sciendo

Proceedings on Privacy Enhancing Technologies ; 2021 (3):394–412

Yana Dimova, Gunes Acar, Lukasz Olejnik, Wouter Joosen, and Tom Van Goethem

The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion

Abstract: Online tracking is a whack-a-mole game between trackers who build and monetize behavioral user profiles through intrusive data collection, and anti-tracking mechanisms that are deployed as browser extensions, DNS resolvers, or built-in to the browser. As a response to pervasive and opaque online tracking, more and more users adopt anti-tracking measures to preserve their privacy. Consequently, as the information that trackers can gather on users is being curbed, some trackers are looking for ways to evade these protections. In this paper we report on a large-scale longitudinal evaluation of an anti-tracking evasion scheme that leverages CNAME records to include tracker resources in a same-site context, which effectively bypasses anti-tracking measures that rely on fixed hostname-based block lists. Using historical HTTP Archive data we find that this tracking scheme is rapidly gaining traction, especially among high-traffic websites. Furthermore, we report on several privacy and security issues inherent to the technical setup of CNAME-based tracking that we detected through a combination of automated and manual analyses. We find that some trackers are using the technique against the Safari browser, which is known to include strict anti-tracking configurations. Our findings show that websites using CNAME trackers must take extra precautions to avoid leaking sensitive information to third parties.

Keywords: tracking, CNAME, evasion

DOI 10.2478/popets-2021-0053
Received 2020-11-30; revised 2021-03-15; accepted 2021-03-16.

Yana Dimova: imec-DistriNet, KU Leuven, E-mail: yana.dimova@cs.kuleuven.be
Gunes Acar: imec-COSIC, KU Leuven, E-mail: gunes.acar@esat.kuleuven.be
Lukasz Olejnik: European Data Protection Supervisor, independent researcher, E-mail: me@lukaszolejnik.com
Wouter Joosen: imec-DistriNet, E-mail: wouter.joosen@cs.kuleuven.be
Tom Van Goethem: imec-DistriNet, E-mail: tom.vangoethem@cs.kuleuven.be

1 Introduction

Websites use trackers for various purposes including analytics, advertising and marketing. Although tracking may help websites in monetization of their content, the use of such methods may often come at the expense of users' privacy, for example when it involves building detailed behavioral profiles of users. As a reaction to the omnipresence of online tracking, many countermeasures have been developed, including specialised browser extensions, DNS resolvers, and built-in browser protections. As of today, all major browsers (except Google Chrome) include some forms of anti-tracking measures. Safari's Intelligent Tracking Prevention (ITP) includes multiple features to thwart various forms of tracking and circumvention techniques [60]; Firefox' Enhanced Tracking Protection (ETP) and the tracking prevention mechanism in Edge rely on blocklists to exclude trackers [35, 61].

As a counter-reaction to the increased use of anti-tracking measures, several trackers have resorted to new techniques in an attempt to circumvent these measures. Prominent and well-studied examples of these evasion techniques include browser fingerprinting [6, 23, 24, 28, 42], leveraging various browser mechanisms to persist a unique identifier [11, 26, 53], and creating a fingerprint from hardware anomalies [19, 37, 64]. A notable example for the use of evasion techniques is the case of Criteo, one of the tracking actors we study in this paper. In 2015, Criteo was found to use HTTP redirections to set first-party cookies [14, 45], and later abused the HTTP Strict-Transport-Security mechanism [26, 53], both in an effort to circumvent Safari's Intelligent Tracking Protection (ITP). Our study complements these past reports with an observation that Criteo is applying a specialised form of first-party tracking to Safari browsers.

The evasion technique that we study has been known for several years, but recently gained more attention, presumably due to the increased protection against third-party tracking. This tracking scheme takes advantage of a CNAME record on a subdomain such that it is same-site to the including website. As such, de-

fenses that block third-party cookies are rendered ineffective. Furthermore, because custom subdomains are used, these are unlikely to be included in blocklists (instead of blocking the tracker for all sites, blocklists would have to include every instance for each website including the CNAME-based tracker).

Using the HTTP Archive dataset, supplemented with results from custom crawls, we report on a large-scale evaluation of the CNAME-based tracking ecosystem, involving 13 manually-vetted tracking companies. We find that this type of tracking is predominantly present on popular websites: 9.98% of the top 10,000 websites employ at least one CNAME-based tracker.

The use of such tracking is rising. Through a historical analysis of the ecosystem, we show that the number of websites that rely on this type of tracking is steadily growing, especially compared to similarly-sized tracking companies which have experienced a decline in number of publishers. We find that CNAME-based tracking is often used in conjunction with other trackers: on average 28.43 third-party tracking scripts can be found on websites that also use CNAME-based tracking. This abundance and complexity of trackers result in unexpected privacy leaks. For instance, trackers get access to each other's first-party cookies that are set via the `document.cookie` interface. We find that such practices lead to wide-spread cookie leaks, as they bypass origin-based web security policies enforced by the browsers. Using automated methods we measure such cookie leaks to CNAME-based trackers, and identify cookie leaks on 95% of the sites embedding CNAME-based trackers. Although most of these leaks are due to first-party cookies set by other third-party scripts, we also find cases of cookie leaks to CNAME-based trackers in POST bodies and in URL parameters, which indicates a more active involvement by the CNAME-based trackers.

Furthermore, through a series of experiments, we report on the increased threat surface that is caused by including the tracker as same-site. Specifically, we find several instances where requests are sent to the tracking domain over an insecure connection (HTTP) while the page was loaded over a secure channel (HTTPS). This allows an attacker to alter the response and inject new cookies, or even alter the HTML code effectively launching a cross-site scripting attack against the website that includes the tracker. Same attacks would have negligible consequences if the tracking iframe was included from a cross-site domain. Finally, we detected two vulnerabilities in the tracking functionality of CNAME-based trackers, which could expose visitors' data on *all* publisher websites through cross-site scripting and session-fixation attacks.

In summary, we make the following contributions:

- We provide a large-scale analysis of the CNAME-based tracking scheme, based on a custom detection method that allows us to discover previously unknown trackers.
- Through a longitudinal analysis we find that this form of first-party tracking is becoming increasingly popular and is often used to complement third-party tracking.
- We perform a series of experiments to identify security and privacy threats that are intrinsic to CNAME-based tracking. We identify numerous issues, including the extensive leakage of cookies set by third-party trackers.
- We discuss the various countermeasures that have recently been developed to thwart this type of tracking, and assess to what extent these are resistant to further circumvention techniques.

2 Background

2.1 Web browser requests

Upon visiting a web page, the browser will make various requests to fetch embedded resources such as scripts, style sheets and images. Depending on the relation between the embedding website and the site that the resources are hosted on, these can be *same-origin*, *same-site* or *cross-site*. If the resource shares the same scheme (i.e. http or https), host (e.g. www.example.com) and port (e.g. 80 or 443) as the embedding site, it is considered same-origin. In case there is no exact match for the host, but the resource is located on the same registrable domain name, the effective top level domain plus one (*eTLD+1*), as the embedding website (e.g. www.example.com and foo.example.com), it is considered same-site. Finally, resources that have a different eTLD+1 domain from the including website are considered cross-site, i.e., resources from *tracker.com* included on *example.com* are cross-site.

Prior to making the connection to the server, the browser first resolves the server's domain name to an IP address. In the most straightforward case, the DNS resolution of the domain name returns an `A` record containing the IP address. However, the domain could also use a `CNAME` record to refer to any other domain name. This can be an iterative process as the new domain name

can also refer to another `CNAME` record. This process continues until an `A` record is found. Through this indirection of CNAMEs, the host that the browser connects to may belong to a different party, such as a tracker, than the domain it actually requests the resource from. This means that requests to *xxx.example.com* may actually be routed to a different site, such as *yyy.tracker.com*.

Cookie scoping Before a request is sent, the browser will first determine which cookies to attach to the HTTP request. This includes all cookies that were set on the same (sub)domain as the one where the request will be sent to. Also included in the requests are cookies that were set by a same-site resource, i.e. either on another subdomain, or on the top domain, and had the `Domain` attribute set to the top domain, for instance by the following response header from https://sub.example.com/

```
Set-Cookie: cookie=value; Domain=example.com
```

Cookies that were set without the `Domain` attribute will only be included on requests that are same-origin to the response containing the `Set-Cookie` header. The `SameSite` attribute on cookies determines whether a cookie will be included if the request is cross-site. If the value of this attribute is set to `None`, no restrictions will be imposed; if it is set to `Lax` or `Strict`, it will not be included on requests to resources that are cross-site to the embedding website; the latter imposes further restrictions on top-level navigational requests. Several browser vendors intend to move to a configuration that assigns `SameSite=Lax` to all cookies by default [15, 36, 57]. As such, for third-party tracking to continue to work, the cookies set by the trackers explicitly need to have the `SameSite=None` attribute. However, the transition to SameSite cookies has no effect on CNAME-based trackers, as their tracking requests appear to be same-site.

2.2 Tracking

2.2.1 Third-party tracking

In a typical tracking scenario, websites include resources from third-party trackers in a cross-site context. When a user visits a website with a particular third party, the third party may set a cookie in the user's browser. The next time the user visits a website on which the same tracker is embedded, the browser will include the cookie in the request to the tracker. This scheme allows trackers to identify users across different websites to build detailed profiles of their browsing behavior. Such tracking has triggered privacy concerns and has resulted in substantial research effort to understand the complexity of the tracking ecosystem [25, 38] and its evolution [34].

2.2.2 First-party tracking

In first-party tracking, the script and its associated requests are loaded from or sent to a same-site origin. Consequently, any cookie that is set by the first-party tracker will only be included in the requests to the same site. Historically, one method that was used to bypass this restriction was cookie matching [44, 46]. However, requests that are used to match cookies can be blocked by anti-tracking tools based on simple matching rules. Instead, CNAME-based tracking uses a delegation of the domain name, which circumvents the overwhelming majority of anti-tracking mechanisms currently available to users.

2.2.3 CNAME-based tracking

General overview In the typical case of third-party tracking, a website will include a JavaScript file from the tracker, which will then report ads and analytics related information by sending (cross-site) requests to the tracker domain. With CNAME-based tracking, the same operations are performed, except that the domain that the scripts are included from and where the data is sent to, is a subdomain of the website. For example, the website `example.com` would include a tracking script from `track.example.com`, thus effectively appearing as same-site to the including website. Typically, the subdomain has a CNAME record that points to a server of the tracker. An overview of the CNAME-based tracking scheme is shown in Figure 1.

Bypassing anti-tracking measures The CNAME tracking scheme has direct implications for many anti-tracking mechanisms. Because the requests to the tracking services are same-site (i.e. they point to the same eTLD+1 domain as the visited website), countermeasures that aim to block third-party cookies are effectively circumvented. To address CNAME-based tracking, blocklists such as EasyPrivacy [22] or Disconnect.me [20] would need to contain a unique subdomain for every website that uses CNAME-based tracking, instead of a single entry per tracker. Anti-tracking mechanisms that rely on such blocklists will have greater performance costs with the growing blocklists.

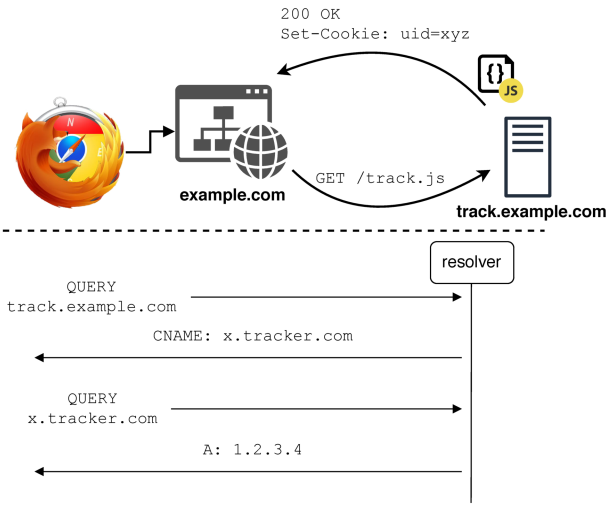


Fig. 1. Overview of CNAME-based tracking.

On the other hand, CNAME-based tracking has certain limitations compared to traditional third-party tracking. For instance, users' visits across different websites cannot be easily tracked using a third-party cookie.

3 Detecting CNAME-based tracking

In this section we describe the datasets and methods we used to detect CNAME-based trackers, and the websites that include them.

3.1 Dataset

In order to analyze the CNAME-based tracking at a scale, we leveraged the (freely available) crawling data from HTTP Archive [10]. The HTTP Archive dataset is based on visiting the home page of all origins from the Chrome User Experience Report (CrUX), which consists of websites (including those hosted on subdomains) frequently visited by Chrome users. The results reported in this section are based on HTTP Archive's desktop crawl performed in October 2020, consisting of 5,506,818 visited web pages from 4,218,763 unique eTLD+1 domains. The dataset includes HTTP headers of all requests and responses (507M in total) that were made when visiting the CrUX web pages with the latest Chrome browser. As the dataset only contains the IP address of the remote servers, we extended the dataset with DNS records (in particular CNAME) obtained by running zdns [63] on all first-party subdomains.

3.1.1 Methodology

Discovering trackers To detect services that offer CNAME-based tracking, we used a three-pronged approach that leverages features intrinsic to this mechanism, combining both automated and manual analysis. First we filtered all requests from HTTP Archive's dataset and only considered the requests that were same-site, but not same-origin — i.e. the same eTLD+1 but not the same origin as the visited web page. Furthermore, we only retained requests to domain names that returned a CNAME record referring (either directly or indirectly after redirection of other CNAME records) to a different eTLD+1 domain in our DNS data. We aggregated these requests on the eTLD+1 of the CNAME record, and recorded a variety of information, such as the average number of requests per website, variation of response sizes, percentage of requests that contain a cookie, or responses that set a cookie. In Appendix A we elaborate on these features and discuss how they could be used to assist or automate the detection of CNAME-based tracking. Out of the resulting 46,767 domains, we only consider the ones that are part of a CNAME-chain on at least 100 different websites, which leaves us with 120 potential CNAME-based trackers.

In the second phase, we performed a manual analysis to rule out services that have no strict intention to track users. Many services that are unrelated to tracking, such as CDNs, use a same-site subdomain to serve content, and may also set a cookie on this domain, thus giving them potential tracking *capabilities*. For instance, Cloudflare sets a `_cfduid` cookie in order to detect malicious visits, but does not intend to track users with this cookie (user information is kept less than 24 hours) [16]. For each of the 120 domains, we visited the web page of the related organization (if available) and gathered information about the kind of service(s) it provides according to the information and documentation provided on its website. Based on this information, we then determined whether tracking was the main service provided by this company, either because it explicitly indicated this, or tracking would be required for the main advertised product, e.g. in order to provide users with personalized content. For instance one such provider, Pardot offers a service named "Marketing Automation", which they define as "a technology that helps businesses grow by automating marketing processes, tracking cus-

tomer engagement, and delivering personalized experiences to each customer across marketing, sales, and service"[1], indicating that customers (website visitors) may be tracked. Finally, we validate this based on the requests sent to the purported tracker when visiting a publisher website: we only consider a company to be a tracker when a uniquely identifying parameter is stored in the browser (e.g. via cookies or localStorage) and sent along with subsequent requests. Using this method, we found a total of five trackers. Furthermore, we extended the list with eight trackers from the CNAME cloaking blocklist by NextDNS [17, 41]. Four of the trackers we detected in our manual analysis were not included in the blocklist. We left two of the trackers from the NextDNS's list out of consideration, as they were not included in the DNS data. We considered the remaining 13 CNAME-based trackers in the study.

Detecting the prevalence of CNAME-based tracking By examining request information to hostnames having a CNAME record to one of the identified trackers, we manually constructed a *signature* for all tracking requests for each of the 13 trackers, based on the DNS records and request/response information (e.g. the same JavaScript resource being accessed or a request URL according to a specific pattern). This allows us to filter out instances where a resource was included from a tracking provider but is unrelated to tracking, as the providers may offer various other services and simply relying on DNS data to detect CNAME publisher domains may lead to an overestimation (we justify this claim in Section 5.2). Using this approach, we detected a total of 10,474 websites (eTLD+1) that used at least one of the trackers. We explore these publishers that use CNAME tracking in more detail in Section 4.2.

3.2 Alternative user agent

A limitation of the HTTP Archive dataset, is that all websites were visited with the Chrome User-Agent string, a browser that does not have built-in tracking protection. Furthermore, only the home page of each website was visited. To evaluate whether these limitations would affect our results, we performed a crawling experiment on the Tranco top 10,000 websites[2] [33]. For each website, we visited up to 20 web pages (totaling 146,397 page visits). We performed the experiment twice: once with the Chrome User-Agent string, and once with Safari's. The latter is known for its strict policies towards tracking, and thus may receive a different treatment. We used a headless Chrome instrumented through the Chrome DevTools Protocol [48] as our crawler. A comparative analysis of these two crawls showed that one tracker, namely Criteo, would only resort to first-party tracking for Safari users. Previously, this tracker was found to abuse top-level redirections [45] and leverage the HTTP Strict Transport Security (HSTS) mechanism to circumvent Safari's ITP [26, 53].

3.3 Coverage

Finally, to evaluate the representativeness of our results and determine whether the composition of the HTTP Archive dataset affected our detection, we performed a comparative analysis with our custom crawl. In the 8,499 websites that were both in the Tranco top 10k, and the HTTP Archive dataset, we found a total of 465 (5.47%) websites containing a CNAME-based tracker. These included 66 websites that were not detected to contain CNAME-based tracking based on the data from HTTP Archive (as it does not crawl through different pages). On the other hand, in the HTTP Archive dataset we found 209 websites that were detected to contain a CNAME-based tracker, which could not be detected as such based on our crawl results. This is because the HTTP Archive dataset also contains popular subdomains, which are not included in the Tranco list. As such, we believe that the HTTP Archive dataset provides a representative view of the state of CNAME-based tracking on the web. We note however that the numbers reported in this paper should be considered lower bounds, as certain instances of tracking can only be detected when crawling through multiple pages on a website.

4 CNAME-based tracking

In this section, we provide an in-depth overview of the CNAME-based tracking ecosystem through a large-scale analysis.

1 https://www.pardot.com/what-is-marketing-automation/

2 Generated on 17 May 2020, available at https://tranco-list.eu/list/6WGX/10000

4.1 CNAME-based trackers

An overview of the detected trackers can be found in Table 1. For every tracker we indicated the number of publishers, counted as the number of unique eTLD+1 domains that have at least one subdomain set up to refer to a tracker (typically with a CNAME record). Furthermore, we estimated the total number of publishers by levering DNS information from the SecurityTrails API [54]. More precisely, all CNAME-based trackers either require the publishers that include them to set a CNAME record to a specific domain, or the trackers create a new subdomain for every publisher. As such, the estimated number of publishers could be determined by finding the domains that had a CNAME record pointing to the tracker, or by listing the subdomains of the tracker domain and filtering out those that did not match the pattern that was used for publishers. For Ingenious Technologies we were unable to estimate the total number of publishers as they use a wildcard subdomain (and thus it could not be determined whether a subdomain referred to an actual publisher using CNAME tracking).

We noted the price of the services offered by the tracker suppliers when such information was available, either from the tracker's website or through third-party reviews. All trackers except TraceDock offered a range of services including analytics and marketing. TraceDock, on the other hand, focuses on providing mechanisms for circumvention of anti-tracking techniques.

Finally, for every tracker we determined whether tracking requests would be blocked by three relevant anti-tracking solutions: uBlock Origin (version 1.26) on both Firefox and Chrome, and the NextDNS CNAME blocklist [40], which was used to extend the list of trackers we considered. As of version 1.25, uBlock Origin on Firefox implements a custom defense against CNAME-based tracking [5]. The defense is based on resolving the domain name of requests that are originally not filtered by the standard blocklist, and checking again the resolved CNAME records against the blocklist. Because Chrome does not support a DNS resolution API for extensions, the defense could not be deployed by uBlock Origin in this browser. Consequently, we find that four of the CNAME-based trackers (Oracle Eloqua, Eulerian, Criteo, and Keyade) are blocked by uBlock Origin on Firefox but not on the Chrome version of the same extension.

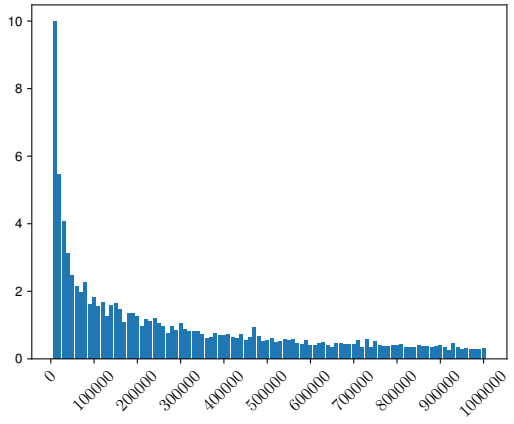


Fig. 2. Percentage of websites using CNAME-based tracking per bin of 10,000 ranks.

4.2 Tracking publishers

As a result of our analysis of the HTTP Archive dataset, we detected 10,474 eTLD+1 domains that had a subdomain pointing to at least one CNAME-based tracker, with 85 publishers referring to two different trackers. We find that for 9,501 publisher eTLD+1s the tracking request is included from a same-site origin, i.e., the publisher website has the same eTLD+1 as the subdomain it includes tracker content from. Furthermore, on 18,451 publisher eTLD+1s we found the tracker was included from a cross-site origin; these were typically sites that were related in some way, e.g. belonging to the same organization. Although these instances cannot circumvent countermeasures where all third-party cookies are blocked, e.g. the built-in protection of Safari, they still defeat blocklists.

Figure 2 displays the percentage of publisher eTLD+1s involved in CNAME-based tracking, both in a same-site or cross-site context, for bins of 10,000 Tranco-ranked websites. As can be seen the use of CNAME-based tracking is heavily biased towards more popular websites. 10% of the top 10,000 Tranco websites refer to a tracker via a CNAME record, while this ration drops to less than 1% for the least popular sites. The ratio of same-site to cross-site CNAME-based tracking is consistently between 50% and 65% for all bins. Because our dataset only contains information about the homepage of websites, and does not include results from Criteo, the reported number should be considered a lower bound.

Using the categorization service by McAfee [55], we determined the most popular categories among CNAME-based tracking publishers, as shown in Figure 3. As a baseline comparison, we also include the distribution of categories in the Tranco top 10k. Because

Table 1. Overview of the analyzed CNAME-based trackers, based on the HTTP Archive dataset from October 2020.

Tracker	Detected # publishers	Est. total # publishers	Pricing (min. /mo)	requests to tracker is blocked by: uBlock Origin Firefox	uBlock Origin Chrome	NextDNS CNAME blocklist
Pardot	5,993	21,759	$1,250	✔*	✔*	✖
Adobe Experience Cloud	2,612	9,029	$5,000†	✔	✔	✔
Act-On Software	1,041	2,533	$900	✔	✔	✖
Oracle Eloqua	304	3,743	$2,000†	✔	✖	✖
Eulerian	253	1,501	?	✔	✖	✔
Webtrekk	101	822	?	✔	✔	✔
Ingenious Technologies	41	-	?	✖	✖	✔
TraceDock	49	69	€49	✖	✖	✔
<intent>	14	124	?	✖	✖	✔
AT Internet	31	74	€355	✖	✖	✔
Criteo	16	13,082	?	✔	✖	✔
Keyade	12	86	?	✔	✖	✔
Wizaly	12	55	$2000†	✖	✖	✔

†: Pricing information does not originate from original source, but as reported in reviews of the product.
*: Requests made to the CNAME subdomain triggered by a third-party analytics script hosted on `pardot.com`; the blocklist prevents the analytics script from loading. If this script was loaded from the CNAME domain, it would not be blocked.

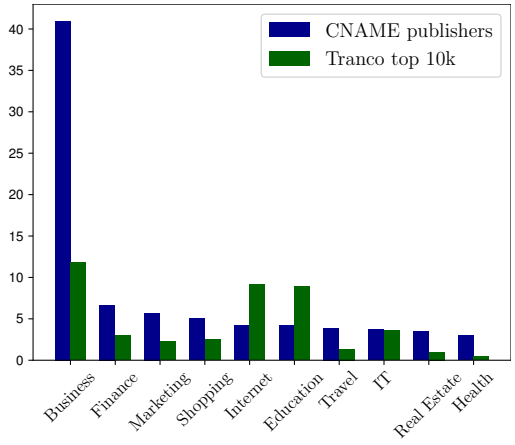


Fig. 3. Most popular categories among CNAME-based tracking publishers.

of the strong financial motives to perform tracking, e.g. marketing and attribution of online purchases, it is not surprising that publishers are mainly financially-focused, with approximately 40% of the publisher's websites being categorized as Business.

Finally, we explored to what extent publishers that employ CNAME-based tracking also include third-party trackers. To this end we analyzed all requests using the EasyPrivacy blocklist [22] to determine the number of trackers that would be blocked by this list. We find that the vast majority of websites that include a CNAME-based tracker (93.97%) also included at least one third-party tracker. On average these sites had 28.43 third-party tracking requests. This indicates that CNAME-based tracking is almost always used in conjunction with other types of tracking. This co-existence may allow CNAME trackers to read first-party cookies set by other trackers via JavaScript. We explore this issue in more detail in Section 6.

5 Historical evolution

In this section we report on various analyses we performed to capture the longitudinal evolution of CNAME-based tracking.

5.1 Uptake in CNAME-based tracking

First, we explore the change in prevalence of CNAME-based tracking over time. To achieve this, we leverage the HTTP Archive dataset, which is collected on a monthly basis and dates back several years. We consider the datasets from December 2018, when the pages from the Chrome User Experience Report started to be used as input for their crawler, until October 2020.

To determine the number of publishers using CNAME tracking over time, we used an iterative approach as shown in Figure 4. Starting from the most recent month (October 2020), we obtained the domain names and associated IP addresses that were used to connect to the CNAME-trackers. Next, we use the HTTP Archive's data from the previous month to determine all IP addresses that (confirmed) CNAME domains resolve to, allowing us to capture changes of IP addresses by trackers. We add these IP addresses to

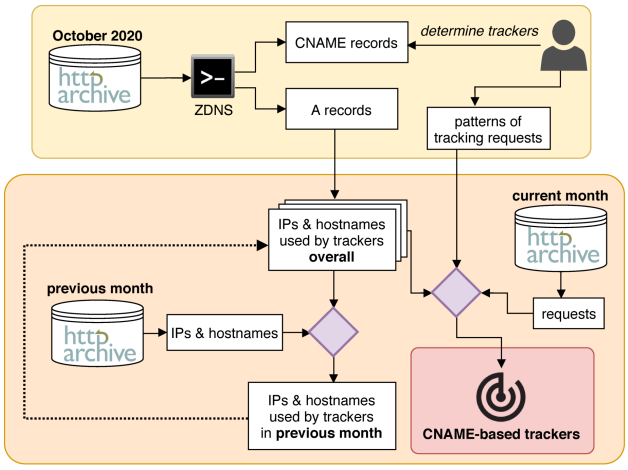


Fig. 4. Overview of the methodology that was used to determine CNAME-based trackers over time.

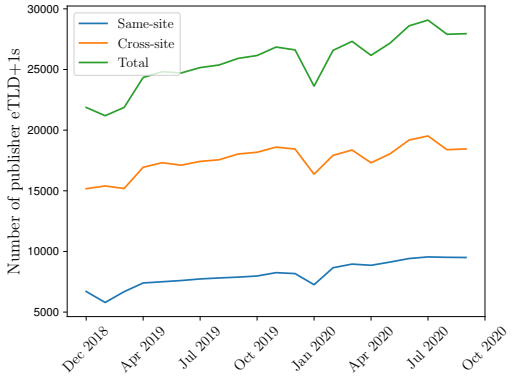


Fig. 5. Number of eTLD+1 domains that include CNAME-based tracking in a same-site and cross-site context.

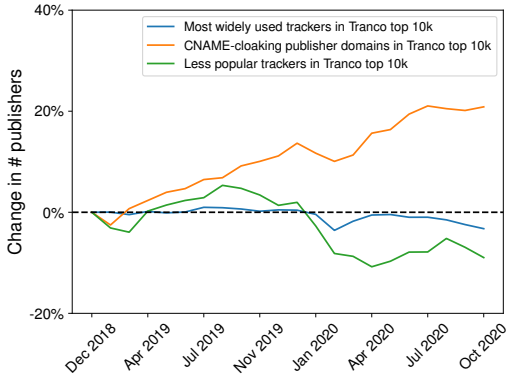


Fig. 6. Relative percentage, based on the state as of December 2018, of the number of publishers of popular and less popular trackers and CNAME-based trackers.

the list of IPs we found in October through a scan with `zdns`. Through this iterative process we obtain a set of IP addresses that were ever used by the different CNAME trackers. Furthermore, whenever we notice that a tracker is using IPs within a certain range for the tracking subdomains, we added the whole range to the set of used IPs (e.g. Eulerian uses IP addresses from the range 109.232.192.0/21 for its tracking subdomains). Relying just on the IP information would likely lead to false positives as the trackers provide various other services which may be hosted on the same IP address, and ownership of IP addresses may change over time. To avoid marking unrelated services as tracking, we rely on our manually-defined *request signatures* (as defined in Section 3.1.1) to filter out any requests that are unrelated to tracking. Using the domain names of the confirmed tracking requests and the set of IP addresses associated with tracking providers, we can apply the same approach again for the previous month. We repeat this process for every month between October 2020 and December 2018.

Figure 5 shows the total number of publisher eTLD+1s using CNAME-based tracking, either in a same-site or cross-site context. The sudden drop in number of cross-site inclusions of CNAME trackers in October 2019 is mainly due to a single tracker (Adobe Experience Cloud). We suspect it is related to changes it made with regard to CCPA regulations (the HTTP Archive crawlers are based in California) [9]. Overall, we find that the number of publisher sites that employ CNAME-based tracking is gradually increasing over time.

To further explore how the adoption of CNAME-based tracking changed over time, we compare it to the evolution of third-party tracking on the web. More specifically, for the ten most popular tracking companies according to WhoTracks.me [30], and fifteen randomly selected less popular trackers with between 50 and 15,000 publishers as of October 2020 (similar to the customer base we observed for the CNAME-based trackers), we determined the number of publishers in the Tranco top 10k list[3], between December 2018 and October 2020. To this end we used the EasyPrivacy blocklist, and only used the rules that match the selected trackers. For the three cases (popular trackers, less popular trackers and CNAME-based trackers) we computed the relative increase or decrease in number of publishers for the Tranco top 10k websites. As the point of reference, we take the first entry of our dataset: December 2018.

3 Generated on 01 July 2020, available at https://tranco-list.eu/list/Z7GG/10000

The relative changes in the number of publishers are shown in Figure 6, and indicate that the customer base of less popular trackers declines whereas popular trackers retain a stable customer base. This is in line with the findings of a study by Cliqz and Ghostery [58]. Our results clearly show that compared to third-party trackers, the CNAME-based trackers are rapidly gaining popularity, with a growth of 21% over the past 22 months (compared to a change of −3% for popular trackers and −8% for less popular trackers).

5.2 Method evaluation

In this section, we evaluate the method we used to detect CNAME-based tracking throughout time for correctness and completeness. For this analysis, we make use of historical DNS data provided by Rapid7 [49]. We try to determine both the web pages that were incorrectly considered to be using CNAME-based tracking, as well as publishers that we might have missed by using our method.

Correctness To assess the correctness of our approach, we looked for subdomains that we considered to be using CNAME tracking for each month of our analysis (December 2018 until October 2020), but that did not have a CNAME record pointing to a tracker in the corresponding month in the historical Rapid7 DNS dataset. We found 81 publishers, 0.46% of the 17,633 publishers that we determined over the whole period, that could potentially be labeled incorrectly. Upon a closer examination, we find that all of these 81 publishers were indeed correctly marked by our method.

These 81 publishers can be divided in three major groups based on the reason that caused the mismatch in the datasets. *First*: Because of the timing difference between the HTTP Archive dataset and the Rapid7 dataset, the tracking domain of 21 publishers did not yet appear in the Rapid7 DNS dataset in the first month of starting to use CNAME-based tracking. *Second*: 15 CNAME-based tracking domains incorrectly configured their DNS records, causing them to send tracking requests to an non-existent or typo domain. For instance, several CNAME records pointed to a `.207.net` domain instead a `.2o7.net` domain. *Third*: We found 42 publisher tracking subdomains that did not have a CNAME record pointing to a known tracking domain. Instead, it pointed to another domain that would still resolve to the same IP address used by the tracker. This occurs when the tracker adds a new tracking domain but the publisher that included it did not yet update their CNAME records. For example, we observe nine publisher subdomains that have a CNAME record pointing to `.ca-eulerian.net`, whereas the currently used domain is `.eulerian.net`. On the other hand, as of October 2020, Adobe Experience Cloud added a new tracking domain, namely `data.adobedc.net`; in the dataset of this month we found 33 tracking subdomains that already started referring to it. As our method is agnostic of the domain name used in the CNAME record of the publisher subdomain (the domain name may change over time), it can detect these instances, in contrast to an approach that is purely based on CNAME records. Finally, for the remaining three publishers, we found that a DNS misconfiguration on the side of the publisher caused the CNAME record to not correctly appear in the Rapid7 dataset. Although tracking requests were sent to the tracking subdomain, these subdomains would not always resolve to the correct IP address, or return different results based on the geographic location of the resolver.

As a result, we conclude that all of the publishers were correctly categorized as using CNAME-based tracking. Moreover, our method is robust against changes in tracking domains used by CNAME trackers.

Completeness We evaluate the completeness of our method by examining domain names that we did not detect as publishers, but that do have a CNAME record to a tracking domain. Our detection method uses an accumulating approach starting from the most recent month's data (October 2020) and detecting CNAME-based tracking for each previous month, based on the current month's data. For this reason, we only consider publisher subdomains that we might have missed in the final month of our analysis (December 2018), where the missed domains error would be most notable. Out of the 20,381 domain names that have a CNAME record in the Rapid7 dataset pointing to a tracking domain, 12,060 (59.2%) were not present in the HTTP Archive dataset. From the remaining domain names, 7,866 (38.6%) were labeled as publishers by us, leaving 455 (2.2%) domain names that we *potentially* missed as a consequence of using our method. After examining the HTTP Archive dataset for these domains, we find that for 195 hostnames the IP address is missing in the dataset. For the remaining 260 domains, we find that the majority (196) does not send any tracking-related request to the tracker, which could indicate that the tracking service is not actively being used. For 41 domain names, we find that the sent requests do not match our request pattern, and further examination shows that these are in fact using another service, unrelated to tracking, from

one of the providers. The remaining 22 domain names were missed as publishers in our method since these resolved to an IP address that was not previously used for CNAME-based tracking.

Our results show that relying solely on DNS data to detect CNAME-based tracking leads to an overestimation of the number of publishers. Furthermore, our method missed only 0.28% of CNAME-based tracking publishers due to irregularities in the set of IP addresses used by CNAME-based tracking providers. A downside of our method is that it cannot automatically account for changes of the request signature used by CNAME trackers throughout time. However, we note that in the analysis spanning 22 months, we did not encounter changes in the request signature for any of the 13 trackers.

Tracker domain ownership Lastly, we verify whether the ownership of the IP-addresses used by the thirteen trackers changes throughout time. To achieve this, we examine PTR records of the IP-addresses used for tracking in December 2018 and check whether the owner company of the resulting domains has changed since then, by using Rapid7's reverse DNS dataset [50] and historical WHOIS data [59]. We find that all of the IP addresses point to domains owned by the corresponding tracker. Furthermore, for 7 trackers, the ownership of the tracking domains has not changed since December 2018. 6 trackers had redacted their WHOIS information due to privacy, out of which 1 was not updated throughout our measurement period. The other 5 have been updated recently and therefore we cannot conclude that their owner has remained the same. We do suspect this is the case however, since all of the domains were owned by the corresponding tracker before the details became redacted.

5.3 Effects on third-party tracking

In order to gather more insight on the reasons as to why websites adopt CNAME-based tracking, we performed an additional experiment. We posed the hypothesis that if the number of third-party trackers employed by websites decreases after they started using the CNAME-based tracking services, this would indicate that the CNAME-based tracking is used as a replacement for third-party tracking. A possible reason for this could be privacy concerns: without any anti-tracking measures, third-party tracking allows the tracker to build profiles of users by following them on different sites, whereas CNAME-based tracking only tracks users on a specific site (assuming that the tracker acts in good faith). Conversely, if the number of third-party trackers remains stable or even increases, this would indicate that CNAME-based tracking is used in conjunction with third-party tracking, e.g. to still obtain information on users that employ anti-tracking measures.

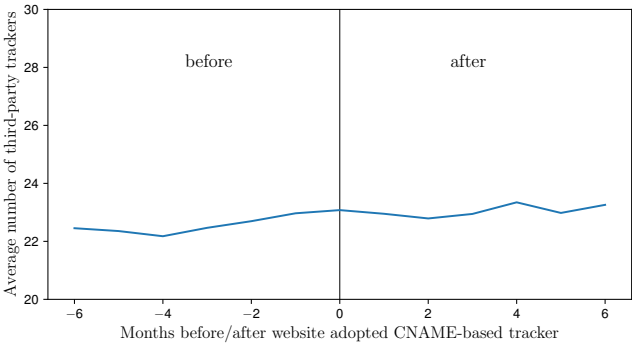


Fig. 7. Number of third-party trackers adopted by publishers in the six months before and after they adopted a CNAME-based tracker.

To measure the evolution of the number of third-party trackers on publisher sites that recently adopted CNAME-based tracking, we again use the measurements ranging between December 2018 and October 2020 from the HTTP Archive dataset. We consider a publisher website including a CNAME tracker to be *new* if for six consecutive months it did not refer to this tracker through a CNAME record on a subdomain, and then for the following six months always included a resource from this tracker. In total we found 1,129 publishers started using CNAME tracking in our analysis period. For these publishers, we determined the number of third-party trackers based on the EasyPrivacy blocklist for the six months before and after the time the publishers adopted CNAME-based tracking. The average number of third-party trackers over this time period is shown in Figure 7. We find that the adoption of CNAME-based tracking services does not significantly affect the third-party trackers that are in use, suggesting that CNAME-based trackers are used to complement the information obtained from other trackers.

6 Implications of first-party inclusion

In this section we investigate how CNAME-based tracking can expand a website's attack surface.

6.1 Transport security

When visiting a website that employs CNAME-based tracking, various types of requests are made to the tracker-controlled subdomain. We find that most commonly, the web page makes a request to report analytics data, typically via an asynchronous request or a pixel. Additionally, we find that in most cases the tracking script is also included from the CNAME subdomain. To ensure that a man-in-the-middle attacker cannot modify these scripts in transit, a secure HTTPS connection should be used. Based on the HTTP Archive dataset from July 2020, we find that the vast majority (92.18%) of websites that use CNAME-based tracking support TLS, and in almost all cases the tracker requests are sent over secure connections. Nevertheless, we did identify 19 websites where active content, i.e. HTML or JavaScript, was requested from the tracker over an insecure connection. Although most modern browsers block these requests due to mixed content policies, users with outdated browsers would still be susceptible to man-in-the-middle attacks.

On 72 websites we found that analytics requests to CNAME-based trackers were sent unencrypted over HTTP while the web page itself was loaded over HTTPS. In this case, the request is not blocked but instead the browser warns the user that the connection is insecure. Because this is a same-site request, cookies that are scoped to the eTLD+1 domain, and that do not contain the `Secure` attribute, are attached to this request. Consequently these potentially identifying cookies can be intercepted by network eavesdroppers. Furthermore an attacker could exploit unencrypted HTTP responses. Specifically, the adversary could inject arbitrary cookies in `Set-Cookie` headers to launch a session-fixation attack [31, 51]. In the remainder of this section, we explore the privacy and security threats associated with including the tracker as first party in more detail.

6.2 Tracker vulnerabilities: case studies

To further explore how the security of websites and their visitors is affected by including a CNAME-based tracker, we performed a limited security evaluation of the trackers that are included on publisher websites. For up to maximum 30 minutes per tracker, we analyzed the requests and responses to/from the CNAME subdomain for client-side web vulnerabilities. In most cases, we found that only a single request was made, and an empty response was returned. Despite the time-limited nature of our analysis, we did identify vulnerabilities in two different trackers that affect all publishers that include them. We reported the vulnerabilities to the affected trackers and actively worked with them to mitigate the issues. Unfortunately, in one instance the tracker did not respond to repeated attempts to report the vulnerability, leaving hundreds of websites exposed. We still hope to be able to contact this vendor through one of their customers.

6.2.1 Vulnerability 1: session fixation

The first vulnerability is caused by the tracker's functionality to extend the lifetime of first-party advertising and analytics cookies, such as Facebook's `_fbp` cookie or the `_ga` cookie by Google Analytics. Because these cookies are set by a cross-site script through the `document.cookie` API, Safari's ITP limits their lifespan to seven days [1]. To overcome these limits, the tracker provides a specific endpoint on the CNAME subdomain that accepts a POST request with a JSON payload containing the cookie names and values whose lifetime should be extended. In the response, the tracker's server includes several `Set-Cookie` headers containing the tracking cookies. Consequently, these cookies are no longer set via the DOM API and would have an extended lifetime under Safari's ITP policies for cookies. We note that this circumvention is disabled as of late 2020, thanks to Safari's recent ITP update targeting CNAME-based trackers [2]. This update caps the lifetime of HTTP cookies from CNAME trackers to seven days, which matches the lifetime of cookies set via JavaScript.

We found that the tracker endpoint did not adequately validate the origin of the requests, nor the cookie names and values. This makes it possible to launch a session-fixation attack through the functionality provided by the tracker, which is enabled by default on all the websites that include the tracker in a first-party context. For example, on a shopping site the attacker could create their own profile and capture the cookies associated with their session. Subsequently, the attacker could abuse the session-fixation vulnerability to force the victim to set the same session cookie as the one from the attacker, resulting in the victim being logged in as the attacker. If at some point the victim would try to make a purchase and enter their credit card information, this would be done in the attacker's profile. Finally, the attacker can make purchases using

the victim's credit card, or possibly even extract the credit card information.

The impact of this vulnerability highlights the increased threat surface caused by using the CNAME-based tracking scheme. If a third-party tracker that was included in a cross-site context would have the same vulnerability, the consequences would be negligible. The extent of the vulnerability would be limited to the setting of an arbitrary cookie on a tracking domain (as opposed to the first-party visited website) which would have no effect on the user. However, because in the CNAME-tracking scheme the tracking domain is a subdomain of the website, cookies set with a `Domain` attribute of the eTLD+1 domain (this was the default in the detected vulnerability), will be attached to all requests of this website and all its subdomains. As a result, the vulnerability does not only affect the tracker, but introduces a vulnerability to all the websites that include it.

6.2.2 Vulnerability 2: cross-site scripting

The second vulnerability that we identified affects publishers that include a different tracker, and likewise it is directly related to a tracker-specific functionality. In this case, the tracker offers a method to associate a user's email address with their fingerprint (based on IP address and browser properties such as the User-Agent string). This email address is later reflected in a dynamically generated script that is executed on every page load, allowing the website to retrieve it again, even if the user would clear their cookies. However, because the value of the email address is not properly sanitized, it is possible to include an arbitrary JavaScript payload that will be executed on every page that includes the tracking script. Interestingly, because the email address is associated with the user's browser and IP fingerprint, we found that the payload will also be executed in a private browsing mode or on different browser profiles. We tested this vulnerability on several publisher websites, and found that all could be exploited in the same way. As such, the issue introduced by the tracking provider caused a persistent XSS vulnerability in several hundreds of websites.

6.3 Sensitive information leaked to CNAME-based trackers

CNAME-based trackers operate on a subdomain of publisher websites. It is therefore possible that cookies sent to the tracker may contain sensitive information, such as personal information (name, email, location) and authentication cookies, assuming these sensitive cookies are scoped to the eTLD+1 domain of the visited website (i.e. `Domain=.example.org`). Furthermore, it is possible that websites explicitly share personal information with the CNAME-based trackers in order to build a better profile on their users.

To analyze the type of information that is sent to trackers and to assess the frequency of occurrence, we performed a manual experiment on a random subset of publishers. Based on data from a preliminary crawl of 20 pages per website, we selected up to ten publisher websites per tracker that had at least one HTML form element with a password field. We limited the number of websites in function of the manual effort required to manually register, login, interact with it, and thoroughly analyze the requests that were sent. We looked for authentication cookies (determined by verifying that these were essential to remain logged on to the website), and personal information such as the name and email that was provided during the registration process.

Out of the 103 considered websites, we were able to successfully register and log in on 50 of them. In total, we found that on 13 of these websites sensitive information leaked to a CNAME tracker. The leaked information included the user's full name (on 1 website), location (on 2 websites), email address (on 4 websites, either in plain-text or hashed), and the authentication cookie (on 10 websites). We note that such leaks are the result of including the trackers in a first-party context. Our limited study indicates that the CNAME tracking scheme negatively impacts users' security (authentication cookie leaks) and privacy (personal data leaks).

6.4 Cookie leaks to CNAME-based trackers

Next we perform an automated analysis to investigate cookies that are inadvertently sent to CNAME trackers. In June 2020, we conducted an automated crawl of 8,807 websites that we, at that time, identified as using CNAME-based tracking following the methodology outlined in Section 4.2. In this crawl, we searched for cookies sent to the CNAME subdomain while excluding the cookies set by the CNAME tracker itself (either through its subdomain or its third-party domains).

The crawler We built our crawler by modifying the DuckDuckGo Tracker Radar Collector [21], a Puppeteer-based crawler that uses the Chrome Dev-

Tools Protocol (CDP). We extended the crawler by adding capabilities to capture HTTP request cookies, POST data, and document.cookie assignments. The Tracker Radar Collector uses the Chrome DevTools Protocol to capture the access by scripts to Web API methods and browser properties that may be relevant to browser fingerprinting and tracking. We used this JavaScript instrumentation to identify scripts that set cookies using JavaScript.

For each website, we loaded the homepage using a fresh profile. We instructed the crawler to wait ten second on each website, and then reload the page. This allowed us to capture the leaks of cookies that were set after the request to the CNAME-based tracker domain. We also collected HTTP headers, POST bodies, JavaScript calls, and cookies from the resulting profile. When crawling, we used a Safari User-Agent string, as we found at least one CNAME-based tracker (Criteo) employing first-party tracking for Safari users only.

Data analysis To identify the cookie leaks, we first built the list of cookies sent to the CNAME subdomain. From the resulting list, we excluded session cookies, short cookies (less than 10 characters), and cookies that contain values that occur on multiple visits (to exclude cookies that are not uniquely identifying). To determine the latter, we first built a mapping between the distinct cookie values and the number of sites they occur on.

Next, we identified the *setter* of the cookies. First, we searched the cookie name and value in `Set-Cookie` headers in HTTP responses. When the cookie in question was sent in the corresponding request, we excluded its response from the analysis. For JavaScript cookies, we searched for the name-value pair in assignments to `document.cookie` using the JavaScript instrumentation data. We then used the JavaScript stack trace of the assignment to determine the origin of the script. After determining the setter, we excluded cookies set by the CNAME-based tracker itself.

Leaks in HTTP Cookie headers

We identified one or more cookie leaks on 7,377 sites (95%) out of the 7,797 sites where we could identify the presence of at least one CNAME-based tracker. Table 2 shows the five origins with most cookies leaked to CNAME-based trackers. The overwhelming majority of cookie leaks (31K/35K) are due to third-party analytics scripts setting cookies on the first-party domain.

The leakage of first-party cookies containing unique IDs may not reveal any additional information to CNAME-based trackers, since these trackers may already have an ID for the users in their own cookies. However, cookies containing other information such as ad campaign information, emails, authentication cookies may also leak to the CNAME-based trackers (as shown in Section 6.3). Moreover, our analysis found that on 4,006 sites, a cookie set by a third-party domain is sent to the CNAME-based tracker's subdomain. 3,898 of these sites are due to Pardot, which sets the same cookie on its first-party subdomain and its third-party domain. To set the same cookie on both domains, Pardot sends its unique ID in a URL parameter called `visitor_id` to its first-party subdomain.

Table 2. Five origins with most leaked cookies to CNAME-based trackers. The right column indicates the number of distinct sites cookies we observed one or more cookie leaks set by the scripts from these origins.

Cookie origin	Purpose	Num. of distinct sites
www.google-analytics.com	Analytics	5,970
connect.facebook.net	FB Pixel	3,287
www.googletagmanager.com	Tag management	2,376
bat.bing.com	Advertising	1,182
assets.adobedtm.com	Tag management	887

Leaks in POST request bodies Cookie leaks discussed above may happen inadvertently, without the knowledge or the cooperation of the CNAME trackers. However, we identified two other types of cookie leaks that involve more active participation by the CNAME trackers. First, we studied cookie values sent in POST request bodies, again excluding the cookies set by the CNAME tracker itself, session cookies, and cookies that occur on multiple sites, as described above. We found that 166 cookies (on 94 distinct sites) set by another party were sent to a CNAME tracker's subdomain in a POST request body. The majority of these cases were due to TraceDock (46 sites) and Adobe Experience Cloud (30 sites), while Otto Group and Webtrekk caused these cookie leaks on 11 and 7 sites respectively.

We used the request "initiators" field to identify the *senders* of the requests. The "initiators" field contains the set of script addresses that triggered an HTTP request, derived from JavaScript stack traces. In 78 of the 166 instances, the CNAME subdomain or the tracker's third-party domains were among the initiators of the POST request. In the remaining cases, the CNAME tracker's script was served on a different domain (e.g. Adobe Experience Cloud, assets.adobedtm.com), a different subdomain that also belongs to the CNAME tracker (e.g. Otto Group uses tp.xyz.com subdomain for its scripts and te.xyz.com for the endpoint), or the re-

quest was triggered by a tag manager script, or a combined script that contains the CNAME tracker's script.

These findings suggest that certain CNAME tracker scripts actively read and exfiltrate cookies that belong to other parties. Although the content of the cookies may not always reveal additional information, our manual analysis presented above revealed sensitive information such as email addresses, authentication cookies and other personal information is leaking to the CNAME trackers.

Leaks in request URLs Next we investigate the cookies sent to CNAME tracker subdomains in the request URLs. To detect such leaks we searched for cookies in the request URLs (and URL-decoded URLs) excluding the scheme and the hostname. We excluded the same set of cookies as the previous two analyses – cookies set by CNAME tracker itself, short cookies, session cookies and cookies with non-identifying values.

We found 1,899 cookie leaks in request URLs to CNAME subdomains on 1,295 distinct sites. 1,566 of the cookies were sent to Adobe Experience Cloud's subdomain, while Pardot's and Eularian's subdomains received 130 and 101 cookies, respectively. In addition, in 4,121 cases (4,084 sites), a cookie set by Pardot's third-party domain was sent to its CNAME subdomain, confirming the finding above that Pardot syncs cookies between its third-party domain and its CNAME subdomain. Overall, in 378 cases the leaked cookie was set by a third-party domain, indicating that cookies were synced or simply exchanged between the domains.

Our automated analysis of cookie leaks, in combination with the deeper manual analysis presented above indicates that passive and active collection of cookies by the CNAME trackers is highly prevalent and have severe privacy and security implications including the collection of email addresses, unique identifiers and authentication cookies. Further, our results show that certain CNAME-based trackers use third-party cookies for cross-site tracking and at times receive cookies set by other third-party domains, allowing them to track users across websites.

7 Discussion

While CNAME-based tracking exists for several years, our study shows that recently it is gaining substantial popularity, especially on frequently-visited websites. In this section we explore the current countermeasures against this form of tracking, and discuss their effectiveness and potential circumvention techniques that trackers may use in the future.

Countermeasures In response to a report that a tracker was using CNAMEs to circumvent privacy blocklists[4], uBlock Origin released an update for its Firefox version that thwarts CNAME cloaking [27]. The extension blocks requests to CNAME trackers by resolving the domain names using the `browser.dns.resolve` API to obtain the last CNAME record (if any) before each request is sent. Subsequently, the extension checks whether the domain name matches any of the rules in its blocklists, and blocks requests with matching domains while adding the outcome to a local cache. Although uBlock Origin has also a version for Chromium-based browsers, the same defense cannot be applied because extensions on Chromium do not have access to a similar DNS API.

As we explain in Section 4, uBlock Origin for Chrome, which does not have a defense for CNAME-based tracking, still manages to block several trackers. This is because the requests to the trackers matched an entry of the blocklist with a URL pattern that did not consider the hostname. Unfortunately, it is fairly straightforward for the tracker to circumvent such a fixed rule-based measure, e.g. by randomizing the path of the tracking script and analytics endpoint, as is evidenced by the various trackers that could only be blocked by uBlock Origin on Firefox. An alternative strategy for browser extensions that do not have access to a DNS API could be to analyze the behavior or artifacts of tracking scripts. However, the tracker's code could be dynamic and include many variations, making detection arduous and performance-intensive.

Thanks to the increasing attention to CNAME-based tracking, Safari and Brave recently followed uBlock Origin's suit, and implemented countermeasures against CNAME-based tracking. Safari limited the expiry of cookies from CNAME trackers to seven days, which is the same limit they use for all cookies set by JavaScript [2]. Brave, on the other hand, started recursively checking for CNAME records of the network requests against their blocklists [3]. Mozilla is working on implementing a similar defense in Firefox [4].

Other tracking countermeasures include DNS sinkholes that return a false IP address, (e.g. 127.0.0.1) when the domain name matches an entry from the blocklist. As this type of countermeasure work at the DNS level, it considers all the intermediary resolutions

4 https://github.com/uBlockOrigin/uBlock-issues/issues/780

to CNAME records, and effectively blocks the domains that match a blocklist. Examples of DNS-based tools that adopted defenses against CNAME cloaking include NextDNS [47], AdGuard [8], and Pi-hole [56].

Circumvention Both anti-tracking solutions, i.e. browser extensions and DNS resolvers, rely on blocklists, and can thus only block trackers whose domain names are on the list. Updating CNAME records using randomized domain names may bypass these blocklists. However, this requires publishers to frequently update their CNAME records, which may be impractical for many websites. Another circumvention option is to directly refer to the IP address of the tracker through an A record instead of a CNAME record. We found the pool of IP addresses used by CNAME-based trackers to be relatively stable over time, and in fact found that several (35) publishers already use this method. At the time of this writing, using IP addresses (and A records) circumvents blocklists, which do not use IP addresses to identify trackers.

While IP addresses can be added to blocklists, changing IP addresses as soon as they are added to blocklists would be practically infeasible, as it requires all publishers to update their DNS records. Nevertheless, a tracker could request their publishers to delegate authority for a specific subdomain/zone to the tracker by setting an NS record that points to the tracker. As such, the tracker could dynamically generate A record responses for any domain name within the delegated zone, and thus periodically change them to avoid being added to blocklists. For anti-tracking mechanisms to detect this circumvention technique, this would require obtaining the NS records to determine whether they point to a tracker. Although it may be feasible to obtain these records, it may introduce a significant overhead for the browser extensions and DNS-based anti-tracking mechanisms.

In general, as long as the anti-tracking mechanism can detect the indirection to the third-party tracker, it is possible to detect and block requests to the tracker, albeit at a certain performance cost. Trackers could try to further camouflage their involvement in serving the tracking scripts and collecting the analytics information. For instance, they could request the publishers that include tracking scripts to create a reverse proxy for a specific path that points to the tracker, which could be as easy as adding a few lines in the web server configuration, or adjusting the settings of the CDN provider. In such a situation, the tracking-related requests would appear, from a user's perspective, to be sent to the visited website, both in terms of domain name as well as IP address. Thus, current tracking defenses would not be able to detect or block such requests. As the perpetual battle between anti-tracking mechanisms and trackers continues, as evidenced by the increasing popularity of CNAME-based tracking, we believe that further empirical research on novel circumvention techniques is warranted.

Limitations As stated in Section 5, the method we use to detect CNAME-based tracking in historical data cannot account for changes in the request signature used by trackers. In practise, these signatures remained the same during our measurement period. Furthermore, part of the experiments we conducted in Section 6 required substantial manual analysis, making it infeasible to perform on a larger set of websites.

8 Related work

In 2009, Krishnamurthy and Wills provided one of the first longitudinal analyses of user information flows to third-party sites (called *aggregators*) [32]. The authors also observed a trend of serving third-party tracking content from first-party contexts, pointing out the challenges for countermeasures based on blocklists. Meyer and Mitchell studied the technology and policy aspects of third-party tracking [38]. Englehardt and Narayanan [24] measured tracking on Alexa top million websites using OpenWPM and discovered new fingerprinting techniques such as AudioContext API-based fingerprinting.

The CNAME tracking scheme was anecdotally mentioned by Bau et al. in 2013 [13], but the authors did not focus on the technique in their study. To our knowledge, the first systematic analysis of the CNAME scheme used to embed third-party trackers in first-party content is the work of Olejnik and Casteluccia [43], in which they identified this special arrangement as part of the real-time bidding setup. The authors also reported leaks of first-party cookies to such third parties. In our paper, we extensively expand such analyses. Although cookies were most commonly used for cross-site tracking, more advanced mechanisms have been used in practice and studied by researchers. Browser fingerprinting [23], where traits of the host [62], system, browser and graphics stack [39] are extracted to identify the user is one of the stateless tracking vectors that does not need cookies to operate. Fingerprinting on the web was measured at scale by Acar et al. [6, 7], Nikiforakis et al.[42], and Englehardt and Narayanan [24]. As demonstrated first by Samy Kamkar, combining multiple tracking vectors may

enable evercookies (or supercookies), that can be used to regenerate removed identifiers [29]. Over the years, many information exfiltration or tracking vectors have been studied, including Cache Etag HTTP header [11], WebSockets [12], ultrasound beacons [37], and fingerprinting sensors calibrations on mobile devices [64].

Similar to these studies we measure the prevalence of a tracking mechanism that tries to circumvent existing countermeasures. However our work uses novel methods to identify CNAME-based trackers in historical crawl data, allowing us to perform a longitudinal measurement.

In a concurrent study, Dao et al. explored the ecosystem of CNAME-based trackers [18]. Based on a crawl of the Alexa top 300k, they find 1,762 CNAME-based tracking domains as of January 2020, which are detected by matching the CNAME domain with EasyPrivacy. In our work, we detected 9,273 sites that leverage CNAME-based tracking in a same-site context and an additional 19,226 websites that use it in a cross-site context. We rely on an approach that combines historical DNS records (A records) with manually constructed fingerprints. The latter is used to filter out any potential false positives that may be caused by changes in the IP space ownership, or because the CNAME- or A-records may be used to other services of the same provider unrelated to tracking. Based on the evaluation of our method in Section 5.2, we find that it is important to use request-specific information to prevent incorrectly marking domains as using CNAME-based tracking. Furthermore, relying on filter lists, and in particular on the eTLD+1 domains that are listed, could result in the inclusion of non-tracking domains. For instance, sp-prod.net is the second most popular tracker considered by Dao et al., but was excluded in our work as it is part of a "Consent Management Platform" that captures cookie consent for compliance with GDPR [52]. Additionally, filter lists may be incomplete, resulting in trackers being missed: for example, Pardot, the tracker we find to be most widely used, was not detected in prior work. Consequently, relying on filter lists also prevents the detection of new trackers, this limitation is not applicable to our method.

Dao et al. also perform an analysis of the historical evolution of CNAME-based tracking, based on four datasets of the Alexa top 100k websites collected between January 2016 and January 2020. As the used OpenWPM datasets do not include DNS records, the researchers rely on a historical forward DNS dataset provided by Rapid7 [49], which does not cover all domains over time. By using the HTTP Archive dataset, which includes the IP address that was used, we were able to perform a more granular analysis, showing a more accurate growth pattern. We also show that this growth is rapidly increasing, significantly outperforming third-party trackers with a comparable customer base. Finally, to the best of our knowledge, we are the first to perform an analysis of the privacy and security threats associated with the CNAME-based tracking scheme.

9 Conclusion

Our research shed light on the emerging ecosystem of CNAME-based tracking, a tracking scheme that takes advantage of a DNS-based cloaking technique to evade tracking countermeasures. Using HTTP Archive data and a novel method, we performed a longitudinal analysis of the CNAME-based tracking ecosystem using crawl data of 5.6M web pages. Our findings show that unlike other trackers with similar prevalence, CNAME-based trackers are becoming increasingly popular, and are mostly used to supplement "typical" third-party tracking services. We evaluated the privacy and security threats that are caused by including CNAME trackers in a same-site context. Through manual analysis we found that sensitive information such as email addresses and authentication cookies leak to CNAME trackers on sites where users can create accounts. Furthermore, we performed an automated analysis of cookie leaks to CNAME trackers and found that cookies set by other parties leak to CNAME trackers on 95% of the websites that we studied. Finally we identified two major web security vulnerabilities that CNAME trackers caused. We disclosed the vulnerabilities to the respective parties and have worked with them to mitigate the issues. We hope that our research helps with addressing the security and privacy issues that we highlighted, and inform development of countermeasures and policy making with regard to online privacy and tracking.

Acknowledgements

This research is partially funded by the Research Fund KU Leuven, and by the Flemish Research Programme Cybersecurity with reference number VR20192203. We would like to thank Steve Englehardt and the reviewers for their constructive comments. Gunes Acar holds a Postdoctoral fellowship of the Research Foundation Flanders (FWO).

References

[1] 2019. Intelligent Tracking Prevention 2.1. https://webkit.org/blog/8613/intelligent-tracking-prevention-2-1 [Online; accessed 24. Feb. 2021].

[2] 2020. CNAME Cloaking and Bounce Tracking Defense. https://webkit.org/blog/11338/cname-cloaking-and-bounce-tracking-defense [Online; accessed 23. Feb. 2021].

[3] 2020. What's Brave Done For My Privacy Lately? Episode #6: Fighting CNAME Trickery | Brave Browser. https://brave.com/privacy-updates-6 [Online; accessed 23. Feb. 2021].

[4] 2021. 1598969 - Block trackers using CNAME Cloaking (1st-party tracker blocking). https://bugzilla.mozilla.org/show_bug.cgi?id=1598969 [Online; accessed 24. Feb. 2021].

[5] Lawrence Abrams. 2019. uBlock Origin Now Blocks Sneaky First-Party Trackers in Firefox. https://www.bleepingcomputer.com/news/security/ublock-origin-now-blocks-sneaky-first-party-trackers-in-firefox/.

[6] 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 (CCS '14)*. ACM, New York, NY, USA, 674–689. https://doi.org/10.1145/2660267.2660347

[7] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. 2013. FPDetective: dusting the web for fingerprinters. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 1129–1140.

[8] AdGuard. 2019. Disguised trackers threat and how we will address it. https://adguard.com/en/blog/disguised-trackers.html.

[9] Adobe Experience Cloud. 2019. Adobe Experience Cloud Release Notes - October 2019. https://docs.adobe.com/content/help/en/release-notes/experience-cloud/previous/2019/10102019.html.

[10] HTTP Archive. 2020. State Of The Web Report. https://httparchive.org/.

[11] Mika D Ayenson, Dietrich James Wambach, Ashkan Soltani, Nathan Good, and Chris Jay Hoofnagle. 2011. Flash cookies and privacy II: Now with HTML5 and ETag respawning. *Available at SSRN 1898390* (2011).

[12] Muhammad Ahmad Bashir, Sajjad Arshad, Engin Kirda, William Robertson, and Christo Wilson. 2018. How tracking companies circumvented ad blockers using websockets. In *Proceedings of the Internet Measurement Conference 2018*. ACM, 471–477.

[13] Jason Bau, Jonathan Mayer, Hristo Paskov, and John C Mitchell. 2013. A promising direction for web tracking countermeasures. *Proceedings of W2SP* (2013).

[14] Omar Benguerah. 2017. Setting first-party cookies by redirection. US Patent 9,723,051.

[15] Google Chrome. 2020. Cookies default to SameSite=Lax. https://www.chromestatus.com/feature/5088147346030592.

[16] Cloudflare. 2020. Understanding the Cloudflare Cookies. https://support.cloudflare.com/hc/en-us/articles/200170156-Understanding-the-Cloudflare-Cookies.

[17] Romain Cointepas. 2019. CNAME Cloaking, the dangerous disguise of third-party trackers. https://medium.com/nextdns/cname-cloaking-the-dangerous-disguise-of-third-party-trackers-195205dc522a.

[18] Ha Dao, Johan Mazel, and Kensuke Fukuda. 2020. Characterizing CNAME Cloaking-Based Tracking on the Web. In *Proceedings of IFIP/IEEE Traffic Measurement Analysis Conference (TMA)*. 9 pages.

[19] Sanorita Dey, Nirupam Roy, Wenyuan Xu, Romit Roy Choudhury, and Srihari Nelakuditi. 2014. AccelPrint: Imperfections of Accelerometers Make Smartphones Trackable.. In *NDSS*. Citeseer.

[20] Disconnect. 2020. Privacy Solutions. https://disconnect.me/.

[21] duckduckgo. 2020. tracker-radar-collector. https://github.com/duckduckgo/tracker-radar-collector [Online; accessed 10. Jun. 2020].

[22] EasyPrivacy. 2020. Filter List That Completely Removes All Forms Of Tracking From The Internet. https://easylist.to/index.html.

[23] Peter Eckersley. 2010. How unique is your web browser?. In *International Symposium on Privacy Enhancing Technologies Symposium*. Springer, 1–18.

[24] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 1388–1401.

[25] 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*. 289–299.

[26] Brent Fulgham. 2018. Protecting Against HSTS Abuse. https://webkit.org/blog/8146/protecting-against-hsts-abuse.

[27] Raymond Hill. 2020. uBlock Origin - 1.25.0. https://github.com/gorhill/uBlock/releases/tag/1.25.0.

[28] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2020. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. *arXiv preprint arXiv:2008.04480* (2020).

[29] Samy Kamkar. 2010. Evercookie-virtually irrevocable persistent cookies. *His Blog* 9 (2010).

[30] Arjaldo Karaj, Sam Macbeth, Rémi Berson, and Josep M Pujol. 2018. WhoTracks. Me: Shedding light on the opaque world of online tracking. *arXiv preprint arXiv:1804.08959* (2018).

[31] Mitja Kolšek. 2002. Session fixation vulnerability in web-based applications. *Acros Security* 7 (2002).

[32] Balachander Krishnamurthy and Craig Wills. 2009. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of the 18th international conference on World wide web*. ACM, 541–550.

[33] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*

(NDSS 2019). https://doi.org/10.14722/ndss.2019.23386

[34] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 2016. Internet jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*.

[35] Scott Low and Joe Martin. 2020. Tracking Prevention in Microsoft Edge (Chromium). https://docs.microsoft.com/en-us/microsoft-edge/web-platform/tracking-prevention.

[36] Andrea Marchesini. 2019. Enable sameSite=lax by default on Nightly. https://bugzilla.mozilla.org/show_bug.cgi?id=1604212.

[37] Vasilios Mavroudis, Shuang Hao, Yanick Fratantonio, Federico Maggi, Christopher Kruegel, and Giovanni Vigna. 2017. On the privacy and security of the ultrasound ecosystem. *Proceedings on Privacy Enhancing Technologies* 2017, 2 (2017), 95–112.

[38] Jonathan R Mayer and John C Mitchell. 2012. Third-party web tracking: Policy and technology. In *2012 IEEE Symposium on Security and Privacy*. IEEE, 413–427.

[39] Keaton Mowery and Hovav Shacham. 2012. Pixel perfect: Fingerprinting canvas in HTML5. *Proceedings of W2SP* (2012), 1–12.

[40] NextDNS. 2020. CNAME Cloaking Blocklist. https://github.com/nextdns/cname-cloaking-blocklist.

[41] NextDNS. 2020. NextDNS CNAME Cloaking Blocklist. https://github.com/nextdns/cname-cloaking-blocklist.

[42] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. 2013. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *2013 IEEE Symposium on Security and Privacy*. IEEE, 541–555.

[43] Lukasz Olejnik and Claude Castelluccia. 2014. Analysis of openx-publishers cooperation. In *In 7th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2014)*.

[44] Lukasz Olejnik, Tran Minh-Dung, and Claude Castelluccia. 2014. Selling off privacy at auction. In *In Proceedings of the 2014 Symposium on Network and Distributed System Security*.

[45] Mike O'Neill. 2015. Discovered In The Wild: A New Method Bypassing Safari's Third-Party Cookie Blocking. https://baycloud.com/blog/PostDetail?slug=discovered-in-the-wild-a-new-method-bypassing-safaris-third-party-cookie-blocking.

[46] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. 2019. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *The World Wide Web Conference*. 1432–1442.

[47] Olivier Poitrey. 2019. NextDNS first to support blocking of ALL third-party trackers disguised as first-party. https://medium.com/nextdns/nextdns-added-cname-uncloaking-support-becomes-the-first-cross-platform-solution-to-the-problem-e3f437f84342.

[48] Chrome DevTools Protocol. 2020. Instrument, Inspect, Debug And Profile Chromium. https://chromedevtools.github.io/devtools-protocol/.

[49] Rapid7. 2020. DNS 'ANY', 'A', 'AAAA', 'TXT', 'MX', and 'CNAME' responses for known forward DNS names. https://opendata.rapid7.com/sonar.fdns_v2/.

[50] Rapid7. 2020. DNS IPv4 PTR responses. https://opendata.rapid7.com/sonar.rdns_v2/.

[51] Michael Schrank, Bastian Braun, Martin Johns, and Joachim Posegga. 2010. Session fixation–the forgotten vulnerability? *Sicherheit 2010. Sicherheit, Schutz und Zuverlässigkeit* (2010).

[52] SourcePoint. 2020. Consent Management Platform. https://help.sourcepoint.com/en/collections/1255107-consent-management-platform.

[53] Alan Toner. 2017. Safari in Arms Race Against Trackers - Criteo Feels the Heat. https://www.eff.org/deeplinks/2017/12/arms-race-against-trackers-safari-leads-criteo-30.

[54] Security Trails. 2020. Robust APIs & Data Services for Security Teams. https://securitytrails.com/.

[55] Pelayo Vallina, Victor Le Pochat, Álvaro Feal, Marius Paraschiv, Julien Gamba, Tim Burke, Oliver Hohlfeld, Juan Tapiador, and Narseo Vallina-Rodriguez. 2020. Mis-shapes, Mistakes, Misfits: An Analysis of Domain Classification Services. In *Proceedings of the 2020 Internet Measurement Conference (IMC 2020)*. https://doi.org/10.1145/3419394.3423660

[56] Adam Warner. 2020. Pi-hole v5.0 is here! https://pi-hole.net/2020/05/10/pi-hole-v5-0-is-here/.

[57] Mike West. 2020. Incrementally Better Cookies. https://tools.ietf.org/html/draft-west-cookie-incrementalism-01.

[58] WhoTracks.me. 2018. GDPR - What happened? https://whotracks.me/blog/gdpr-what-happened.html.

[59] Whoxy. 2020. WHOIS Lookup API for Domain Names. https://www.whoxy.com/.

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

[61] Marissa Wood. 2019. Today's Firefox Blocks Third-Party Tracking Cookies and Cryptomining by Default. https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookies-and-cryptomining-by-default/.

[62] Ting-Fang Yen, Yinglian Xie, Fang Yu, Roger Peng Yu, and Martin Abadi. 2012. Host Fingerprinting and Tracking on the Web: Privacy and Security Implications.. In *NDSS*, Vol. 62. 66.

[63] ZDNS. 2020. Command-line Utility That Provides High-speed DNS Lookups. https://github.com/zmap/zdns.

[64] Jiexin Zhang, Alastair R. Beresford, and Ian Sheret. 2019. SensorID: Sensor Calibration Fingerprinting for Smartphones. In *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*. IEEE.

A Assisted detection

First-party subdomains referring to third-parties are by no means exclusive to CNAME-based tracking: services such as CDNs rely on a similar setup. Many websites hosting various services utilize CNAMEs to connect website domains to third-party hosts. Furthermore, a variety of different kinds of services provide

third-party content in a first-party context by using CNAME records. Examples include Consent Management Providers or domain parking services and traffic management platforms.

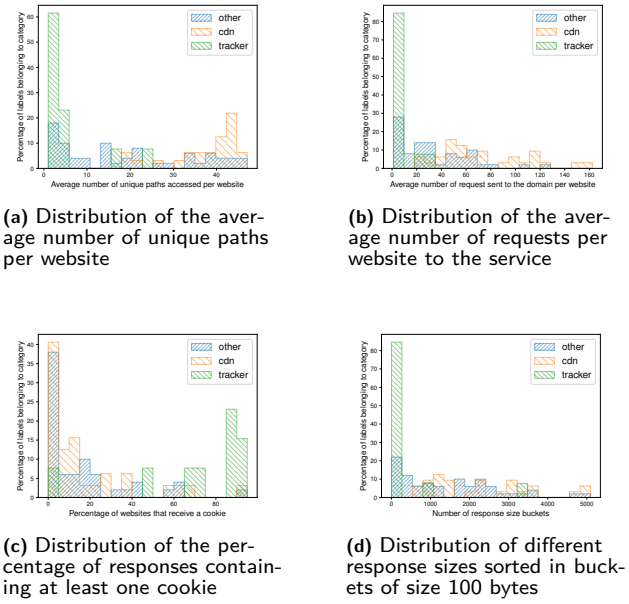
In our approach to distinguish the various kinds of first-party services we collected features that help us characterize a resource. For each of the 120 services we considered, we measured the number of websites the first-party is active on, the number of different hostnames a request to the service originates from, and the number of unique paths occurring in requests to the service. Furthermore, we captured the body size of the response, its content type (i.e. an image, script, video or html resource) and the average number of requests per website using the service. Lastly, we detected the percentage of requests and websites that sent and received cookies from the service.

To measure the uniformity of the response sizes of potential first-party trackers we sorted the sizes in buckets, each bucket with a size of 100 bytes. We then considered the number of buckets as a possible feature for distinction between different kinds of services. A low number of buckets would indicate that the service has a similar response to each request (e.g. the same script) which would increase the likelihood of the service being a tracker.

After manually visiting the websites of each of the considered services, we were able to classify them in three different categories: *trackers*, *Content Distribution Networks* (CDNs) and *other*. Any service that did not mention being explicitly a CDN or a tracker on their website, was categorized as "other".

To gain a better understanding of the features we collected, we analyzed their distribution across the different categories. Figure 8 shows the features that are the least overlapping for the three categories.
As can be deduced from Figure 8d and Figure 8a, the number of response size buckets and the number of unique paths accessed by the website is much lower for trackers than for CDNs and other services. This was in line with our expectation that customer websites access a similar resource each time. Furthermore, tracking services receive a low number of requests per website and often respond with a cookie.

Given the fact that we had a small list of confirmed trackers only, it was not feasible to build a classifier with the purpose of distinguishing tracking services from other types of services. However, our findings are still useful for performing assisted detection of tracking services. They form a simple heuristic for ruling out some companies from being trackers. With more data, the features that we gathered could likely be used for automatic detection.



(a) Distribution of the average number of unique paths per website

(b) Distribution of the average number of requests per website to the service

(c) Distribution of the percentage of responses containing at least one cookie

(d) Distribution of different response sizes sorted in buckets of size 100 bytes

Fig. 8. Features distinguishing trackers from other types of services