Tracing Information Flows Between Ad Exchanges Using Retargeted Ads

Muhammad Ahmad Bashir
Northeastern University
ahmad@ccs.neu.edu

Sajjad Arshad
Northeastern University
arshad@ccs.neu.edu

William Robertson
Northeastern University
wkr@ccs.neu.edu

Christo Wilson
Northeastern University
cbw@ccs.neu.edu

Abstract

Numerous surveys have shown that Web users are concerned about the loss of privacy associated with online tracking. Alarmingly, these surveys also reveal that people are also unaware of the amount of data sharing that occurs between ad exchanges, and thus underestimate the privacy risks associated with online tracking.

In reality, the modern ad ecosystem is fueled by a flow of user data between trackers and ad exchanges. Although recent work has shown that ad exchanges routinely perform *cookie matching* with other exchanges, these studies are based on brittle heuristics that cannot detect all forms of information sharing, especially under adversarial conditions.

In this study, we develop a methodology that is able to detect client- and server-side flows of information between arbitrary ad exchanges. Our key insight is to leverage *retargeted ads* as a tool for identifying information flows. Intuitively, our methodology works because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*. Using crawled data on 35,448 ad impressions, we show that our methodology can successfully categorize four different kinds of information sharing behavior between ad exchanges, including cases where existing heuristic methods fail.

We conclude with a discussion of how our findings and methodologies can be leveraged to give users more control over what kind of ads they see and how their information is shared between ad exchanges.

1 Introduction

People have complicated feelings with respect to online behavioral advertising. While surveys have shown that some users prefer relevant, targeted ads to random, untargeted ads [60, 14], this preference has caveats. For example, users are uncomfortable with ads that are targeted based on sensitive Personally Identifiable Information (PII) [44, 4] or specific kinds of browsing history (*e.g.,* visiting medical websites) [41]. Furthermore, some users are universally opposed to online tracking, regardless of circumstance [46, 60, 14].

One particular concern held by users is their "digital footprint" [33, 65, 58], *i.e.,* which first- and third-parties are able to track their browsing history? Large-scale web crawls have repeatedly shown that trackers are ubiquitous [24, 19], with DoubleClick alone being able to observe visitors on 40% of websites in the Alexa Top-100K [11]. These results paint a picture of a balkanized web, where trackers divide up the space and compete for the ability to collect data and serve targeted ads.

However, this picture of the privacy landscape is at odds with the current reality of the ad ecosystem. Specifically, ad exchanges routinely perform *cookie matching* with each other, to synchronize unique identifiers and share user data [2, 54, 21]. Cookie matching is a precondition for ad exchanges to participate in *Real Time Bidding* (RTB) auctions, which have become the dominant mechanism for buying and selling advertising inventory from publishers. Problematically, Hoofnagle *et al.* report that users naïvely believe that privacy policies prevent companies from sharing user data with third-parties, which is not always the case [32].

Despite user concerns about their digital footprint, we currently lack the tools to fully understand how information is being shared between ad exchanges. Prior empirical work on cookie matching has relied on heuristics that look for specific strings in HTTP messages to identify flows between ad networks [2, 54, 21]. However, these heuristics are brittle in the face of obfuscation: for example, DoubleClick cryptographically hashes their cookies before sending them to other ad networks [1]. More fundamentally, analysis of *client-side* HTTP messages are insufficient to detect *server-side* information flows between ad networks.

In this study, we develop a methodology that is able to detect client- and server-side flows of information between arbitrary ad exchanges that serve *retargeted ads*. Retargeted ads are the most specific form of behavioral ads, where a user is targeted with ads related to the exact products she has previously browsed (see § 2.2 for definition). For example, Bob visits `nike.com` and browses for running shoes but decides not to purchase them. Bob later visits `cnn.com` and sees an ad for the exact same running shoes from Nike.

Our key insight is to leverage retargeted ads as a mechanism for identifying information flows. This is possible because the strict conditions that must be met for a retarget to be served allow us to infer the precise flow of tracking information that facilitated the serving of the ad. Intuitively, our methodology works because it relies on the *semantics* of how exchanges serve ads, rather than focusing on specific cookie matching *mechanisms*.

To demonstrate the efficacy of our methodology, we conduct extensive experiments on real data. We train 90 *personas* by visiting popular e-commerce sites, and then crawl major publishers to gather retargeted ads [9, 12]. Our crawler is an instrumented version of Chromium that records the *inclusion chain* for every resource it encounters [5], including 35,448 chains associated with 5,102 unique retargeted ads. We use carefully designed pattern matching rules to categorize each of these chains, which reveal 1) the pair of ad exchanges that shared information in order to serve the retarget, and 2) the mechanism used to share the data (*e.g.,* cookie matching).

In summary, we make the following contributions:

- We present a novel methodology for identifying information flows between ad networks that is content- and ad exchange-agnostic. Our methodology allows to identify four different categories of information sharing between ad exchanges, of which cookie matching is one.
- Using crawled data, we show that the heuristic methods used by prior work to analyze cookie matching are unable to identify 31% of ad exchange pairs that share data.
- Although it is known that Google's privacy policy allows it to share data between its services [26], we provide the first empirical evidence that Google uses this capability to serve retargeted ads.
- Using graph analysis, we show how our data can be used to automatically infer the roles played by different ad exchanges (*e.g.,* Supply-Side and Demand-Side Platforms). These results expand upon prior work [25] and facilitate a more nuanced understanding of the online ad ecosystem.

Ultimately, we view our methodology as a stepping stone towards more balanced privacy protection tools for users, that also enable publishers to earn revenue. Surveys have shown that users are not necessarily opposed to online ads: some users are just opposed to tracking [46, 60, 14], while others simply desire more nuanced control over their digital footprint [4, 41]. However, existing tools (*e.g.,* browser extensions) cannot distinguish between targeted and untargeted ads, thus leaving users with no alternative but to block all ads. Conversely, our results open up the possibility of building in-browser tools that just block cookie matching, which will effectively prevent most targeted ads from RTB auctions, while still allowing untargeted ads to be served.

Open Source. As a service to the community, we have open sourced all the data from this project. This includes over 7K labeled behaviorally targeted and retargeted ads, as well as the inclusion chains and full HTTP traces associated with these ads. The data is available at:

`http://personalization.ccs.neu.edu/`

2 Background and Definitions

In this section, we set the stage for our study by providing background about the online display ad industry, as well as defining key terminology. We focus on techniques and terms related to Real Time Bidding and retargeted ads, since they are the focus of our study.

2.1 Online Display Advertising

Online display advertising is fundamentally a matching problem. On one side are *publishers* (*e.g.,* news websites, blogs, *etc.*) who produce content, and earn revenue by displaying ads to users. On the other side are advertisers who want to display ads to particular users (*e.g.,* based on demographics or market segments). Unfortunately, the online user population is fragmented across hundreds of thousands of publishers, making it difficult for advertisers to reach desired customers.

Ad networks bridge this gap by aggregating *inventory* from publishers (*i.e.,* space for displaying ads) and filling it with ads from advertisers. Ad networks make it possible for advertisers to reach a broad swath of users, while also guaranteeing a steady stream of revenue for publishers. Inventory is typically sold using a Cost per Mille (CPM) model, where advertisers purchase blocks of 1000 *impressions* (views of ads), or a Cost per Click (CPC) model, where the advertiser pays a small fee each time their ad is clicked by a user.

Ad Exchanges and Auctions. Over time, ad networks are being supplanted by *ad exchanges* that rely on an auction-based model. In Real-time Bidding (*RTB*) exchanges, advertisers bid on individual impressions, in real-time; the winner of the auction is permitted to serve

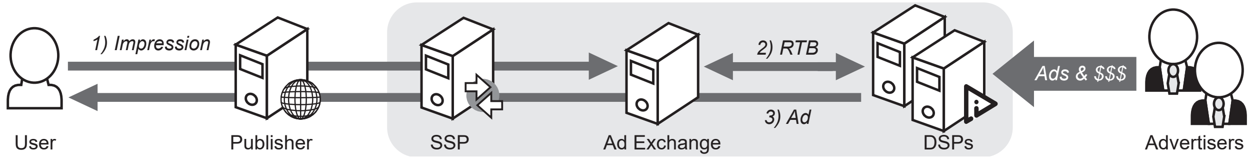


Figure 1: The display advertising ecosystem. Impressions and tracking data flow left-to-right, while revenue and ads flow right-to-left.

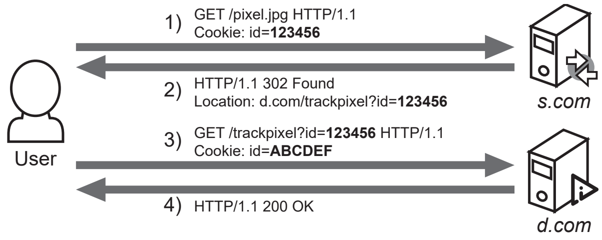
an ad to the user. Google's DoubleClick is the largest ad exchange, and it supports RTB.

As shown in Figure 1, there is a distinction between Supply-side Platforms (*SSPs*) and Demand-side Platforms (*DSPs*) with respect to ad auctions. SSPs work with publishers to manage their relationships with multiple ad exchanges, typically to maximize revenue. For example, OpenX is an SSP. In contrast, DSPs work with advertisers to assess the value of each impression and optimize bid prices. MediaMath is an example of a DSP. To make matters more complicated, many companies offer products that cross categories; for example, Rubicon Project offers SSP, ad exchange, and DSP products. We direct interested readers to [45] for more discussion of the modern online advertising ecosystem.

2.2 Targeted Advertising

Initially, the online display ad industry focused on generic brand ads (*e.g.,* "Enjoy Coca-Cola!") or *contextual ads* (*e.g.,* an ad for Microsoft on StackOverflow). However, the industry quickly evolved towards *behavioral targeted ads* that are served to specific users based on their browsing history, interests, and demographics.

Tracking. To serve targeted ads, ad exchanges and advertisers must collect data about online users by tracking their actions. Publishers embed JavaScript or invisible "tracking pixels" that are hosted by tracking companies into their web pages, thus any user who visits the publisher also receives third-party cookies from the tracker (we discuss other tracking mechanisms in § 3). Numerous studies have shown that trackers are pervasive across the Web [38, 36, 55, 11], which allows advertisers to collect users' browsing history. All major ad exchanges, like DoubleClick and Rubicon, perform user tracking, but there are also companies like BlueKai that just specialize in tracking.



1) GET /pixel.jpg HTTP/1.1
Cookie: id=**123456**

2) HTTP/1.1 302 Found
Location: d.com/trackpixel?id=**123456**

3) GET /trackpixel?id=**123456** HTTP/1.1
Cookie: id=**ABCDEF**

4) HTTP/1.1 200 OK

Figure 2: SSP *s* matches their cookie to DSP *d* using an HTTP redirect.

Cookie Matching. During an RTB ad auction, DSPs submit bids on an impression. The amount that a DSP bids on a given impression is intrinsically linked to the amount of information they have about that user. For example, a DSP is unlikely to bid highly for user *u* whom they have never observed before, whereas a DSP may bid heavily for user *v* who they have recently observed browsing high-value websites (*e.g.,* the baby site `TheBump.com`).

However, the Same Origin Policy (SOP) hinders the ability of DSPs to identify users in ad auctions. As shown in Figure 1, requests are first sent to an SSP which forwards the impression to an exchange (or holds the auctions itself). At this point, the SSP's cookies are known, but not the DSPs. This leads to a catch-22 situation: a DSP cannot read its cookies until it contacts the user, but it cannot contact the user without first bidding and winning the auction.

To circumvent SOP restrictions, ad exchanges and advertisers engage in *cookie matching* (sometimes called *cookie syncing*). Cookie matching is illustrated in Figure 2: the user's browser first contacts ad exchange `s.com`, which returns an HTTP redirect to its partner `d.com`. `s` reads its own cookie, and includes it as a parameter in the redirect to `d`. `d` now has a mapping from its cookie to `s`'s. In the future, if `d` participates in an auction held by `s`, it will be able to identify matched users using `s`'s cookie. Note that some ad exchanges (including DoubleClick) send cryptographically hashed cookies to their partners, which prevents the ad network's true cookies from leaking to third-parties.

Retargeted Ads. In this study, we focus on *retargeted ads*, which are the most specific type of targeted display ads. Two conditions must be met for a DSP to serve a retargeted ad to a user *u*: 1) the DSP must know that *u* browsed a specific product on a specific e-commerce site, and 2) the DSP must be able to uniquely identify *u* during an auction. If these conditions are met, the DSP can serve *u* a highly personalized ad reminding them to purchase the product from the retailer. Cookie matching is crucial for ad retargeting, since it enables DSPs to meet requirement (2).

3 Related Work

Next, we briefly survey related work on online advertising. We begin by looking at more general studies of the advertising and tracking ecosystem, and conclude with a more focused examination of studies on cookie matching and retargeting. Although existing studies on cookie matching demonstrate that this practice is widespread and that the privacy implications are alarming, these works have significant methodological shortcomings that motivate us to develop new techniques in this work.

3.1 Measuring the Ad Ecosystem

Numerous studies have measured and broadly characterized the online advertising ecosystem. Guha *et al.* were the first to systematically measure online ads, and their carefully controlled methodology has been very influential on subsequent studies (including this one) [27]. Barford *et al.* take a much broader look at the *adscape* to determine who the major ad networks are, what fraction of ads are targeted, and what user characteristics drive targeting [9]. Carrascosa *et al.* take an even finer grained look at targeted ads by training *personas* that embody specific interest profiles (*e.g.,* cooking, sports), and find that advertisers routinely target users based on sensitive attributes (*e.g.,* religion) [12]. Rodriguez *et al.* measure the ad ecosystem on mobile devices [61], while Zarras *et al.* analyzed malicious ad campaigns and the ad networks that serve them [66].

Note that **none** of these studies examine retargeted ads; Carrascosa *et al.* specifically excluded retargets from their analysis [12].

Trackers and Tracking Mechanisms. To facilitate ad targeting, participants in the ad ecosystem must extensively track users. Krishnamurthy *et al.* have been cataloging the spread of trackers and assessing the ensuing privacy implications for years [38, 36, 37]. Roesner *et al.* develop a comprehensive taxonomy of different tracking mechanisms that store state in users' browsers (*e.g.,* cookies, HTML5 LocalStorage, and Flash LSOs), as well as strategies to block them [55]. Gill *et al.* use large web browsing traces to model the revenue earned by different trackers (or *aggregators* in their terminology), and found that revenues are skewed towards the largest trackers (primarily Google) [24]. More recently, Cahn *et al.* performed a broad survey of cookie characteristics across the Web, and found that less than 1% of trackers can aggregate information across 75% of websites in the Alexa Top-10K [11]. Falahrastegar *et al.* expand on these results by comparing trackers across geographic regions [20], while Li *et al.* show that most tracking cookies can be automatically detected using simple machine learning methods [42].

Note that **none** of these studies examine cookie matching, or information sharing between ad exchanges.

Although users can try to evade trackers by clearing their cookies or using private/incognito browsing modes, companies have fought back using techniques like *Evercookies* and *fingerprinting*. Evercookies store the tracker's state in many places within the browser (*e.g.,* FlashLSOs, etags, *etc.*), thus facilitating regeneration of tracking identifiers even if users delete their cookies [34, 57, 6, 47]. Fingerprinting involves generating a unique ID for a user based on the characteristics of their browser [18, 48, 50], browsing history [53], and computer (*e.g.,* the HTML5 canvas [49]). Several studies have found trackers in-the-wild that use fingerprinting techniques [3, 52, 35]; Nikiforakis *et al.* propose to stop fingerprinting by carefully and intentionally adding more entropy to users' browsers [51].

User Profiles. Several studies specifically focus on tracking data collected by Google, since their trackers are more pervasive than any others on the Web [24, 11]. Alarmingly, two studies have found that Google's Ad Preferences Manager, which is supposed to allow users to see and adjust how they are being targeted for ads, actually hides sensitive information from users [64, 16]. This finding is troubling given that several studies rely on data from the Ad Preferences Manager as their source of ground-truth [27, 13, 9]. To combat this lack of transparency, Lecuyer *et al.* have built systems that rely on controlled experiments and statistical analysis to infer the profiles that Google constructs about users [39, 40]. Castelluccia *et al.* go further by showing that adversaries can infer users' profiles by passively observing the targeted ads they are shown by Google [13].

3.2 Cookie Matching and Retargeting

Although ad exchanges have been transitioning to RTB auctions since the mid-2000s, only three empirical studies have examined the cookie matching that enables these services. Acar *et al.* found that hundreds of domains passed unique identifiers to each other while crawling websites in the Alexa Top-3K [2]. Olejnik *et al.* noticed that ad auctions were leaking the winning bid prices for impressions, thus enabling a fascinating behind-the-scenes look at RTB auctions [54]. In addition to examining the monetary aspects of auctions, Olejnik *et al.* found 125 ad exchanges using cookie matching. Finally, Falahrastegar *et al.* examine the clusters of domains that all share unique, matched cookies using crowdsourced browsing data [21]. Additionally, Ghosh *et al.* use game theory to model the incentives for ad exchanges to match cookies with their competitors, but they provide no empirical measurements of cookie matching [23].

Several studies examine retargeted ads, which are directly facilitated by cookie matching and RTB. Liu *et al.* identify and measure retargeted ads served by DoubleClick by relying on unique AdSense tags that are embedded in ad URLs [43]. Olejnik *et al.* crawled specific e-commerce sites in order to elicit retargeted ads from those retailers, and observe that retargeted ads can cost advertisers over $1 per impression (an enormous sum, considering contextual ads sell for <$0.01) [54].

Limitations. The prior work on cookie matching demonstrates that this practice is widespread. However, these studies also have significant methodological limitations, which prevent them from observing all forms of information sharing between ad exchanges. Specifically:

1. All three studies identify cookie matching by locating unique user IDs that are transmitted to multiple third-party domains [2, 54, 21]. Unfortunately, this will miss cases where exchanges send permuted or obfuscated IDs to their partners. Indeed, DoubleClick is known to do this [1].
2. The two studies that have examined the behavior of DoubleClick have done so by relying on specific cookie keys and URL parameters to detect cookie matching and retargeting [54, 43]. Again, these methods are not robust to obfuscation or encryption that hide the content of HTTP messages.
3. Existing studies cannot determine the precise information flows between ad exchanges, *i.e.,* which parties are sending or receiving information [2]. This limitation stems from analysis techniques that rely entirely on analyzing HTTP headers. For example, a script from `t1.com` embedded in `pub.com` may share cookies with `t2.com` using dynamic AJAX, but the referrer appears to be `pub.com`, thus potentially hiding `t1`'s role as the source of the flow.

In general, these limitations stem from a reliance on analyzing specific *mechanisms* for cookie matching. In this study, one of our primary goals is to develop a methodology for detecting cookie matching that is agnostic to the underlying matching mechanism, and instead relies on the fundamental *semantics* of ad exchanges.

4 Methodology

In this study, our primary goal is to develop a methodology for detecting flows of user data between arbitrary ad exchanges. This includes client-side flows (*i.e.,* cookie matching), as well as server-side flows.

In this section, we discuss the methods and data we use to meet this goal. First, we briefly sketch our high-level approach, and discuss key enabling insights. Second, we introduce the instrumented version of Chromium that we use during our crawls. Third, we explain how we designed and trained shopper *personas* that view products on the web, and finally we detail how we collected ads using the trained personas.

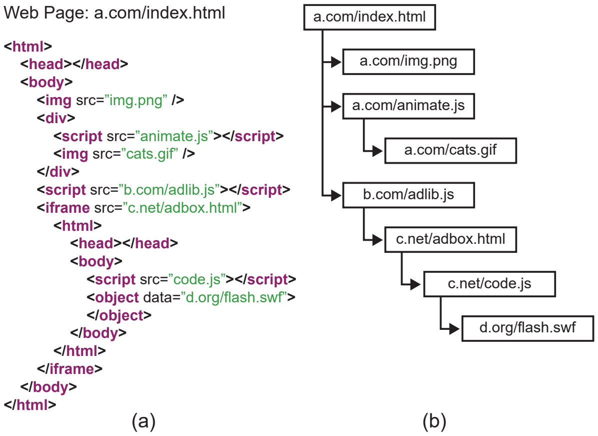
4.1 Insights and Approach

Although prior work has examined information flow between ad exchanges, these studies are limited to specific types of cookie matching that follow well-defined patterns (see § 3.2). To study arbitrary information flows in a mechanism-agnostic way, we need a fundamentally different methodology.

We solve this problem by relying on a key insight: in most cases, if a user is served a retargeted ad, this proves that ad exchanges shared information about the user (see § 6.1.1). To understand this insight, consider that two preconditions must be met for user u to be served a retarget ad for *shop* by DSP d. *First*, either d directly observed u visiting *shop*, or d must be told this information by SSP s. If this condition is not met, then d would not pay the premium price necessary to serve u a retarget. *Second*, if the retarget was served from an ad auction, SSP s and d must be sharing information about u. If this condition is not met, then d would have no way of identifying u as the source of the impression (see § 2.2).

In this study, we leverage this observation to reliably infer information flows between SSPs and DSPs, regardless of whether the flow occurs client- or server-side. The high-level methodology is quite intuitive: have a clean browser visit specific e-commerce sites, then crawl publishers and gather ads. If we observe retargeted ads, we know that ad exchanges tracking the user on the *shopper-side* are sharing information with exchanges serving ads on the *publisher-side*. Specifically, our methodology uses the following steps:

- § 4.2: We use an instrumented version of Chromium to record *inclusion chains* for all resources encountered during our crawls [5]. These chains record the precise origins of all resource requests, even when the requests are generated dynamically by JavaScript or Flash. We use these chains in § 6 to categorize information flows between ad exchanges.
- § 4.3: To elicit retargeted ads from ad exchanges, we design *personas* (to borrow terminology from [9] and [12]) that visit specific e-commerce sites. These sites are carefully chosen to cover different types of products, and include a wide variety of common trackers.



```
<html>
  <head></head>
  <body>
    <img src="img.png" />
    <div>
      <script src="animate.js"></script>
      <img src="cats.gif" />
    </div>
    <script src="b.com/adlib.js"></script>
    <iframe src="c.net/adbox.html">
      <html>
        <head></head>
        <body>
          <script src="code.js"></script>
          <object data="d.org/flash.swf">
          </object>
        </body>
      </html>
    </iframe>
  </body>
</html>
```

Figure 3: (a) DOM Tree, and (b) Inclusion Tree.

- § 4.4: To collect ads, our personas crawl 150 publishers from the Alexa Top-1K.
- § 5: We leverage well-known filtering techniques and crowdsourcing to identify retargeted ads from our corpus of 571,636 unique crawled images.

4.2 Instrumenting Chromium

Before we can begin crawling, we first need a browser that is capable of recording detailed information about the provenance of third-party resource inclusions in webpages. Recall that prior work on cookie matching was unable to determine which ad exchanges were syncing cookies in many cases because the analysis relied solely on the contents of HTTP requests [2, 21] (see § 3.2). The fundamental problem is that HTTP requests, and even the DOM tree itself, do not reveal the true sources of resource inclusions in the presence of dynamic code (JavaScript, Flash, *etc.*) from third-parties.

To understand this problem, consider the example DOM tree for `a.com/index.html` in Figure 3(a). Based on the DOM, we might conclude that the chain $a \rightarrow c \rightarrow d$ captures the sequence of inclusions leading from the root of the page to the Flash object from `d.org`.

However, direct use of a webpage's DOM is misleading because the DOM does not reliably record the inclusion relationships between resources in a page. This is due to the ability of JavaScript to manipulate the DOM at run-time, *i.e.,* by adding new inclusions dynamically. As such, while the DOM is a faithful syntactic description of a webpage *at a given point in time*, it cannot be relied upon to extract relationships between included resources. Furthermore, analysis of HTTP request headers does not solve this problem, since the `Referer` is set to the first-party domain even when inclusions are dynamically added by third-party scripts.

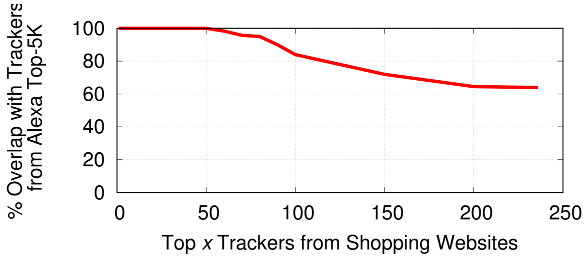


Figure 4: Overlap between frequent trackers on e-commerce sites and Alexa Top-5K sites.

To solve this issue, we make use of a heavily instrumented version of Chromium that produces *inclusion trees* directly from Chromium's resource loading code [5]. Inclusion trees capture the semantic inclusion structure of resources in a webpage (*i.e.,* which objects cause other objects to be loaded), unlike DOM trees which only capture syntactic structures. Our instrumented Chromium accurately captures relationships between elements, regardless of where they are located (*e.g.,* within a single page or across frames) or how the relevant code executes (*e.g.,* via an inline `<script>`, `eval()`, or an event handler). We direct interested readers to [5] for more detailed information about inclusion trees, and the technical details of how the Chromium binary is instrumented.

Figure 3(b) shows the inclusion tree corresponding to the DOM tree in Figure 3(a). From the inclusion tree, we can see that the true *inclusion chain* leading to the Flash object is $a \rightarrow b \rightarrow c \rightarrow c \rightarrow d$, since the `IFrame` and the Flash are dynamically included by JavaScript from `b.com` and `c.net`, respectively.

Using inclusion chains, we can precisely analyze the provenance of third-party resources included in webpages. In § 6, we use this capability to distinguish client-side flows of information between ad exchanges (*i.e.,* cookie matching) from server-side flows.

4.3 Creating Shopper Personas

Now that we have a robust crawling tool, the next step in our methodology is designing shopper personas. Each persona visits products on specific e-commerce sites, in hope of seeing retargeted ads when we crawl publishers.

Since we do not know a priori which e-commerce sites are conducting retargeted ad campaigns, our personas must cover a wide variety of sites. To facilitate this, we leverage the hierarchical categorization of e-commerce sites maintained by Alexa[1]. Although Alexa's hierarchy

[1] `http://www.alexa.com/topsites/category/Top/Shopping`

has 847 total categories, there is significant overlap between categories. We manually selected 90 categories to use for our personas that have minimal overlap, as well as cover major e-commerce sites (*e.g.,* Amazon and Walmart) and shopping categories (*e.g.,* sports and jewelry).

For each persona, we included the top 10 e-commerce sites in the corresponding Alexa category. In total, the personas cover 738 unique websites. Furthermore, we manually selected 10 product URLs on each of these websites. Thus, each persona visits 100 products URLs.

Sanity Checking. The final step in designing our personas is ensuring that the e-commerce sites are embedded with a representative set of trackers. If they are not, we will not be able to collect targeted ads.

Figure 4 plots the overlap between the trackers we observe on the Alexa Top-5K websites, compared to the top *x* trackers (*i.e.,* most frequent) we observe on the e-commerce sites. We see that 84% of the top 100 e-commerce trackers are also present in the trackers on Alexa Top-5K sites[2]. These results demonstrate that our shopping personas will be seen by the vast majority of major trackers when they visit our 738 e-commerce sites.

4.4 Collecting Ads

In addition to selecting e-commerce sites for our personas, we must also select publishers to crawl for ads. We manually select 150 publishers by examining the Alexa Top-1K websites and filtering out those which do not display ads, are non-English, are pornographic, or require logging-in to view content (*e.g.,* Facebook). We randomly selected 15 URLs on each publisher to crawl.

At this point, we are ready to crawl ads. We initialized 91 copies of our instrumented Chromium binary: 90 corresponding to our shopper personas, and one which serves as a control. During each *round* of crawling, the personas visit their associated e-commerce sites, then visit the 2,250 publisher URLs (150 publishers $*$ 15 pages per publisher). The control *only* visits the publisher URLs, *i.e.,* it does not browse e-commerce sites, and therefore should never be served retargeted ads. The crawlers are executed in tandem, so they visit the publishers URLs in the same order at the same times. We hard-coded a 1 minute delay between subsequent page loads to avoid overloading any servers, and to allow time for the crawler to automatically scroll to the bottom of each page. Each round takes 40 hours to complete.

We conducted nine rounds of crawling between December 4 to 19, 2015. We stopped after 9 rounds because we observed that we only gathered 4% new images during the ninth round. The crawlers recorded inclusion trees, HTTP request and response headers, cookies, and images from all pages. At no point did our crawlers click on ads, since this can be construed as click-fraud (*i.e.,* advertisers often have to pay each time their ads are clicked, and thus automated clicks drain their advertising budget). All crawls were done from *Northeastern University's* IP addresses in Boston.

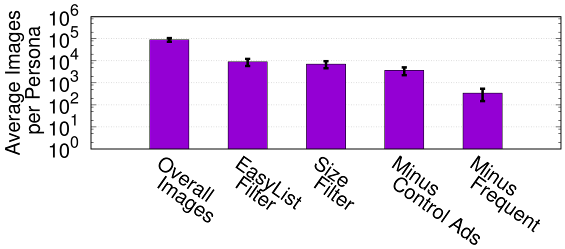


Figure 5: Average number of images per persona, with standard deviation error bars.

5 Image Labeling

Using the methodology in § 4.4, we collected 571,636 unique images in total. However, only a small subset are retargeted ads, which are our focus. In this section, we discuss the steps we used to filter down our image set and isolate retargeted ads, beginning with standard filters used by prior work [9, 42], and ending with crowdsourced image labeling.

5.1 Basic Filtering

Prior work has used a number of techniques to identify ad images from crawled data. First, we leverage the *EasyList* filter[3] provided by *AdBlockPlus* to detect images that are likely to be ads [9, 42]. In our case, we look at the inclusion chain for each image, and filter out those in which none of the URLs in the chain are a hit against EasyList. This reduces the set to 93,726 unique images.

Next, we filter out all images with dimensions $< 50 \times 50$ pixels. These images are too small to be ads; most are 1×1 tracking pixels.

Our final filter relies on a unique property of retargeted ads: they should only appear to personas that visit a specific e-commerce site. In other words, any ad that was shown to our control account (which visits no e-commerce sites) is either untargeted or contextually targeted, and can be discarded. Furthermore, any ad shown to >1 persona may be behaviorally targeted, but it cannot be a retarget, and is therefor filtered out[4].

[2] We separately crawled the resources included by the Alexa Top-5K websites in January 2015. For each website, we visited 6 pages and recorded all the requested resources.

[3] https://easylist-downloads.adblockplus.org/easylist.txt

[4] Several of our personas have retailers in common, which we account for when filtering ads.

Figure 5 shows the average number of images remaining per persona after applying each filter. After applying all four filters, we are left with 31,850 ad images.

5.2 Identifying Targeted & Retargeted Ads

At this point, we do not know which of the ad images are retargets. Prior work has identified retargets by looking for specific URL parameters associated with them, however this technique is only able to identify a subset of retargets served by DoubleClick [43]. Since our goal is to be mechanism and ad exchange agnostic, we must use a more generalizable method to identify retargets.

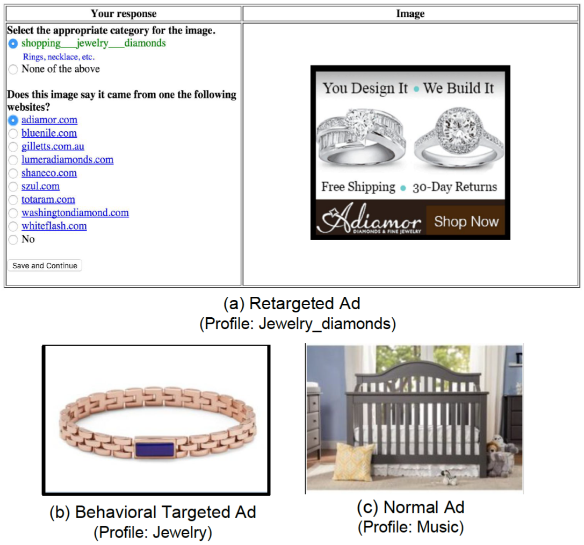
Crowdsourcing. Given the large number of ads in our corpus, we decided to crowdsource labels from workers on Amazon Mechanical Turk (AMT). We constructed Human Intelligence Tasks (HITs) that ask workers to label 30 ads, 27 of which are unlabeled, and 3 of which are known to be retargeted ads and serve as controls (we manually identified 1,016 retargets from our corpus of 31,850 to serve as these controls).

Figure 6(a) shows a screenshot of our HIT. On the right is an ad image, and on the left we ask the worker two questions:

1. Does the image belong to one of the following categories (with “None of the above” being one option)?
2. Does the image say it came from one of the following websites (with “No” being one option)?

The purpose of question (1) is to isolate behavioral and retargeted ads from contextual and untargeted ads (*e.g.,* Figure 6(c), which was served to our *Music* persona). The list for question (1) is populated with the shopping categories associated with the persona that crawled the ad. For example, as shown in Figure 6(a), the category list includes “shopping_jewelry_diamonds” for ads shown to our *Diamond Jewelry* persona. In most cases, this list contains exactly one entry, although there are rare cases where up to 3 categories are in the list.

If the worker does not select “None” for question (1), then they are shown question (2). Question (2) is designed to separate retargets from behavioral targeted ads. The list of websites for question (2) is populated with the e-commerce sites visited by the persona that crawled the ad. For example, in Figure 6(a), the ad clearly says “Adiamor”, and one of the sites visited by the persona is `adiamor.com`; thus, this image is likely to be a retarget.



Your response	Image
Select the appropriate category for the image. ● shopping___jewelry___diamonds Rings, necklace, etc. ○ None of the above **Does this image say it came from one the following websites?** ● adiamor.com ○ bluenile.com ○ gilletts.com.au ○ lumeradiamonds.com ○ shaneco.com ○ szul.com ○ totaram.com ○ washingtondiamond.com ○ whiteflash.com ○ No Save and Continue	

(a) Retargeted Ad
(Profile: Jewelry_diamonds)

(b) Behavioral Targeted Ad
(Profile: Jewelry)

(c) Normal Ad
(Profile: Music)

Figure 6: Screenshot of our AMT HIT, and examples of different types of ads.

Quality Control. We apply four widely used techniques to maintain and validate the quality of our crowdsourced image labels [63, 29, 56]. *First*, we restrict our HITs to workers that have completed ≥50 HITs and have an approval rating ≥95%. *Second*, we restrict our HITs to workers living in the US, since our ads were collected from US websites. *Third*, we reject a HIT if the worker mislabels ≥2 of the control images (*i.e.,* known retargeted ads); this prevents workers from being able to simply answer “None” to all questions. We resubmitted rejected HITs for completion by another worker. Overall, the workers correctly labeled 87% of the control images. *Fourth* and finally, we obtain two labels on each unlabeled image by different workers. For 92.4% of images both labels match, so we accept them. We manually labeled the divergent images ourselves to break the tie.

Finding More Retargets. The workers from AMT successfully identified 1,359 retargeted ads. However, it is possible that they failed to identify some retargets, *i.e.,* there are false negatives. This may occur in cases like Figure 6(b): it is not clear if this ad was served as a behavioral target based on the persona’s interest in jewelry, or as a retarget for a specific jeweler.

To mitigate this issue, we manually examined all 7,563 images that were labeled as behavioral ads by the workers. In addition to the images themselves, we also looked at the inclusion chains for each image. In many cases, the URLs reveal that specific e-commerce sites visited by our personas hosted the images, indicating that the ads are retargets. For example, Figure 6(b) is actually part of a retargeted ad from `fossil.com`. Our manual analysis uncovered an additional 3,743 retargeted ads.

These results suggest that the number of false negatives from our crowdsourcing task could be dramatically reduced by showing the URLs associated with each ad image to the workers. However, note that adding this information to the HIT will change the dynamics of the task: false negatives may go down but the effort (and therefore the cost) of each HIT will go up. This stems from the additional time it will take each worker to review the ad URLs for relevent keywords.

In § 6.2, we compare the datasets labeled by the workers and by the authors. Interestingly, although our dataset contains a greater *magnitude* of retargeted ads versus the worker's dataset, it does not improve *diversity*, *i.e.,* the smaller dataset identifies 96% of the top 25 most frequent ad networks in the larger dataset. These networks are responsible for the vast majority of retargeted ads and inclusion chains in our dataset.

Final Results. Overall, we submitted 1,142 HITs to AMT. We paid $0.18 per HIT, for a total of $415. We did not collect any personal information from workers. In total, we and workers from AMT labeled 31,850 images, of which 7,563 are behavioral targeted ads and 5,102 are retargeted ads. These retargets advertise 281 distinct e-commerce websites (38% of all e-commerce sites).

5.3 Limitations

With any labeling task of this size and complexity, it is possible that there are false positives and negatives. Unfortunately, we cannot bound these quantities, since we do not have ground-truth information about known retargeted ad campaigns, nor is there a reliable mechanism to automatically detect retargets (*e.g.,* based on special URL parameters, *etc.*).

In practice, the effect of false positives is that we will erroneously classify pairs of ad exchanges as sharing information. We take measures to mitigate false positives by running a control crawl and removing images which appear in multiple personas (see § 5.1), but false positives can still occur. However, as we show in § 6, the results of our classifier are extremely consistent, suggesting that there are few false positives in our dataset.

False negatives have the opposite effect: we may miss pairs of ad exchanges that are sharing information. Fortunately, the practical impact of false negatives is low, since we only need to correctly identify a single retargeted ad to infer that a given pair of ad exchanges are sharing information.

6 Analysis

In this section, we use the 5,102 retargeted ads uncovered in § 5, coupled with their associated inclusion chains (see § 4.2), to analyze the information flows between ad exchanges. Specifically, we seek to answer two fundamental questions: *who* is sharing user data, and *how* does the sharing take place (*e.g.,* client-side via cookie matching, or server-side)?

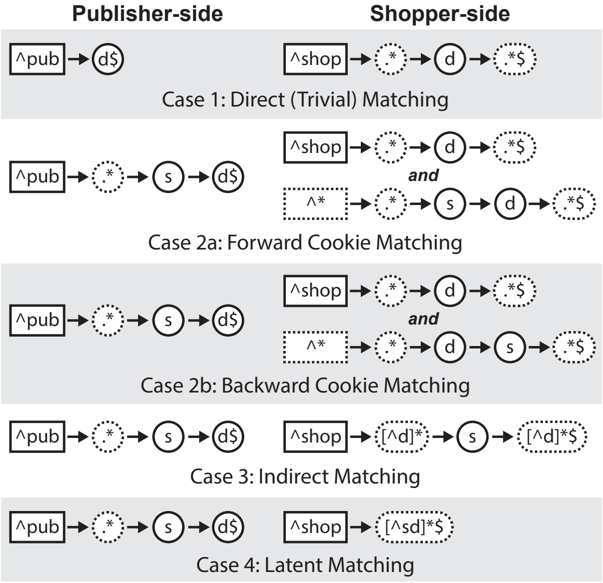


Figure 7: Regex-like rules we use to identify different types of ad exchange interactions. *shop* and *pub* refer to chains that begin at an e-commerce site or publisher, respectively. *d* is the DSP that serves a retarget; *s* is the predecessor to *d* in the publisher-side chain, and is most likely an SSP holding an auction. Dot star (.∗) matches any domains zero or more times.

We begin by *categorizing* all of the retargeted ads and their associated inclusion chains into one of four classes, which correspond to different mechanisms for sharing user data. Next, we examine specific pairs of ad exchanges that share data, and compare our detection approach to those used in prior work to identify cookie matching [43, 2, 54, 21]. We find that prior work may be missing 31% of collaborating exchanges. Finally, we construct a graph that captures ad exchanges and the relationships between them, and use it to reveal nuanced characteristics about the roles that different exchanges play in the ad ecosystem.

6.1 Information Flow Categorization

We begin our analysis by answering two basic questions: *for a given retargeted ad, was user information shared between ad exchanges, and if so, how?* To answer these questions, we categorize the 35,448 *publisher-side* inclusion chains corresponding to the 5,102 retargeted ads in our data. Note that 1) we observe some retargeted ads multiple times, resulting in multiple chains, and 2) the chains for a given unique ad may not be identical.

We place publisher-side chains into one of four categories, each of which corresponds to a specific information sharing mechanism (or lack thereof). To determine the category of a given chain, we match it against carefully designed, regular expression-like rules. Figure 7 shows the pattern matching rules that we use to identify chains in each category. These rules are mutually exclusive, *i.e.,* a chain will match one or none of them.

Terminology. Before we explain each classification in detail, we first introduce shared terminology that will be used throughout this section. Each retargeted ad was served to our persona via a *publisher-side* chain. *pub* is the domain of the publisher at the root of the chain, while *d* is the domain at the end of the chain that served the ad. Typically, *d* is a DSP. If the retarget was served via an auction, then an SSP *s* must immediately precede *d* in the publisher-side chain.

Each retarget advertises a particular e-commerce site. *shop* is the domain of the e-commerce site corresponding to a particular retargeted ad. To categorize a given publisher-side chain, we must also consider the corresponding *shopper-side* chains rooted at *shop*.

6.1.1 Categorization Rules

Case 1: Direct Matches. The first chain type that we define are *direct matches*. Direct matches are the simplest type of chains that can be used to serve a retargeted ad. As shown in Figure 7, for us to categorize a publisher-side chain as a direct match, it must be exactly length two, with a direct resource inclusion request from *pub* to *d*. *d* receives any cookies they have stored on the persona inside this request, and thus it is trivial for *d* to identify our persona.

On the shopper-side, the only requirement is that *d* observed our persona browsing *shop*. If *d* does not observe our persona at *shop*, then *d* would not serve the persona a retargeted ad for *shop*. *d* is able to set a cookie on our persona, allowing *d* to re-identify the persona in future.

We refer to direct matching chains as "trivial" because it is obvious how *d* is able to track our persona and serve a retargeted ad for *shop*. Furthermore, in these cases no user information needs to be shared between ad exchanges, since there are no ad auctions being held on the publisher-side.

Case 2: Cookie Matching. The second chain type that we define are *cookie matches*. As the name implies, chains in this category correspond to instance where an auction is held on the publisher-side, and we observe direct resource inclusion requests between the SSP and DSP, implying that they are matching cookies.

As shown in Figure 7, for us to categorize a publisher-side chain as cookie matching, *s* and *d* must be adjacent at the end of the chain. On the shopper-side, *d* must observe the persona at *shop*. Lastly, we must observe a request from *s* to *d* or from *d* to *s* in some chain before the retargeted ad is served. These requests capture the moment when the two ad exchanges match their cookies. Note that $s \rightarrow d$ or $d \rightarrow s$ can occur in a publisher- or shopper-side chain; in practice, it often occurs in a chain rooted at *shop*, thus fulfilling both requirements at once.

For the purposes of our analysis, we distinguish between *forward* ($s \rightarrow d$) and *backward* ($d \rightarrow s$) cookie matches. Figure 2 shows an example of a forward cookie match. As we will see, many pairs of ad exchanges engage in both forward and backward matching to maximize their opportunities for data sharing. To our knowledge, no prior work examines the distinction between forward and backward cookie matching.

Case 3: Indirect Matching. The third chain type we define are *indirect matches*. Indirect matching occurs when an SSP sends meta-data about a user to a DSP, to help them determine if they should bid on an impression. With respect to retargeted ads, the SSP tells the DSPs about the browsing history of the user, thus enabling the DSPs to serve retargets for specific retailers, even if the DSP never directly observed the user browsing the retailer (hence the name, *indirect*). Note that no cookie matching is necessary in this case for DSPs to serve retargeted ads.

As shown in Figure 7, the crucial difference between cookie matching chains and indirect chains is that *d never* observes our persona at *shop*; only *s* observes our persona at *shop*. Thus, by inductive reasoning, we must conclude that *s* shares information about our persona with *d*, otherwise *d* would never serve the persona a retarget for *shop*.

Case 4: Latent Matching. The fourth and final chain type that we define are *latent matches*. As shown in Figure 7, the defining characteristic of latent chains is that neither *s* nor *d* observe our persona at *shop*. This begs the question: how do *s* and *d* know to serve a retargeted ad for *shop* if they never observe our persona at *shop*? The most reasonable explanation is that some other ad exchange *x* that is present in the shopper-side chains shares this information with *d* behind-the-scenes.

We hypothesize that the simplest way for ad exchanges to implement latent matching is by having *x* and *d* share the same unique identifiers for users. Although *x* and *d* are different domains, and are thus prevented by the SOP from reading each others' cookies, both ad exchanges may use the same deterministic algorithm for generating user IDs (*e.g.,* by relying on IP addresses or browser fingerprints). However, as we will show, these synchronized identifiers are not necessarily visible from the client-side (*i.e.,* the values of cookies set by *x* and *d* may be obfuscated), which prevents trivial identification of latent cookie matching.

	Unclustered		Clustered	
Type	Chains	%	Chains	%
Direct	1770	5%	8449	24%
Forward Cookie Match	24575	69%	25873	73%
Backward Cookie Match	19388	55%	24994	70%
Indirect Match	2492	7%	178	1%
Latent Match	5362	15%	343	1%
No Match	775	2%	183	1%

Table 1: Results of categorizing publisher-side chains, before and after clustering domains.

Note: Although we do not expect to see cases 3 and 4, they can still occur. We explain in § 6.1.2 that indirect and latent matching is mostly performed by domains belonging to the same company. The remaining few instances of these cases are probably mislabeled behaviorally targeted ads.

6.1.2 Categorization Results

We applied the rules in Figure 7 to all 35,448 publisher-side chains in our dataset twice. First, we categorized the raw, unmodified chains; then we *clustered* domains that belong to the same companies, and categorized the chains again. For example, Google owns `youtube.com`, `doubleclick.com`, and `2mdn.net`; in the clustered experiments, we replace all instances of these domains with `google.com`. Appendix A.1 lists all clustered domains.

Table 1 presents the results of our categorization. The first thing we observe is that cookie matching is the most frequent classification by a large margin. This conforms to our expectations, given that RTB is widespread in today's ad ecosystem, and major exchanges like DoubleClick support it [17]. Note that, for a given (s, d) pair in a publisher-side chain, we may observe $s \to d$ and $d \to s$ requests in our data, *i.e.,* the pair engages in forward and backward cookie matching. This explains why the percentages in Table 1 do not add up to 100%.

The next interesting feature that we observe in Table 1 is that indirect and latent matches are relatively rare (7% and 15%, respectively). Again, this is expected, since these types of matching are more exotic and require a greater degree of collaboration between ad exchanges to implement. Furthermore, the percentage of indirect and latent matches drops to 1% when we cluster domains. To understand why this occurs, consider the following real-world example chains:

Publisher-side: $pub \to rubicon \to googlesyndication$

Shopper-side: $shop \to doubleclick$

According to the rules in Figure 7, this appears to be a latent match, since Rubicon and Google Syndication do not observe our persona on the shopper-side. However, after clustering the Google domains, this will be classified as cookie matching (assuming that there exists at least one other request from Rubicon to Google).

The above example is extremely common in our dataset: 731 indirect chains become cookie matching chains after we cluster the Google domains *alone*. Importantly, this finding provides strong evidence that Google does in fact use latent matching to share user tracking data between its various domains. Although this is allowed in Google's terms of service as of 2014 [26], our results provide direct evidence of this data sharing with respect to serving targeted ads. In the vast majority of these cases, Google Syndication is the DSP, suggesting that on the server-side, it ingests tracking data and user identifiers from all other Google services (*e.g.,* DoubleClick and Google Tag Manager).

Of the remaining 1% of chains that are still classified as indirect or latent after clustering, the majority appear to be false positives. In most of these cases, we observe s and d doing cookie matching in other instances, and it seems unlikely that s and d would also utilize indirect and latent mechanisms. These ads are probably mislabeled behaviorally targeted ads.

The final takeaway from Table 1 is that the number of uncategorized chains that do not match any of our rules is extremely low (1-2%). These publisher-side chains are likely to be false positives, *i.e.,* ads that are not actually retargeted. These results suggest that our image labeling approach is very robust, since the vast majority of chains are properly classified as direct or cookie matches.

6.2 Cookie Matching

The results from the previous section confirm that cookie matching is ubiquitous on today's Web, and that this information sharing fuels highly targeted advertisements. Furthermore, our classification results demonstrate that we can detect cookie matching without relying on semantic information about cookie matching mechanisms.

In this section, we take a closer look at the pairs of ad exchanges that we observe matching cookies. We seek to answer two questions: *first*, which pairs match most frequently, and what is the directionality of these relationships? *Second*, what fraction of cookie matching relationships will be missed by the heuristic detection approaches used by prior work [43, 2, 54, 21]?

Who Is Cookie Matching? Table 2 shows the top 25 most frequent pairs of domains that we observe matching cookies. The arrows indicate the direction of matching (forward, backward, or both). "Ads" is the number of unique retargets served by the pair, while "Chains" is the total number of associated publisher-side chains. We present both quantities as observed in our complete dataset (containing 5,102 retargets), as well as the subset

			All Data		AMT Only		
Participant 1		**Participant 2**	**Chains**	**Ads**	**Chains**	**Ads**	**Heuristics**
criteo	↔	googlesyndication	9090	1887	1629	370	↔: US
criteo	↔	doubleclick	3610	1144	770	220	→: E, US ←: DC, US
criteo	↔	adnxs	3263	1066	511	174	↔: E, US
criteo	↔	googleadservices	2184	1030	448	214	→: E, US ←: US
criteo	↔	rubiconproject	1586	749	240	113	↔: E, US
criteo	↔	servedbyopenx	707	460	111	71	↔: US
mythings	↔	mythingsmedia	478	52	53	1	→: E, US ←: US
criteo	↔	pubmatic	363	246	64	37	→: E, US ←: US
doubleclick	↔	steelhousemedia	362	27	151	16	→: US ←: E, US
mathtag	↔	mediaforge	360	124	63	13	↔: E, US
netmng	↔	scene7	267	162	45	32	→: E ←: -
criteo	↔	casalemedia	200	119	54	31	→: E, US ←: US
doubleclick	↔	googlesyndication	195	81	101	62	↔: US
criteo	↔	clickfuse	126	99	14	13	↔: US
criteo	↔	bidswitch	112	78	25	15	→: E, US ←: US
googlesyndication	↔	adsrvr	107	29	102	24	↔: US
rubiconproject	↔	steelhousemedia	86	30	43	19	↔: E
amazon-adsystem	↔	ssl-images-amazon	98	33	33	7	-
googlesyndication	↔	steelhousemedia	47	22	36	16	-
adtechus	→	adacado	36	18	36	18	-
googlesyndication	↔	2mdn	40	19	39	18	→: US ←: -
atwola	→	adacado	32	6	28	5	-
adroll	↔	adnxs	31	8	26	7	-
googlesyndication	↔	adlegend	31	22	29	20	-
adnxs	↔	esm1	46	1	0	0	→: US ←: -

Table 2: Top 25 cookie matching partners in our dataset. The arrow signifies whether we observe forward matches (→), backward matches (←), or both (↔). The heuristics for detecting cookie matching are: *DC* (match using DoubleClick URL parameters), *E* (string match for exact cookie values), *US* (URLs that include parameters like "usersync"), and - (no identifiable mechanisms). Note that the HTTP request formats used for forward and backward matches between a given pair of exchanges may vary.

that was identified solely by the AMT workers (containing 1,359 retargets).

We observe that cookie matching frequency is heavily skewed towards several heavy-hitters. In aggregate, Google's domains are most common, which makes sense given that Google is the largest ad exchange on the Web today. The second most common is Criteo; this result also makes sense, given that Criteo specializes in retargeted advertising [15]. These observations remain broadly true across the AMT and complete datasets: although the relative proportion of ads and chains from less-frequent exchange pairs differs somewhat between the two datasets, the heavy-hitters do not change. Furthermore, we also see that the vast majority of exchange pairs are identified in both datasets.

Interestingly, we observe a great deal of heterogeneity with respect to the directionality of cookie matching. Some boutique exchanges, like Adacado, only ingest cookies from other exchanges. Others, like Criteo, are omnivorous, sending or receiving data from any and all willing partners. These results suggest that some participants are more wary about releasing their user identifiers to other exchanges.

Comparison to Prior Work. We observe many of the same participants matching cookies as prior work, including DoubleClick, Rubicon, AppNexus, OpenX, MediaMath, and myThings [2, 54, 21]. Prior work identifies some additional ad exchanges that we do not (*e.g.,* Turn); this is due to our exclusive focus on participants involved in retargeted advertising.

However, we also observe participants (*e.g.,* Adacado and AdRoll) that prior work does not. This may be because prior work identifies cookie matching using heuristics to pick out specific features in HTTP requests [43, 2, 54, 21]. In contrast, our categorization approach is content and mechanism agnostic.

To investigate the efficacy of heuristic detection methods, we applied three of them to our dataset. Specifically, for each pair (s, d) of exchanges that we categorize as cookie matching, we apply the following tests to the HTTP headers of requests between s and d or vice-versa:

1. We look for specific keys that are known to be used by DoubleClick and other Google domains for cookie matching (*e.g.,* "google_nid" [54]).
2. We look for cases where unique cookie values set by one participant are included in requests sent to the other participant[5].

[5] To reduce false positives, we only consider cookie values that have length >10 and <100.

			Degree			Position p in Chains (%)			# of Shopper	
		Domain	In	Out	In/Out Ratio	p_2	p_{n-1}	p_n	Websites	# of Ads
DSPs		criteo	35	6	5.83	9.28	0.00	68.8	248	3,335
		mediaplex	8	2	4.00	0.00	85.7	0.07	20	14
		tellapart	6	1	6.00	25.0	100.0	0.18	33	9
		mathtag	12	6	2.00	0.00	90.9	0.06	314	2
		mythingsmedia	1	0	-	0.00	0.00	1.41	1	59
		steelhousemedia	8	0	-	0.00	0.00	16.8	40	89
		mediaforge	5	0	-	0.00	0.00	1.28	29	143
SSPs		pubmatic	5	9	0.56	3.17	74.2	0.01	362	4
		rubiconproject	19	22	0.86	23.5	62.8	0.01	394	3
		adnxs	18	20	0.90	94.2	91.9	0.16	476	12
		casalemedia	9	10	0.90	1.30	90.0	0.00	298	0
	AOL	atwola	4	19	0.21	84.6	18.2	0.01	62	2
		advertising	4	4	1.00	0.00	75.0	0.10	337	17
		adtechus	17	16	1.06	1.58	27.3	0.09	328	15
	OpenX	servedbyopenx	6	11	0.55	7.2	83.8	0.00	2	0
		openx	10	9	1.11	0.95	9.83	0.00	390	0
		openxenterprise	4	4	1.00	40.0	20.0	0.00	1	0
Google		googletagservices	44	2	22.00	93.7	0.00	0.00	65	0
		googleadservices	4	17	0.24	2.94	33.5	0.00	485	0
		2mdn	3	1	3.00	0.00	0.00	1.35	62	125
		googlesyndication	90	35	2.57	70.1	62.7	19.8	84	638
		doubleclick	38	36	1.06	38.8	63.1	0.22	675	19

Table 3: Overall statistics about the connectivity, position, and frequency of ad domains in our dataset.

3. We look for keys with revealing names like "user-sync" that frequently appear in requests between participants in our data.

As shown in the "Heuristics" column in Table 2, in the majority of cases, heuristics are able to identify cookie matching between the participants. Interestingly, we observe that the mechanisms used by some pairs (*e.g.,* Criteo and DoubleClick) change depending on the directionality of the cookie match, revealing that the two sides have different cookie matching APIs.

However, for 31% of our cookie matching partners, the heuristics are unable to detect signs of cookie matching. We hypothesize that this is due to obfuscation techniques employed by specific ad exchanges. In total, there are 4.1% cookie matching chains that would be completely missed by heuristic tests. This finding highlights the limitations of prior work, and bolsters the case for our mechanism-agnostic classification methodology.

6.3 The Retargeting Ecosystem

In this last section, we take a step back and examine the broader ecosystem for retargeted ads that is revealed by our dataset. To facilitate this analysis, we construct a graph by taking the union of all of our publisher-side chains. In this graph, each node is a domain (either a publisher or an ad exchange), and edges correspond to resource inclusion relationships between the domains. Our graph formulation differs from prior work in that edges denote actual information flows, as opposed to simple co-occurrences of trackers on a given domain [25].

Table 3 presents statistics on the top ad-related domains in our dataset. The "Degree" column shows the in- and out-degree of nodes, while "Position" looks at the relative location of nodes within chains. p_2 is the second position in the chain, corresponding to the first ad network after the publisher; p_n is the DSP that serves the retarget in a chain of length n; p_{n-1} is the second to last position, corresponding to the final SSP before the DSP. Note that a domain may appear in a chain multiple times, so the sum of the p_i percentages may be $>100\%$. The last two columns count the number of unique e-commerce sites that embed resources from a given domain, and the unique number of ads served by the domain.

Based on the data in Table 3, we can roughly cluster the ad domains into two groups, corresponding to SSPs and DSPs. DSPs have low or zero out-degree since they often appear at position p_n, *i.e.,* they serve an ad and terminate the chain. Criteo is the largest source of retargeted ads in our dataset by an order of magnitude. This is not surprising, given that Criteo was identified as the largest retargeter in the US and UK in 2014 [15].

In contrast, SSPs tend to have in/out degree ratios closer to 1, since they facilitate the exchange of ads between multiple publishers, DSPs, and even other SSPs. Some SSPs, like Atwola, work more closely with publishers and thus appear more frequently at p_2, while others, like Mathtag, cater to other SSPs and thus appear almost exclusively at p_{n-1}. Most of the SSPs we observe also function as DSPs (*i.e.,* they serve some retargeted ads), but there are "pure" SSPs like Casale Media and OpenX that do not serve ads. Lastly, Table 3 reveals that

SSPs tend to do more user tracking than DSPs, by getting embedded in more e-commerce sites (with Criteo being the notable exception).

Google is an interesting case study because its different domains have clearly delineated purposes. `googletagservices` is Google's in-house SSP, which funnels impressions directly from publishers to Google's DSPs: `2mdn`, `googlesyndication`, and `doubleclick`. In contrast, `googleadservices` is also an SSP, but it holds auctions with third-party participants (*e.g.,* Criteo). `googlesyndication` and `doubleclick` function as both SSPs and DSPs, sometimes holding auctions, and sometimes winning auctions held by others to serve ads. Google Syndication is the second most frequent source of retargeted ads in our dataset behind Criteo.

7 Concluding Discussion

In this study, we develop a novel, principled methodology for detecting flows of tracking information between ad exchanges. The key insight behind our approach is that we re-purpose retargeted ads as a detection mechanism, since their presence reveals information flows between ad exchanges. Our methodology is content-agnostic, and thus we are able to identify flows even if they occur on the server-side. This is a significant improvement over prior work, which relies on heuristics to detect cookie matching [2, 54, 21]. As we show in § 6, these heuristics fail to detect 31% of matching pairs today, and they are likely to fail more in the future as ad networks adopt content obfuscation techniques.

Implications for Users. Ultimately, our goal is not just to measure information flows between ad exchanges, but to facilitate the development of systems that balance user privacy against the revenue needs of publishers.

Currently, users are faced with unsatisfactory choices when deciding if and how to block ads and tracking. Whitelisting approaches like NoScript are effective at protecting privacy, but are too complicated for most users, and deprive publishers of revenue. Blocking third-party cookies is ineffective against first-party trackers (*e.g.,* Facebook). AdBlockPlus' controversial "Acceptable Ads" program is poorly governed and leaves users vulnerable to unscrupulous ad networks [62]. DNT is DOA [8]. Although researchers have proposed privacy preserving ad exchanges, these systems have yet to see widespread adoption [22, 28, 7].

We believe that data about information flows between ad exchanges potentially opens up a new middle ground in ad blocking. One possibility is to develop an automated system that uses the methodology developed in this paper to continuously crawl ads, identify cookie matching flows, and construct rules that match these flows. Users could then install a browser extension that blocks flows matching these rules. The advantage of this extension is that it would offer improved privacy protection relative to existing systems (*e.g.,* Ghostery and Disconnect), while also allowing advertising (as opposed to traditional ad blockers). However, the open challenge with this system design would be making it cost effective, since it would still rely crowdsourced labor.

Another possibility is using our data as ground-truth for a sophisticated blocker that relies on client-side Information Flow Control (IFC). There exist many promising, lightweight approaches to implementing JavaScript IFC in the browser [30, 10, 59, 31]. However, IFC alone is not enough to block cookie matching flows: as we have shown, ad networks obfuscate data, making it impossible to separate benign from "leaky" flows in general. Instead, we can use information gathered using our methodology as ground-truth to mark data in specific incoming flows, and rely on IFC to enforce restrictions that prevent outgoing flows from containing the marked data.

Acknowledgements

We thank our shepherd, Nektarios Leontiadis, and the anonymous reviewers for their helpful comments. This research was supported in part by NSF grants CNS-1319019 and CHS-1408345. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] Real-time bidding protocol, February 2016. `https://developers.google.com/ad-exchange/rtb/cookie-guide`.

[2] Acar, G., Eubank, C., Englehardt, S., Juarez, M., Narayanan, A., and Diaz, C. The web never forgets: Persistent tracking mechanisms in the wild. In *Proc. of CCS* (2014).

[3] Acar, G., Juarez, M., Nikiforakis, N., Diaz, C., Gürses, S., Piessens, F., and Preneel, B. Fpdetective: Dusting the web for fingerprinters. In *Proc. of CCS* (2013).

[4] Agarwal, L., Shrivastava, N., Jaiswal, S., and Panjwani, S. Do not embarrass: Re-examining user concerns for online tracking and advertising.

[5] Arshad, S., Kharraz, A., and Robertson, W. Include me out: In-browser detection of malicious third-party content inclusions. In *Proc. of Intl. Conf. on Financial Cryptography* (2016).

[6] Ayenson, M., Wambach, D. J., Soltani, A., Good, N., and Hoofnagle, C. J. Flash cookies and privacy ii: Now with html5 and etag respawning. *Available at SSRN 1898390* (2011).

[7] Backes, M., Kate, A., Maffei, M., and Pecina, K. Obliviad: Provably secure and practical online behavioral advertising. In *Proc. of IEEE Symposium on Security and Privacy* (2012).

[8] BALEBAKO, R., LEON, P. G., SHAY, R., UR, B., WANG, Y., AND CRANOR, L. F. Measuring the effectiveness of privacy tools for limiting behavioral advertising. In *Proc. of W2SP* (2012).

[9] BARFORD, P., CANADI, I., KRUSHEVSKAJA, D., MA, Q., AND MUTHUKRISHNAN, S. Adscape: Harvesting and analyzing online display ads. In *Proc. of WWW* (2014).

[10] BICHHAWAT, A., RAJANI, V., GARG, D., AND HAMMER, C. Information flow control in webkit's javascript bytecode. In *Proc. of Principles of Security and Trust* (2014).

[11] CAHN, A., ALFELD, S., BARFORD, P., AND MUTHUKRISHNAN, S. An empirical study of web cookies. In *Proc. of WWW* (2016).

[12] CARRASCOSA, J. M., MIKIANS, J., CUEVAS, R., ERRAMILLI, V., AND LAOUTARIS, N. I always feel like somebody's watching me: Measuring online behavioural advertising. In *Proc. of ACM CoNEXT* (2015).

[13] CASTELLUCCIA, C., KAAFAR, M.-A., AND TRAN, M.-D. Betrayed by your ads!: Reconstructing user profiles from targeted ads. In *Proc. of PETS* (2012).

[14] CHANCHARY, F., AND CHIASSON, S. User perceptions of sharing, advertising, and tracking.

[15] Criteo ranking by Econsultancy. `http://www.criteo.com/resources/e-consultancy-display-retargeting-buyers-guide/`.

[16] DATTA, A., TSCHANTZ, M. C., AND DATTA, A. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *Proc. of PETS* (2015).

[17] Double Click RTB explained. `https://developers.google.com/ad-exchange/rtb/`.

[18] ECKERSLEY, P. How unique is your web browser? In *Proc. of PETS* (2010).

[19] ENGLEHARDT, S., REISMAN, D., EUBANK, C., ZIMMERMAN, P., MAYER, J., NARAYANAN, A., AND FELTEN, E. W. Cookies that give you away: The surveillance implications of web tracking. In *Proc. of WWW* (2015).

[20] FALAHRASTEGAR, M., HADDADI, H., UHLIG, S., AND MORTIER, R. The rise of panopticons: Examining region-specific third-party web tracking. In *Proc of. Traffic Monitoring and Analysis* (2014).

[21] FALAHRASTEGAR, M., HADDADI, H., UHLIG, S., AND MORTIER, R. Tracking personal identifiers across the web. In *Proc. of PAM* (2016).

[22] FREDRIKSON, M., AND LIVSHITS, B. Repriv: Re-imagining content personalization and in-browser privacy. In *Proc. of IEEE Symposium on Security and Privacy* (2011).

[23] GHOSH, A., MAHDIAN, M., MCAFEE, P., AND VASSILVITSKII, S. To match or not to match: Economics of cookie matching in online advertising. In *Proc. of EC* (2012).

[24] GILL, P., ERRAMILLI, V., CHAINTREAU, A., KRISHNAMURTHY, B., PAPAGIANNAKI, K., AND RODRIGUEZ, P. Follow the money: Understanding economics of online aggregation and advertising. In *Proc. of IMC* (2013).

[25] GOMER, R., RODRIGUES, E. M., MILIC-FRAYLING, N., AND SCHRAEFEL, M. C. Network analysis of third party tracking: User exposure to tracking cookies through search. In *Proc. of IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)* (2013).

[26] GOODALE, G. Privacy concerns? what google now says it can do with your data. Christian Science Monitor, April 2014. `http://www.csmonitor.com/USA/2014/0416/Privacy-concerns-What-Google-now-says-it-can-do-with-your-data-video`.

[27] GUHA, S., CHENG, B., AND FRANCIS, P. Challenges in measuring online advertising systems. In *Proc. of IMC* (2010).

[28] GUHA, S., CHENG, B., AND FRANCIS, P. Privad: Practical privacy in online advertising. In *Proc. of NSDI* (2011).

[29] HANNAK, A., SAPIEŻYŃSKI, P., KAKHKI, A. M., KRISHNAMURTHY, B., LAZER, D., MISLOVE, A., AND WILSON, C. Measuring Personalization of Web Search. In *Proc. of WWW* (2013).

[30] HEDIN, D., BIRGISSON, A., BELLO, L., AND SABELFELD, A. JSFlow: Tracking Information Flow in JavaScript and Its APIs. In *Proc. of Symposium on Applied Computing* (2014).

[31] HEULE, S., STEFAN, D., YANG, E. Z., MITCHELL, J. C., AND RUSSO, A. IFC inside: Retrofitting languages with dynamic information flow control. In *Proc. of Principles of Security and Trust* (2015).

[32] HOOFNAGLE, C. J., AND URBAN, J. M. Alan westin's privacy homo economicus. *49 Wake Forest Law Review 261* (2014).

[33] HOWELL, D. How to protect your privacy and remove data from online services. Tech Radar, January 2015. `http://www.techradar.com/news/internet/how-to-protect-your-privacy-and-remove-data-from-online-services-1291515`.

[34] KAMKAR, S. Evercookie - virtually irrevocable persistent cookies., September 2010. `http://samy.pl/evercookie/`.

[35] KOHNO, T., BROIDO, A., AND CLAFFY, K. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing 2*, 2 (2005), 93–108.

[36] KRISHNAMURTHY, B., NARYSHKIN, K., AND WILLS, C. Privacy diffusion on the web: A longitudinal perspective. In *Proc. of WWW* (2009).

[37] KRISHNAMURTHY, B., AND WILLS, C. Privacy leakage vs. protection measures: the growing disconnect. In *Proc. of W2SP* (2011).

[38] KRISHNAMURTHY, B., AND WILLS, C. E. Generating a privacy footprint on the internet. In *Proc. of IMC* (2006).

[39] LÉCUYER, M., DUCOFFE, G., LAN, F., PAPANCEA, A., PETSIOS, T., SPAHN, R., CHAINTREAU, A., AND GEAMBASU, R. Xray: Enhancing the web's transparency with differential correlation. In *Proc. of USENIX Security Symposium* (2014).

[40] LECUYER, M., SPAHN, R., SPILIOPOLOUS, Y., CHAINTREAU, A., GEAMBASU, R., AND HSU, D. Sunlight: Fine-grained targeting detection at scale with statistical confidence. In *Proc. of CCS* (2015).

[41] LEON, P. G., UR, B., WANG, Y., SLEEPER, M., BALEBAKO, R., SHAY, R., BAUER, L., CHRISTODORESCU, M., AND CRANOR, L. F. What matters to users?: Factors that affect users' willingness to share information with online advertisers.

[42] LI, T.-C., HANG, H., FALOUTSOS, M., AND EFSTATHOPOULOS, P. Trackadvisor: Taking back browsing privacy from third-party trackers. In *Proc. of PAM* (2015).

[43] LIU, B., SHETH, A., WEINSBERG, U., CHANDRASHEKAR, J., AND GOVINDAN, R. Adreveal: Improving transparency into online targeted advertising. In *Proc. of HotNets* (2013).

[44] MALHEIROS, M., JENNETT, C., PATEL, S., BROSTOFF, S., AND SASSE, M. A. Too close for comfort: A study of the effectiveness and acceptability of rich-media personalized advertising. In *Proc. of CHI* (2012).

[45] MAYER, J. R., AND MITCHELL, J. C. Third-party web tracking: Policy and technology. In *Proc. of IEEE Symposium on Security and Privacy* (2012).

[46] McDonald, A. M., and Cranor, L. F. Americans' attitudes about internet behavioral advertising practices. In *Proc. of WPES* (2010).

[47] McDonald, A. M., and Cranor, L. F. A survey of the use of adobe flash local shared objects to respawn http cookies. *ISJLP 7*, 639 (2011).

[48] Mowery, K., Bogenreif, D., Yilek, S., and Shacham, H. Fingerprinting information in JavaScript implementations. In *Proc. of W2SP* (2011).

[49] Mowery, K., and Shacham, H. Pixel perfect: Fingerprinting canvas in html5. In *Proc. of W2SP* (2012).

[50] Mulazzani, M., Reschl, P., Huber, M., Leithner, M., Schrittwieser, S., and Weippl, E. Fast and reliable browser identification with JavaScript engine fingerprinting. In *Proc. of W2SP* (2013).

[51] Nikiforakis, N., Joosen, W., and Livshits, B. Privaricator: Deceiving fingerprinters with little white lies. In *Proc. of WWW* (2015).

[52] Nikiforakis, N., Kapravelos, A., Joosen, W., Kruegel, C., Piessens, F., and Vigna, G. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *Proc. of IEEE Symposium on Security and Privacy* (2013).

[53] Olejnik, L., Castelluccia, C., and Janc, A. Why Johnny Can't Browse in Peace: On the Uniqueness of Web Browsing History Patterns. In *Proc. of HotPETs* (2012).

[54] Olejnik, L., Minh-Dung, T., and Castelluccia, C. Selling off privacy at auction. In *Proc of NDSS* (2014).

[55] Roesner, F., Kohno, T., and Wetherall, D. Detecting and defending against third-party tracking on the web. In *Proc. of NSDI* (2012).

[56] Soeller, G., Karahalios, K., Sandvig, C., and Wilson, C. Mapwatch: Detecting and monitoring international border personalization on online maps. In *Proc. of WWW* (2016).

[57] Soltani, A., Canty, S., Mayo, Q., Thomas, L., and Hoofnagle, C. J. Flash cookies and privacy. In *AAAI Spring Symposium: Intelligent Information Privacy Management* (2010).

[58] Spector, L. Online privacy tips: 3 ways to control your digital footprint. PC World, January 2016. `http://www.pcworld.com/article/3020163/internet/online-privacy-tips-3-ways-to-control-your-digital-footprint.html`.

[59] Stefan, D., Yang, E. Z., Marchenko, P., Russo, A., Herman, D., Karp, B., and Mazières, D. Protecting users by confining JavaScript with COWL. In *Proc. of OSDI* (2014).

[60] Ur, B., Leon, P. G., Cranor, L. F., Shay, R., and Wang, Y. Smart, useful, scary, creepy: Perceptions of online behavioral advertising.

[61] Vallina-Rodriguez, N., Shah, J., Finamore, A., Grunenberger, Y., Papagiannaki, K., Haddadi, H., and Crowcroft, J. Breaking for commercials: Characterizing mobile advertising. In *Proc. of IMC* (2012).

[62] Walls, R. J., Kilmer, E. D., Lageman, N., and McDaniel, P. D. Measuring the impact and perception of acceptable advertisements. In *Proc. of IMC* (2015).

[63] Wang, G., Mohanlal, M., Wilson, C., Wang, X., Metzger, M., Zheng, H., and Zhao, B. Y. Social turing tests: Crowdsourcing sybil detection. In *Proc. of NDSS* (2013).

[64] Wills, C. E., and Tatar, C. Understanding what they do with what they know. In *Proc. of WPES* (2012).

[65] Wolpin, S. International privacy day: Protect your digital footprint. The Huffington Post, January 2015. `http://www.huffingtonpost.com/stewart-wolpin/international-privacy-day_b_6551012.html`.

[66] Zarras, A., Kapravelos, A., Stringhini, G., Holz, T., Kruegel, C., and Vigna, G. The dark alleys of madison avenue: Understanding malicious advertisements. In *Proc. of IMC* (2014).

A Appendix

A.1 Clustered Domains

We clustered the following domains together when classifying publisher-side chains in § 6.1.2.

Google: google-analytics, googleapis, google, doubleclick, gstatic, googlesyndication, googleusercontent, googleadservices, googletagmanager, googletagservices, googlecommerce, youtube, ytimg, youtube-mp3, googlevideo, 2mdn

OpenX: openxenterprise, openx, servedbyopenx

Affinity: affinitymatrix, affinity

Ebay: ebay, ebaystatic

Yahoo: yahoo, yimg

Mythings: mythingsmedia, mythings

Amazon: cloudfront, amazonaws, amazon-adsystem, images-amazon

Tellapart: tellapart, tellaparts