.3 Criteo. Criteo provides *Criteo Identity Graph* for identity resolution [20]. Criteo Identity Graph is built from four different sources: (i) data contributed by advertisers, (ii) data collected from publisher websites by Criteo itself, (iii) data provided by Criteo partners such as LiveRamp and Oracle, (iv) and predictions on existing data by Criteo's machine learning models. Criteo claims that its identity graph is able to stitch together identifiers from more than 2 billion users across the world, and that it contains persistent deterministic identifiers for 96% of the users [20]. Similar to other identity resolution services, Criteo generates an ID, based on identifiers, such as hashed emails, mobile device IDs, cookie IDs, and stores it in first-party storage as `cto_bundle`. Their documentation shows that Criteo makes use of both first-party cookies and localStorage for storing `cto_bundle` cookie. We consider this to be one of the fundamental behaviors of first-party ATS cookies. As described in Section 5.1, CookieGraph's graph representation abstracts storage to refer to both Cookies and localStorage. We also

```
data: {
  behaviorIds: [1,2,3],
  behaviors: {
    int: ['behaviorName', 'behaviorName2'],
    act: ['behaviorName']
  },
  ruleBuilder: {
    key1: ['value 1a', 'value 1b']
  },
  thirdParty: {
    namespace: 'NAMESPACE',
    value: 'TPID_VALUE'
  }
}
```

Code 1: Example of data sent structure sent to Lotame during a user's first visit.

```
{
   "created_at":"2022-02-09T11:42:40.817811Z",
   "id5_consent":true,
   "original_uid":"ID5*
        FnFOGLkYzdJjuoK3KvAecVW2oFpZ7OrZiW7h-M0H
       ACAHYuWkxQrEGcpWuOkQUXbHB2OO8Rj0wt94jllT
       WHQ6wdkqOwSbnYea8cesuONCF4HZeIoDaB_TwBsy
       lKrs3tHB2Y87ZwP0DrpYlGz1OG1Fgdn0YdgqoSGU
       SGxzS1gUzsHaMIUBVqf2I08es6aUULEB2n48oyL0
       nGnRtstVqtcQQdquS3Aay4Hhgbzh9gIZyYHa_nLT
       d5rjbbR0ZXwkXDzB2yU1XUC2dukip1J_clVAgdtt
       xC_xaRRBOLi0fnvp9cHbqr_pWihTtaUMS_R6eLuB
       2_AMExt1UdhJZBe2mcXZAdwm9lcbeMMvlpg3MBrC
       oHcUgzNypi-5xLUqBD8GC4B3KsefcNkiDvI4n9ZL
       7OjAdzqB9PD-KczAx63Ck0gEIHdNPfQeEi-f5VaO
       OEhf6B3VUqDoL11hqVoIuDhKJbgd2kp0mgXabhwJ
       tPO7sgWwHd_vz_uIYYmqQBTbH-JFVB3h1-kI9GQv
       dby2PyftDawd596ho3tuOsKtoDOk4S4Her2Uw-_u
       BYRxrt6YzVYqB3tRwTVI3Fxm8cGJyjdmYAd88lom
       BIpkOeg2Ok4VTNc",
   "universal_uid":"ID5*
        HGH7W7iMpMu3-EPZCXUuqNBB7fFHUUVcbSddSSG
       Fu5UHYucsBxMz2jncvKS7rkwlB2MWiiPupapPxa
       79eieMAdkTyMQz82s1vIekPr28DEHZbqTCrapj9
       Fb9K0x4zjlB2YHOKNDwQY6mZwxk_1mwAdna3wWna
       hrpMEUrPxJSnAHaPYB-InS5DXGpQgqbqirB2nHFI
       D4j9i9BgCP3k0VygdqdtFHsT7eeDfFYuB8EQ0Ha4
       -yV9Ifvbvi5oxmtH7HB2xg-mmmOeyVOPBYGi2tfw
       dtREZnUE83cfn_LHvHvu4HbvkLkwEFJiddOEp4PT
       ZbB2-de_VPyKHax5JtpO46xwdwZ_0UMgANOsZygV
       0SrrMHcZ37qQB-LkCO4tWoTbv_B3KMGCMrebcfLE
       TeCn0AEgdzIR1utDJzM6AaiL9KVkAHdPtrAtTv73
       ZyDg92Rq-_B3XeRNOOc7b2CEBsilXOlQd2sfmR36
       NyW-dsK9CUmd4Hd3vcrlAWzfYEfw01Q5J1B3ibAF
       UYrA0XWMl-D9jSlAd5iX1tGA4vPu0wdZkXVOEHek
       q2xibOm9XwN2nSdZjbB3v8nOyzGuF9QgwI67pMGQ
       d85BszRCJDUkiiu-tv5BQ",
   "signature":"ID5_Ab6tnGgmCcjKo-qFGVKszuNpNePqkOHZT
       rbCmpuktLLOlNOCALhmY_91AHP8LU0BvfJT2Q
       JQWlsUEfynB1hBGZc",
   "link_type":1,
   "cascade_needed":true,
   "privacy":{
      "jurisdiction":"other",
      "id5_consent":true
   }
}
```

Code 2: Example of data structure received from ID5 during a user's first visit.

include a count of localStorage accesses in the feature set computed from the graph representation. Inclusion of these features help CookieGraph effectively model first-party ATS cookies behavior.

A.2 Breakage Analysis

Following sites were used for breakage analysis:

- bidswitch.com
- kirkusreviews.com
- csdn.net
- dropbox.com
- trello.com
- deodap.com
- baidu.com
- stvincent.edu
- microsoft.com
- promopult.ru
- twitch.tv
- pikiran-rakyat.com
- etsy.com
- jar-download.com
- seoreviewtools.com
- planvital.cl
- snnp.co.th
- telfar.net
- tribunnews.com
- gop.edu.tr
- coinbase.com
- uideck.com
- vk.com
- amssoft.ru
- generateblocks.com
- swissid.ch
- zhanqi.tv
- google.com.hk
- withgoogle.com
- hindifire.com
- aparat.com
- ebay.com
- shopee.co.th
- chase.com
- medium.com
- box.com
- calendardate.com
- weibo.com
- zoom.us
- google.com
- castlelearning.com
- office.com
- rocketpunch.com
- freepik.com
- aliexpress.us
- huanqiu.com
- plati.market
- daysmartspa.com
- klett.pl
- stackoverflow.co