Synthoid: Endpoint User Profile Control

Marcel Flores
Department of EECS
Northwestern University
Evanston, Illinois 60208
Email: marcel-flores@u.northwestern.edu

Aleksandar Kuzmanovic
Department of EECS
Northwestern University
Evanston, Illinois 60208
Email: akuzma@cs.northwestern.edu

***Abstract*—Web tracking is a practice that has grown in scope and complexity as the Internet has expanded. In particular, advertising networks have been continually advancing web tracking in order to gather the most information on users. On the other hand, users have often reacted negatively to tracking schemes, expressing concern about their lack of control. We present Synthoid, a system which returns control to users, without a need for tracker cooperation or broad adoption.**

Synthoid achieves this by actively fetching content of a selected set of topics, imprinting the tracking network profile for the user with these topics. It eliminates the need for users to place trust in web trackers by directly controlling the signal trackers measure, without inconveniencing users. Additionally, as traffic is generated directly from the users' machines, Synthoid is able to function with any method of tracking, even those which use advanced techniques.

We examine the effectiveness of Synthoid with real web trackers. We demonstrate that it is successfully able to imprint a target profile with low volume and when faced with significant interference. Finally, we show that it is able to entirely alter browsing profiles when run alongside user traces. The key behind Synthoid's performance is a strong, artificial, yet carefully constructed and semantically determined, signal.

I. Introduction

In recent years the field of user tracking on the web has grown significantly. Beyond simply analytics, advertising networks have been expanding their capabilities with the goal of most efficiently presenting effective ads to users [17], [20]. These advancements have come in many forms, from context aware advertising to the building of user profiles to deliver custom ad experiences [11]. In order to build a meaningful user profile, these networks have developed complex techniques to gather user information.

Many of these techniques have created tension with users and consumer advocates, who are concerned about the collection of so much information in a manner which users are unable to control and which reveals a portion of a user's browsing history. As a result, users have pursued the development of methods to counter the collection of this information. These methods include the blocking of tracking content through web browser plugins such as AdBlock [1], as well as more aggressive methods, such as regularly clearing browser cookies and local storage data. However, many of these techniques complicate the users' web browsing experience, causing many sites not to function correctly. This can range from lost login sessions, often requiring painful re-authorization procedures, to sites simply refusing to operate if the browser does not allow cookies. Furthermore, these approaches only address a limited number of tracking mechanisms, leaving trackers to develop new methods.

In this paper, we present a system which returns control of a user's appearance directly to the user. We present Synthoid, which allows a user to select a set of topics and automatically visits sites of these topics from the user's machine. Doing so actively imprints the user's desired topics in tracking profiles. Such a setup allows the user to place whatever information they desire into these profiles by targeting tracking at the source. This method does not force users to place trust in the tracking networks.

Since the system functions by generating direct traffic, it stands to work with *all* networks. As traffic is generated directly from the user's machine, Synthoid can function with any method of tracking, even those which use techniques beyond simple cookie interactions, *i.e.*, [33]. By generating web traffic, Synthoid is able to control these systems, filling the tracking history with traffic which reflects the user's desired behavior. By browsing only sites of focused topics, Synthoid generates strong signals to trackers.

Under pressure from regulators, some of the largest networks, *e.g.*, Google (DoubleClick), Yahoo!, and Microsoft, provided interfaces for users to view or edit their profiles. However, there are several issues. First, users are left to trust in the effectiveness of these interfaces, but have no means to audit their effectiveness. Users have no way of knowing which parts of their browsing history are ultimately stored by the networks, used for selecting advertisements, or sold to other organizations. Second, nothing compels the networks to honor these requests. Worse still, they have significant incentive not to honor them, as limiting the information they store about a user can dramatically reduce their revenue [3], [9], [10]. Third, numerous other trackers do not even provide such interfaces, leaving users without any mechanism to regulate their profiles. Synthoid offers recourse in the above scenarios, allowing users to influence their profile with information of their choosing. This is done comprehensively for all trackers, yet without placing any trust in, utilizing mechanisms from, or expecting the cooperation of tracking networks.

We demonstrate experimentally that Synthoid is successfully able to imprint user profiles. First, we demonstrate that Synthoid is able to rapidly generate a successfully imprinted profile in under 2 days with low browsing traffic. Second, we explore the effect of adjusting the amount of interference on the ability of Synthoid to imprint profiles and find that even when faced with 8 times its volume in focused interference,

no less than half of target topics still appear in the interest profiles. Third, we use traffic traces from actual users and demonstrate that when run alongside these traces, Synthoid is successfully able to control the profiles as desired and is capable of generating an entirely disjoint profile. Finally, we consider the generality of our system, and find that it is effective at imprinting profiles in multiple tracking services.

In the next section, we present a brief background on web tracking and the current approaches to consumer protection. We then describe the design and operation of our system in Section III. We provide an evaluation of the effectiveness of our system in a number of key scenarios in Section IV. Finally we discuss the generalizability of Synthoid in Section V, and conclude in Section VI.

II. Background

Web tracking has become prominent in today's Internet, and one of the primary mechanisms and purposes for web tracking is advertising. The increased importance of advertising in the Internet economy has put advertisers in a position to collect information on users behavior across the web.

For example, suppose there are two websites, both of which contain advertisements from the same ad tracking network. If the sites managed their ads themselves, each site could only collect information about its own users. Now, however, the presence of the tracker on both sites means the user will download content from the ad network's servers. If the tracker is able to reliably identify users, for example by cookies, flash data, or browser fingerprint, they are able to recognize whenever a particular user has visited a participating site. This information is then processed by the tracker and becomes part of the calculation of the user's profile using the trackers categorization of the site, potentially indicating that the user is interested in the topics of the participating sites.

This careful design allows popular advertisers and trackers to collect large amounts of information from users. Since there is very little cost associated with this tracking, they can be created any time a user who doesn't have a profile is detected. Since the users need never directly interact with any part of the system, it can operate over long periods of time without the users' knowledge or consent.

The ecosystem of trackers is subject to significant variation [26]. Some exist as large advertising networks, managing the collection and analysis of user data, the generation of ad profiles, and the distribution of ads. Other services are more focused, performing only tracking and analysis. Others still are primarily social networks. For the purposes of Synthoid, these models are effectively the same, as they all collect and analyze user browsing information. Therefore, we refer to such networks and trackers interchangeably.

A. Current Approaches

Users who do not wish to participate have explored various techniques to interfere with tracking. Common techniques include blocking certain domains (i.e., preventing the loading of `iframes` from known trackers), preventing certain JavaScript from executing without user permission, and blocking cookies [26]. However, such steps may inhibit the proper functioning of a site, either because the site detects that ads are being interrupted, or the site depends on these technologies for normal operation. Furthermore, trackers are left to develop replacement techniques.

Several existing solutions have proposed systems which sit between advertisers and users, preventing information from flowing freely between them, often by moving interest mining to users [5], [8], [12], [25], [31]. While these systems appear effective, some require infrastructure, such as auction platforms, proxies, or hardware, in order to function. Even the most scalable systems require the deployment and maintenance of such infrastructure. Worse still, [25] requires a trusted third party to operate portions of its infrastructure. Finally, all of these systems depend on the willing participation of ad networks. Since advertisers and trackers stand to gain the most information at the least cost with the current model, they have no incentive to be involved in the deployment of such systems.

The Do Not Track header [7], and similar regulatory efforts, rely on the cooperation of tracking networks or outside regulatory bodies. Moreover, the technology of tracking often outpaces such efforts. Additionally, it is not unheard of that a tracker may simply ignore such mechanisms, as they only restrict the information the tracker can collect [3]. A lack of effective auditing methods mean that users are unable to detect when such systems are being ignored.

Services, such as TRUSTe and The Network Advertising Initiative [23], [32], have attempted to provide a central interface by which users are able to opt-out of tracking. However, users are only able to opt-out on select services. Furthermore, opting-out is generally the only choice: users are not able to exert any real control over their profiles.

Other attempted solutions have considered the approach of poisoning users' profiles by pooling tracking cookies among several users, attempting to provide a protection by aggregating user behavior [6]. However, such solutions are extremely easy to detect, as a tracker could easily observe a single cookie performing rapid travel between physically distant locations. Moreover, even when cookies are not used, accurate fingerprinting is still feasible [33], allowing trackers to develop new mechanisms to subvert such tools.

In light of these restrictions and challenges, Synthoid runs directly on a user's computer. It allows a user to modify their tracking profile as desired by generating synthetic traffic traces designed to be indistinguishable from a user's regular traffic from the network's perspective. By running solely on the user's machine, Synthoid avoids the need for tracking network participation and instead takes advantage of existing tracking methods. It does not require that the user perform any manipulation to the low level communications, or otherwise interfere with their browsing experience.

III. Synthoid

In order to empower the user to determine their own online profiles in a universal and effective manner, we present Synthoid. Synthoid generates web traffic alongside a user's normal web browsing. It does so at a sufficient rate and consistency that any third party observing the traffic is left with information generated both by the user's behavior and

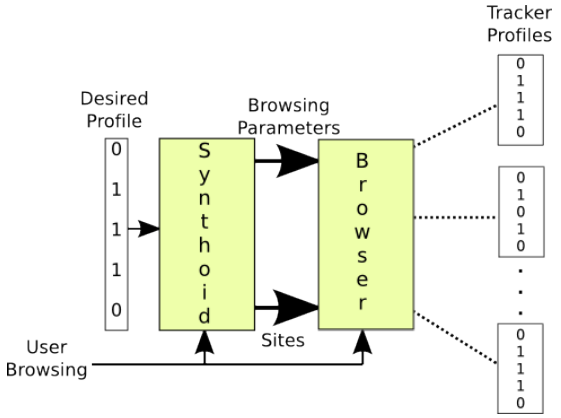

Fig. 1. Synthoid system design. The binary vectors represent potential interest profiles.

information the user chose to place in their profile, regardless of the tracking mechanism. This allows the user to control the trackers impression of their interests on two levels. First, in terms of the aggregate profile, the user can imprint topics entirely separate from their real interests. At the individual site level, the tracker is unable to distinguish real from synthetic traffic components. Together, these allow the user to control the tracker's perception of their interests and behaviors.

A. Overview

Figure 1 presents an overview of Synthoid. In order to use Synthoid the user selects a set of topics from an available list, or allows Synthoid to select them at random, which they would like trackers to observe. Doing so designates a *desired profile*. Synthoid then selects an appropriate set of sites from the indicated categories, as well as control parameters dictating the browsing behavior. We describe the site selection process in more detail in Section III-A1. Synthoid then passes these inputs to the browser. Details of how the browser performs these visits and the techniques used to generate a regular traffic pattern are discussed in Section III-A2.

1) Site Selection: In order for our system to generate the described traces, we require a database of websites that are semantically categorized. We chose the Open Directory Project (ODP) to seed our traces. The ODP claims to be the largest such web directory with over 5 million sites, and its data are easily accessed. We note that our system is designed to work with any such classification, and is not limited to the ODP.

The system builds a list of sites from the ODP database which match the *designated profile*, building a separate list for each of the input categories. In order to provide consistency in its traces, Synthoid allows the user to specify the number of sites to use for each topic list. Setting the list size to a low value causes the system to visit individual sites more frequently. A larger value allows the visits to be spread across a greater number of sites, increasing the variety for a topic. We explore the effects of this parameter in Section IV-C. These lists persist for the duration of the operation of Synthoid and form the pool of sites which the system is able to visit.

2) Browsing: To begin browsing, Synthoid randomly selects one of the provided topics. Next, the system randomly samples a single site from the list corresponding to that topic and loads it. After the page has loaded completely, the browsing module pauses for an interval sampled from an exponential distribution. It then selects a random link from the loaded page and follows it. Once the new page has loaded, a new link is selected and the process continues. The mean of the pause-time distribution, and the number of links followed in this fashion are both tunable parameters to the system. In our evaluations in Section IV we use a mean of 5 seconds and follow 4 links. After the final link, the system repeats, selecting a new topic and corresponding page.

By default the system performs the above browsing pattern every 2 hours between 8 AM and 12 AM, though these times are user editable. The user is further able to set how much traffic is generated by specifying a duty cycle. In particular, the user is able to specify what fraction of the 2 hour period is devoted to browsing, including page load time and the time between links. A smaller duty cycle corresponds to a lower number of sites visited. The browsing is performed at the beginning of each period; once the duty cycle is reached, the system refrains from selecting a new topic and site until the start of the next period. We explore the effects of adjusting the duty cycle experimentally in Section IV-B. The flexibility of these parameters allows users to configure the system so that it closely replicates their usual browsing behavior.

Throughout the process, the browsing module also maintains the cookies and browser state for the trace, ensuring consistent identification for trackers. In our implementation, all system controls are implemented in Python. For browsing, a version of Webkit [30] was used for its ease of customizability. However, Synthoid can be configured to use the Selenium Browser Automation Tool [27] to perform the trace with the the user's default browser, allowing the user browser fingerprint to match exactly. Alternatively, a simple browser plugin could be used to perform the browsing in the background.

B. Feedback

1) Tracker Feedback: To demonstrate that our system is successfully manipulating the user profile, we require the ability to see the profile that a tracker has generated. However, this is not a standard feature of trackers. Even in the rare case in which viewing the user profile is possible, significant effort may still be required. For example, Microsoft requires the creation of an account to view one's profile. At the time of writing, this left only Yahoo!, Blue Kai, and DoubleClick as viable options. We have chosen the DoubleClick network as our primary source of feedback. To ensure that our system is as general as possible, we confirm in Section IV-F that our imprints were indeed successful in the other networks.

Figure 2 provides an example of the type of profile which DoubleClick makes visible to the user. The profile comes in the form of a list of topics in a shallow hierarchy that is generally 3 or 4 topics deep. The top level of this hierarchy is made up of 25 categories, covering a wide variety of possible user interests. We refer to the profile that we have collected from the tracker as the *observed profile*.

2) Scoring System: In order to measure the performance of our system, we developed a method to determine how well the observed profile matches the desired profile. This

Fig. 2. An example interest profile from DoubleClick. A number of category hierarchies can be observed.

task is complicated by the fact that many sites actually fall into multiple topic categories and may therefore occur with sufficient regularity that these secondary topics are imprinted on the profile. As the appearance of additional topics is part of the goals of Synthoid, we require that our scoring system not count their appearance negatively, accepting that extra topics are a normal feature.

Topic Mapping The task of developing a scoring system is further complicated by disparities between the topics in the ODP and the observed DoubleClick topics. In order to remove the need to map all possible sub-categories between the two models, we consider only the top-level categories. We map the input ODP categories to a top-level category from the DoubleClick model, as the latter is more permissive. The inputs are mapped using longest prefix matching, so the input topic in "Arts/Literature" would map to the DoubleClick top-level category "Books and Literature", where "Arts" would map simply to "Arts and Entertainment".

Cosine Similarity We represent the user profiles with a binary vector. Each dimension of the vector corresponds to an input topic that has been mapped to its top-level DoubleClick counterpart. First, the vector is initialized to 0. If the topic appears in the resulting profile, a value of 1 is set in the corresponding dimension. We then generate an ideal vector of the same dimensionality, where we set the value of the desired topic dimensions to 1. We then compute the cosine similarity of these vectors, therefore measuring how similar the resulting observed profile is to the input desired profile. Later, we will refer to the *score* of evaluations to mean this cosine similarity.

IV. Evaluation

In order to further understand how Synthoid would function in real world use scenarios, we perform a number of experiments, generating actual web traffic and measuring the resulting profiles. In particular, we hope to understand how Synthoid must operate in order to successfully imprint the user selected topics on the tracking profile.

A. Topic Selection

As a preliminary step, a desired profile was selected. In order for this selection to offer a proper representation of the functionality of Synthoid we performed a random sample, without replacement, of 10 topics from the set of all topics on the ODP which were no more than 2 layers deep (i.e. /Topic/Subtopic) and contained at least $2,000$ sites. To obtain a better understanding of the resulting profile, we apply the measurement methods from [26] to measure the rate of appearance of DoubleClick ads. We present our selected topics and their ad prevalence rates in Table I. The rates offer a significant spread: from 6% of sites having DoubleClick ads in "Marketing and Advertising" to 42% in "Baseball." We make use of these chosen topics throughout our evaluations.

Topic	Ad Rate
Arts - Performing Arts	31%
Business - Accounting	6%
Business - Marketing and Advertising	6%
Health - Conditions and Diseases	24%
Home - Food	8%
Home - Gardening	18%
Recreation Travel	15%
Science - Astronomy	14%
Sports - Basketball	40%
Sports - Baseball	42%

TABLE I. The selected topics and their prevalence of DoubleClick ads in a sample of 200 sites.

B. Duty Cycle

First, we consider the effect adjusting the total amount of browsing performed by Synthoid. We consider 6 duty cycle settings, which range from 1% to 100%. In all cases, Synthoid was given the previously described profile. We further set Synthoid to visit 4 links on each site and set the mean of the wait-time distribution to 5 seconds. Each instance ran for 1 week, after which the profile was collected every 30 minutes for an additional day.

Figure 3 shows the average scores over 3 identical instances of each type at the end of the final day of collection, where the error bars indicate the standard deviation. We see that the system is able to strongly influence the profile in all cases and is also able to implement a target profile even at lower duty cycles. Furthermore we note there is very little variation between the rates, each scoring in a very similar range.

While every topic succeeded in appearing in the profile of at least one instance, certain topics have some difficulty, failing to appear more often than not: "Astronomy" and "Conditions and Diseases." While their appearance at all indicates that they are capable of being imprinted, we suspect they failed to appear due to competition with more common topics. In particular, the "Arts and Entertainment" and "Business" categories dominated the profiles, accounting for as much as half of the topics in the profile. While we chose to keep the topic ratios equal for these experiments, Synthoid can account for this by increasing the fraction of traffic devoted to less popular topics.

For additional perspective, we consider the score for each duty cycle over time. To do so, we examine profile observations that were taken after periods of inactivity, to avoid including transient profile features in the score.

Figure 4 shows the scores over the course of the experiment.[1] We see that all duty cycles except the 1% reached a

[1]For clarity, we have selected a single run of each duty cycle, but note that all runs of the same duty cycle behaved in a similar fashion.

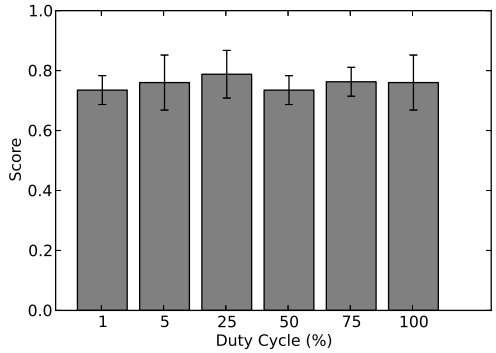

Fig. 3. The average score for instances of varying duty cycle. The system is able to imprint the majority of desired topics at low volumes.

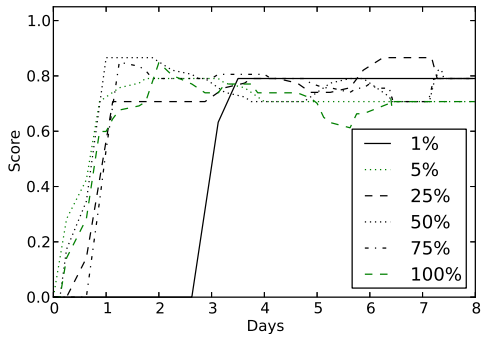

Fig. 4. The score over the course of the week for various Synthoid duty cycles. Only the 1% duty cycle run seems to lag behind.

steady plateau after only a single day of operation. The 1% instances took slightly longer, requiring a full 3 days. Rather than the profile failing to imprint correctly at low rates, it merely takes longer for the tracker to detect. We see at duty cycles as low as 5%, the system was already able to influence the profile as fast as all other duty cycles. Therefore the system is able to achieve its fastest rate at relatively low traffic rates.

C. Restricted Site List

The next parameter that we consider is the size of the topic lists. As mentioned in Section III-A1, a natural hypothesis is that trackers pay particular attention to consistency: if a certain site is visited regularly, its corresponding topics are more likely to appear in the tracking profile than if many different sites of the same topic are visited over time.

In order to explore the potential effects of this parameter, we perform week-long runs of Synthoid using varying sizes of the pool of sites per topic, considering 25, 50, 100, 200, 400, and 800 sites. The experiments were performed at a duty cycle of 37.5%, with all other parameters as they were in the previous experiment and each repeated 3 times. Recall that a list size of 25 means that the system has 25 possible sites to choose from for each topic, the number of pages loaded is still determined by the duty cycle and load times.

We find that the scores are largely unaffected by the list size parameter, achieving an average score of 0.74 across all runs, with a 16% maximum difference from the mean and a variance of 0.8%. These observations suggest that Synthoid can successfully operate with a relatively small list. This has several direct benefits: First, it greatly reduces the footprint of Synthoid, as it need not store large lists for each topic. Second, the task of generating a semantic classification of websites becomes easier, as a large set is no longer necessary.

We observe no significant difference between the list sizes when we consider their scores over the week. As in the previous experiments, the system takes a single day to reach a stable profile. This suggests that reducing the list size will not negatively affect the time it takes to imprint a profile.

In light of these observations, we use a list size of 100 for the remainder of the experiments, as we saw it should not affect performance, while reducing the resources used by the system and still providing sufficient tracker occurrences.

D. Interference

Next, we explore how the system performs when there is additional web traffic using the same cookie, such as the user's normal browsing, by exposing Synthoid to different rates of interference. In order to properly understand these effects, we consider two experimental setups. The first maintains a fixed amount of traffic devoted to imprinting the profile, and considers increasing amounts of interference (Section IV-D1). The second maintains a constant total traffic, but considers increasing proportions of traffic devoted to interference (Section IV-D2).

1) Total-Traffic Dependent Analysis: For this experiment, we divide Synthoid's efforts in two: one devoted to imprinting the same selection of topics used previously; and a second devoted to a set of 5 topics randomly selected from the remaining ODP topics not used in the first 10. We begin with system generating equal amount of traffic for all 15 topics. As before, we allow Synthoid to run for a week and collect profile information for 1 additional day. We then repeat the setup on a fresh cookie with double the duty cycle of the interference topics, then consider 4× the interference, and finally 8×. When running at equal rates, the system operates with a duty cycle of 25%. As we want the score to measure the occurrence of the original 10 topics despite the addition of the interfering traffic, we exclude the additional topics from the desired profile when calculating the score for these experiments.

Figure 5 shows that there is very little change across the levels of interference. Even in the face of up to 8 times its volume in interference, Synthoid is still successfully able to imprint a portion of its target topics. On average, Synthoid imprinted 0.5 topics fewer than when running unencumbered. Observing the profiles indicated that the the size of the profile was quite large, often up to 20 topics. This suggests the trackers are responding to the diverse high volume traffic, allowing for a larger, more varied profile.

Figure 6 indicates a slight delay in the first 3 days of profile imprinting. After day 3, the system again reached a stable profile. When faced with 4× and 8× as much interference, the initial startup was particularly slow, and the scores did not recover to quite as high a level. However, all other setups were able to achieve successful profile imprinting similar to those seen when Synthoid was run alone.

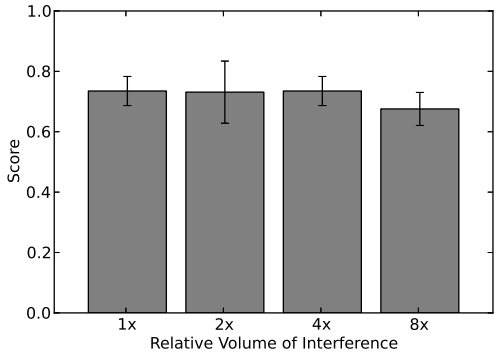

Fig. 5. The average scores for increasing volumes of interference topics. The final score does not seem to be significantly affected.

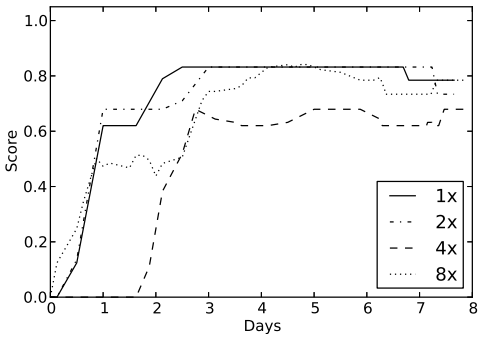

Fig. 6. The scores of total-traffic dependent interference runs over the course of the week. The $4\times$ and $8\times$ runs took an additional day to imprint.

2) Total-Traffic Independent Analysis: As a further analysis on the effects of interference, we run Synthoid at a duty cycle of 50% and alter the proportion devoted to interference traffic. As we increase the proportion of browsing devoted to interference, the traffic devoted to the target topics decreases. This allows us to study the effects of interference separate from changes that happen as a result of changing entire system browsing rate.

Figure 7 shows the performance of Synthoid for each rate of interference. While we see a small dip in the performance of the runs with $4\times$ the interference, we note that it seemed to recover in the $8\times$ case. In these interference scenarios, Synthoid imprinted an average of 0.9 topics fewer than when running alone.

Considering the scores over time, we observed that in all cases the profiles took between 2.5 and 3 days to reach steady state, generally staying near 0 until the second day. The scores for all levels of interference were marginally lower than those seen in the unhindered runs. This is likely caused by a combination of heavy interference and lower volume.

E. Case Studies

As is often the case, there is potential for significant complexity to be added when considering actual user traffic, rather than synthetic traffic. In order to try and understand how Synthoid performs when running alongside real web traffic, we perform a number of case studies using traffic generated by

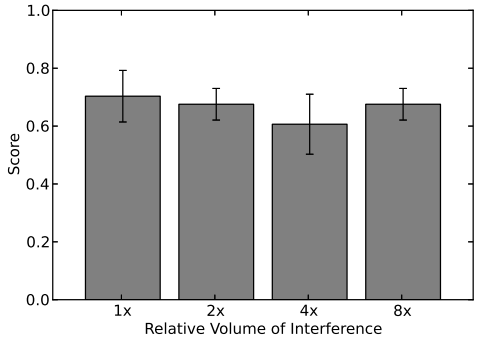

Fig. 7. The average scores with fixed total traffic and variable ratio of interference. The decrease at $4\times$ seems to be noise.

User	Number of Pages	Unique Domains
1	2019	251
2	559	92
3	1031	186
4	1772	120
5	2369	147

TABLE II. AN OVERVIEW OF THE CHARACTERISTICS OF THE USER SUBMITTED TRACES.

actual users. A small set of users anonymously submitted traces of their last 7 days of traffic. Table II contains an overview of the submitted user traces. We see that the number of sites visited varies dramatically between users. We also estimate the number of unique domains in the set using the current Mozilla public suffix list [24]. The rate of unique domains varies widely between users, suggesting significant variation in how each of the users browses the internet.

We recreate each user trace on two fresh DoubleClick cookies. The first is done as a control in order to see what profile results from only the user's behavior. The second is performed alongside Synthoid running on the original set of 10 randomly sampled topics which we used previously. All browsing parameters were set to match the previous runs, and the system operated at a duty cycle of 25%.

Figure 8 shows the final score of Synthoid when used alongside the human traces. Despite the variation in the human behavior, Synthoid regularly scored above 0.8, imprinting most of the profile. Furthermore, there is no significant difference between the runs: Synthoid functioned equally well alongside all 5 traces.

Figure 9 indicates Synthoid is again rapidly able to imprint the profile. All instances performed similarly, despite relatively large variations in the number of pages and domains in the user trace. Such performance suggests robustness to varied user behaviors. Moreover, Synthoid performed comparably to previous unencumbered experiments, suggesting that it was relatively unaffected by the human traffic.

Manual comparison of the topics in the final profiles revealed significant differences between the profile produced by recreating the user trace alone and alongside Synthoid. In particular, all runs for users 1, 2, and 3 generated entirely disjoint profiles, sharing no topics with the original. Users 4 and 5 shared 2 topics with their original profiles, however this was the result of these topics, forms of "Food" and "Performing Arts", both being in Synthoid's target profile.

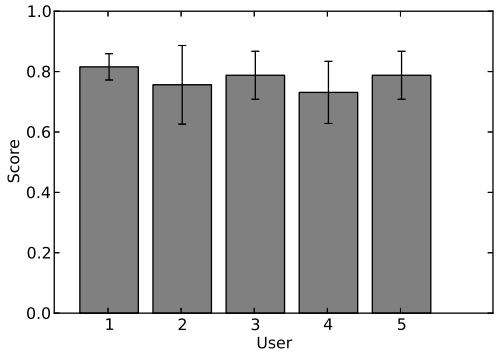

Fig. 8. The average scores of Synthoid when run alongside human traces. Strong performance is seen for all 5 users.

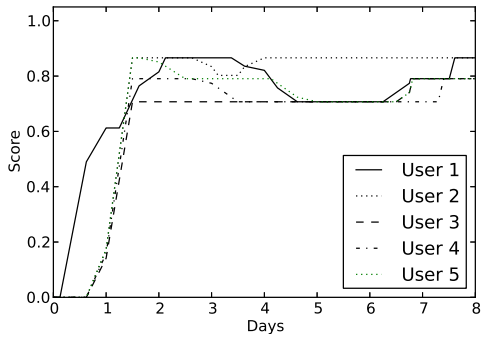

Fig. 9. The scores over time for Synthoid alongside each human trace. A stable imprint is achieved after only 1.5 days.

F. Other Trackers

In order to understand how our system interacts with other ad trackers, we also consider the resulting profiles from the Yahoo! and Blue Kai ad networks. In a sample of Yahoo! profiles resulting from Section IV-B, we found that the majority of the profile was imprinted, with "Travel," "Performing Arts," and "Accounting" or "Marketing and Advertising" appearing in all checked profiles. "Baseball" or "Basketball" appeared in a majority of checked profiles. Surprisingly, "Food" and "Gardening" failed to appear more often than not, topics DoubleClick had no difficulty with, suggesting differences in the types of sites which use each tracker. "Astronomy" proved the most difficult, never appearing in a profile. Samples of profiles from Section IV-D revealed nearly identical results.

In examining the Yahoo! profiles, we noted that the profiles were in general slightly noisier than their DoubleClick counterparts, often including seemingly unrelated topics which did not appear in our earlier analysis. This suggests that the algorithms employed by each network vary significantly, and likely the set of participating sites differ greatly. Despite this difference, Synthoid functions well in both cases.

A review of the Blue Kai profiles of the same experiments revealed much smaller profiles, the largest containing only 11 topics, and frequently containing only 3 or 4. Various forms of "Performing Arts" and "Accounting" appeared in nearly every profile checked. "Sports" occurred in roughly half the observed profiles. Similar topics, such as "Software" and "Online communities" were observed as well, but in general there were very few additional unrelated topics. Again, we suspect these differences arise largely from differences in the sites visible to the tracker and the specific algorithms they employ. While these profiles were much smaller, their contents were dominated by topics targeted by Synthoid.

V. Discussion

A. Generalizability

Central to Synthoid's design principles is its ability to operate effectively on any network, regardless of the tracking method and profile creation algorithm. While we have demonstrated its effectiveness for networks which offer feedback, many networks offer users no control or information. However, Synthoid should still prove effective in these environments as long as it is able to visit sites which lie in the appropriate behavioral tracking networks.

We demonstrated that Synthoid can alter a tracker profile completely. Still, all of a user's traffic remains in the tracker's system. Nonetheless, in the presence of careful synthetic traffic, trackers are unable to differentiate between such traffic, causing the tracker to effectively discard certain user interests.

Another advantage of Synthoid's endpoint design is its ability to cooperate with fingerprinting techniques. Since it operates closely to the user, techniques that attempt to measure peculiarities of a user's browser will recognize that Synthoid's traffic comes directly from the user, and therefore treat it's traffic as the user's. Even a third party snooping on a user's traffic on a local network would be unable to readily differentiate between a user's normal traffic and Synthoid traffic .

We showed Synthoid is capable of imprinting user profiles with 25MB/day. Such volumes would be reasonable even in a mobile setting. Further adjustments could be made to optimize mobile use, such as only browsing when connected to Wi-Fi or while charging, preventing wasteful data and power use.

B. Related Work

There has been a significant body of work studying the interactions between users, advertising networks, and online tracking services. Jensen, et al. [17] developed a web crawler that hunted for the presence of trackers and other user tracking mechanisms. Others have studied the prevalence and mechanisms by which user information can be collected [20]. In particular, many of these consider methods beyond traditional cookies, including Flash, HTML5, and JavaScript [4], [16], [29]. More recently, Roesner, et al. [26] examined several of the mechanisms used by popular trackers. They further measured the prevalence of tracking behaviors on popular websites. Yen, et al. [33] considered how much personal information can be revealed via information collected by trackers. While understanding the methods and prevalence of trackers is important, Synthoid operates in a generic fashion that is able to inform any tracker which listens to a user's traffic, regardless of the particular mechanism.

A handful of solutions have been proposed which aim to resolve the conflict between the needs for effective advertising and for user control. These range from moving the actual tracking to the users local machine [8], [18], [31] to developing

intermediary layers or auctions for releasing user data [5], [12], [25]. However, these solutions require participation on the part of advertising networks. Synthoid overcomes these requirements by making use of the tracking systems currently in place to imprint trackers with a user selected profile and therefore needs no additional structure. Since the traffic appears as ordinary web traffic, trackers themselves need not opt-in to the system, so no explicit cooperation is required.

More generally a significant body of work has been devoted to the study of user web information control, studying how a user's information may be leaked through a number of vectors, from simple browser partitioning [2], [14], [15], [28] to the use of online social networks and other services [22], [13], [19], [21]. Synthoid works to return control directly to the user, allowing them to control external profiles of their information.

VI. Conclusions

We have presented Synthoid, a system designed to imprint web trackers with information selected directly by users. Synthoid imprints a user's desired profile on tracking profiles by regularly visiting a series of web pages that match the user's selected topics. This endpoint design enables a user to select the contents of their profile for all trackers from a single interface. Furthermore, this approach is not limited to those trackers which currently allow editing and functions for any service which observes a user's traffic. Finally, it does not require users to place trust in the trackers, allowing users to provide the tracker with information of their choosing.

We have demonstrated Synthoid's effectiveness and robustness to interference through a number of experiments using the profiles generated by a real world tracker and real web traffic. Furthermore, we showed that Synthoid can completely alter a profile when run alongside collected web traces from human users. Finally, we showed that the system is general, and functions well in multiple tracking services. It is able to do so by generating strong, synthetic signal in the form of browser traffic. Synthoid provides users with a direct methodology which can be implemented and employed effectively today.

Acknowledgements

This project is supported by the National Science Foundation (NSF) via grant CNS-1319086.

References

[1] AdBlock Plus, http://adblockplus.org/en/.

[2] G. Aggrawal, E. Bursztein, C. Jackson, and D. Boneh, "An analysis of private browsing modes in modern browsers," in *Proc. of 19th Usenix Security Symposium*, 2010.

[3] J. Angwin, "Google, ftc near settlement on privacy," *The Wall Street Journal*, July 2012.

[4] M. Ayenson, D. Wambach, A. Soltani, N. good, and C. Hoffnagle, "Flash cookies and privacy II: Now with HTML5 and etag respawning," in *Social Science Research Network Working Paper Series*, 2011.

[5] M. Backes, A. Kate, M. Maffei, and K. Pecina, "Obliviad: Provably secure and practical online behavioral advertising," in *Proc. SP 2012*, 2012, pp. 257–271.

[6] "Cookie Cooker," http://www.cookiecooker.de/index_alt_en.html.

[7] Federal Trade Commission, "Protecting consumer privacy in an era of rapid change," March 2012.

[8] M. Fredrikson and B. Livshits, "Repriv: Re-imagining content personalization and in-browser privacy," in *Proc. SP 2011*, 2011.

[9] P. Gill, V. Erramilli, A. Chaintreau, B. Krishnamurthy, K. Papagiannaki, and P. Rodriguez, "Follow the money: Understanding economics of online aggregation and advertising," in *Proc. IMC '13*, 2013.

[10] A. Goldfarb and C. Tucker, "Privacy regulation and online advertising," *Management Science*, 2011.

[11] Google, "How it works: Ads help," http://support.google.com/ads/bin/answer.py?hl=en&answer=2662749, September 2012.

[12] S. Guha, B. Cheng, and P. Francis, "Privad: practical privacy in online advertising," in *Proc. of NSDI '11*, 2011.

[13] K. Gummadi, B. Krishnamurthy, and A. Mislove, "Addressing the privacy management crisis in online social networks," in *Proceedings of the IAB Workshop on Internet Privacy*, 2010.

[14] C. Jackson, A. Bortz, D. Boneh, and J. Mitchell, "Protecting browser state from web privacy attacks," in *Proceedings of the 15th international conference on World Wide Web*, ser. WWW '06, 2006.

[15] A. Janc and L. Olejnik, "Feasibility and real-world implications of web browser history detection." in *W2SP*, 2010.

[16] D. Jang, R. Jhala, S. Lerner, and H. Shacham, "An empirical study of privacy-violating information flows in javascript web applications," in *Proceedings of the 17th ACM conference on Computer and communications security*, ser. CCS '10, 2010.

[17] C. Jensen, C. Sarkar, C. Jensen, and C. Potts, "Tracking website data-collection and privacy practices with the iwatch web crawler," in *Proceedings of the 3rd symposium on Usable privacy and security*, ser. SOUPS '07, 2007.

[18] A. Juels, "Targeted advertising ... and privacy too," in *Proceedings of the 2001 Conference on Topics in Cryptology: The Cryptographer's Track at RSA*, ser. CT-RSA 2001, 2001.

[19] B. Krishnamurthy and C. Wills., "Characterizing privacy in online social networks," in *Proceedings of the first workshop on Online social networks*, ser. WOSN '08, 2008.

[20] B. Krishnamurthy and C. Wills, "Privacy diffusion on the web: a longitudinal perspective," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09, 2009.

[21] B. Krishnamurthy and C. Wills., "Privacy leakage in mobile online social networks," in *Proceedings of the 3rd conference on Online social networks*, ser. WOSN'10, 2010.

[22] B. Krishnamurthy, K. Naryshkin, and C. E. Wills, "Privacy leakage vs. protection measures: the growing disconnect." in *W2SP*, 2011.

[23] Networking Advertising Initiative, http://www.networkadvertising.org.

[24] Public Suffix List, http://publicsuffix.org/.

[25] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez, "For sale : your data: by : you," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*, ser. HotNets-X, 2011.

[26] F. Roesner, T. Kohno, and D. Wetherall, "Detecting and defending against third-party tracking on the web," in *NSDI'12: Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, 2012.

[27] Selenium, http://seleniumhq.org.

[28] K. Singh, A. Moshchuk, H. Wang, and W. Lee, "On the incoherencies in web browser access control policies," in *Proc. SP 2010*, 2010.

[29] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. Hoofnagle, "Flash cookies and privacy," in *Social Science Research Network Working Paper Series*, 2009.

[30] The Webkit Open Source Project, http://www.webkit.org.

[31] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas, "Adnostic: Privacy preserving targeted advertising." in *NDSS*, 2010.

[32] TRUSTe, http://www.truste.com.

[33] T.-F. Yen, Y. Xie, F. Yu, R. Yu, and M. Abadi, "Host fingerprinting and tracking on the web: Privacy and security implications," in *NDSS*, 2012.