Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels

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

Imane fouad*, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic

Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels

Abstract: Web tracking has been extensively studied over the last decade. To detect tracking, previous studies and user tools rely on filter lists. However, it has been shown that filter lists miss trackers. In this paper, we propose an alternative method to detect trackers inspired by analyzing behavior of invisible pixels. By crawling 84,658 webpages from 8,744 domains, we detect that third-party invisible pixels are widely deployed: they are present on more than 94.51% of domains and constitute 35.66% of all third-party images. We propose a fine-grained behavioral classification of tracking based on the analysis of invisible pixels. We use this classification to detect new categories of tracking and uncover new collaborations between domains on the full dataset of $4,216,454$ third-party requests. We demonstrate that two popular methods to detect tracking, based on EasyList&EasyPrivacy and on Disconnect lists respectively miss 25.22% and 30.34% of the trackers that we detect. Moreover, we find that if we combine all three lists, $379,245$ requests originated from 8,744 domains still track users on 68.70% of websites.

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

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

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 works have shown that advertising networks and data brokers use a wide range of techniques to track users across the Web [2, 9, 22, 24, 31, 37, 38, 42, 43, 46], from standard stateful cookie-based tracking [25, 43], to stateless fingerprinting [2, 13, 24, 37].

In the last decade, numerous studies measured prevalence of third-party trackers on the Web [2, 11, 12, 24, 31–33, 37, 43, 49]. Web Tracking is often considered in the context of targeted behavioral advertising, but it's not limited to ads. Third-party tracking has become deeply integrated into the Web contents that owners include in their websites.

But what makes a tracker? How to recognize that a third-party request is performing tracking? To detect trackers, the research community applied a variety of methodologies. The most known Web tracking technique is based on *cookies*, but only some cookies contain unique identifiers and hence are capable of tracking the users. Some studies detect trackers by analysing cookie storage, and third-party requests and responses that set or send cookies [31, 43], while other works measured the mere presence of third-party cookies [32, 33]. To measure *cookie syncing*, researchers applied various heuristics to filter cookies with unique identifiers [1, 24, 25]. However, this approach has never been applied to detect tracking at large scale. Overall, previous works provide different methods to identify third-party requests that are responsible for tracking [43, 49]. Detection of identifier cookies and analysing behaviors of third-party domains is a complex task. Therefore, most of the state-of-the-art works that aim at measuring trackers at large scale rely on *filter lists*. In particular, EasyList [20] and EasyPrivacy [21] (EL&EP) and Disconnect [17] lists became the *de facto* approach to detect third-party tracking requests in privacy and measurement communities [10–12, 23, 24, 28–30, 42][1]. EasyList and EasyPrivacy are the most popular publicly maintained blacklist of known advertising and tracking requests, used by the popular blocking extensions AdBlock Plus [5] and uBlockOrigin [47]. Disconnect is another very popular list for detecting domains known for tracking, used in Disconnect browser extension [16] and in tracking protection of Firefox browser [26].

Nevertheless, filter lists detect only known tracking and ad-related requests. Therefore, a tracker can avoid this detection by using a different subdomain for track-

1 We summarize the usage of filter lists in security, privacy and web measurement community in Table 12 in the Appendix.

ing, or wholly register a new domain if the filter list block the entire domain. Even though, the second option is quite challenging because in such case, all the associated publishers would need to update their pages. Third parties can also incorporate tracking behavior into functional website content, which is never blocked by filter lists because blocking functional content would harm user experience. Therefore, it is interesting to evaluate how effective are filter lists at detecting trackers, how many trackers are missed by the research community in their studies, and whether filter lists should still be used as the *default tools* to detect trackers at scale.

Our contributions: To evaluate the effectiveness of filter lists, we propose a new, fine-grained behavior-based tracking detection. Our results are based on a stateful dataset of 8K domains with a total of 800K pages generating 4M third-party requests. We make the following contributions:

1- We analyse all the requests and responses that lead to invisible pixels (by "invisible pixels" we mean 1×1 *pixel images or images without content).* Pixels are routinely used by trackers 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. This makes invisible pixels *the perfect suspects for tracking* and propose a new classification of tracking behaviors. Our results show that pixels are still widely deployed: they are present on more than 94% of domains and constitute 35.66% of all third-party images. We found out that pixels are responsible only for 23.34% of tracking requests, and the most popular tracking content are scripts: a mere loading of scripts is responsible for 34.36% of tracking requests.

2- We uncover hidden collaborations between third parties. We applied our classification on more than 4M third-party requests collected in our crawl. We have detected new categories of tracking and collaborations between domains. We show that domains sync first party cookies through a *first to third party cookie syncing*. This tracking appears on 67.96% of websites.

3- We show that filter lists miss a significant number of cookie-based tracking. Our evaluation of the effectiveness of EasyList&EasyPrivacy and Disconnect lists shows that they respectively miss 25.22% and 30.34% of the trackers that we detect. Moreover, we find that if we combine all three lists, 379,245 requests originating from 8,744 domains still track users on 68.70% of websites.

4- We show that privacy browser extensions miss a significant number of cookie-based tracking. By evaluating the popular privacy protection extensions: Adblock, Ghostery, Disconnect, and Privacy Badger, we show that Ghostery is the most efficient among them and that all extensions fail to block at least 24% of tracking requests.

2 Methodology

To track users, domains deploy different mechanisms that have different impacts 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 sites. In our study, by "Web tracking" we refer to both within-site and cross-site tracking.
To detect Web tracking, we first collect data from Alexa top 10,000 domains, then by analyzing the invisible pixels we define a new classification of Web tracking behaviors that we apply to the full dataset. In this section, we explain the data collection process and the criteria we used to detect identifier cookies and cookie sharing.

2.1 Data collection

Two stateful crawls: We performed passive Web measurements using the OpenWPM platform [24]. It uses the Firefox browser, and provides browser automation by converting high-level commands into automated browser actions. We launched *two stateful crawls on two different machines with different IP addresses.* For each crawl, we used one browser instance and saved the state of the browser between websites. In fact, measurement of Web tracking techniques such as cookie syncing is based on re-using cookies stored in the browser, and hence it is captured more precisely in a stateful crawl.
Full dataset: We performed a stateful crawl of Alexa top $10,000$ domains in February 2019 in France [7] from two different machines. Due to the dynamic behavior of the websites, the content of a same page might differs every time this page is visited. To reduce the impact of this dynamic behavior and reduce the difference between the two crawls, we launched the two crawls at the same time. For each domain, we visited the home page and the first 10 links from 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 $10,000$ Alexa top domains, we successfully crawled $8,744$ domains with a total of $84,658$ 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 with a *content-length* less than 100 KB. We made this choice to save storage space. Moreover, in addition to HTTP requests, responses and cookies, we were only interested in the storage of invisible pixels. In our first dataset, named *full dataset*, we capture all HTTP requests, responses, and cookies.

Prevalence of invisible pixels: As a result of our crawl of 84,658 pages, we have collected 2,297,716 images detected using the field *content-type* in the HTTP header. We only stored images with a *content-length* less than 100 KB. These images represent 89.83% of the total number of delivered images. Even though we didn't store all the images, we were able to get the total number of delivered images using the content-type HTTP header extracted from the stored HTTP responses.

Figure 1 shows the distribution of the number of pixels in all collected images. We notice that invisible pixels (1×1 pixels and images with no content) represent 35.66% of the total number of collected images.

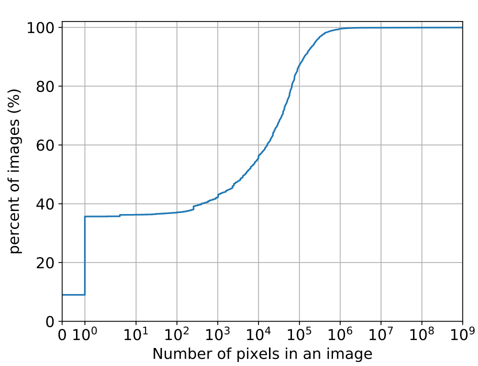


Fig. 1. Cumulative function of the number of pixels in images with a *content-length* less than 100 KB. 35.66% *of the images are invisible pixels,* 9.00% *have no content (they are shown as zero-pixel images), and* 26.66% *are of size* 1×1 *pixel.*

We found that out of 8,744 successfully crawled domains, 8,264 (94.51%) domains contain at least one page with one invisible pixel. By analyzing webpages independently, we found that 92.85% out of 84,658 visited 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. Moreover, the user is unaware of their existence. Hence, every invisible pixel represents a threat to the user privacy. We consider the set of requests and 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.

2.2 Detecting identifier cookies

Cookies are a classical way to track users in the Web. A key task to detect this kind of tracking is to be able to detect cookies used to store identifiers. We will refer to these cookies as *identifier cookies*. In order to detect identifier cookies, we analyzed data extracted from the two simultaneous crawls performed from two different machines. We refer to the owner of the cookie as host, and we define a cookie instance as (host, key, value).

We compare cookies instances between the two crawls: A tracker associates different identifiers to different users in order to distinguish them. Hence, an identifier cookie should be unique per user (user specific). We analyzed the 8,744 crawled websites where we have a total of 607,048 cookies instances belonging to 179,580 (host, key) pairs. If an identical cookie instance appears in the two crawls, that is, the host, key and value of both cookies are identical, we consider that the cookie is not used for tracking. We refer to such cookies as *safe cookies*. We extracted 108,252 safe cookies from our dataset. They represent 17.83% of the total number of cookies instances.

Due to the dynamic behavior of websites, not all cookies appear in both crawls. We mark the cookies (host, name) that appear only in one crawl as unknown cookies. In total, we found 15,386 unknown cookies (8.56%). We exclude these cookies from our study.

We don't consider the cookie lifetime: The lifetime of the cookie is used to detect identifier cookies in related works [1, 24, 25]. Only cookies that expire at least a month after being placed are considered as identifier cookies. In our study, we don't put any boundary on the cookie 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.

Detection of cookies with identifier cookie as key: We found that some domains store the identifier cookie as part of the cookie key. To detect this behavior,

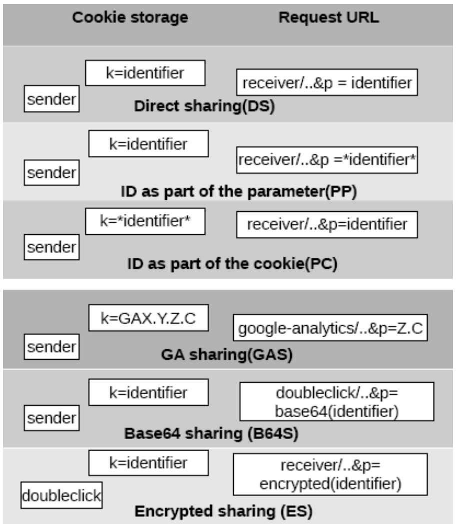


Fig. 2. Detecting identifier sharing. "sender" is the domain that owns the cookie and triggers the request, "receiver" is the domain that receives this request, and "identifier" is the *identifier cookie* value. "*" represents any string.

we analyzed the cookies with the same host and value and different keys across the two crawls. We found 5,295 (0.87%) cookies instances with identifier cookie as key. This behavior was performed by 966 different domains. Table 9 in Appendix presents the top 10 domains involved. The cookies with identifier cookie as key represent only 0.87% of the total number of cookies. Therefore, we will exclude them from our study.

2.3 Detecting identifier sharing

Third party trackers not only collect data about the users, but also exchange this data to build richer users profiles. Cookie syncing is a common technique used to exchange user identifiers stored in cookies. To detect such behaviors, we need 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 [45]. Therefore, trackers need to pass identifiers through the URL parameters.

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, we take inspiration from [1, 24]. We split cookies and URL parameter values using as delimiters any character not in [a-zA-Z0-9,'-','_','.']. Figure 2 shows six different techniques we deployed to detect identifier sharing. The first three methods are generic: either the identifier is sent as the parameter value, as part of the parameter value or it's stored as part of the cookie value and sent as parameter value.

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 prevalent in serving invisible pixels across websites (see Figure 13 in Appendix). 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 identifier 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 [19] we infer that encrypted cookie was shared(**Encrypted sharing**). For more details see the Section 11.1 in the Appendix.

2.4 Limitations

We detected six different techniques used to share the identifier cookie. However, trackers may encrypt the cookie before sharing it. In this work, we only detected encrypted cookies when it's shared following a specific semantic set by `doubleclick` [19].

We do not inspect the payload of POST requests that could be used to share the identifier cookie. For example, it's known that `google-analytics.com` sends the identifier cookie as part of the URL parameters with GET requests or in the payload of the POST requests [8] – we do not detect such a case in this work.

To detect the sender of the request in case of inclusion, we use the referer field. Therefore, we may miss to interpret who is the effective initiator of the request, it can be either the first party or an included script.

3 Overview of tracking behaviors

In Section 2.1, 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 the *invisible pixels dataset*.

In total, we have 747,816 third party requests leading to invisible images. By analyzing these requests, we detected 6 categories of tracking behaviors in 636,053 (85.05%) requests that lead to invisible images.

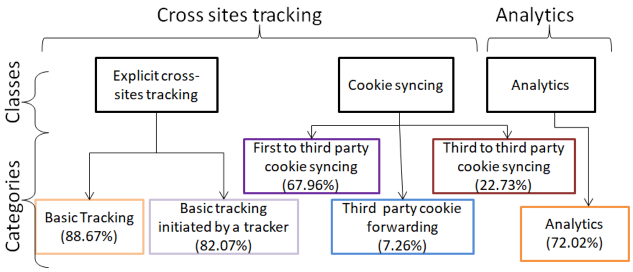


Fig. 3. Classification overview. *(%) represents the percentage of domains out of 8,744 where we detected the tracking behavior. A tracking behavior is performed in a domain if it's detected in at least one of its pages.*

We further group these categories into three main classes: explicit cross-site tracking (Section 4.1), cookie syncing (Section 4.2), and analytics (Section 4.3). In the following, we call BehaviorTrack our detection method of these behaviors.

After defining our classification using the *invisible pixels dataset*, we apply it on the *full dataset* where we have a total of 4,216,454 third-party requests collected from 84,658 pages on the 8,744 domains successfully crawled. By analyzing these requests, we detected 6 tracking behaviors in 2,724,020 (64.60%) requests.

Figure 3 presents an overview of all classes (black boxes) and categories of tracking behaviors and their prevalence in the full dataset. Out of 8,744 crawled domains, we identified at least one form of tracking in 91.92% domains. We further analyzed prevalence of each tracking category that we report in Section 4. We found out that *first to third party cookie syncing (see Sec. 4.2.3) appears on 67.96% of the domains!*

In addition, we analyzed the most prevalent domains involved in either cross-site tracking, analytics, or both behaviors. Figure 4 demonstrates that a third party domain may have several behaviors. For example, we detect that `google-analytics.com` exhibits both cross-site tracking and analytics behavior. This variance of behaviors is due to the web site developer, as it's the case for cookie syncing and analytics behaviors. It can also be due to the domain's partners as it's the case for cookie forwarding. Google-analytics in that case is included by another third party, the developer is not necessarily aware of this practice.

We found that not all the tracking detected in the *full dataset* is based on invisible pixels. We extracted the type of the content served by the tracking requests using the HTTP header *Content-Type*. Table 1 presents the top 5 types of content used for tracking. Out of the 2,724,020 requests involved in at least one tracking behavior in the full dataset, the top content delivered by tracking requests is scripts (34.36%), while the second most common content is invisible pixels (23.34%). We also detected other content used for tracking purposes such as visible images.

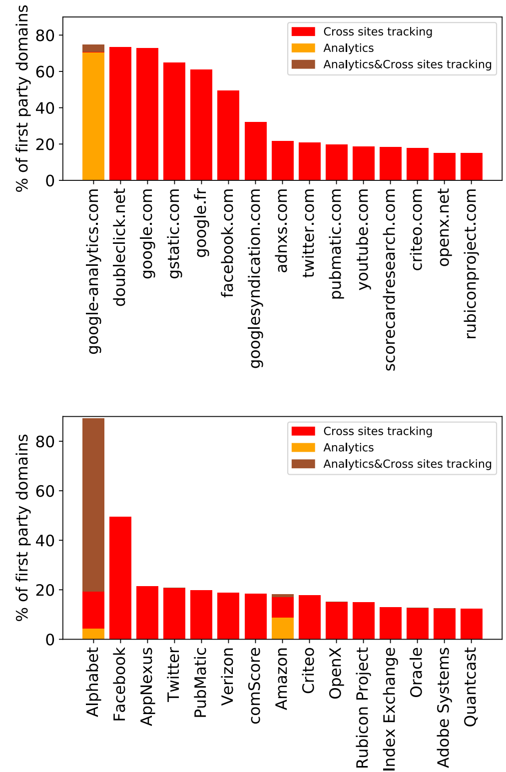


Fig. 4. Top 15 domains and companies involved in analytics, cross-site tracking, or both on the same first-party domain.

Content type	% requests
Script	34.36%
Invisible images	23.34 %
Text/html	20.01%
Big images	8.54 %
Application/json	4.32%

Table 1. Top 5 types of content used in the 2,724,020 third party tracking requests.

4 Classification of tracking

In this Section, we explain all the categories of tracking behaviors presented in Figure 3 that we have uncovered by studying the *invisible pixels* dataset. For each category, we start by explaining the tracking behavior, we then give its privacy impact on the user's privacy, and finally we present the results from the *full dataset*.

4.1 Explicit cross-site tracking

Explicit cross-site 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 4.2.

4.1.1 Basic tracking

Basic tracking is the most common tracking category 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 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`.

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

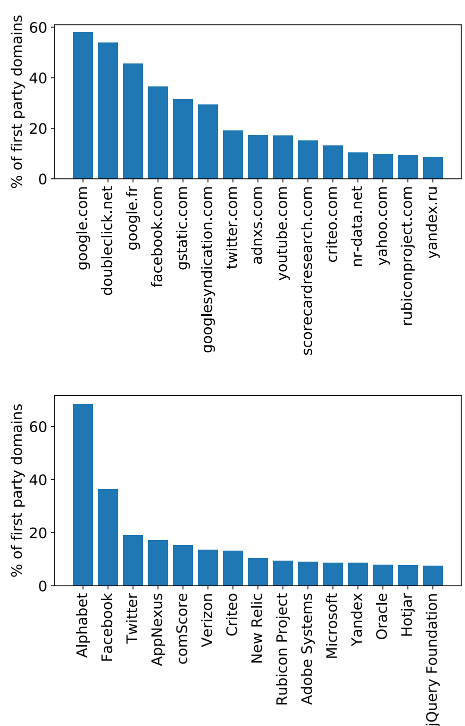


Fig. 5. Basic tracking: Top 15 cross-site trackers and companies in charge of the trackers included in $8,744$ domains.

Results: We detected basic tracking in 88.67% of visited domains. In total, we found $5,421$ distinct third parties performing basic tracking. Figure 5 shows the top domains involved in basic tracking. We found that `google.com` alone is tracking the user on over $5,079$ (58.08%) domains. This percentage becomes more important if we consider the company instead of the domain (Figure 5). By considering companies instead of domains, we found that, by only using the *basic tracking* `Alphabet` (the owner of `Google`) is tracking users in 68.30% of Alexa top 8K websites.

4.1.2 Basic tracking initiated by a tracker

When the user visits a website that includes content from a third party, the third party can redirect the request to a second third party tracker or include it. The second tracker 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 her.

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

Privacy 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 behaviors 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 have collected using basic tracking. Consequently, through *basic tracking initiated by a tracker*, trackers get to know the website visited by the users, without being included in it as long as this website includes one of the tracker's partners. Hence, through this tracking technique, the user's browsing history is shared instantly without syncing cookies.

Results: We detected Basic tracking initiated by a tracker in 82.07% of the domains. From Figure 6, we can notice that `google.com` is the top tracker included by other third parties. By only relying on its partners, without being directly included by the developer, `google.com` is included in over $5,374$ (61.45%) of the Alexa top 8k domains and its owner company `Alphabet` is included in over 71.56% of the visited domains. `Google.com` is included by 295 different third party trackers in our dataset. In our results, we found that `doubleclick.net` and `googlesyndication.com`, both owned by Google, are the top domains includ-

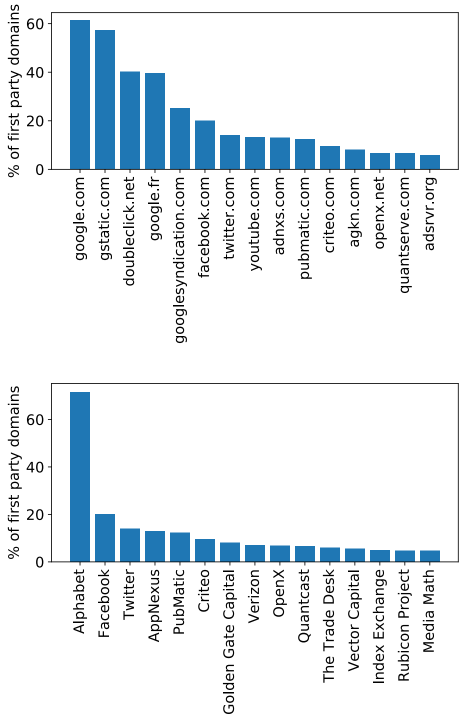


Fig. 6. Basic tracking initiated by a tracker: Top 15 trackers and companies included in $8,744$ domains.

Partners	# requests
pubmatic.com ↔ doubleclick.net	4,392
criteo.com ↔ doubleclick.net	2,258
googlesyndication.com ↔ adnxs.com	1,508
googlesyndication.com ↔ openx.net	1,344
adnxs.com ↔ doubleclick.net	1,199
rubiconproject.com ↔ googlesyndication.com	1,199
doubleclick.net ↔ yastatic.net	979
doubleclick.net ↔ demdex.net	790
adnxs.com ↔ amazon-adsystem.com	760
rfihub.com ↔ doubleclick.net	685

Table 2. Basic tracking initiated by a tracker: Top 10 pairs of partners from different companies that include each other. (↔) both ways inclusion.

ing each other (176,295 requests in our dataset). Table 2 presents the top 10 pairs of partners from different companies that are mutually including each other on websites. Note that in Table 2 we don't report mutual inclusion of domains that belong to the same company.

4.2 Cookie syncing

To create a more complete profile of the user, third party domains need to merge profiles they collected on different websites. One of the most known techniques to do so is cookie syncing. We separate the previously known technique of cookie syncing [1, 24] into two distinct categories, *third to third party cookie syncing* and *third party cookie forwarding*, because of their different privacy impact. We additionally detect a new type of cookie syncing that we call *first to third party cookies syncing.*

4.2.1 Third to third party cookie syncing

When two third parties have an identifier cookie in a user's browser and need to merge user profile, they use third to third party cookie syncing.

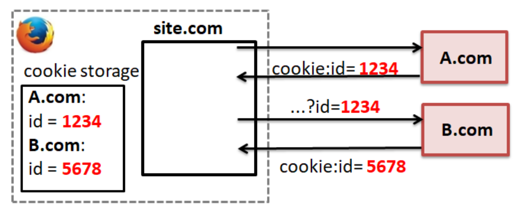


Fig. 7. Third to third party cookie syncing behavior.

Tracking explanation: Figure 7 demonstrates cookie syncing[2]. The first party domain includes a content 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 content, `A.com` decides to redirect to `B.com` and in the redirection request sent to `B.com`, `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 its cookie *id = 5678*, which will allow `B.com` to link the two identifiers to the same user.

Privacy impact: *Third to third party cookie syncing* is one of the most harmful tracking techniques that impact the user's privacy. In fact, third party cookie syncing can be seen as a set of trackers performing *basic tracking* and then exchanging the data they collected about the user. It's true that a cross sites 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 does 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 its partners are included. What makes this

2 Notice that in figures that explain the tracking behaviors, 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.

practice even more harmful is when a third party has several partners with whom it syncs cookies. One example of such behavior is rubiconproject.com, that syncs its identifier cookie with 7 partners: tapad.com, openx.net, imrworldwide.com, spotxchange.com, casalemedia.com, pubmatic.com and bidswitch.net.

Partners	# requests	Sharing technique
adnxs.com → criteo.com	1,962	→DS
doubleclick.net → facebook.com	789	→DS
casalemedia.com → adsrvr.org	778	→DS
mathtag.com↔adnxs.com	453	→DS
pubmatic.com → lijit.com	321	→DS
adobedtm.com → facebook.com	269	→DS
doubleclick.net ↔ criteo.com	250	→ DC, PCS; ← DS
mmstat.com → cnzz.com	233	→ DC
sharethis.com → agkn.com	233	→ DC
mathtag.com → lijit.com	109	→ DC

Table 3. Third to third party cookie syncing: Top 10 partners. The arrows represent the flow of the cookie synchronization, (→) one way matching or (↔) both ways matching. DS (Direct 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 22.73% websites. We present in Table 3 the top 10 partners that we detect as performing cookie syncing. In total, we have detected 1,263 unique partners performing cookie syncing. The syncing could be done in both ways, as it's the case for doubleclick.net and criteo.com, or in one way, as it's the case for adnxs.com and criteo.com. 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 party cookie syncing that involves a large variety of sharing techniques.

4.2.2 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. Cookie forwarding has always been called "syncing" while instead it simply enables a third party to reuse an identifier of a tracker, without actually syncing its own identifier.

Tracking explanation: The first party domain site.com includes A.com's content. To get the image, a request is sent to A.com along with its cookie. A.com then redirects the request to its partner (B.com) and sends

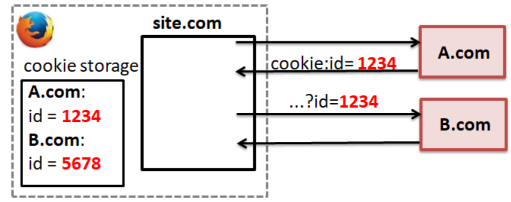


Fig. 8. Third party cookie forwarding behavior.

the identifier cookie it associated to the user (*1234*) as part of the URL parameters (Figure 8).

Third party cookie forwarding differs from *Third to third party cookie syncing* depending on whether there 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 works [43] [31], in the sense that we have a collaboration of third-party advertisers. However, in our study we check that the second tracker do not use its own cookie to identify the user. This means that this tracker (B.com) is 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.

Privacy impact: *Third party cookie forwarding* allows trackers to instantly share the browsing history of the user. A.com in Figure 8 does not only associate an identifier cookie to the user, but it also redirect and shares 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 passively receives the user's browsing history from the first tracker.

Results: We detected third party cookie forwarding in 7.26% of visited websites. To our surprise, the top domain receiving identifier cookie from third parties is google-analytics.com (Figure 14 in Appendix). Google-analytics.com is normally included by domains owners to get analytics of their websites, it's known as a *within domain tracker*. But in this case, google-analytics.com is used by the third party domains. The third party is forwarding its third party cookie to google-analytics.com on different websites, consequently google-analytics.com in this case is tracking the user across websites. This behavior was discovered by Roesner *et al.* [43]. They reported this behavior in only a few instances, but in our dataset we found 386 unique partners that forward cookies, among which 271 are forwarding cookies to google-analytics.com. In Table 10 (Appendix), we present the top 10 third parties forwarding cookies to google-analytics service.

4.2.3 First to third party cookie syncing

In this category, we detect that first party cookie get synced with third party domain.

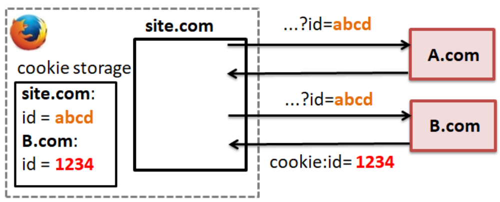


Fig. 9. First to third party cookie syncing behavior.

Tracking explanation: Figure 9 demonstrates the cookie syncing of the first party cookie. The first party domain `site.com` includes a content from `A.com?id=abcd`, where `A.com` is a third party and *abcd* is the first party identifier cookie of the user set for `site.com`. `A.com` receives the first party cookie *abcd* in the URL parameters, and then redirects the request to `B.com`. As part of the request redirected to `B.com`, `A.com` includes the first party identifier cookie. `B.com` sets its own identifier cookie *1234* in the user's browser. Using these two identifiers (the first party's identifier *abcd* received in the URL parameters and its own identifier *1234* sent in the cookie), `B.com` can create a matching table that allows `B.com` to link both identifiers to the same user.

The first party cookie can also be shared directly by the first party service (imagine Figure 9 where `A.com` is absent). In that case, `site.com` includes content from `B.com` and as part of the request sent to `B.com`, `site.com` sends the first party identifier cookie *1234*. `B.com` sets its own identifier cookie *1234* in the user's browser. `B.com` can now link the two identifiers to the same user.

Privacy impact: In our study, we differentiate the case when cookie shared is a first party cookie and when it is a third party cookie. We made this distinction because, the kind and the sensitivity of the data shared differs in the two cases. Using this tracking technique, first party websites get to sync cookies with third parties. Moreover, pure analytic services allow to sync in-site history with cross-site history.

Results: We detected first to third party cookie syncing in 67.96% of visited domains. In Table 4, we present the top 10 partners syncing first party cookies. We differentiate the two cases: (1) first party cookie synced through an intermediate service (as shown in Figure 9) and (2) first party cookie synced directly from the first party domain. In total we found

Partners	# requests
First party cookie synced through an intermediate service	
google-analytics.com → doubleclick.net	8,297
Direct First to third party cookie syncing	
hibapress.com → criteo.com	460
alleng.org → yandex.ru	332
arstechnica.com → condenastdigital.com	243
thewindowsclub.com → doubleclick.net	228
digit.in → doubleclick.net	224
misionesonline.net → doubleclick.net	221
wired.com → condenastdigital.com	219
newyorker.com → condenastdigital.com	218
uol.com.br → tailtarget.com	198

Table 4. First to third party cookie syncing: Top 10 partners.

17,415 different partners involved. The top partners are `google-analytics.com` and `doubleclick.net`. We found that `google-analytics.com` first receives the cookie as part of the URL parameters. Then, through a redirection process, `google-analytics.com` transfers the first party cookie to `doubleclick.net` that inserts or receives an identifier cookie in the user's browser. We found out that `google-analytics.com` is triggering such first party cookie syncing on 38.91% of visited websites.

4.3 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.

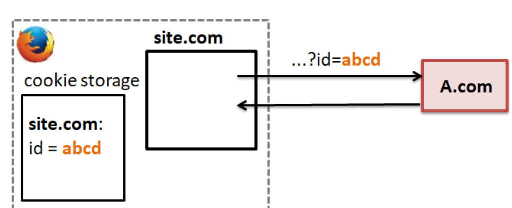


Fig. 10. Analytics behavior.

Tracking explanation: Figure 10 shows the analytics category where the domain directly visited by the user (`site.com`) owns a cookie containing a unique identifier in the user's browser. Such cookie is called a first 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 content and uses this request to share the first party identifier cookie.

Privacy impact: In analytic behavior, the third party domain is not able to track the user across websites because it does not set its own cookie in the 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. The analytics service 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.

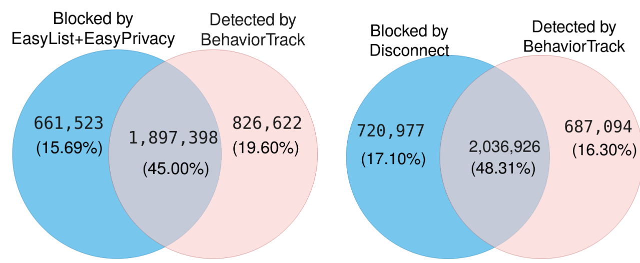
Results: We detected analytics in 72.02% of the visited domains. We detect that google-analytics.com is the most common analytics service. It's used on 69.25% of the websites. The next most popular analytics is alexametrics.com, it's prevalent on 9.10% of the websites (see Figure 15 in the Appendix).

5 Are filter lists effective at detecting trackers?

Most of the state-of-the-art works that aim at measuring trackers at large scale rely on *filter lists*. In particular, EasyList [20], EasyPrivacy [21] and Disconnect [17] lists became the *de facto* approach to detect third-party tracking requests in the privacy and measurement communities [10–12, 23, 24, 28–30, 42]. Nevertheless, filter lists detect only known tracking and ad-related requests, therefore a tracker can easily avoid this detection by registering a new domain. Third parties can also incorporate tracking behavior into functional website content, which could not be blocked by filter lists because blocking functional content would harm user experience. Therefore, it is interesting to evaluate how effective are filter lists at detecting trackers, how many trackers are missed by the research community in their studies, and whether filter lists should still be used as the *default tools* to detect trackers at scale.

In this Section, we analyze how effective are filter lists at detecting third-party trackers. Contrary to Merzdovnik et al.'s work [35], which measured blocking of third party requests without identifying whether such requests are tracking or not, we compare all the cross-site tracking and analytics behavior reported in Section 4 (that we unite under one detection method, that we call BehaviorTrack) with the third-party trackers detected by filter lists. EasyList and EasyPrivacy (EL&EP) and Disconnect filter lists in our comparison were extracted in April 2019. We use the Python library *adblockparser* [4], to determine if a request would have been blocked by EL&EP. For Disconnect we compare to the domain name of the requests (the Disconnect list contains full domain names, while EL&EP are lists of regular expressions that require parsing).

For the comparison, we used the *full dataset* of 4,216,454 third party requests collected from 84,658 pages of 8,744 successfully crawled domains.



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

Fig. 11. Effectiveness of filter lists at detecting trackers on 4,216,454 third party requests from 84,658 pages.

Measuring tracking requests We apply filter lists on requests to detect which requests are blocked by the lists, as it has been done in previous works [24]. We then use the filter lists to classify follow-up third-party requests that would have been blocked by the lists. This technique has been extensively used in the previous works [23, 28–30] (for more details, see Table 12 in the Appendix). We classify a request as blocked if it matches one of the conditions:
- the request directly matches the list.
- the request is a consequence of a redirection chain where an earlier request was blocked by the list.
- the request is loaded in a third-party content (an iframe) that was blocked by the list (we detect this case by analyzing the referrer header).

Figure 11 provides an overview of third party requests blocked by filter lists or detected as tracking requests according to BehaviorTrack. Out of all 4,216,454 third party requests in the *full dataset*, 2,558,921 (60.7%) requests were blocked by EL&EP, 2,757,903 (65.4%) were blocked by Disconnect, and 2,724,020 (64.6%) were detected as performing tracking by BehaviorTrack.

Requests blocked only by filter lists: Figure 11 shows that EL&EP block 661,523 (15.69% out

Filter list(s)	# missed requests	% of 4.2M third-party requests	% of 2.7M tracking requests	# domains responsible for missed requests	# trackers follow up	# effective missed tracking requests
EL&EP	$826,622$	19.60%	25.22%	$5,136$	$118,314$	$708,308$
Disconnect	$687,094$	16.30%	30.34%	$6,189$	$46,285$	$640,809$

Table 5. Overview of third-party requests missed by the filter lists and detected as tracking by BehaviorTrack.

of $4,216,454$) requests that were not detected as performing tracking by BehaviorTrack. These requests originate from $2,121$ unique third party domains. Disconnect blocks $720,977$ (17.10%) requests not detected by BehaviorTrack. These requests originate from $1,754$ distinct third party domains.

These requests are missed by BehaviorTrack because they do not contain any identifier cookie. Such requests may contain other non user-specific cookies (identical across two machines, see Sec. 2.2), however we assume that such cookies are not used for tracking. EL&EP and Disconnect block these requests most likely because they are known for providing analytics or advertising services, or because they perform other types of tracking through scripts such as fingerprinting, which is out of the scope of our study.

5.1 Tracking missed by the filter lists

Table 5 gives an overview of third-party requests missed by EL&EP and Disconnect filter lists and detected by BehaviorTrack as performing tracking. The number of third party domains involved in tracking detected only by BehaviorTrack (e.g., $6,189$ for Disconnect) is significantly higher than those only detected by filter lists (e.g., $1,754$ for Disconnect as reported earlier in this section). We define the term *trackers follow up* as the requests using identifying cookies set by previous requests blocked by the filter lists (note that our crawler is stateful). As a result, by simulating the blocking behavior of the filter lists, these cookies should be blocked and not included in the analysis of the following requests. Consequently, the follow up requests should not be categorized as tracking requests.
By further analyzing the requests only detected as tracking by BehaviorTrack and missed by EL&EP, we found that $118,314$ requests (14.31% of the requests detected only by BehaviorTrack) are trackers follow up. Similarly, we found that $46,285$ requests (6.73% of the requests detected only by BehaviorTrack) missed by Disconnect are trackers follow up. We exclude these requests from the following analysis and we further analyze the remaining $708,308$ requests missed by EL&EP and the $640,809$ missed by Disconnect.

BehaviorTrack detects all kind of trackers including the less popular ones that are under the bar of detection of filter list. Because less popular trackers are less prevalent, they generate fewer requests and therefore remain unnoticed by filter lists. This is the reason why we detect a large number of domains responsible for tracking.

5.1.1 Tracking enabled by useful content

Content type	Missed by EL&EP	Missed by Disconnect
script	33.38%	35.27 %
big images	20.62%	21.73 %
text/html	13.77%	14.73 %
font	8.79%	0.09 %
invisible images	6.68%	12.21 %
stylesheet	6.17%	3.05 %
application/json	4.00%	4.83 %
others	6.59%	8.12%

Table 6. Top content type detected by BehaviorTrack and not by filter lists on the $708,308$ requests missed by EL&EP and the $640,809$ missed by Disconnect

We analyzed the type of content provided by the remaining tracking requests. Table 6 presents the top content types used for tracking and not blocked by the filter lists. We refer to images with dimensions larger than 50×50 pixels as Big images. These kinds of images, texts, fonts and even stylesheets are used for tracking. The use of these types of contents is essential for the proper functioning of the website. That makes the blocking of responsible requests by the filter lists impossible. In fact, the lists are explicitly allowing content from some of these trackers to avoid the breakage of the website, as it's the case for `cse.google.com`.

We categorized the top 30 third party services not blocked by the filter lists but detected by BehaviorTrack as performing tracking using Symantec's WebPulse Site Review [14]. Unlike in previous sections, where we analyzed the 2^{nd}-level TLD, such as `google.com`, here we report on full domain names, such as `cse.google.com`.

Service category	EL&EP	Disconnect
Content Servers	23.33 %	23.33 %
Social Networking	16.67 %	0.00%
Web Ads/Analytics	13.33 %	23.33 %
Search Engines/Portals	13.33 %	23.33 %
Technology/Internet	13.33 %	10.00 %
Consent frameworks	3.33 %	3.33 %
Travel	3.33 %	3.33 %
Non Viewable/Infrastructure	3.33 %	0.00%
Shopping	3.33 %	3.33 %
Business/Economy	3.33 %	6.67 %
Audio/Video Clips	3.33 %	0.00%
Suspicious	0.00%	3.33 %

Table 7. Categories of the top 30 tracking services detected by BehaviorTrack and missed by the filter lists.

That gives more information about the service provided. New domains such as `consensu.org` are not categorized properly so we manually added a new category called "Consent frameworks" to our categorization for such services. Table 7 represents the results of this categorization. Web Ads/Analytics represents 13.33% of the services missed by EL&EP and 23.33% of those missed by Disconnect. However, the remaining services are mainly categorized as content servers, search engines and other functional categories. They are tracking the user, but not blocked by the lists. This is most likely not to break the websites.

5.1.2 Why useful content is tracking the user

Tracking enabled by a first party cookie: A cookie set in the first party context can be considered as a third party cookie in a different context. For example, a `site.com` cookie is a first party cookie when the user is visiting `site.com`, but it becomes a third party when the user is visiting a different website that includes content from `site.com`. Whenever a request is sent to a domain, say `site.com`, the browser automatically attaches all the cookies that are labeled with `site.com` to this request.

For example, when a user visits `google.com`, a first party identifier cookie is set. Later on, when a user visits `w3school.com`, a request is sent to the service `cse.google.com` (Custom Search Engine by Google). Along with the request, Google's identifier cookie is sent to `cse.google.com`. The filter list cannot block such a request, and is incapable of removing the first party tracking cookies from it. In our example, filter lists do not block the requests sent to `cse.google.com` on 329 different websites. In fact, blocking `cse.google.com` breaks the functionality of the website. Consequently, an identifier cookie is sent to the `cse.google.com`, allowing it to track the user across websites.

By analyzing the requests missed by the lists, we found that this behavior explains a significant amount of missed requests: 44.61% requests ($316,008$ out of $708,308$) missed by EL&EP and 32.00% requests ($205,088$ out of $640,809$) missed by Disconnect contain cookies initially set in a first party context.

Tracking enabled by large scope cookies. A cookie set with a 2^{nd}-level TLD domain can be accessed by all its subdmains. For example, a third party `sub.tracker.com` sets a cookie in the user browser with `tracker.com` as its domain. The browser sends this cookie to another subdomain of `tracker.com` whenever a request to that subdomain is made. As a result of this practice, the identifier cookie set by a tracking subdomain with 2^{nd}-level TLD domain is sent to all other subdomains, even the ones serving useful content.

Large scope cookies are extremely prevalent among requests missed by the filter lists. By analyzing the requests missed by the lists, we found that 77.08% out of 22,606 third-party cookies used in the requests missed by EL&EP and 75.41% out of 24,934 cookies used in requests missed by Disconnect were set with a 2^{nd}-level TLD domain (such as `tracker.com`).

5.2 Panorama of missed trackers

To study the effectiveness of EL&EP and Disconnect combined, we compare requests blocked by these filter lists with requests detected by BehaviorTrack as tracking according to the classification from Figure 3. These results are based on the dataset of $4,216,454$ third-party requests collected from $84,658$ pages of $8,744$ domains.

Overall, $379,245$ requests originating from 9,342 services (full third-party domains) detected by BehaviorTrack are not blocked by EL&EP and Disconnect. Yet, these requests are performing at least one type of tracking, they represent 9.00% of all $4,2$M third-party requests and appear in 68.70% of websites.

We have detected that the $379,245$ requests detected by BehaviorTrack perform at least one of the tracking behaviors presented in Figure 3. Table 8 shows the distribution of tracking behaviors detected by BehaviorTrack. We notice that the most privacy-violating behavior that includes setting, sending or syncing third-party cookies is represented by the basic tracking that is present in (83.90%) of missed requests.

Table 11 in Appendix presents the top 15 domains detected as trackers and missed by the filter lists. For

Tracking behavior	Prevalence
Basic tracking	83.90%
Basic tracking initiated by a tracker	13.50%
First to third party cookie syncing	1.42%
Analytics	1.00%
Third to third party cookie syncing	0.09%
Third party cookie forwarding	0.08%

Table 8. Distribution of tracking behaviors in the $379,245$ requests missed by EL&EP and Disconnect.

each domain, we extract its category, owners and country of registration using the whois library [48] and manual checks. We also manually analyzed all the cookies associated to tracking: out of the 15 presented domains, 7 are tracking the user using persistent first party cookies. The cookies of the search engine Baidu expires within 68 years, whereas the cookies associated to Qualtrics, an experience management company, expires in 100 years.

We found that content from `code.jquery.com`, `s3.amazonaws.com`, and `cse.google.com` are explicitly allowed by the filter lists on a list of predefined first-party websites to avoid the breakage of these websites. We identified `static.quantcast.mgr.consensu.org` by IAB Europe that rightfully should not be blocked because they provide useful functionality for GDPR compliance. We detect that the cookie values seemed to be unique identifiers, but are set without expiration date, which means such cookies will get deleted when the user closes her browser. Nevertheless, it is 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 (remember that we did not emit any user behavior, like clicking on buttons or links during our crawls).

We identified tag managers – these tools are designed to help Web developers to manage marketing and tracking tags on their websites and can't 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 have an explicit exception in EasyList.

6 Are browser extensions effective at blocking trackers?

In this Section, we analyze how effective are the popular privacy protection extensions in blocking the privacy leaks detected by BehaviorTrack. We study the following extensions: Adblock [3], Ghostery [27], Disconnect [16],

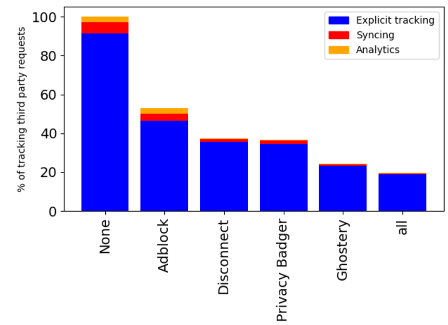


Fig. 12. Percentage of third party requests allowed by privacy protecting browser extensions out of 2,924,480 tracking requests.

and Privacy Badger [41]. The latest version of uBlock Origin 1.22.2 is not working correctly with OpenWPM under Firefox 52, which is the latest version of Firefox running on OpenWPM that supports both web extensions and stateful crawling. Hence we didn't include uBlock Origin in our study.

We performed simultaneous stateful Web measurements of the Alexa top 10K websites using OpenWPM in November 2019 from servers located in France. For each website, we visit the homepage and 2 randomly chosen links on the homepage from the same domain. Selection of links was made in advance.

We consider the following measurement scenarios:

1. Firefox with no extension.
2. Firefox with Adblock 3.33.0 (default settings).
3. Firefox with Ghostery 8.3.4 (activated blocking).
4. Firefox with Disconnect 5.19.3 (default settings).
5. Firefox with Privacy Badger 2019.7.11 (trained on the homepage and 2 random links from this homepage for the top 1,000 Alexa websites).
6. Firefox with all previous extensions combined.

Out of $30,000$ crawled pages, $25,485$ were successfully loaded by all the crawls. The analysis in the following is done on this set of pages.

Figure 12 represents the effectiveness of the extensions in blocking the tracking requests detected by BehaviorTrack. Our results show that Ghostery is the most efficient among them. However, it still fails to block 24.38% of the tracking requests. All extensions miss trackers in the three classes, However, Disconnect and Privacy Badger have an efficient Analytics blocking mechanism: they are missing Analytics behavior on only 0.36% and 0.27% of the pages respectively. Most tracking requests missed by the extensions are performing Explicit tracking.

Conclusion: Similarly to Merzdovnik et al. [35], we show that tracker blockers (Disconnect, Ghostery and

Badger) are more efficient than adblockers (Adblock) in blocking tracking behaviors. However, all studied extensions miss at least 24.38% of the tracking detected by BehaviorTrack. This shows that even though the extensions reduce the amount of tracking performed, they do not solve the problem of protecting users from tracking.

7 Discussion

Our results show that there are numerous problems in the cookie-based third party tracking. In this Section, we discuss these problems with respect to different actors.

Browser vendors. We observed that first party cookies can be exploited in a third party context to perform cross site tracking. In its Intelligent Tracking Prevention 2.0 introduced in 2018, Safari allowed cookies to be used in a third party context only in the first 24 hours of the cookie lifetime. Such time frame could be limited even further, however this approach requires rigorous testing with end users. Other browser vendors should follow Safari and prohibit the usage of cookies in a third party context.

Web standardization organizations. While third-party content provides useful features to the website, it is also capable of tracking users. We have shown that third party domains serving functional content such as Content Servers or Search Engines may track the user with identifier cookies. We have noticed that we detect such tracking because the domain behind such functional content does not set but only receives identifier cookies that are already present in the browser and were initially set with the 2^{nd}-level domain as host, which makes the cookie accessible by all subdomains. Even if the tracking is not intentional, and the domain is not using the identifier cookie it receives to create user's profile currently, this cookie leakage is still a privacy concern that could be exploited by the service anytime. We therefore believe that Web standardization bodies, such as the W3C, could propose to limit the scope of the cookies and not send it to all the subdomains.

Supervisory bodies. When a supervisory body, such as a Data Protection Authority in the EU, has to investigate and find the responsible party for the tracking happening on a website, it is a very complex task to identify who is liable for setting or sending the identifier cookie. In our work, we have identified *tracking initiators* – third party domains that only redirect or include other domains that perform tracking. Such tracking initiators, that we detected on 11.24% of websites, are partially liable for tracking. Another example are CDNs, we have observed that requests or responses for fetching a jQuery library from `code.jquery.com` contains identifier cookies. We found that it is the Cloudflare CDN that inserts a cookie named `__cfduid` into its traffic in order "to identify malicious visitors to their Customers' websites".

Conclusion. Our work raises numerous concerns in the area of tracking detection and privacy protection of Web users. We believe that our work can be used to improve existing tracking detection approaches, but nevertheless various actors need to revise their practices when it comes to the scope and usage of cookies, and third parties should exclude third party tracking from the delivery of functional website content.

8 Related Work

In this Section, we first provide an overview of previous works on measuring invisible pixels. We then examine state-of-the-art techniques to detect online tracking: behavior-based techniques and methods leveraging the filter lists.

8.1 Invisible pixels, known as web bugs

Invisible pixels are extensively studied starting from 2001 [6, 18, 34, 36, 44]. Invisible pixels, called "web bugs" in previous works, were primarily used to set and send third-party cookies attached to the request or response when the browser fetches such image. In 2003, Martin et al. [34] found that 58% of the 84 most popular websites and 36% of 289 random websites contain at least one web bug. In 2002, Alsaid and Martin [6] deployed a tool (*Bugnosis*) to detect the web bugs. The main goal of the tool was to raise awareness among the public. It was used by more than 100,000 users. However, it was only generating warning messages without actively blocking the bugs and was only supported by Internet Explorer 5, that is deprecated today. Dobias [18] showed that web bugs lead to new privacy threats, such as fingerprinting.

Ruohonen and Leppänen [44] studied the presence of invisible pixels in Alexa's top 500 websites. They showed that invisible pixels are still widely used. 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. Such a method misses an important number of images that are dynamically loaded.

The significant number of studies on invisible pixels shows that it is a well known problem. The goal of our study is different: *we aim to use invisible pixels that are still widely present on the Web to detect different tracking behaviors and collaborations.*

8.2 Detection of online tracking

Detection of trackers by analysing behavior: Roesner et al. [43] and Lerner et al. [31] were the first to analyze trackers based on their behavior. They have proposed a classification of tracking behaviors that makes a distinction between analytics and cross-domain tracking. We, however, 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 Section 3). Yu et al. [49] identify trackers by detecting unsafe data without taking into account the behavior of the third party domain and the communications between trackers.

Previous studies [1, 11, 24, 38, 39] measured cookie syncing on websites and users.

Olejnik et al. [38] considered cookies with sufficiently long values to be identifiers. If such identifier is shared between domains, then it is classified as cookie syncing. Additionally, Olejnik et al. [38] studied the case of `doubleclick.net` to detect cookie syncing based on the URL patterns. In our study, we show that domains are using more complex techniques to store and share identifier cookies. We base our technique for detecting identifier cookies on the work of Acar et al. [1], and Englehardt and Narayanan [24], who only checked for the identifiers that are stored and shared in a clear text. In our work, we detect more cases of cookie synchronization because we detect encoded cookies and even encrypted ones in the case of `doubleclick.net`. Bashir et al. [11], used retargeted ads to detect cookie syncing. To detect these ads, authors filtered out all images with dimensions lower than 50×50 pixels, then they studied the information flow leading to these images. Which limit their study to chains resulting to a retargeted ad. In our work, we analyse all kind of requests.

Papadopoulos et al. [39] used a year-long dataset from real mobile users to study cookie syncing. The authors detect not only syncing done through clear text, but encrypted cookie syncing as well. Hence, they cover DS, PC and ES sharing techniques detected by BehaviorTrack (see Figure 2), but they miss the remaining techniques that represent 39.03% of the cookie sharing that we detect. Moreover, they only focus on cookie syncing, while we conduct a more in-depth study of different tracking behaviors extracted from invisible pixels dataset, and we compare our tracking detection to filter lists and the most popular privacy extensions.

Detection of trackers with filter lists: To detect domains related to tracking or advertisement, most of the previous studies [10–12, 23, 24, 28–30, 42] rely on filter lists, such as EasyList [20] and EasyPrivasy [21] (EL&EP) that became the *de facto* approach to detect trackers. From the last three years alone, we identified *9 papers that rely on EL&EP* to detect third-party tracking and advertising (see Table 12 in the Appendix).

Englehardt and Narayanan [24] seminal work on measuring trackers on 1 million websites relies on EL&EP as a ground truth to detect requests sent to trackers and ad-related domains. Three papers by Bashir et al. [10–12] customize EL&EP to detect 2^{nd}-level domains of tracking and ad companies: to eliminate false positives, a domain is considered if it appears more than 10% of the time in the dataset. Lauinger et al. [30] 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. [42] use EasyList as an input to their classifier to identify advertising and tracking domains in Web and mobile ecosystems. Ikram et al. [28] analysed how many tracking JavaScript libraries are blocked by EL&EP on 95 websites. Englehardt et al. [23] apply EL&EP on third-party leaks caused by invisible images in emails. Iordanou et al. [29] rely on EL&EP as a ground truth for detecting ad- and tracking-related third party requests. Only one work by Papadopoulos et al. [40] uses Disconnect list [17] to detect tracking domains. To the best of our knowledge, *we are the first to compare the behavior-based detection method to filter lists extensively used in the literature.*

Effectiveness of filter lists: Merzdovnik et al [35] studied the effectiveness of the most popular tracking blocking extensions. They evaluate how many third party requests are blocked by each extension. In their evaluation, they don't distinguish tracking third party requests from non tracking ones, which affects their evaluation. In our work, we detect trackers using a behavior-based detection method and then we evaluate how much of these trackers are blocked. Das et.al [15] studied the effectiveness of filter lists against tracking scripts that misuse sensors on mobile. They show that filter lists fail to block the scripts that access the sensors. We instead evaluate effectiveness of filter lists against third party requests in web applications that contain identifier cookies.

9 Conclusion

Web tracking remains an important problem for the 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, 91.92% of websites incorporate at least one type of cookie-based tracking.

In this paper, we define a new classification of Web tracking behaviors, thanks to a large scale study of invisible pixels collected from $84,658$ 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 and browser extensions fail to detect these complex behaviors. Therefore, behavior-based tracking detection should be more widely adopted.

10 Acknowledgment

First of all, we would like to thank the valuable comments and suggestions of the anonymous reviewers of our paper. This work is supported by ANSWER project PIA FSN2 (P159564-266178\DOS0060094).

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 offcial website. https://getadblock.com/.

[4] Adblock list parser. https://github.com/scrapinghub/adblockparser.

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

[6] David Martin Adil Alsaid. Detecting web bugs with bugnosis: Privacy advocacy through education. In *Privacy Enhancing Technologies*, 2002.

[7] Alexa official website. https://www.alexa.com/.

[8] Google-analytics service. https://developers.google.com/analytics/devguides/collection/protocol/v1/devguide.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Symantec categorization service. http://sitereview.bluecoat.com/#/.

[15] Anupam Das, Gunes Acar, Nikita Borisov, and Amogh Pradeep. The web's sixth sense: A study of scripts accessing smartphone sensors. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 2018.

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

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

[18] Jaromir Dobias. Privacy effects of web bugs amplified by web 2.0. In *Privacy and Identity Management for Life - 6th IFIP WG 9.2, 9.6/11.7, 11.4, 11.6/PrimeLife International Summer School, Helsingborg, Sweden, August 2-6, 2010, Revised Selected Papers*, pages 244–257, 2010.

[19] Doubleclick cookie syncing. https://developers.google.com/ad-exchange/rtb/cookie-guide.

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

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

[22] 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.

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

[24] 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.

[25] 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.

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

[27] Ghostery Official website. https://www.ghostery.com/.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] David Martin, Hailin Wu, and Adil Alsaid. Hidden surveillance by web sites: Web bugs in contemporary use. 2003.

[35] Georg Merzdovnik, Markus Huber, Damjan Buhov, Nick Nikiforakis, Sebastian Neuner, Martin Schmiedecker, and Edgar Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *2nd IEEE European Symposium on Security and Privacy*, Paris, France, 2017. To appear.

[36] Steve Nichols. Big brother is watching: An update on web bugs. In *SANS Institute*, 2001.

[37] 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.

[38] 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.

[39] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 1432–1442, 2019.

[40] 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.

[41] Privacy Badger Official website - Electronic Frontier Foundation. https://www.eff.org/privacybadger.

[42] 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.

[43] 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.

[44] 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.

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

[46] 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.

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

[48] Whois library. https://pypi.org/project/whois/.

[49] 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.

11 Appendix

11.1 Detecting identifier sharing

GA sharing: `Google-analytics` serves invisible pixels on 69.89% of crawled domains as we show in Figure 13. 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 it [19]. 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 it, which makes the detection of the identifier cookie sharing impossible. Instead, we rely on the semantic defined by doubleclick to share this identifier [19].

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 URL that contains `doubleclick.net` as destination and a parameter *google_nid*. Using the value of the *google_nid* parameter, `doubleclick.net` get to know 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.

11.2 Additional results

Figure 13 represents the Top 20 domains involved in invisible pixels inclusion in the 8,744 domains.

Table 9 presents the top 10 domains using their cookie key to store the identifier.

Figure 14 represents the Top 15 third parties receiving the identifier cookies. Google-analytics is the top domain receiving identifiers in over 4% of the visited websites. Table 10 presents the top 10 third parties sharing their identifiers with google-analytics.

Figure 15 presents the top 15 analytics domains in our dataset of 8,744 domains.

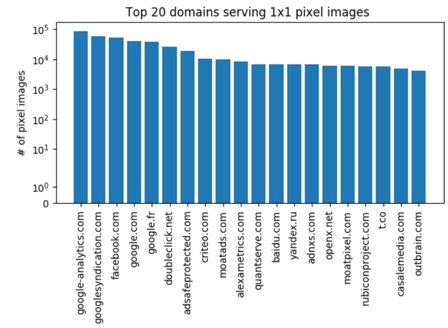


Fig. 13. Top 20 domains responsible for serving invisible pixels

Host	# cookies instances
lpsnmedia.net	583
i-mobile.co.jp	223
rubiconproject.com	83
justpremium.com	72
juicyads.com	64
kinoafisha.info	64
aktualne.cz	63
maximonline.ru	61
sexad.net	47
russian7.ru	45

Table 9. Top 10 domains storing the identifier as key.

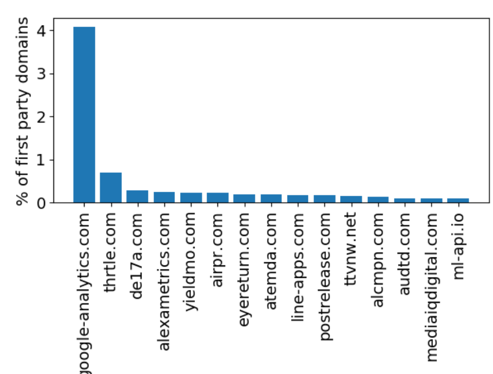


Fig. 14. Third party cookie forwarding : Top 15 receivers in 8,744 domains.

Table 11 presents the top 15 domains detected as trackers and missed by the filter lists. For each domain, we extract its category, owners and country of registration

Table 12 summarizes the usage of EL&EP lists in the previous works that we describe in Section 8.

Third parties	# requests
adtrue.com	298
google.com	123
architonic.com	120
bidgear.com	80
akc.tv	76
insticator.com	73
coinad.com	64
performgroup.com	52
chaturbate.com	47
2mdnsys.com	40

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

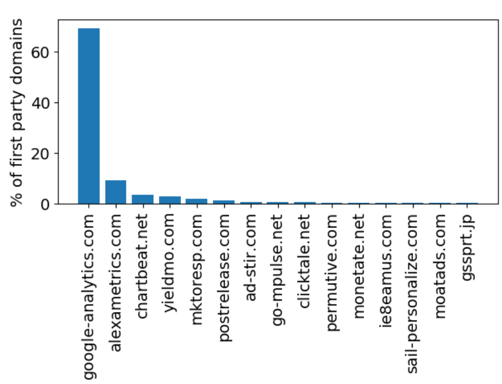


Fig. 15. Analytics: Top 15 receivers in the $8,744$ domains.

Full domain	Prevalence of tracking in first-parties	Cookie name	Cookie expiration	Category	Company	Country
Tracking enabled by a first party cookie						
code.jquery.com	756 (8.65 %)	___cfduid	1 years	Technology/Internet	jQuery Foundation	US
s3.amazonaws.com	412 (4.71 %)	s_fid	5 years	Content Servers	Amazon	US
ampcid.google.com	282 (3.23 %)	NID	6 months	Search Engines	Google LLC	US
cse.google.com	307 (3.51 %)	NID	1 year	Search Engines	Google LLC	US
use.fontawesome.com	221 (2.53 %)	___stripe_mid	1 years	Technology/Internet	WhoisGuard Protected	_
siteintercept.qualtrics.com	99 (1.13 %)	t_uid	100 years	Business/Economy	Qualtrics, LLC	US
push.zhanzhang.baidu.com	98 (1.12 %)	BAIDUID	68 years	Search Engines	Beijing Baidu Netcom Science Technology Co., Ltd.	CN
Tracking enabled in a third party context						
assets.adobedtm.com	427 (4.88 %)	_gd_visitor	20 years	Technology/Internet	Adobe Inc.	US
yastatic.net	303 (3.47 %)	cto_lwid	1 year	Technology/Internet	Yandex N.V.	RU
s.sspqns.com	278 (3.18 %)	tuuid	6 months	Web Ads/Analytics	HI-MEDIA	FR
tags.tiqcdn.com	276 (3.16 %)	utag_main	1 year	Content Servers	Tealium Inc	US
cdnjs.cloudflare.com	206 (2.36 %)	___cfduid	1 year	Content Servers	Cloudflare	US
static.quantcast.mgr.consensu.org	157 (1.80 %)	_cmpQc3pChkKey	Session	Consent frameworks	IAB Europe	BE
a.twiago.com	133 (1.52 %)	deuxesse_uxid	1 month	Office/Business Applications	REDACTED FOR PRIVACY	_
g.alicdn.com	121 (1.38 %)	_uab_collina	10 years	Content Servers	Alibaba Cloud Computing Ltd.	CN

Table 11. Top 15 domains missed by EL&EP and Disconnect but detected by BehaviorTrack to perform tracking.

Table 12. Usage of EL&EP lists in security, privacy, and web measurement communities (venues from 2016-2018). "Detection" describes how EL&EP was used to detect trackers: whether the filterlists were applied on all requests, ("Req"), on requests and follow-up requests that would be blocked, ("Req.+Follow") or whether filterlists were further customised before being applied to the dataset ("Custom"). In the dependency column, "Rely" means that the authors use the EL&EP to build their results, "verify" means that the authors only use EL&EPlists to verify their results.

Paper	Venue	EasyList	EasyPrivacy	Detection	Dependency
Englehardt and Narayanan [24]	ACM CCS 2016	✓	✓	Req.	Rely
Bashir et al. [11]	USENIX Security 2016	✓		Custom.	Rely
Lauinger et al. [30]	NDSS 2017	✓	✓	Req.+Follow	Rely
Razaghpanah et al. [42]	NDSS 2018	✓		Custom.	Rely
Ikram et al. [28]	PETs 2017	✓		Req.+Follow	Verif.
Englehardt et al.[23]	PETs 2018	✓	✓	Req.+Follow	Verif.
Bashir and Wilson [12]	PETs 2018	✓	✓	Custom.	Rely+Verif.
Bashir et al.[10]	IMC 2018	✓	✓	Custom.	Rely
Iordanou et al.[29]	IMC 2018	✓	✓	Req.+Follow	Rely