Tracking the Pixels: Detecting Web Trackers via Analyzing Invisible Pixels

Imane Fouad, Nataliia Bielova, Arnaud Legout, Natasa Sarafijanovic-Djukic

Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic

Tracking the Pixels: Detecting Web Trackers via Analyzing Invisible Pixels

Abstract: Web tracking has been extensively studied over the last decade. To detect tracking, most of the research studies and user tools rely on consumer protection lists. However, there was always a suspicion that lists miss unknown trackers. In this paper, we propose an alternative solution to detect trackers by analyzing behavior of invisible pixels that are perfect suspects for tracking. By crawling 829,349 webpages, we detect that third-party invisible pixels are widely deployed: they are present on more than 83% of domains and constitute 37.22% of all third-party images. We then propose a fine-grained classification of tracking based on the analysis of invisible pixels and use this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of 34,952,217 third-party requests. We demonstrate that two blocking strategies – based on EasyList&EasyPrivacy and on Disconnect lists – each miss 22% of the trackers that we detect. Moreover, we find that if we combine both strategies, 238,439 requests (11%) originated from 7,773 domains that still track users on 5,098 websites.

Keywords: online tracking; ad-blocker; cookie synching; invisible pixels

DOI foobar

1 Introduction

The Web has become an essential part of our lives: billions are using Web applications on a daily basis and while doing so, are placing *digital traces* on millions of websites. Such traces allow advertising companies, as well as data brokers to continuously profit from collecting a vast amount of data associated to the users. Recent research has shown that advertising networks and data brokers use a wide range of techniques to track users across the Web and Mobile [2, 7, 17, 19, 25, 28, 29, 31, 32, 35], from standard stateful cookie-based tracking [20, 32], up to advanced cross-browser device fingerprinting [11, 28]. In the last decade, numerous studies measured prevalence of third-party trackers on the Web [2, 9, 10, 19, 25–28, 32, 38].

But what makes a tracker? How to recognize that a third-party request is performing tracking? There is no uniform answer in the research community, and different works took a variaty of methodologies to identify third-party requests as tracking.

The most known Web tracking technology is based on *cookies*, but not all cookies are useful for cross-site tracking, and not all of them contain unique identifiers. One method to detect trackers is to analyse behaviour of HTTP requests and responses that set or send cookies [25, 32] and identify different classes of tracking, such as analytics or cross-domain tracking. Other studies measured the mere presence of third-party cookies [26, 27]. These studies were based on collecting *all third-party cookies* and analysing behavior associated to them. However, it is well-known that cookies are used for various functionalities, and may contain non-unique values that are not useful for tracking. Therefore, measuring only a number of third-party cookies definitely leads to a high number of false positives. Several other works [1, 19, 20] proposed heuristics to filter cookies that are likely to contain unique identifiers. However this detection was applied at large scale only in the context of measuring *cookie syncing*, that allows third-parties to merge users' data across websites [1, 19]. The previous works demonstrate that there is no unified method to detect which third-party requests are tracking.

Relying on consumer protection lists: Detection of identifier cookies and analysing behaviors of third-party domains is a complex task in itself, and we notice that most of the previous works that aim at measuring trackers at large scale instead rely on *consumer protection lists*. EasyList [15] and EasyPrivacy [16] (EL&EP) are the most popular publicly maintained blacklist of know advertising and tracking domains, used by the popular browser extensions AdBlock Plus [3] and uBlockOrigin [36]. According to PageFair

Imane Fouad: INRIA, imane.fouad@inria.fr
Nataliia Bielova: INRIA, nataliia.bielova@inria.fr
Arnaud Legout: INRIA, arnaud.legout@inria.fr
Natasa Sarafijanovic-Djukic: IRIS, natasa.sarafijanovic@gmail.com

report of 2017 [4], AdBlock Plus is installed on more than 600 million devices in the world. Disconnect [13] is another very popular list for detecting domains known for tracking, used in Disconnect browser extension [12] and in tracking protection of Firefox browser [21].

Relying on EL&EP or Disconnect became the *de facto* approach to detect third-party tracking requests in privacy and measurement community [8–10, 18, 19, 22–24, 31]. However it is well-known that these lists detect only known tracking and ad-related requests, and a tracker can easily avoid this detection by registering a new domain or changing the parameters of the request.

Invisible pixels are perfect suspects for tracking: In this work, to detect trackers, we propose a new technique based on the analysis of invisible pixels[1]. These images are routinely used by trackers in order to send information or third-party cookies back to their servers: the simplest way to do it is to create a URL containing useful information, and to dynamically add an image tag into a webpage. Since invisible pixels do not provide any useful functionality, we consider them *perfect suspects for tracking*. By analysing a dataset of invisible pixels, collected from 829, 349 webpages, we observe that invisible pixels are widely used: more than 83% of pages incorporate at least one invisible pixel.

Overall, we make the following key contributions:

1. **We define a new classification of Web tracking behaviors based on the analysis of invisible pixels.** By analyzing behavior associated to the delivery of invisible pixels, we propose a new fine-grained classification of tracking behaviors, that consists of 8 categories of tracking. To our knowledge, *we are the first to analyse tracking behavior based on invisible pixels that are present on 83% of the webpages*. We present an overview of the classification of tracking behaviors in Section 5.
2. **We apply our classification to a full dataset and uncover new collaborations between third-party domains.** We detect new relationships between third-party domains beyond basic cookie syncing detected in the past. In particular, we discovered that *first to third party cookie syncing* is the most prevalent tracking behavior performed by 50,812 distinct domains. Finally, we find that 76.23% of requests responsible for tracking originate from loading other resources than invisible images. To our knowledge, *we are the first to discover a highly prevalent first to third party syncing behavior detected on 51.54% of all crawled domains*. We present all the categories of tracking in Section 6.
3. **We show that the consumer protection lists cannot be considered as ground truth to identify trackers.** We find out that the browser extensions based on EasyList and EasyPrivacy (EL&EP) and Disconnect each miss 22% of tracking requests we detect. Moreover, if we combine all the lists, 238,439 requests originated from 7,773 domains are unknown to these lists and hence still track users on 5,098 webpages even if tracking protection is installed. We also detect instances of cookie syncing in domains unknown to these lists and therefore likely unrelated to advertising. To our knowledge, *we are the first to detect that EL&EP and also Disconnect lists used in majority of Web Tracking detection literature are actually missing tracking requests to 7,773 distinct domains.*

2 Background

In this section we present an overview of cookie-based tracking and explain what is cookie syncing.

The websites are composed of the content provided by the owner of the website and numerous third-party content, such as advertisements, web analytic scripts, social widgets or images. Following the standard naming [25], for a given website we distinguish two kinds of domains: *first-party* domain, which is the domain of the website owner, and *third-party* domains that host *third-party* content served on the website.

Cookies: Any domain the browser is interacting with can set an HTTP cookie that stores a unique identifier associated with the browser. Cookies can be set either through the "Set-cookie" HTTP header or programmatically by a JavaScript library included in the website. To protect cookies from being stolen, browsers implement an access control mechanism to ensure that only the server that set the cookies will be able to receive them: when cookies are stored in the browser, they are marked with the domain and path that set them and are therefore sent with requests to the same domain and path. Additionally, the Same Origin Policy (SOP) [34] ensures that cookies from one origin cannot be programmatically accessed by scripts in any other origin.

Tracking via cookies: *First party* cookies are set in the users browser by the site explicitly visited by the user or programmatically by the third party script in-

1 By "invisible pixels" we mean 1x1 pixel images or images without content.

cluded in the website (due to SOP). *Third party* cookies are instead set either (1) in the HTTP response by any third party content (images, html files or even at the delivery of scripts); or (2) programmatically via scripts operating from a third party iframe. While first-party cookies allow third parties to track users *within one website*, third party cookies are known to allow trackers to collect user's browsing history *cross-domains* [32].

Even though cookies is the most common mechanism for Web tracking today, researchers have recently documented the pervasiveness other types of tracking, such as via local storage and Flash objects [7, 32, 35] and various forms of browser fingerprinting [1, 2, 19, 28].

Cookie syncing: Every cookie stored in the user's browser is only accessible to the domain that set it (or to the origin in case the cookies were set programmatically). However, third parties are interested in merging information they collected about the users and recreate a more complete history of the user's browsing. To do so, domains use the mechanism of *cookie syncing* or *cookie matching* [1, 9, 19, 29] that shares identifiers of the same user among several third-party domains. Cookie syncing is known to be used in the Real-Time-Bidding auction for targeted advertisement.

3 Related Work

To measure prevalence of Web trackers, previous studies use two methods to detect trackers: (1) based on behavior of third-party content and analysis of third-party cookies, (2) based on consumer protection lists. In this section we present the related works based on these two alternative methods.

Detection of trackers by analysing behavior: Roesner et al. [32] and Lerner et al. [25] were the first to analyze trackers based on their behavior. They have proposed a classification of tracking behaviors that make a distinction between analytics and cross-domain tracking. With respect to this previous work, we propose a more *fine-grained classification of tracking behaviors* that includes not only previously known behaviors but also specific categories of cookie sharing and syncing (see more details in Section 5). Yu et al.[38] identify trackers by detecting unsafe data without taking into account the behavior of the third party domain and the communications between trackers.

Previous studies by Acar et al. [1], Olejnik et al. [29], Englehardt and Narayanan [19] and Bashir et al. [9] were measuring cookie syncing on thousands of websites. Olejnik et al. [29] consider as potential identifier every cookie sufficiently long. If the value of this cookie is shared between domains, such sharing is classified as cookie syncing. Additionally, Olejnik et al. studied the particular case of `doubleclick` to detect cookie matching based on the URL patterns.

In our study we show that domains are using more complex techniques to store and share the identifier cookie. We build our technique based on the study of Acar et al. [1] and Englehardt and Narayanan [19], where they show that companies may use various technique to store the identifier in the cookie. However, they only check for the identifiers that are stored and shared in a clear text. We further detect synchronization of encoded cookies and even encrypted ones for the case of `doubleclick.net`. Bashir et al. [9] used ads to detect cookie syncing. The authors filter out all images with dimensions $< 50\times50$ pixels excluding by that the invisible pixels that are the basis of our work. We show that cookie syncing is used with invisible images as well.

Detection of invisible pixels Ruohonen and Leppänen [33] studied the presence of invisible pixels in Alexa's top 500 websites. Differently from our work, where we detect all effectively delivered images from the response headers, the authors analyze the source code of landing HTML page and extract images from the <img> tag. We believe such method misses an important number of images that are dynamically loaded.

To compare to their work, we crawled the first 500 Alexa's top websites with all the links, creating a dataset similar to the original work [33]. We have detected $2,698,177$ images from which $746,974$ are invisible pixels (27.68%). Ruohonen and Leppänen instead only extracted $30,572$ images from a similar dataset, from which they detected 324 invisible pixels (1.05%). This comparison shows that detection by the <img> tag indeed misses a significant number of invisible images.

Detection of trackers with consumer protection lists To detect domains related to tracking or advertisement, most of the previous studies [8–10, 18, 19, 22–24, 31] rely on consumer protection lists, such as EasyList [15] and EasyPrivasy [16] (EL&EP). EL&EP became *de facto* approach to detect trackers in security, privacy and web measurement community. Only from the latest three years venues, we identified *nine papers that rely on EL&EP* to detect or block third-party tracking and advertising (see Table 5 in Appendix 9 for details).

Englehardt and Narayanan [19] seminal work on measuring trackers on 1 million top Alexa websites relies on EL&EP as a ground truth to detect requests to

trackers and ad-related domains. We believe that this work has set up a methodology to detect trackers that then have been used in the number of follow-up studies.

Three papers by Bashir et al. [8–10] customize EL&EP to detect 2^{nd}-level domains of tracking and ad companies: to eliminate false positives, a domain is considered only if it appears more than 10% of the time in the dataset. Lauinger et al. [24] use EL&EP to identify advertising and tracking content in order to detect what content has included outdated and vulnerable JavaScript libraries in Web applications. Razaghpanah et al. [31] use EasyList as an input to their classifier to identify advertising and tracking domains in Web and mobile ecosystems. Ikram et al. [22] analysed how many tracking JavaScript libraries are blocked by EL&EP lists based on 95 websites. They have found that EasyList is able to block 44% of tracking scripts. Englehardt et al. [18] apply EL&EP lists on third-party leaks caused by invisible images in emails. Iordanou et al. [23] rely on EL&EP as a ground truth for detecting ad- and tracking-related third party requests.

Only one work by Papadopoulos et al. [30] use Disconnect list [13] to detect tracking domains.

Summary: To the best of our knowledge, *we are the first who compares the new detection method based on invisible pixels and cookies with the consumer protection lists extensively used in the literature.*

4 Methodology

To track the user, domains deploy different mechanisms that has a different impact on the user's privacy. While some domains are only interested in tracking the user within the same website, others are recreating her browsing history by tracking her across domains. In our study, by "web tracking" we mean all behaviors, including both within-site and cross-domain tracking. To detect web tracking we rely on invisible pixels that are perfect suspects for tracking. In this section, we explain the data collection process and the criteria we used to detect identifier cookies and cookie sharing.

4.1 Data collection

We first performed passive Web measurements using the OpenWPM platform [19]. OpenWPM provides browser automation by converting high-level commands into automated browser actions. It allowed us to crawl websites in parallel by opening multiple browsers instances at the same time. In order to distribute the crawling, we spread the load at the start of the crawling to increase the number of browsers. We run our measurements in a distributed way on our institution's cluster located in France. For our crawling, we used nodes with at least 12 GB of memory per node. On every node we lunched 20 browsers instances in parallel.

Full dataset: We crawled the top 100,000 domains according to Alexa ranking in October 2018 in France [5]. For each domain we visited the home page and the first 10 links pointing to pages in the same domain. The timeout for loading a homepage is set up to 90s, and the timeout for loading a link on the homepage is set up to 60s. Out of 100,000 Alexa top domains, we successfully crawled 84,094 domains with a total of 829,349 pages.

For every page we crawl, we store the HTTP request (url, method, header, date and time), the HTTP response (url, method, status code, header, date and time), and the cookies (both set/sent and a copy of the browser cookie storage) to be able to capture the communication between the client and the server. We also store the body of the HTTP response if it's an image, whose *content-length* field in the header is less than 100 KB. We made this choice in order to save storage space and because we were only interested in the detection of invisible pixels. In our first dataset, named *full dataset*, we capture all HTTP requests, responses and cookies.

We have designed our crawl to be semi-stateful. We keep the state (cookies and other browser storage) of the crawl only while visiting pages from the same domain in order to detect cookie synching. When we finish crawling one domain, we kill the instance of the browser and open a new fresh instance with empty storage to start crawling the next domain. This trade-off allowed us to crawl a large amount of pages in a stateful fashion within domain and i statelessly across domains.

Prevalence of invisible pixels: As a result of our crawl of 829,349 pages, we have collected 20,375,723 images with a *content-length* less than 100 KB, that represent 92.85% of the total number of delivered images.

Figure 1 shows the distribution of the number of pixels in all collected images. We notice that invisible pixels represent 37.22% of the total number of collected images: 27.66% are 1x1 pixel images and 9.56% have no content. If we consider all images (also those larger than 100KB), then invisible pixels would still represent 34.25% of the total number of images.

Out of 84,094 successfully crawled domains, 75,393 (89.65%) domains contain at least one page with one

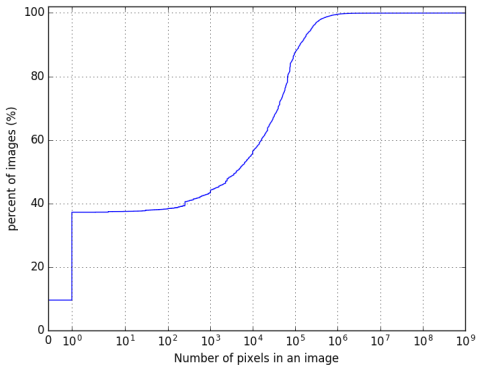


Fig. 1. Cumulative function of the number of pixels in images with a *content-length* less than 100 KB. *9.56% of images have no content (they are shown as zero-pixel images),* 27.66% *of images are of size 1x1 pixels.*

invisible pixel. If we analyse webpages, then 83.61% of all 829,349 pages include at least one invisible pixel.

Invisible pixels subdataset: The invisible pixels do not add any content to the Web pages. However, they are widely used in the web. They generally allow the third party to send some information using the requests sent to retrieve the images. Hence, every invisible pixel represents a threat to a user privacy, because the user is totally unaware of their existence. We consider the set of requests and the responses used to serve the invisible pixels as a ground-truth dataset that we call *invisible pixels dataset*. The study of this *invisible pixels dataset* allow us to excavate the tracking behaviors of third party domains in the web.

In total we have found 6,806 different third parties that serve invisible pixels. We notice that `google-analytics` is the top domain that serves invisible pixels in 58,777 (69.89%) websites out of 84,094 successfully crawled websites. (Figure 18 in the Appendix presents the top 20 such domains.)

4.2 Detecting identifier cookies

Cookies are a classical way to track the user in the web. A key task to detect this kind of tracking is to be able to detect cookies that store an identifier. We will refer to them as *identifier cookie*. In our work, we consider both first party and third party cookies – this helps us to detect new categories of tracking, such as *first to third party cookie syncing* (see Section 6.2.1). In the following we define several steps to define an identifier cookie.

We analyzed the 829,349 crawled pages where we have a total of 15,973,353 cookies. To identify a user, a cookie value should be user specific. We define a cookie value to be *safe* if it's shared between multiple domains. Remember that in our dataset every domain is crawled in a new instance of the browser with empty cookie storage. If the same cookie value appears in multiple domains, then it's unlikely used to distinguish between users. Notice that safe cookies can be used for browser fingerprinting, but this is out of the scope of our study.

The cookie length should be at least 6: We grouped the cookies based on their value length and computed the percentage of cookies with a *safe* value for each cookie length (for a detailed plot, see Figure 17 in Appendix 9). For every length smaller than 6, at least 26% of the cookies have a safe value. Based on that, we consider as potential *identifier cookie* the cookies with a value length at least equal to 6.

We exclude safe cookies: To be considered as a *identifier cookie* the cookie value should not only be sufficiently long, but it should also be user specific. Given the size of our dataset, we couldn't make two crawls and check which cookies remain the same as in [1, 19, 20]. Instead, we excluded cookies with *safe* values from all cookies whose value length is at least 6.

We don't consider the cookie lifetime: The lifetime of the cookie is an important criteria for the identifier detection in related works [1, 19, 20]: only cookies that expire at least a month after being placed are considered. However, we don't put any boundary on the cookie's lifetime because domains can continuously update cookies with a short lifetime and do the mapping of these cookies on the server side which will allow a long term tracking.

4.3 Detecting identifier sharing

Third party trackers not only collect data about the users, but also exchange it to build richer users profiles. Cookie syncing is a common technique used to exchange user identifier stored in cookies. The key to detect such behaviors is to detect the *identifier cookies* shared between domains. A cookie set by one domain cannot be accessed by another domain because of the cookie access control and Same Origin Policy[34]. Therefore, trackers need to pass identifiers through the browser and profit from HTTP redirection or explicit content inclusion.

Identifier sharing can be done in different ways: it can be sent in clear as a URL parameter value, or in a specific format, encoded or even encrypted. To detect identifiers that are only a part of cookie and parameter values, we take inspiration from [1, 19], and split

all cookie and URL parameter values using delimiters which are any character not in [a-zA-Z0-9,'-','_','.'].

We have deployed six different techniques to detect identifier sharing, described in Figure 2. The first three methods are generic: either the identifier is sent as the parameter value (**Direct sharing**), as part of the parameter value (**ID as part of the parameter**) or it's stored as part of the cookie value and sent as parameter value (**ID as part of the cookie**).

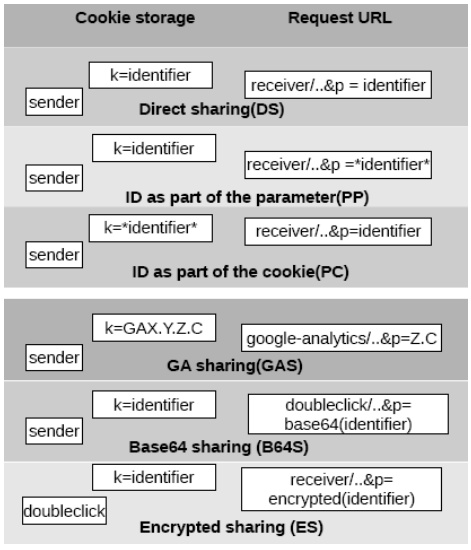


Fig. 2. Detecting identifier sharing. "Sender" is the domain that owns the cookie and triggers the request, "receiver" is the domain that receives this request and "identifier" is a cookie value that we detected as *identifier cookie*. "*" represents any string.

We noticed that the requests for invisible images, where we still didn't detect any cookie sharing originate mostly from `google-analytics.com` and `doubleclick.net`. Indeed, these domains are the most prevalent in serving invisible pixels across websites (see Figure 18 in Appendix 9). We therefore base the next techniques on these two use cases.

First, we notice that first party cookies set by `google-analytics.com` have the format GAX.Y.Z.C, but the cookies that are sent to it are of the form Z.C. We therefore detect this particular type of cookies, that were not detected in previous works that rely on delimiters (**GA sharing**). Second, by base64 decoding the value of the parameter sent to `doubleclick.net` we detect the encoded sharing(**Base64 sharing**). Finally, by relying on Doubleclick documentation [14] we infer that encrypted cookie was shared(**Encrypted sharing**). For more details see the Section 9.1 in Appendix 9.

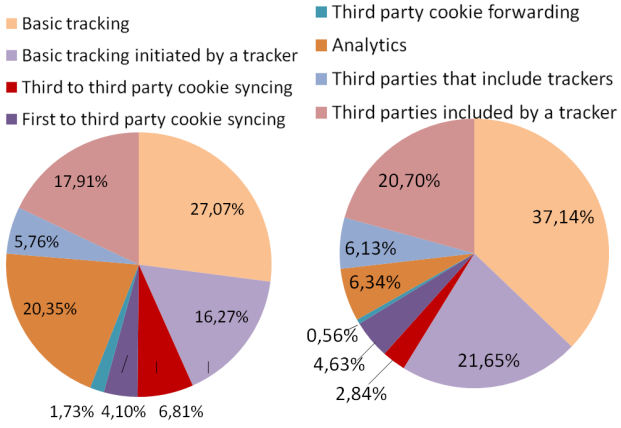
Limitation: The method we used is not without limitations. In our study we don't check the payload of the post requests which could be used to share the cookie as it's the case for `google-analytics`. In fact `google-analytics` provide two ways to send the identifier cookie either by sending it as part of the URL parameter value in case of get request, or in the payload if the URL is sent in a POST request, in our analysis we will not be detecting the second case [6].

5 Overview of tracking behaviors

In section 4.1 we detected that invisible pixels are widely present on the web and are perfect suspects for tracking. In this section we detect the different tracking behaviors by analyzing *invisible pixels dataset* from Section 4.1.

In total we have 5, 562, 902 third party requests leading to invisible images. By analyzing these requests, we detected 8 categories of different tracking behaviors in 4,358,637 (78.36%) requests that lead to invisible images. Figure 3a shows the distribution of all the tracking categories we detect in *invisible pixels dataset*.

We further group the categories into four main classes: explicit cross-domain tracking (Section 6.1), cookie syncing (Section 6.2), analytics category (Section 6.3) and implicit cross-domain tracking (Section 6.4).



(a) Invisible pixels dataset **(b)** Tracking in full dataset

Fig. 3. Classification of tracking behaviors in 4,358,637 requests from invisible pixels dataset (a) and of 18, 336, 172 third party requests from the full dataset (b).

Figure 4 presents an overview of all classes (black boxes) and categories (eight colorful boxes correspond to categories in Figure 3) of tracking behaviors.

After defining our classification using the *invisible pixels dataset*, we apply it on the *full dataset* where we have a total of 34,952,217 third-party requests collected

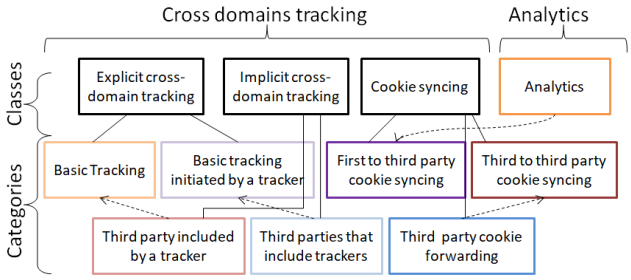


Fig. 4. Classification overview. $\dashrightarrow$ *represent the potential relation between categories in a stateful crawl.*

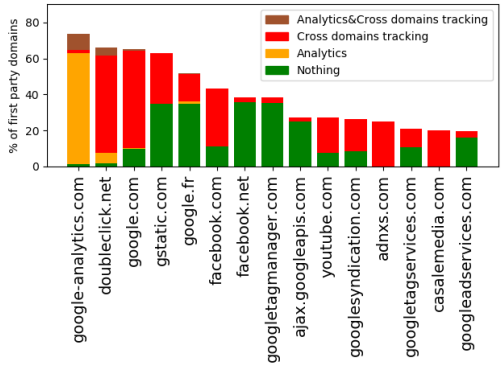


Fig. 5. Top 15 third parties involved in analytics, cross-domain tracking or both behaviors on the same first-party domain.

from 829,349 pages from 84,094 domains successfully crawled. Figure 3b shows the distribution of the different tracking behaviors for full dataset.

Out of 84,094 of successfully crawled domains, we identified at least one form of tracking in 77,310 (91.93%) websites. We have further analyzed prevalence of each tracking category that we report in Section 6. We found out that *first to third party cookie syncing that has never been detected in the previous works, actually appears on 51.54% of the crawled domains!*

In addition, we analysed the most prevalent domains involved in either cross-domain tracking, analysics or both behaviors. Figure 5 demonstrated that a third party domain often has several behaviors. For example, we detect that `google-analytics.com` exhibits both cross-domain tracking and analytics behavior.

We found that not all the tracking detected in the *full dataset* is based on invisible pixels. In fact, in the 18,336,172 requests involved in at least one tracking behavior in the full dataset, the top content delivered is JavaScript remote scripts (27.28%), while the second most common content are invisible pixels (23.77%). We also detected other content used for tracking purposes such as html files and visible images.

6 Classification of tracking

In this section, we explain all the categories of tracking behaviors presented in Figure 4 that we have uncovered by studying the *invisible pixels* dataset. We then detect all the categories of tracking in the *full dataset*, which allows us to uncover tracking domains and the collaboration between them. For each category we start by explaining the tracking behavior, we then give the impact of the behavior on the user's privacy, and finally we present the results from the *full dataset*.

6.1 Explicit cross-domain tracking

Explicit cross-domain tracking class includes two categories: *basic tracking* and *basic tracking initiated by a tracker*. In both categories we do not detect cookie syncing that we analyze separately in Section 6.2.

6.1.1 Basic tracking

Basic tracking is the most common tracking category both in *invisible pixels dataset* and *the full dataset*, as we see from Figure 3.

Tracking behavior: Basic tracking happens when a third party domain, say `A.com`, sets an identifier cookie in the user's browser. Upon a visit to a webpage with some content from `A.com`, a request is sent to `A.com` with its cookie. Using this cookie, `A.com` identifies the user across all websites that include content from `A.com`.

Impact: *Basic tracking* is the best known tracking technique, that allows third parties to track the user across sites, hence to recreate her browsing history. However, third parties are able to identify the user only one the websites where their content is present.

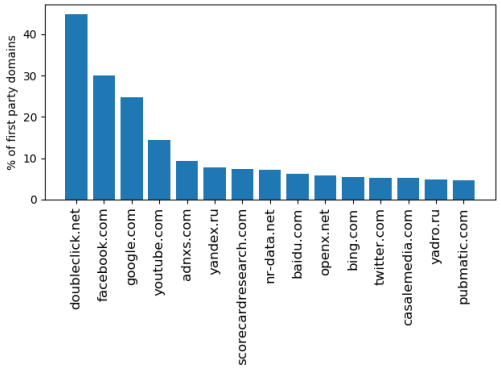


Fig. 6. Basic tracking: Top 15 cross-domain trackers in 84,094 successfully crawled domains.

Results: We detected basic tracking in 83.66% of domains. In total, we found 22,270 distinct third parties making basic tracking. Figure 6 shows the top domains involved in basic tracking. We see that `doubleclick.net` alone is tracking the user on over 37,702 (45%) domains. By only using the *basic tracking*, `doubleclick.net` recreates 36.7% of the user's browsing history. This percentage becomes more important if we consider the company instead of the domain. By only looking at the top 20 domains, we can already notice that 3 among them belong to Google (`doubleclick.net`,`youtube.com` and `google.com`) and so using these three domains `Google` can recreate 44.05% of user's browsing history using only the basic tracking.

6.1.2 Basic tracking initiated by a tracker

When the user visits a website that includes content from a third party, the third party may not only track the user with its cookie but it can also redirect either by inclusion or by redirection process to another tracker that will associate his own identifier cookie to the user. In this case the second tracker is not directly embedded by the first party and yet it can track the user.

Tracking behavior: *Basic tracking initiated by a tracker* happens when a basic tracker is included in a website by another basic tracker.

Impact: By redirecting to each other, trackers trace the user activity on a larger number of websites. They gather the browsing history of the user on websites where at least one of them is included. The impact of these behavior on the user's privacy could be similar to the impact of cookie syncing. In fact, by mutually including each other on websites, each tracker can recreate the combination of what both partners could have collected using basic tracking. Consequently, through *basic tracking initiated by a tracker* trackers could share the user's browsing history without syncing cookies.

Results: We detected Basic tracking initiated by a tracker in 64.82% of domains. From Figure 7 we can notice that `doubleclick.net` is the top tracker included by other third parties. It's included in over 29,642 domains. By only relying on it's partners, without being directly included by the developer, `doubleclick.net` can recreate 17.23% of the user's browsing history.

Note that a domain can implement multiple kind of tracking on the same page. `Doubleclick.net` is included by 1,425 different third party trackers in our dataset. In Table 1 we present the top 10 partners that are mutually including each other on websites.

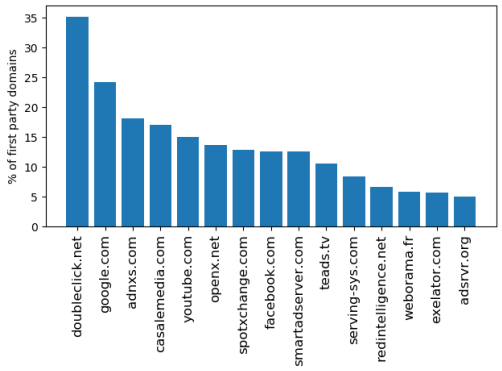


Fig. 7. Basic tracking initiated by a tracker: Top 15 receivers in 84,094 successfully crawled domains.

Partners	# of requests
google.com ↔ doubleclick.net	168,782
adnxs.com ↔ doubleclick.net	52,272
pubmatic.com ↔ doubleclick.net	41,289
youtube.com ↔ google.com	35,683
googlesyndication.com ↔ doubleclick.net	30,577
serving-sys.com ↔ doubleclick.net	28,822
casalemedia.com ↔ doubleclick.net	26,361
weborama.fr ↔ doubleclick.net	24,514
redintelligence.net ↔ doubleclick.net	21,758
openx.net ↔ doubleclick.net	19,576

Table 1. Basic tracking initiated by a tracker: Top 10 partners performing mutual cross domain tracking by inclusion.

6.2 Cookie syncing

In order to create a more complete profile of the user and better target her, third party domains could merge the data they collected. To do so they first need to synchronize the identifiers they associated to her which could be done through the cookie syncing mechanism. We define three categories that involves at least two parties that share their cookies. In previous works the first two categories *third to third party cookie syncing* and *third party cookie forwarding* were measured together. *First to third party cookies syncing* is the new category that was never detected in the past. Cookie forwarding has been always called "syncing" while instead it simply enables a third party to reuse an identifier of a tracker, without actually syncing its own identifier.

6.2.1 First to third party cookie syncing

First to third party cookie syncing involves a first and a third party domain.

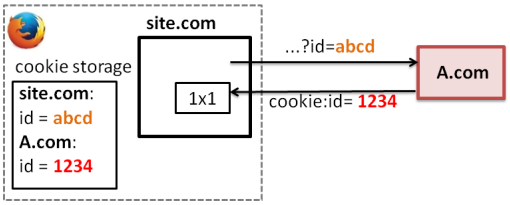


Fig. 8. First to third party cookie syncing behavior. *site.com includes A.com, a request is sent to A.com with site.com's cookie as part of the URL along with it's own cookie.* This behavior allows the first party website and the third party domain to combine the data collected about the user.

Tracking explanation: Figure 8 demonstrates the cookie syncing performed by a first and third party domains[2]. As shown in Figure 8. The first party domain includes an invisible pixel. The invisible pixel has as source A.com?id=abcd where `A.com` is the partner and *abcd* is the identifier cookie of the user set by `site.com`. If the tracker already set a cookie in user's browser, this cookie will be sent as part of the request.

Using these two identifiers (the first party's identifier received in the URL parameter and it's own identifier sent in the cookie) the third party domain can do the matching and create a matching table. Once the matching table is built, `A.com` can link the identifiers *1234* and *abcd* to the same user.

Impact: In our study, we differentiate the case when one of the collaborating domains is the first party visited directly by the user and when the two domains are third parties. We made this distinction because the kind and the sensitivity of the data shared differs in the two cases. When the user visits a first party, she could provide a number of sensitive information such as name, address, age, etc. After syncing cookies with the third party, the first party could potentially share these information with the tracker.

Results: We detected First to third party cookie syncing in 51.54% of visited websites. In Table 2, we present the top 10 partners syncing first party cookies. In total we found 50,812 different partners involved. The top partners are `google-analytics` and `doubleclick`. In fact, through redirection process, `google-analytics` transfer the first party cookie to `doubleclick` that insert a, identifier cookie into the user's browser. So `google-analytics` is not only performing analytics but it's also triggering other tracking techniques.

2 Notice that in figures that explain a tracking behavior, we show cookies only in the response, and never in a request. This actually represents both cases when cookies are sent in the request and also set in the response.

Partners	# of requests
google-analytics.com → doubleclick.net	63,403
livejournal.com → yandex.ru	2,645
livejournal.com → doubleclick.net	2,226
livejournal.com → rambler.ru	1,370
uol.com.br → doubleclick.net	982
nexac.com → addthis.com	747
empr.com → lytics.io	597
uol.com.br → tailtarget.com	552
wmagazine.com → condenastdigital.com	494
codeforwin.org → doubleclick.net	491

Table 2. First to third party cookie syncing: Top 10 partners.

Limitation: Since we are looking at the referer er to detect who the sender is in case of inclusion we may be miss interpreting who is the effective initiator of this syncing, it can be either the first party domain or a script included.

6.2.2 Third to third party cookie syncing

Third to third party cookie syncing involves at least two third party domains who synchronize their cookies..

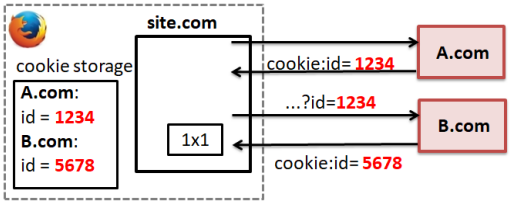


Fig. 9. Third to third party cookie syncing behavior. *site.com includes A.com, a request is then sent to A.com along with it's cookie. A.com redirects the request to B.com and includes it's identifier in the request URL, B.com receives the request along with it's cookie.* This behavior allows `A.com` and `B.com` to combine the data collected about the user.

Tracking explanation: Figure 9 demonstrates cookie syncing. The first party domain includes an invisible pixel having as source the first third party `A.com`. A request is then sent to `A.com` to fetch the content. Instead of sending the invisible, `A.com` decides to redirect to `B.com` and in the redirection request sent to `B`, `A.com` includes the identifier it associated to the user. In our example `B.com` will receive the request B.com?id=1234 where *1234* is the identifier associated by `A.com` to the user. Along with the request `B.com` will receive it's cookie *id = 5678* which will allow `B.com` to link the two identifiers to the same user.

Impact: *Third to third party cookie syncing* is one of the most harmful tracking techniques that impacts

the user's privacy. In fact, third party cookie syncing can be seen as set of trackers performing *basic tracking* and then exchanging the data they collected about the user. It's true that a cross domain tracker recreates part of the user's browsing history but this is only possible on the websites on which it was embedded. Using cookie syncing a tracker not only log the user's visit to the websites where it's included but it can also log her visits to the websites where it's partners are included. What makes this practice even more harmful is that a third party can have more than one partner with whom it syncs cookies and therefore the user's browsing history collected in that case is even more important. As example of this behavior we detected a cookie syncing between `adsrvr` and `rubiconproject`. `Adsvr` synced the same cookie with `doubleclick`, `casalemedia` and `adnxs`.

Partners	# of requests	Sharing technique
adnxs.com↔doubleclick.net	12,216	→DS,PCS,B64S; ← DS, PCS, ES
adnxs.com↔criteo.com	10,616	↔ DS
everesttech.net→doubleclick.net	9,330	→ B64S
mathtag.com↔adnxs.com	9,118	↔ DS
smartadserver.com	8,641	↔ PCS
mathtag.com	8,537	↔ DS, PPS
adnxs.com↔adsrvr.org	6,198	↔ DS
mathtag.com→openx.net	5,756	→ DS
adnxs.com↔adform.net	5,462	→DS; ← DS, PCS
adnxs.com→spotxchange.com	5,314	→ DS

Table 3. Third to third party cookie syncing: Top 10 partners. The arrows represent the follow of the cookie synchronization, (→) one way matching or (↔) both ways matching. DS (Direct sharing), B64S (Base64 sharing), ES (Encrypted sharing), PCS (ID as part of the cookie), PPS (ID sent as part of the parameter) are different sharing techniques described in Figure 2.

Results: We detected third to third party cookie syncing in 28.90% websites. We present in Table 3 the top 10 partners that we detect as performing cookie syncing. In total we have detected 3,727 unique partners performing cookie syncing. The syncing could be done in both ways as it's the case for `adnxs` and `doubleclick` or in one way as it's the case for `mathtag` and `openx`. In case of two ways matching we noticed that the two partners can perform different identifier sharing techniques. We see the complexity of the third to third cookie syncing that involves a large variety of sharing techniques. We also noticed that cookie syncing can be done between two subdomains from the same domain as it's the case for `smartadserver` and `mathtag`.

6.2.3 Third party cookie forwarding

The purpose of the collaboration between third party domains in *third party cookie forwarding* is to instantly share the browsing history.

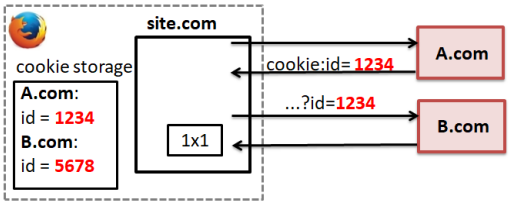


Fig. 10. Third party cookie forwarding behavior. *site.com sends request to A.com, where A.com's cookies are attached. A.com will then redirect the request to B.com and includes it's cookie as part of the request URL.* This allows `B.com` to track the user across domains using `A.com`'s identifier cookie

Tracking explanation: The first party domain `site.com` includes `A.com`'s invisible pixel. To get the image a request is sent to `A.com` along with it's cookie. `A.com` then redirects the request to it's partner (`B.com`) and send as part of the URL parameters the identifier cookie it associated to the user (*1234*) (Figure 10).

Third party cookie forwarding differs from *Third to third party cookie syncing* depending on whether their is a cookie set by the receiver in the browser or not. This category is similar to Third party advertising networks in Roesner et al.and Lerner et al.'s work [32] [25], in the sense that we actually have a collaboration of third party advertisers. However, in our study we check that the second tracker do not use it's own cookie to identify the user. This means that this tracker (`B.com`) is entirely relying on the first one (`A.com`) to track the user. In fact `B.com` uses `A.com`'s identifier to recreate her browsing history.

Impact: *Third party cookie forwarding* allows trackers to instantly share the browsing history of the user. `A.com` in Figure 10 do not only associate an identifier cookie to the user but it also redirect and share this identifier cookie with it's partner. This practice allows both `A.com` and `B.com` to track the user across websites. From a user privacy point of view, *third party cookie forwarding* is not as harmful as cookie syncing because the second tracker in this case does not contribute in the user's profile creation but it passively receives the user's browsing history from the first tracker.

Results: We detected Third party cookie forwarding in 14.83% of visited websites. To our surprise, the top domain receiving identifier cookie from third parties is `google-analytics` (figure 11). `Google-analytics` is

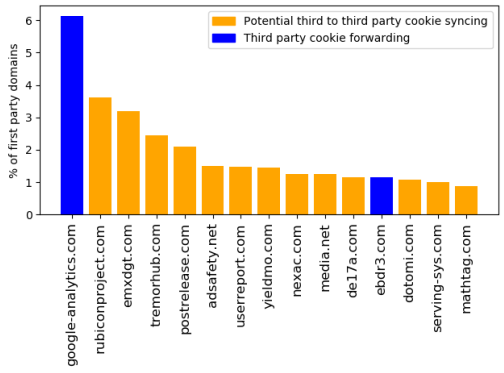


Fig. 11. Third party cookie forwarding : Top 15 receivers in 84,094 successfully crawled domains. *Potential third to third party cookie syncing - the receiver either sets or receives a cookie at least once during the crawl.*

normally included by first party domains to get analytics of their website, it's known as a *within domain tracker*. But in this case `google-analytics` is used by the third party domains. The third party is forwarding it's third party cookie to `google-analytics` on different websites, consequently `google-analytics` in this case is tracking the user across domains. This behavior was discovered by Roesner *et al.* paper [32]. They reported this behavior in only a few instances, but in our dataset we found 4,652 unique partners that forward cookies among which 3,287 are forwarding cookies to `google-analytics`. In Table **??** in the appendix we present the top 10 third parties forwarding cookies to `google-analytics` services.

The domains represented as *potential third to third party cookie syncing* in Figure 11 are the domains that set a cookie at least once during the crawl. In fact, in a stateful crawl those domains will receive the cookie along with the request due to the browser behavior and will consequently be categorized as syncing cookies.

6.3 Analytics category

Instead of measuring website audience themselves, websites today use third party analytics services. Such services provide reports of the website traffic by tracking the number of visits, the number of visited pages in the website, etc. The first party website includes content from the third party service on the pages it wishes to analyze the traffic.

Tracking explanation: Figure 12 shows the analytics category where the domain directly visited by the user (`site.com`) owns a cookie containing a unique identifier in user's browser. Such cookie is called first

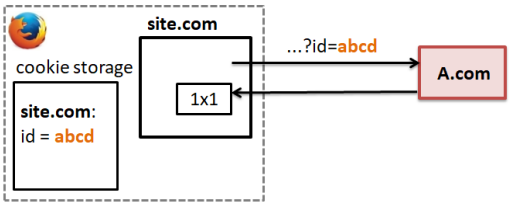


Fig. 12. Analytics behavior. *site.com includes A.com, the request is then sent to A.com with site.com's cookie as part of the URL.* Websites often use third party analytic services to track the users within their website.

party identifier cookie.This cookie is used by the third party (`A.com`) to uniquely identify the visitors within `site.com`. The first party website makes a request to the third party to get the invisible pixel and uses this request to share the first party identifier cookie.

We noticed that the analytic behavior could involve more than one third party. `Google-analytics` for example redirect to `doubleclick` and share the first party identifier cookie with it. `Doubleclick` in this case is behaving as an analytic domain, it's not setting a cookie in the user's browser.

Impact: In analytics behavior, the third party domain is not able to track the user across websites because it does not set it's own cookie in user's browser. Consequently, for this third party the same user will have different identifiers in different websites. However, using the first party identifier cookie shared by the first party, the third party can identify the user within the same website. From a user point of view, analytics behavior is not as harmful as the other tracking methods because the third party domain can not recreate the user's browsing history but it can only track her activity within the same domain which could be really useful for the website developer.

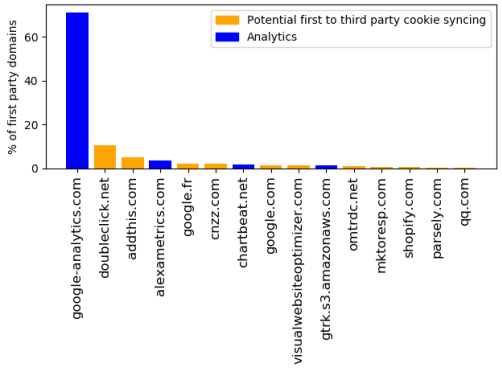


Fig. 13. Analytics: Top 15 receivers in the 84,094 domains. *Potential first to third party cookie syncing - the third party domain either sets or receives a cookie at least once during the crawl; Analytics - the domain is performing analytics.*

Results: We detected analytics in 75.85% of visited websites. From Figure 13, we can notice that `google-analytics` is by far the most popular service used for analytics which was expected. What's surprising is to see `doubleclick` as the second top domain performing analytics in our dataset. In fact this is due to the redirection behavior we explained above. In 80.36% of the cases where `doubleclick` is performing analytics, the first party identifier cookie is shared by `google-analytics` through a redirection and not sent by the first party itself.

In Figure 13, the Potential Tracking domains represent the domains that at some point during the crawl set a cookie. We remind that we are using a cross site stateless crawl (see section 4), we do not keep the cookies set by crawling one domain in another one. However, in a statefull crawl, the Potential Tracking domains would be categorized as receivers of the category first to third party cookie syncing due to the browser behavior. `Google-analytics` is not a potential tracker because it never sets cookies by himself, but it is involved in tracking categories, such as *Third party cookie forwarding* that enables `google-analytics` to track users across sites with the cookies of other third parties.

6.4 Implicit cross-domain tracking

Implicit cross-domain tracking includes two categories: *third parties that include trackers* and *third parties included by a tracker*. In both cases the third party domain is relies on another third party to perform tracking.

6.4.1 Third parties that include trackers

Tracking explanation: The domain directly visited by the user (`site.com`) includes an invisible pixel from a third party `A.com`, but `A.com` is not tracking the user. However, `A.com` redirects to or includes another domain that sets an identifier cookie on the user's browser.

Impact: Through redirection or inclusion, domains can track the user in a website in which they were not embedded. This practice could be very efficient to circumvent the privacy policy. If the developer choose to use a service that is not tracking the user across websites on his webpage, this service could redirect the request to it's partner that tracks the user cross sites.

Results: We detected third parties that include trackers in 35.56% of websites. Figure 14 shows that the top domain redirecting requests is

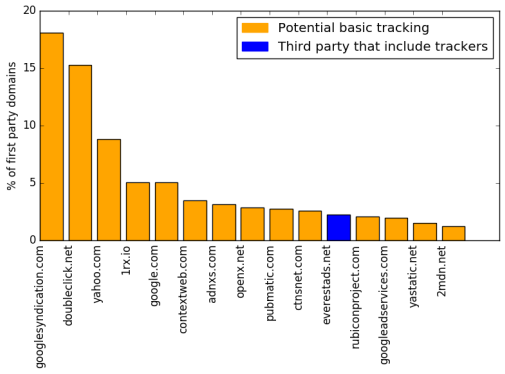


Fig. 14. Third party that include trackers: Top 15 senders in 84,094 domains. *Potential basic tracking - the domain sets or receives a cookie at least once during the crawl.*

Partners	# of requests
googlesyndication.com → adnxs.com	49,961
googlesyndication.com → pubmatic.com	40,472
googlesyndication.com → google.com	34,004
googlesyndication.com → casalemedia.com	16,508
imasdk.googleapis.com → doubleclick.net	16,153

Table 4. Third parties that include trackers: Top 5 partners

`googlesyndication.com`, and most of these requests are redirected to `doubleclick.net`. Table 4 presents presents the top partners involved in this tracking category. We notice that the partners sometimes belong to the same company (Google owns both `googlesyndication.com` and `google.com`) or to different companies (for example, `googlesyndication.com` owned by Google and `adnxs.com` – by AppNexus).

We remind that our crawler is semi-stateful: we keep the cookies only while visiting pages within one domain. However, if we made a stateful crawl, the Potential tracking domains presented in Figure 14 would be categorized as a basic trackers.

6.4.2 Third party included by a tracker

Tracking explanation: We noticed that the tracker included in the website often redirect to another third party. In this category *Third party included by a tracker* unlike the first third party that actually associate an identifier cookie to the user, we don't detect any kind of tracking performed by the second one.

impact: By analyzing the content loaded by these included third parties, we fond that 48.64% of the content is JavaScript. These scripts could be used to dynamically send requests or to perform tracking that is

out of the scope of our study. We don't define the exact role of these domains, but they are definitely players in the tracking ecosystem.

Results: We detected Third party included by a tracker in 61.04% of visited websites. A tracker can include another tracker as it's the case for *Third party tracker inclusion* but it can also include a third party that is not performing a detectable tracking by our method. In a stateful crawl, the potential trackers presented in Figure 15 would become trackers initiated by another tracker 6.1.2.

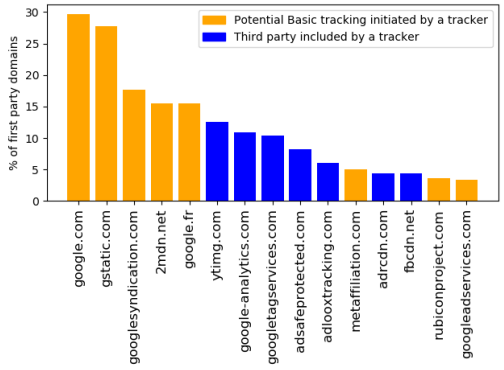


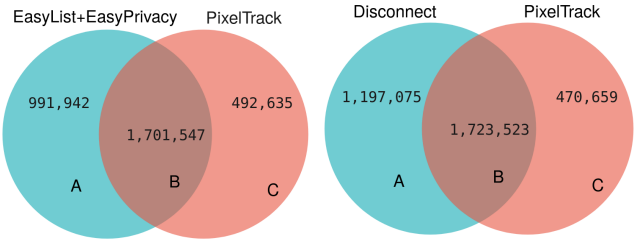
Fig. 15. Third party included by a tracker: Top 15 receivers in 84,094 successfully crawled domains .

7 Do EasyList, EasyPrivacy and Disconnect detect all trackers?

In this section we propose PixelTrack, a classifier that identifies all the tracking behaviors presented in Section 6 by analyzing the HTTP requests and responses, and cookie storage. We then analyse EasyList and EasyPrivacy (EL&EP) and Disconnect consumer lists. EL&EP were extracted on 26 October 2018, while Disconnect on 11 November 2018.: EL&EP are used by the most popular ad-blocking extensions, AdBlock Plus [3] and uBlockOrigin [36], while Disconnect list is used in Disconnect [12] and in tracking protection integrated in the Firefox browser [21]. We first compare PixelTrack to EL&EP and Disconnect separately, and then to all lists combined.

For the comparison, we extract a dataset of Alexa top 10,100 domains from the *full dataset*. Out of these domains, 8,425 were successfully crawled with a total of 83,859 pages and 4,275,704 third party requests.

Measuring blocked requests To simulate protection provided by the consumer lists, we determine



(a) Comparing to EasyList and EasyPrivacy **(b)** Comparing to Disconnect

Fig. 16. A: requests blocked by the filtering list and not detected by our method, B: requests blocked the filtering list and detcted by our method, C; requests detected only by our method. *The analysis is done on the first 8,425 successfully crawled domains with a total of* 4,275,704 *third party requests.*

whether a request would have been blocked by a browser or an extension using these lists, similarly to [18]. We classify a request as blocked if it matches one of the conditions: (i) the requests directly matches the list; (ii) the request is a consequence of a redirection chain where an earlier request was blocked. (iii) the request is loaded in a third-party content (an iframe) that was blocked (we detect this case by analyzing Referrer header).

7.1 EasyList and EasyPrivacy lists

We start the comparison by detecting all the requests blocked by EL&EP and detected as tracking by PixelTrack on our smaller dataset of 8,425 domains. The result of this comparison is presented in Figure 16a.

Requests blocked by EL&EP and not by PixelTrack. Figure 16a shows that EL&EP block 991,942 (23.19%) requests that were not detected as performing tracking by PixelTrack. The 991,942 requests blocked by EL&EP belongs to 2,230 different third party domain.

An explanation of this difference is that a tracker can have different behaviors. By analyzing these 2,230 domains, we found that 1,122 of them were detected at least once as performing tracking by PixelTrack. But EL&EP often blocks by domain name. If a third party is classified as tracker by the list, all the requests to this third party will be blocked. PixelTrack instead detects tracking by analyzing each request separately. Another reason why EL&EP blocks a request could be that the request is known for performing other types of tracking that is out of the scope of our study.

Requests detected only by PixelTrack: From Figure 16a, we also see that 492,635 (11.52 %) of the requests were only detected by PixelTrack. These requests belong to 4,872 different third party domains.

The 4,872 distinct third party domains involved in the tracking detected only by PixelTrack is clearly higher than 2, 230 domains detected only by EL&EP. One reason is the trackers that PixelTrack detects on average generate less requests (they generate on average 162 requests, while domains blocked by EL&EP generate 444 requests). So they are less popular and might be below the bar to be detected to EL&EP.

7.2 Disconnect list

To better study the efficiency of the Disconnect list [13] used in Disconnect extension and Firefox browser, we compared PixelTrack to Disconnect list as well. We compare the requests blocked by Disconnect to the requests detected as performing tracking by PixelTrack on the top 8, 425 successfully crawled domains. The result of this comparison is presented in Figure 16b.

Requests blocked by Disconnect and not by PixelTrack. Disconnect blocks 2, 920, 598 requests in total compared to 2, 693, 489 blocked by EL&EP. We conclude that Disconnect provide a better coverage than EL&EP. Figure 16b shows that Disconnect blocks 1, 197, 075 (27.99%) requests not detected by PixelTrack. These requests belong to 1, 868 distinct third party domains. By analyzing these requests, we found that 709 among the 1, 868 third party domains serving these requests were detected at least once by our method. Disconnect is based on domain names, so if we were to block requests by the domain names, we would have detected 709 out of 1, 868 domains blocked by Disconnect.

Requests detected only by PixelTrack: Figure 16b also shows that 470, 659 (11%) of the requests were only detected by PixelTrack. These 492, 635 requests belong to 5, 894 different third party domains. Similarly to EL&EP, the number of third party domains involved in the tracking detected only by PixelTrack is also higher than 1,868 domains detected only by Disconnect.

7.3 Discovering Unknown Trackers

To compare efficiency of all tools that are based on EL&EP and on Disconnect together, we compare requests blocked by these consumer lists with requests detected by PixelTrack as tracking according to classification from Figure 4. These results are based on the dataset of 4, 275, 704 third-party requests collected from 83, 859 pages that belong to the top 8, 425 domains.

We find that 238, 439 requests originating from 7,773 full third-party domains[3] detected by PixelTrack are not blocked by EL&EP and Disconnect[4] yet these requests are responsible for at least one type of tracking. These 238, 439 requests missed by the consumer protection lists represent 11% of all 2,194,182 third-party requests that we identified as tracking using PixelTrack.

We have detected that the 238, 439 requests detected by PixelTrack perform all classes of tracking behaviors (classes are shown as black boxes in Figure 4). Figure 19 in Appendix 9 shows the distribution of classes of tracking behaviors detected by PixelTrack.

We notice that the most privacy-violating behavior that includes setting, sending or syncing third-party cookies is represented by the *explicit cross-domain tracking* (79.56%) and *cookie syncing* (2.93%) classes, that combined, are present in 196,726 (82.49%) of missed requests. These tracking requests appear in 4,513 (53.56%) first party domains, including popular websites such as `tmall.com` (#8 in Alexa list) and `mail.ru` (#37 in Alexa list). The missed 196,726 requests originate from 6,226 full domains. In the remainder of this section we investigate 19 specific domains to explain why they are not blocked.

Table 8 of Appendix 9 presents a list of 19 domains that perform explicit cross-domain tracking or cookie syncing, as well as owners and country of registration of domains that we extracted using whois library [37] and complementing it with manual search. We also manually analyzed all the cookies associated to tracking, and report examples of the cookie's host and expiration date.

The first 15 domains we have selected for investigation are the most prevalent in performing explicit cross-domain tracking and cookie synching in the first-party domains. We have also added two domains, that are less prevalent, but PixelTrack has detected more than 3,000 tracking requests originating from each of these two domains. Additionally, we included two tag managers who emit tracking behavior on a few websites, but whose content appear on more than 3% of all 8,425 domains. *So why these domains are missed by the consumer lists?*

The answer for Disconnect is straightforward: these domains simply do not appear in the list most likely be-

3 Notice that differently form previous sections, where by "domain" we meant 2^{nd}-level TLD, such as `google.com`, here we report on full domain names, such as `cse.google.com` that give us more information about the purpose of its inclusion.

4 This set of requests corresponds to the intersection of the corresponding sets C in Figures 16a and 16b.

cause these domains are known to provide visible content, such as consent frameworks or tag management. But for EL&EP we have found more interesting explanation by thoroughly analyzing these lists.

Serving content explicitly allowed by the EasyList. We find that `action.metaffiliation.com` belongs to an online advertising company NetAffiliation, whose main domain `netaffiliation.com` is blocked by the EasyList, however EasyList allows `metaffiliation.com` to serve images and any content if it's inside an iframe. Domains `cse.google.com` (Custom Search Engine by Google) and `onesignal.com` are in general blocked by EasyList but they are allowed to serve scripts on a list of predefined first-party websites.

Changing domain name to avoid blocking by EasyPrivacy. We notice that for several domains, EasyPrivacy blocks similar domain names. For `mc.webvisor.org`, EasyPrivacy blocks related domains `webvisor.com` and `webvisor.ru`. A content from a CDN `g.alicdn.com` is not blocked, but we see that another subdomain `atanx.alicdn.com` is explicitly blocked in EasyPrivacy. Similarly, EasyPrivacy doesn't block `ynuf.alipay.com` of Alibaba, but it blocks another similar domain of Alibaba, `ynuf.alibaba.com`. EasyPrivacy blocks a subdomain `tracker.cooster.ru` but doesn't explicitly block `cooster.ru`. Therefore, putting a tracking content on a similar domain, a higher level domain or a subdomain allows to avoid blocking.

Consent Framework systems. We identified `static.quantcast.mgr.consensu.org` by IAB Europe and `consent-pref.trustarc.com` by TrustArc, that rightfully should not be blocked because they provide useful functionality for GDPR compliance. In both cases, we detect that the cookie values seemed to be unique identifiers, but are set without expiraiton date, which means such cookies will get deleted when the user clothes her browser. Nevertheless, it's known that users rarely close browsers, and more importantly, it is unclear why a consent framework system sets identifier cookies even before the user clicks on the consent button[5].

Tag managers. These tools are designed to help Web developers to manage marketing and tracking tags on your their websites and should be blocked not to break the functionality of the website. We detected that two such managers, `tags.tiqcdn.com` by Tealium and `assets.adobedtm.com` by Adobe track users cross-sites, but both of them have an explicit exception in EasyList when it appears on a particular webpage.

For the remaining 8 domains in Table 8 we have not found an explanation and leave it for future work.

Summary. To sum up, tracking detection is a complex task and preventing this tracking is even more complex. Through the paper we have shown that a domain can have different behaviors. Domains can at the same time provide useful functionalities, but also track users. We have also shown that domains serving visible content such as CDNs or Tag managers may track the user as well. However, these domains are not blocked because the website functionality will break otherwise. Blocking domains by name as it's the case for Disconnect or EL&EP is not efficient. These lists are widely used in research literature, in browsers and ad-blocking extensions for tracker detection, but they lead to both false positives, and false negatives. Faced with the driven choice between protecting you privacy or keeping the functionality, we clearly need a more fine-grained approach to detect tracking.

8 Conclusion

Web tracking remains an important problem for privacy of Web users. Even after the General Data Protection Regulation (GDPR) came in force in May 2018, third party companies continue tracking users with various sophisticated techniques based on cookies without their consent. According to our study, 91.93% of websites incorporate at least one type of cookie-based tracking.

In this paper, we defined a new classification of Web tracking behaviors thanks to a large scale study of invisible pixels collected from 829, 349 webpages. This invisible images are frequently used in the web: they are present on more than 83% of the crawled webpages. We then applied our classification to the full dataset which allowed us to uncover different relationships between domains. The redirection process and the different behaviors that a domain can adopt are an evidence of the complexity of these relationships. We show that even the most popular consumer protection lists fail to detect these complex behaviors. We find out that the browser extensions based on EasyList and EasyPrivacy and Disconnect each miss 22% of tracking requests we detect. Therefore, these consumer protection lists should not be considered as ground truth to identify trackers, and industry should complement the usage of the lists with detection of trackers based on their behavior.

5 Remember that we did not emit any user behavior, like clicking on buttons or links during our crawling.

References

[1] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juárez, Arvind Narayanan, and Claudia Díaz. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*, pages 674–689, 2014.

[2] Gunes Acar, Marc Juárez, Nick Nikiforakis, Claudia Díaz, Seda F. Gürses, Frank Piessens, and Bart Preneel. Fpdetective: dusting the web for fingerprinters. In *2013 ACM SIGSAC Conference on Computer and Communications Security (CCS'13)*, pages 1129–1140, 2013.

[3] Adblock Plus Official website. https://adblockplus.org/.

[4] 2017 adblocking report. the pagefair team. https://pagefair.com/blog/2017/adblockreport/.

[5] Alexa. https://www.alexa.com/.

[6] analytics post request. https://developers.google.com/analytics/devguides/collection/protocol/v1/devguide.

[7] Mika D Ayenson, Dietrich James Wambach, Ashkan Soltani, Nathan Good, and Chris Jay Hoofnagle. Flash cookies and privacy ii: Now with html5 and etag respawning. Technical report, Available at SSRN: https://ssrn.com/abstract=1898390orhttp://dx.doi.org/10.2139/ssrn.1898390, 2011.

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

[9] Muhammad Ahmad Bashir, Sajjad Arshad, William Robertson, and Christo Wilson. Tracing Information Flows Between Ad Exchanges Using Retargeted Ads. In *Proceedings of the 25th USENIX Security Symposium*, 2016.

[10] Muhammad Ahmad Bashir and Christo Wilson. Diffusion of User Tracking Data in the Online Advertising Ecosystem. In *Proceedings on Privacy Enhancing Technologies (PETS 2018)*, 2018.

[11] Yinzhi Cao, Song Li, and Erik Wijmans. (cross-)browser fingerprinting via os and hardware level features. In *24th Annual Network and Distributed System Security Symposium, NDSS 2017, San Diego, California, USA, 26 February - 1 March, 2017*, 2017.

[12] Disconnect Official website. https://disconnect.me/.

[13] Disconnect List. https://disconnect.me/trackerprotection/blocked.

[14] cookies doubleclick. https://developers.google.com/ad-exchange/rtb/cookie-guide.

[15] EasyList filter lists. https://easylist.to/.

[16] EasyPrivacy filter lists. https://easylist.to/easylist/easyprivacy.txt.

[17] Peter Eckersley. How Unique is Your Web Browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, PETS'10, pages 1–18. Springer-Verlag, 2010.

[18] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. I never signed up for this! privacy implications of email tracking. In *Privacy Enhancing Technologies*, 2018.

[19] Steven Englehardt and Arvind Narayanan. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security ACM CCS*, pages 1388–1401, 2016.

[20] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *Proceedings of WWW 2015*, pages 289–299, 2015.

[21] The new Firefox. Fast for good. https://www.mozilla.org/en-US/firefox/new/.

[22] Muhammad Ikram, Hassan Jameel Asghar, Mohamed Ali Kaafar, Anirban Mahanti, and Balachandar Krishnamurthy. Towards seamless tracking-free web: Improved detection of trackers via one-class learning. In *Privacy Enhancing Technologies*, 2017.

[23] Costas Iordanou, Georgios Smaragdakis, Ingmar Poese, and Nikolaos Laoutaris. Tracing cross border web tracking. In *ACM Internet Measurement Conference (IMC)*, 2018.

[24] Tobias Lauinger, Abdelberi Chaabane, Sajjad Arshad, William Robertson, Christo Wilson, and Engin Kirda. Thou shalt not depend on me: Analysing the use of outdated javascript libraries on the web. In *Network and Distributed System Security Symposium, NDSS*, 2017.

[25] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska Roesner. 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)*. USENIX Association, 2016.

[26] Timothy Libert, Lucas Graves, and Rasmus Kleis Nielsen. Changes in third-party content on european news websites after gdpr,, 2018. https://timlibert.me/pdf/Libert_et_al-2018-Changes_in_Third-Party_Content_on_EU_News_After_GDPR.pdf.

[27] Timothy Libert and Rasmus Kleis Nielsen. Third-party web content on eu news sites: Potential challenges and paths to privacy improvement, 2018. https://timlibert.me/pdf/Libert_Nielsen-2018-Third_Party_Content_EU_News_GDPR.pdf.

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

[29] Lukasz Olejnik, Minh-Dung Tran, and Claude Castelluccia. Selling off user privacy at auction. In *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*, 2014.

[30] Panagiotis Papadopoulos, Pablo Rodríguez Rodríguez, Nicolas Kourtellis, and Nikolaos Laoutaris. If you are not paying for it, you are the product: how much do advertisers pay to reach you? In *Internet Measurement Conference, IMC*, pages 142–156, 2017.

[31] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Network and Distributed System Security Symposium, NDSS*, 2018.

[32] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Symposium on*

Networked Systems Design and Implementation, NSDI 2012, pages 155–168, 2012.

[33] Jukka Ruohonen and Ville Leppänen. Invisible pixels are dead, long live invisible pixels! In *Workshop on Privacy in the Electronic Society, WPES@CCS*, pages 28–32, 2018.

[34] Same Origin Policy. https://www.w3.org/Security/wiki/Same_Origin_Policy.

[35] Ashkan Soltani, Shannon Canty, Quentin Mayo, Lauren Thomas, and Chris Jay Hoofnagle. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management*, 2010.

[36] uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. https://github.com/gorhill/uBlock.

[37] whois library. https://pypi.org/project/whois/.

[38] Zhonghao Yu, Sam Macbeth, Konark Modi, and Josep M. Pujol. Tracking the trackers. In *International Conference on World Wide Web, WWW*, pages 121–132, 2016.

9 Appendix

9.1 Detecting identifier sharing

GA sharing: `Google-analytics` serves invisible pixels on 69.89% of crawled domains as we show in Figure 18. By analyzing our data, we detect that the cookie set by `google-analytics` script is of the form GAX.Y.Z.C, while the *identifier cookies* sent in the parameter value to `google-analytics` is actually Z.C. This case is not detected by the previous cookie syncing detection techniques for two reasons. First, "." is not considered as a delimiter. Second, even if it was considered as a delimiter, it would create a set of values {GAX, Y, Z, C} which are still different than the real value Z.C used as an identifier by `google-analytics`.

Base64 sharing: When a domain wants to share its *identifier cookie* with `doubleclick.net`, it should encode it in base64 before sending [14]. For example, when `adnxs.com` sends a request to `doubleclick.net`, it includes a random string into a URL parameter. This string is the base64 encoding of the value of the cookie set by `adnxs.com` in the user's browser.

Encrypted sharing: When `doubleclick.net` wants to share its *identifier cookie* with some other domain, it encrypts the cookie before sending, which makes it impossible to detect. Instead we rely on the semantic set by doubleclick to share this identifier that we extract from its documentation [14].

Assume that `doubleclick.net` is willing to share an identifier cookie with `adnxs.com`. To do so, Doubleclick requires that the content of `adnxs.com` includes an image tag, pointing to a RL that contains `doubleclick.net` as destination and a parameter *google_nid*. The value of *google_nid* will tell Doubleclick that `adnxs.com` was the initiator of this request. Upon receiving such request, `doubleclick.net` sends a redirection response pointing to a URL that contains `adnxs.com` as destination with encrypted `doubleclick.net`'s cookies in the parameters. When the browser receives this response, it redirects to `adnxs.com`, who now receives encrypted `doubleclick.net`'s cookie.

We detect such behavior by detecting requests to `doubleclick.net` with *google_nid* parameter and analysing the following redirection. If we notice that the redirection is set to a concrete domain, for example `adnxs.com`, we conclude that `doubleclick.net` has shared its cookie with this domain.

9.2 Figures

Table 5 summarizes the usage of EL&EP lists in the previous works that we describe in Section 3.

Figure 17 presents the percentage of cookies with a *safe* value for each cookie length. This result is used in Section 4.2 to identify the minimal cookie length to detect identifier cookie.

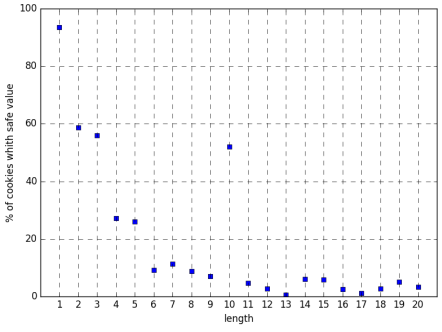


Fig. 17. Percentage of cookies with a safe value per length in 829,349 crawled pages. *For the lengths smaller than 6, from 26% (for 5 characters) to 93.59% (for 1 character) of the cookies have a safe value.*

Table 18 represents the Top 20 domains involved in invisible pixels inclusion in the 84,094 domains.

Table 7 represents the Top 10 third parties forwarding cookies to `google-analytics`.

Figure 19 represents the Classification of tracking behavior in the 238,439 requests missed by EL&EP and Disconnectin Section 7.

Table 5. Usage of EL&EP lists in security, privacy and web measurement community (venues form 2016-2018). "Detection" describes how EL&EP was used to detect trackers: whether the filterlists were applied only on all requests, on requests and follow-up requests that would be blocked, or whether filterlists were further customised before being applied to the dataset. "Dependency" describes whether the results of the paper rely on EL&EP or authors use these lists to only vefiy their results.

Paper	Venue	EasyList	EasyPrivacy	Detection	Dependency
Englehardt and Narayanan [19]	ACM CCS 2016	✓	✓	Req.	Rely
Bashir et al. [9]	USENIX Security 2016	✓		Custom.	Rely
Lauinger et al. [24]	NDSS 2017	✓	✓	Req.+Follow	Rely
Razaghpanah et al. [31]	NDSS 2018	✓		Custom.	Rely
Ikram et al. [22]	PETs 2017	✓		Req.+Follow	Verif.
Englehardt et al.[18]	PETs 2018	✓	✓	Req.+Follow	Verif.
Bashir and Wilson [10]	PETs 2018	✓	✓	Custom.	Rely+Verif.
Bashir et al.[8]	IMC 2018	✓	✓	Custom.	Rely
Iordanou et al.[23]	IMC 2018	✓	✓	Req.+Follow	Rely

Category	**Prevalence in first party domains**
Basic tracking	70,352 (83.66%)
Analytics	63,788 (75.85%)
Basic tracking initiated by a tracker	54,513 (64.82%)
Third party included by a tracker	51,331 (61.04%)
First to third party cookie syncing	43,340 (51.54%)
Third parties that include trackers	29,901 (35.56%)
Third to third party cookie syncing	24,302 (28.90%)
Third party cookie forwarding	12,474 (14.83%)

Table 6. Prevalence of all detected tracking categories observed on 84,094 successfully crawled domains.

Third parties	# of requests
laim.tv	957
spotify.com	780
adtrue.com	562
adsense.az	543
brainfoodmedia.gr	514
google.com	511
push.world	504
chaturbate.com	475
bidgear.com	460
spreaker.com	411

Table 7. Third party cookie forwarding; Top 10 third parties forwarding cookies to `google-analytics`

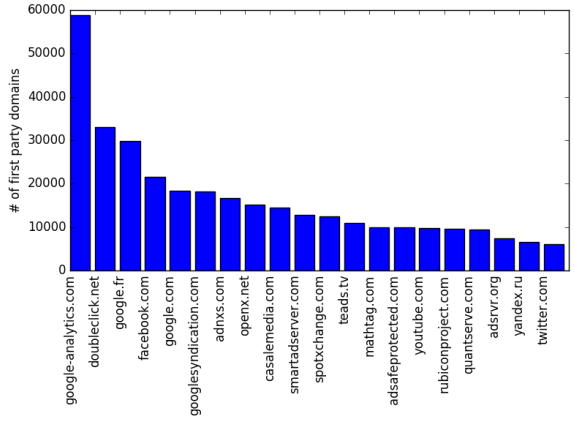


Fig. 18. Top 20 domains involved in invisible pixels inclusion in the $84,094$ successfully crawled domains.

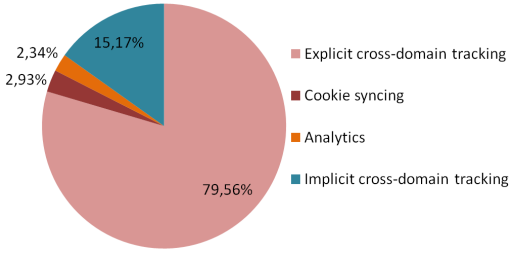


Fig. 19. Classification of tracking behavior in the $238,439$ requests missed by EL&EP and Disconnect.

Table 8 presents example of domains that perform explicit tracking and cookie syncing and that only detected by PixelTrack and not by the filters in Section 7.

Full domain	Prevalence of tracking in first-parties	Cookie host	Cookie expiration	Company	Country
Top 15 most-prevalent domains performing explicit tracking or cookie syncing					
pr-bh.ybp.yahoo.com	298 (3.54%)	.yahoo.com	1 year	Oath Inc.	US
action.metaffiliation.com	274 (3.25%)	.metaffiliation.com	2 months	C2B SA - NetAffiliation	FR
yastatic.net	168 (1.99%)	.yastatic.net	1 years	Yandex N.V.	RU
cse.google.com	160 (1.90%)	.google.com	1 year	Google LLC	US
onesignal.com	142 (1.69%)	.onesignal.com	1 year	Domains By Proxy, LLC	US
static.quantcast.mgr.consensu.org	122 (1.45 %)	static.quantcast.mgr.consensu.org	Session	IAB Europe	BE
mc.webvisor.org	87 (1.03%)	.mc.webvisor.org	10 min	Privacy protection service - whoisproxy.ru	RU
push.zhanzhang.baidu.com	86 (1.02%)	.baidu.com	1 year	Beijing Baidu Netcom Science Technology Co., Ltd.	CN
ampcid.google.com	79 (0.94%)	.google.com	1 year	Google LLC	US
ynuf.alipay.com	76 (0.90%)	ynuf.alipay.com	13 years	Zhejiang Taobao Network Co.,Ltd	CN
g.alicdn.com	72 (0.85%)	g.alicdn.com	10 years	Alibaba Cloud Computing Ltd.	CN
cooster.ru	74 (0.88%)	cooster.ru	10 years	Private Registration	DE
match.rundsp.com	73 (0.87%)	match.rundsp.com	1 years	RUN	US
fourier.alibaba.com	47 (0.56%)	.alibaba.com	10 years	Alibaba Cloud Computing (Beijing) Co., Ltd.	CN
bdimg.share.baidu.com	64 (0.76%)	.baidu.com	1 year	Beijing Baidu Netcom Science Technology Co., Ltd.	CN
Domains with more than 3,000 observed requests					
cdn.discordapp.com	8 (0.09 %)	.discordapp.com	1 year	Discord, Inc.	US
consent-pref.trustarc.com	58 (0.69 %)	consent-pref.trustarc.com	Session	TrustArc	US
Tag managers					
tags.tiqcdn.com	16 (0.19 %)	.tiqcdn.com	1 year	Tealium Inc	US
assets.adobedtm.com	31 (0.37 %)	.adobedtm.com	2 years	Adobe Inc.	US

Table 8. Domains missed by EL&EP and Disconnect but detected by PixelTrack to perform explicit cross-domain tracking and cookie syncing.