TrackAdvisor: Taking back browsing privacy from Third-Party Trackers

Tai-Ching Li[1], Huy Hang[1], Michalis Faloutsos[2], and Petros Efstathopoulos[3]

[1] University of California, Riverside. {tli010,hangh}@cs.ucr.edu
[2] University of New Mexico, Albuquerque. michalis@cs.unm.edu
[3] Symantec Research Lab. petros_efstathopoulos@symantec.com

Abstract. Even though most web users assume that only the websites that they visit directly become aware of the visit, this belief is incorrect. Many website display contents hosted externally by third-party websites, which can track users and become aware of their web-surfing behavior. This phenomenon is called third-party tracking, and although such activities violate no law, they raise privacy concerns because the tracking is carried out without users' knowledge or explicit approval. Our work provides a systematic study of the third-party tracking phenomenon. First, we develop TrackAdvisor, arguably the first method that utilizes Machine Learning to identify the HTTP requests carrying sensitive information to third-party trackers with very high accuracy (100% Recall and 99.4% Precision). Microsoft's Tracking Protection Lists, which is a widely-used third-party tracking blacklist achieves only a Recall of 72.2%. Second, we quantify the pervasiveness of the third-party tracking phenomenon: 46% of the home pages of the websites in Alexa Global Top 10,000 have at least one third-party tracker, and Google, using third-party tracking, monitors 25% of these popular websites. Our overarching goal is to measure accurately how widespread third-party tracking is and hopefully would raise the public awareness to its potential privacy risks.

1 Introduction

Would you feel that your privacy is violated if someone knew which websites you visited last night? Most people would feel uneasy and want to ensure their personal browsing information is not revealed to anyone else but the opposite is exactly what has been happening thanks to a phenomenon called **third-party tracking**. As a user visits a website of interest, third-party websites linked to that website become aware of the user's browsing activities and due to the ubiquitous use of cookies, these third-parties can uniquely identify the user[4]. Although this can be appalling for privacy-sensitive users, there is no violation of laws. The third-party tracker is legitimately contacted by the user's browser, because it hosts resources required by the website that the user wants to visit.

[4] In general, it is more accurate to say that third party tracking can track and identify web-browsers and not end users. In the rest of this document, we will use the term "tracking a user" to imply tracking the browser that is being used.

It is natural to ask why the third-party tracking phenomenon is occurring and how. The answer to the "why" question is money, marketing, and advertising. It is easy to see that knowing how many users watch golf scores and search for luxury cars can help one place ads more effectively. With third-party tracking, ads on a website can be customized based on the user's visits to other websites. If you searched for yachts on one site, you could be shown yacht insurance ads on another site. The answer to the "how" question is the widespread use of: (a) embedded links on a webpage (think Facebook "Like" or Google+ "+1" button) or content being pulled from another site, and (b) cookies. Cookies turn any browser into a silent accomplice as the browser voluntarily provides cookies to the third-party websites. These cookies could have been obtained from a tracking website at an earlier time (e.g. when we logged in to Facebook). The obvious solution would not work: not sending cookies at all will often degrade the user experience or even "break" the interaction with websites.

In our work, we want to answer two main questions: (a) *How can we identify cookie-based third-party tracking accurately?* and (b) *How widespread is the phenomenon of third-party tracking?* To address both questions, we need a method that, when given a website and the HTTP interactions between users and that website, can identify third-party trackers. The challenge lies in identifying features of cookies and of the user interaction in general that can accurately identify third-party trackers. This is non-trivial and there exists no such method in the literature, as we discuss below. For the remainder of this paper, we use the term **privacy** to refer to the right of a web-browsing user to not have a third-party website become aware of websites that the user visits. We focus on cookie-based tracking, because it is still the most prevalent form of tracking, as we discuss in section 7.

There has been very little attention on measuring the pervasiveness of third-party tracking activities, which is our focus here. To the best of our knowledge, the most widely-used approaches to combat the third-party tracking problem rely on black lists of third-party trackers, which are maintained by corporations or communities. Microsoft's **Tracking Protection Lists (TPL)** [7] is one such prominent black list, which aggregates many others. As we show later, these efforts are far from perfect, as they are geared towards blocking the more well-known third-party trackers. We discuss related and complementary research efforts in section 7.

The contribution of this paper is a systematic study of the third-party tracking phenomenon and its extent. We also briefly discuss practical countermeasures to enable users to protect their web-browsing privacy. First, we propose TrackAdvisor, an effective method to detect third-party trackers that surpasses existing third-party tracking lists in terms of both accuracy and detection. Second, we use TrackAdvisor to study the prevalence of third-party tracking among Alexa's Global Top 10K websites. We outline our key contributions and results below.

a. We develop TrackAdvisor, a supervised learning approach that identifies third-party trackers with high accuracy. A key novelty of our approach is that it does not rely on a blacklist of websites; TrackAdvisor focuses

on the collective statistics of all cookies inside an outgoing third-party HTTP request to infer whether the third-party website that receives those cookies is tracking the user. Using Machine Learning techniques and carefully selected features, our method exhibits a Precision of 99.4% and a Recall of 100%.

b. We evaluate the accuracy and completeness of TPL and show it yields a relatively low Recall of 72.2%. Microsoft's Tracking Protection Lists (TPL), which combines many existing blacklists, achieves a Recall of 72.2% although with a high Precision of 96.3%. TPL is incorporated in Internet Explorer and can therefore be thought of as the protection that is readily available to users. As a result, its low Recall is somewhat disconcerting.

c. We show that third-party tracking is prevalent: 46% of Alexa's Global top 10K sites being tracked. We find that close to 46% of the *home pages* of the websites in Alexa's Top 10,000 websites have at least one third-party tracker and on average, one out of every three HTTP requests sent to third-party websites is sent to a third-party tracker. More worrisomely, Google is monitoring 25% of the Alexa sites as a third party tracker through its ad and analytics services. As expected, Facebook and Twitter are also prominent third-party tracking, as Facebook "Like" and Twitter's "Tweet" widgets have become very common, especially on blogs and news-related websites. Interestingly, two lesser known companies, Scorecard Research and QuantServe, are among the top five third-party trackers in our dataset.

2 Background

A. Cookies. In the context of the HTTP protocol and web browsing, a cookie is a small, **local** file (about 4KB in size) that helps a website identify a user and their preferences and it is intended to quickly provide the remote website with information such as language (for rendering the content in the correct language) or geographic location (maybe for nearest store location). Cookies are created by the website and stored on the device by the browser the first time the user visits the website. During every subsequent visit, the browser volunteers the saved information to the website.

There are two main components to the structure of a cookie.

1. A **Name** and **Value** pair, which is explicitly set by the website. The pair can be used to save a user's language preference or geographic location. In the case of a third-party tracker, the value portion will be assigned a string that represents a user's unique ID.
2. **Attributes**, which tell the browser how to handle the cookies. The most common attributes of the cookies are: (a) the **domain** that instructs the browser which cookies to send to which websites upon visit and (b) the **expiration**, which is a timestamp specifying to the browser when to a cookie is to be discarded.

B. Third-party tracking. There are three parties involved in a user's visiting a website: the target website w (the first party) the user wants to visit, the user u (the second party), and the entities (the third party) hosting content

external to the website w. Third parties, in this case, are generally transparent to the users and not all of them are third-party trackers.

As the browser needs to download third-party content, it must send an HTTP request to each of the third parties. We call the ones that collect information about the user at this stage **third-party trackers**.

Tracking mechanism: Although HTTP cookies are not the only means with which third-party trackers keep track of users, they are the most popular. There are three reasons to this. Firstly, all browsers can accept and send cookies. Secondly, other non-HTTP cookies exist and can be used for tracking, but they are inefficient or will create legal issues for the entities who utilize them. Finally, even though third-party websites can track a user by their browser fingerprint [13], this method incurs a much higher overhead, thus is unlikely to adopted widely. We will discuss browser fingerprinting and other tracking mechanisms in more details in section 7.

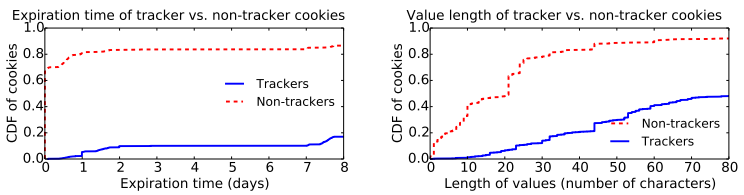
3 Methodology

In this section we will: 1) discuss characteristics of HTTP requests going to trackers and 2) provide an overview of our solution for the problem of detecting third-party trackers.

A. HTTP Requests going to third-party trackers. The key question to ask is whether there are characteristics that differentiate between: (1) HTTP requests carrying information to third-party trackers that can uniquely identify the user, and (2) HTTP requests that carry no such information.

We answer this question positively. The requests going to trackers contain **tracker cookies**, which we define as a cookie that contains a name-value pair that can uniquely identify a user. One such cookie, for instance, may have the name-value pair: `UID=163fkcs65bz` where the value is simply a unique identifier given to the browser by the website. In contrast, there are **non-tracker cookies**, which are used capture user preferences (e.g. display language, timezone), and the browser provides to them to the website in each visit. Because tracker cookies are meant to identify a user, they bear the following characteristics:

1. Their Lifetimes tends to be much longer than non-tracker cookies. A cookie's Lifetime is the time between its creation time and its expiration time.
2. The value part of the name-value pair inside each cookie (recall that each cookie contains only one such pair) must have sufficient **length** to be able to distinguish one user from many others.



(a) Expiration time (b) Length of value

Fig. 1. Difference between tracker and non-tracker cookies

In Figure 1(a), we show the difference in the lifetime values between tracker cookies and non-tracker cookies that we collected and manually labeled (see section 4 for more details on data collection). We can see that while less than 10% of tracker cookies have a lifetime of a single day or less, at least 80% of non-tracker cookies have such short lifetime. Furthermore, Figure 1(b) shows that the length of the value is at least 35 characters for 80% of the tracker cookies, while 80% of the non-tracker cookies have values that are shorter than 35 characters.

The next important question to answer is, then, how we can exploit these characteristics in an effort to correctly classify HTTP requests as either going to third-party trackers and carrying user-identifiable information or harmless and carrying no sensitive information.

B. TrackAdvisor: Identifying trackers, one HTTP request at a time. We present TrackAdvisor, our solution for the problem of identifying third-party trackers. TrackAdvisor looks at *all of the cookies* carried by each outgoing HTTP request, extract collective statistics, and performs classification to determine whether it is heading for a tracker.

TrackAdvisor is a supervised Machine Learning-based application that we envision to reside inside the browser, where it can inspect each outgoing HTTP request and inform the user if the HTTP request carries information that may be able to uniquely identify the user. TrackAdvisor takes as input the cookies exchanged between the browser and the remote websites and identifies the websites that are third-party trackers.

Feature selection: First, we define CookieJar(A, B) as the group of all third-party cookies exchanged between the host A and the remote website B. Note that we exclude the Session cookies because Session cookies are created during a browsing session and are destroyed once the browser is closed. Because of their short-lived nature, Session cookies are unlikely to be used as a tracking mechanism.

Instead of looking at the cookies in CookieJar(A, B) individually, TrackAdvisor looks at CookieJar(A, B) in its entirety, extracts relevant statistics, and performs classification.

We started with considering a large number of features, including maximum Lifetime, minimum ValueLength, mean ValueLength, maximum ValueLength, as well as others. This set of features is then reduced to only three by the Recursive Feature Elimination (RFE) functionality of WEKA [19] which, at a high level, recommends a subset of features that achieves the best accuracy. In our case, the final three features are:

(a) Minimum lifetime: $L^{\min}_{A,B} = \min_c [\text{Lifetime}(c)]$. This feature is selected because trackers, as discussed earlier, tracker cookies tend to have longer lifetime than non-tracker cookies.

(b) Number of third-party cookies in CookieJar(A, B): $N_{A,B}$. This feature is selected because of the trackers' tendency to utilize more cookies than benign third-parties in order to record as much information about the user as possible.

(c) Augmented Lifetime: $L^{\text{aug}}_{A,B} = \sum_c [\text{ValueLength}(c) \times \text{Lifetime}(c)]$. The Augmented Lifetime captures at once captures two important characteristics

of tracker cookies: long Lifetime and long ValueLength, and it is also crucial to future-proofing TrackAdvisor's performance against two possible evasive tactics from third-party trackers: **cookie chunking** and **lifetime reduction**. We will discuss the two techniques, as well as how robust TrackAdvisor is against them at the end of section 4.

The steps that TrackAdvisor executes are:

1. Retain only third-party HTTP requests from the browser. A third-party HTTP request is one that is sent toward an URL that does **not** share the same hostname as the website the user intentionally visits. TrackAdvisor achieves this by looking at the referrer of the request and ignoring requests where the hostnames in the referrer and URL fields are the same.
2. For each CookieJar(A, B) representing an HTTP request sent by host A to website B, TrackAdvisor calculates three features of CookieJar(A, B), that we described above: (a) $L^{\min}_{A,B}$, (b) $N_{A,B}$, and (c) $L^{\text{aug}}_{A,B}$.
3. Use a binary classifier to classify the tuple $\langle L^{\text{aug}}_{A,B}, L^{\min}_{A,B}, N_{A,B} \rangle$.
   A positive output from the classifier means that the tuple belongs in an interaction with a third-party tracker and a negative otherwise. We will discuss how to create the classifier from training data in section 4.
4. If the module returns a positive value, we label B as a third-party tracker and add it to a list that will be presented to the user later.

4 Experiments and Evaluation

In this section we will (a) describe our data collection and preliminary labeling processes and (b) compare the performance of Microsoft's Tracking Protection Lists against that of TrackAdvisor.

A. Data Collection. Our dataset is created by visiting the *landing pages* Alexa's Top 10K Global list [2] during the month of July of 2012. We collected our data using **FourthParty** [4], a Firefox extension that collects data in the background as the user browses the Web. The data that we collected are: a) the header of each HTTP request, b) the header of each HTTP response, and c) the cookie log associated with each request and response. We used the automation framework Selenium [9] with FourthParty installed to collect 563,031 HTTP requests and 99,397 cookies. Of all 563,031 requests, 202,556 were sent to third-party websites and 78,213 contain cookies. Out of 99,397 cookies, 22,270 cookies were sent to third-party websites.

B. Creating training and testing data sets. From the set of all HTTP requests to third-party websites, we created a training and a testing data-set as follows:

- D_{train}: includes 500 randomly chosen requests such that roughly half of them were dispatched to third-party trackers and half were meant to retrieve third-party content and containing no tracking information.
- D_{test}: includes 500 HTTP requests that were randomly chosen in a similar fashion to the ones in D_{train}.

D_{train} and D_{test} are mutually exclusive. The former is used to train TrackAdvisor and the latter will be used for testing both TrackAdvisor and Tracking Protection Lists.

To establish the ground truth, we label the websites in D_{train} and D_{test} (1,000 in total) as either third-party trackers or benign third-party websites using extensive and careful manual evaluation. In our evaluation, we label a website as a third-party tracker by combining the information gained from the three following processes: (a) a manual inspection the website, (b) a consultation with multiple black lists specifically created for third-party tracker, and (c) a careful inspection of cookie properties. To label something as athird-party tracker, we require significant supporting evidence to that effect. We argue that this method is essentially the same used by the contributors to third-party tracking lists. For transparency, we will make our two labelled sets available to the research community.

C. Reference: Microsoft's Tracking Protection List. We compare our approach against Tracking Protection Lists, which is a black list-based component that is used in Microsoft's Internet Explorer. We selected Tracking Protection Lists because: a) it uses the same popular black lists (FanBoy, EasyList, EasyPrivacy, etc.) that empower AdBlock Plus and b) it has been shown that the a combination of the popular black lists achieved comparable performance to Ghostery's [15].

D. Creating a classifier for TrackAdvisor from D_{train}. Recall from the beginning of this section that we have constructed a training dataset and a testing dataset called D_{train} and D_{test}. Also recall that each request in D_{train} is represented by a tuple $\langle L^{\text{aug}}_{A,B}, L^{\text{min}}_{A,B}, N_{A,B}\rangle$. Since each tuple is labeled, we are able to use the WEKA Machine Learning suite [19] to build classifiers. The algorithm that we picked from the suite is Support Vector Machine because it offers the best performance in terms of **Pr**ecision and **Re**call, where Pr = TP / (TP + FP) and Re = TP / (TP + FN). TP is the number of True Positives, FP the number of False Positives, and FN the number of False Negatives.

Before we start the testing, we examine the sensitivity of our approach to the training input by performing a ten-fold cross-validation on D_{train}. The assessment yields a combined Precision of 0.998 and Recall of 0.998 (one FN and one FP). We conclude that our approach is robust to the training data.

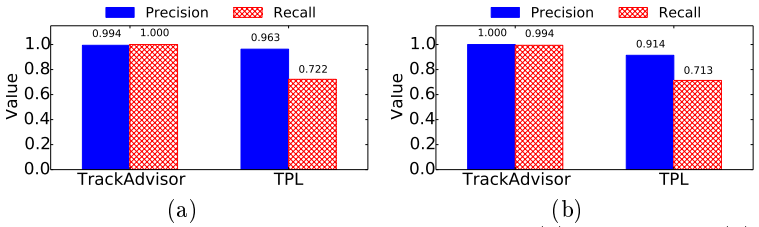


Fig. 2. Classification results for HTTP requests (a) and domains (b)

E. Evaluation of classification on D_{test}. First, we check the URLs of D_{test} against Tracking Protection Lists. As shown in Figure 2(a), TPL achieves a Precision and Recall of 96.3% and 72.2% respectively (13 FPs and 134 FNs).

In contrast, TrackAdvisor achieves perfect Recall and nearly perfect Precision (0 FPs and 2 FNs).

One possible reason why TPL has so many False Negatives could be that TPL is better tuned to recognize the trackers more relatively well-known to the community, as it relies significantly on user reports to populate the list.

F. Possible evasive tactics from third-party trackers: An inquisitive reader may ask why we simply did not use only ValueLength and Lifetime as features for the classifier even though as we have shown in Figure 1 that the ValueLengths and Lifetimes of non-tracker cookies are different from those of tracker cookies. The reason is that a classifier built from only ValueLength and Lifetime is ineffective against two possible evasive tactics from third-party trackers:

T1. Cookie Chunking: Instead of using a single cookie that contains an identifier, third-party trackers can chop it into multiple cookies with different *names* that will be combined later when the HTTP requests are processed at the server. This way, they can reduce the lengths of the cookies and help them avoid detection.

T2. Lifetime Reduction: Instead of setting a large value for the expiration of the cookies, trackers can use smaller values depending on their own *popularity*. For example, a very popular website like Google can set their cookie lifetime to a month or even a week instead of a year because Google knows people visit the site frequently.

We have conducted extensive experiments on the robustness of TrackAdvisor against **T1** and **T2** where we (a) identify every tracker cookie in each HTTP request (in both D_{train} and D_{test}) that we manually label as going to third-party trackers, (b) either split them up according to **T1** or reduce their lifetimes according to **T2**, and (c) re-train our classifier on D_{train} and re-test on D_{test}. We cannot describe the experiments in details due to space limitation but we find that TrackAdvisor's performance is unchanged even when we execute **T1** and **T2**.

5 The pervasiveness of third-party trackers

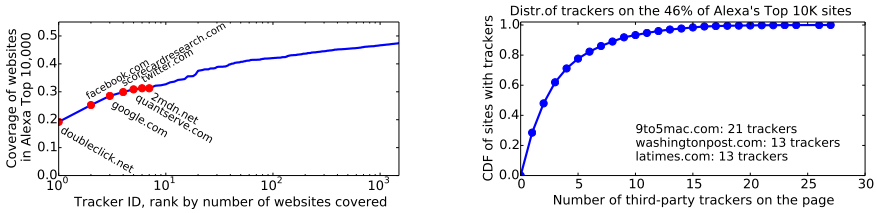
In this section, we quantify the extent of third-party tracking by analyzing the Alexa Top 10K websites. Overall, we find a significant presence of third-party tracking that would be disconcerting to privacy advocates.

A) 46% of the Alexa Top 10K websites have at least one third-party tracker on them. By applying TrackAdvisor on our entire dataset, we found that 46% of the Alexa Top 10K websites had at least one third-party tracker on them. We use the term "target website" to refer to the Alexa website that was explicitly visited by the user in each request as we explained earlier. We plot the cumulative coverage in terms of unique target sites as a function of the number third-party trackers in the order of decreasing activity in Figure 3(a). In more detail, for each third-party tracker t, let S_t be the set of websites in our dataset that are tracked by t. On the x-axis, we order the trackers in decreasing order in terms of the number of sites on which they appear: $|S_{t_i}|$. The y-axis is the

cumulative coverage (C_{t_i}) of the first i trackers in that order. $C_{t_i} = |\cup_{k=1}^{i} S_{t_k}|/N$ where $N = 10,000$ is the total number of target websites.

We can see from Figure 3(a) that:

- 46% of the Alexa Top 10,000 websites have at least one tracker on them.
- The top 5 most common trackers cover 30% of the top 10,000 sites.
- Google alone (`doubleclick.com` and `google.com`) covers 25% of the sites. The `doubleclick.com` domain is responsible for advertisements and `google.com` is where other websites download widgets and libraries.



(a) (b)

Fig. 3. (a) Cumulative coverage of top 10K Alexa sites as a function of third-party trackers in the order of decreasing tracking presence in our dataset. (b) The distribution of the number of trackers on the Alexa top 10K sites.

B. The majority of tracked sites are tracked by more than one tracker. Equally interesting is the fact that a website that has third-party tracking is likely to contain multiple trackers. In Figure 3(b), we plot the CDF of the distribution of third-party trackers on the Alexa websites that have at least one tracker. For example, we see in the plot that 28% of websites have one tracker, which means that there are at least two trackers present on each of the remaining websites (72%).

We also find that 29% of the websites that are tracked by at least five third-party trackers. For a visitor that means that five different entities become aware of her web-surfing preferences. It is equally worrisome to see that some popular websites such as `latimes.com` and `washing-tonpost.com` have upwards of 10 third-party trackers.

The well-known Google Analytics is not on the list in Figure 3(b), because by contract, Google Analytics provides statistics only to the first-party websites and the cookies set by Google Analytics are always associated with the domains of the first-party websites and therefore are not third-party cookies. Furthermore, the same user who visits different websites monitored by Google Analytics will likely receive different IDs, which makes tracking him or her non-trivial.

C. Third-party interactions: 37% tracking versus 63% benign. Recall from section 4 that our dataset contains a total of 202,556 third-party HTTP requests, which includes both third-party tracking and benign third-party interactions. Using our approach, we identify 75,849 (37%) of them as third-party tracking interactions. This is of interest in considering counter-measures to third-party tracking, since there is a large number of interactions with benign third-party websites, as we discuss in the next section.

6 Possible solutions against third-party trackers

Here, we discuss some potential solutions that can be implemented in a browser fairly easily to block third-party trackers from collecting user information.

A. Blocking all third-party cookies. One can consider labeling as trackers all third-party websites that exchange cookies with the user's computer. On the one hand, this type would allow a user to block 100% of the trackers with a false positive rate of 12.6%. On the other hand, that comes at the expense of the degraded browsing experience. There are websites that refuse to display their content unless the user's browser accepts third-party cookies. More specifically, with third-party cookies disabled, iFrames, widely used in third-party games and apps on social networks, cannot read their own cookies [10] and cannot work. As we saw in section 5, the majority (63%) of requests to third-party websites is benign. A complete blocking solution would have unnecessarily blocked them.

B. Removing/Anonymizing the referrer fields in HTTP requests. Apart from the cookies that can uniquely identify users, the values of the referrer fields of the HTTP requests are important to the third-party websites' ability to partially construct a user's browsing history. Therefore, using TrackAdvisor to identify HTTP requests carrying identifying information and then either removing the referrer information or replacing it with bogus values is one way to protect the user's privacy. To the best of our knowledge, third-party websites have tried to withhold content from the users only in the case where the browsers would not accept the cookies and no efforts at all have been invested in validating the referrers as a condition to provide content.

Here we only provide suggestions for possible defense methods against third-party trackers. The full evaluation of the two methods is, however, beyond the scope of this paper and may be tackled in a future work.

7 Related Work

Although much attention has been devoted to studying the phenomenon of third-party trackers using cookies to track users [14, 16, 18], there exists no practical solution that leverages cookies as a means to detect third-party tracking. To the best of our knowledge, all existing practical solutions such as AdBlock Plus [1], Microsoft's Tracking Protection Lists [7], Collusion [3], and Ghostery [5] rely on corporate- and community-maintained black lists (sometimes called block lists) to block HTTP requests to well-known third-party trackers. AdBlock Plus is an improvement to the original AdBlock that also blocks third-party trackers in addition to advertisements. Ghostery and TPL focus on blocking trackers instead.

All other related work have been focused on uncovering other types of cookies (aside from the standard HTTP ones) that could be used to track users but did not propose countermeasures like we did. In [17, 12], the authors documented the use of Flash cookies, which are Locally Shared Objects similar to cookies. Advertisers can create a pair of cookies, an HTTP one and a Flash one, with

identical content, where the latter can "re-spawn" the former even after the former has been deleted. Fortunately, the practice of using Flash cookies have been on the decline because there have been lawsuits against the advertisers, who essentially re-spawned the HTTP cookies against the users' will.

There is a form of cookie-less tracking, which is cache-based and utilizes ETags [12, 6]. An ETag, assigned by the website and unique for each user, is associated with an object on a web page (like an image) that can tell the server if the object in the browser cache is the same as the one on the server. An advertiser then can have exactly the same objects on many websites and track the users just like they would with cookies. This method is not popular, as users can just clear the browser caches frequently.

Most modern browsers offer a "Do Not Track" option which is nothing more than a request and the websites can ignore it if they choose to. The most recent high-profile website that decided to not honor "Do Not Track" is Yahoo [11]. The Electronic Frontier Foundation then responded by releasing Privacy Badger [8], a browser add-on that detects third-party trackers. It keeps tracks of all cookies as the user visits websites and blocks cookies that are *previously seen*. This is a promising development, but, given that this was released only in May 2014, there are no reports yet as to how well Privacy Badger works, if it degrades user experience, and how much overhead it may add in terms of memory due to the large number of cookies that need to be tracked.

Finally, there exists a form of tracking using the *fingerprint* [13] of the browsers. This form of tracking relies on the information that the browser sends to the remote website (such as IP address, User-Agent, System fonts, screen resolution etc.). The remote website then can use all of this information to uniquely identify the browser that the request comes from. However, because the overhead that incurs is very high for browser fingerprinting, we would make the argument that third-party trackers are unlikely to adopt it as a means to track the browsing behaviors of users.

8 Conclusion

We present TrackAdvisor, a Machine Learning-based method designed to detect third-party trackers and become the basis for protecting the users' privacy from third-party trackers. TrackAdvisor's novelty is its focus on the interactions between the browsers and the remote websites to detect when the user's browsing privacy is being leaked instead of relying on black lists. TrackAdvisor exhibits high Precision (99.4%) and Recall (100%) in contrast with a Recall of 72.2% by Microsoft's Tracking Protection Lists, which is a black list-based component in the widely used Internet Explorer.

Towards protecting user privacy, we evaluate two potential countermeasures: a) removing user identity in tracker cookies and b) removing the referrer information from the HTTP requests sent to third-party trackers. We find that the second method achieves the goal of protecting user privacy while not "breaking" the functionalities of the web pages.

Finally, we present a study on the pervasiveness of third-party trackers. Our study shows that 46% of the websites on Alexa's Global Top 10,000 list contain at least one tracker each and 25% of the 10,000 are tracked by a single entity: Google, as its `doubleclick` ad service is very popular and many websites use the code libraries provided by Google itself to add functionalities.

References

1. AdBlock Plus. `https://adblockplus.org`.
2. Alexa, the Web Information Company. `http://www.alexa.com`.
3. Collusion. `https://chrome.google.com/webstore/detail/collusion-for-chrome/ganlifbpkcplnldliibcbegplfmcfigp`.
4. FourthParty Firefox Extension. `http://fourthparty.info`.
5. Ghostery. `https://www.ghostery.com/`.
6. HTTP ETags. `http://en.wikipedia.org/wiki/HTTP_ETag`.
7. Microsoft's Tracking Protection Lists. `http://ie.microsoft.com/testdrive/Browser/p3p/Default.html`.
8. Privacy Badger. `http://www.theregister.co.uk/2014/05/02/eff_privacy_badger/`.
9. Selenium, Web Browser Automation. `http://docs.seleniumhq.org/`.
10. Third-party iFrames can no longer read their own cookies when "Block third-party cookies and site data" is enabled. `https://code.google.com/p/chromium/issues/detail?id=113401`.
11. Yahoo declines to honor "Do not track". `http://yahoopolicy.tumblr.com/post/84363620568/yahoos-default-a-personalized-experience`.
12. Ayenson, M., Wambach, D., Soltani, A., Good, N., and Hoofnagle, C. Flash cookies and privacy II: Now with HTML5 and etag respawning. *Social Science Research Networks* (2011).
13. Eckersley, P. How unique is your web browser? In *Privacy Enhancing Technologies* (2010), Springer, pp. 1–18.
14. Leon, P., Ur, B., Shay, R., Wang, Y., Balebako, R., and Cranor, L. Why Johnny can't opt out: A usability evaluation of tools to limit online behavioral advertising. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2012), ACM, pp. 589–598.
15. Mayer, J. Tracking the Trackers: Self-help tools. `http://cyberlaw.stanford.edu/node/6730`.
16. Mayer, J. R., and Mitchell, J. C. Third-party web tracking: Policy and technology. In *Security and Privacy (SP), 2012 IEEE Symposium on* (2012), IEEE, pp. 413–427.
17. McDonald, A. M., and Cranor, L. F. A survey of the use of adobe flash local shared objects to respawn http cookies. *A Journal of Law and Policy for the Information Society 7* (2012), 639–721.
18. Weinberg, Z., Chen, E. Y., Jayaraman, P. R., and Jackson, C. I still know what you visited last summer: Leaking browsing history via user interaction and side channel attacks. In *Security and Privacy (SP), 2011 IEEE Symposium on* (2011), IEEE, pp. 147–161.
19. Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., and Cunningham, S. J. WEKA: Practical machine learning tools and techniques with Java implementations.