Utilizing Web Trackers for Sybil Defense

MARCEL FLORES, ANDREW KAHN, and MARC WARRIOR, Northwestern University
ALAN MISLOVE, Northeastern University
ALEKSANDAR KUZMANOVIC, Northwestern University

User tracking has become ubiquitous practice on the Web, allowing services to recommend behaviorally targeted content to users. In this article, we design Alibi, a system that utilizes such readily available personalized content, generated by recommendation engines in real time, as a means to tame Sybil attacks. In particular, by using ads and other tracker-generated recommendations as implicit user "certificates," Alibi is capable of creating meta-profiles that allow for rapid and inexpensive validation of users' uniqueness, thereby enabling an Internet-wide Sybil defense service.

We demonstrate the feasibility of such a system, exploring the aggregate behavior of recommendation engines on the Web and demonstrating the richness of the meta-profile space defined by such inputs. We further explore the fundamental properties of such meta-profiles, i.e., their construction, uniqueness, persistence, and resilience to attacks. By conducting a user study, we show that the user meta-profiles are robust and show important scaling effects. We demonstrate that utilizing even a moderate number of popular Web sites empowers Alibi to tame large-scale Sybil attacks.

CCS Concepts: • **Networks** → **Network privacy and anonymity**; *Web protocol security*; • **Information systems** → *Personalization*;

Additional Key Words and Phrases: User tracking, recommendation engines, Sybil attacks

ACM Reference format:
Marcel Flores, Andrew Kahn, Marc Warrior, Alan Mislove, and and Aleksandar Kuzmanovic. 2021. Utilizing Web Trackers for Sybil Defense. *ACM Trans. Web* 15, 2, Article 8 (April 2021), 19 pages.
https://doi.org/10.1145/3450444

1 INTRODUCTION

It is no industry secret that almost every browsing click we make is collected by one or more of numerous information trackers and aggregators associated with a variety of online services. This includes, but is not limited to, trackers associated with recommendations, ad networks, search engines, and online social networks, among others. These trackers have supported the monetization of the Internet from a small network into a gigantic infrastructure with revenues of billions of dollars.

Authors' addresses: M. Flores, A. Kahn, M. Warrior, and A. Kuzmanovic, Northwestern University, McCormick School of Engineering, Northwestern University, Mudd Hall, 2233 Tech Drive, Third Floor, Evanston, IL 60208; emails: {marcelflores2007, andrewkahn2015, marcwarrior2019}@u.northwestern.edu, akuzma@northwestern.edu; A. Mislove, Northeastern University, Khoury College of Computer Science, 440 Huntington Avenue, 202 West Village H, Boston, MA 02115; email: amislove@ccs.neu.edu.

1559-1131/2021/04-ART8 $15.00
https://doi.org/10.1145/3450444

Users have often reacted negatively to tracking systems, expressing concern about a lack of privacy and control over personal data. Nonetheless, despite a substantial effort to expose and control this prevalent behavior both from the research [8, 15, 18, 37, 38] and policy [14, 21, 40] sides, the reality is that users continue to accept updated online privacy policies, which in turn grant the gathering of more user personal data [1–3]. Indeed, most users, although aware of the increasing erosion of online privacy, continue to use these valuable—and often necessary—online services.

In this article, we ask if it is possible to *utilize* the ubiquitous online tracking for the direct benefit of the *users* themselves (as well as for the benefit of numerous distributed systems) to tame multiple identity, or Sybil, attacks [13]. Traditional defenses rely on either trusted identities or assumptions on the structure of the users' social network [26, 39, 49, 50]. Unfortunately, requiring users to present trusted identities runs against the open membership that underlies the success of these services in the first place [42], and recent work has called into question social network assumptions [28, 42].

We propose Alibi, a system that co-opts the work done by online trackers to provide services with a lightweight defense against Sybil attacks. Our key idea is to use the readily available personalized content, generated by online recommendation engines in real time, as a means to verify an online user in a privacy-preserving manner. The system utilizes an abstraction of such tracker-generated personalized content, submitted directly by the user, to construct a multi-tracker user-vector representation and use it in various online verification scenarios. To that end, we explore the properties of such representations, i.e., their construction, uniqueness, resolution, persistency, resilience, and utility, in Sybil defense.

An online service, e.g., a Web site, uses Alibi to verify that a user has not performed a one-time action previously. For example, a user may try to vote for an article on a content aggregation site, a potential Sybil attack target [4, 5]. The service submits a request to Alibi, which then requests that the user fetch tracker-generated content from personalized pages, such as the recommended items on the landing page of a popular retailer, a streaming service, or online news site or aggregator. Thus, Alibi utilizes *per-site recommendation engines*, rather than global advertising trackers, such as DoubleClick. To protect user privacy, Alibi abstracts away individual recommended items, e.g., individual news stories or products, and instead has the user only submit the distribution of the *categories* of recommended items. Alibi computes a vector representation of the user (the *meta-profile*) and compares it to previously logged meta-profiles. Alibi then returns the result to the service, which can react accordingly, for example, by counting or discounting the user's vote.

We demonstrate that Alibi provides users and sites with strong security guarantees and has minimal impact on users' privacy. We verify that Alibi can quickly identify newly created profiles created by attackers, preventing attackers from generating profiles quickly. In comparison to social-network-based single sign-on systems [37] or counter-Sybil systems, e.g., [26, 39, 42, 50], Alibi requires no knowledge about a user's identity nor about a user's social graph. As a result, Alibi provides strictly stronger privacy guarantees than any of these systems and operates at the Internet scale, i.e., is not tied to one system.

Overall, this work makes the following contributions. First, we evaluate a set of popular online services that provide personalized recommendations and characterize the key factors that make them suitable for use in the context of Alibi. Second, we demonstrate that reliable user meta-profiles can be constructed from the information collected from these numerous online sources. Third, we show that these meta-profiles can be consistently identified over time, and that they are resilient to small changes in a user's interests. Fourth, we demonstrate that a user's meta-profile is suitable for recognizing a returning user, with a low likelihood of collision with other users. Fifth, we show that while although interests are biased toward a subset of popular categories on a site, this effect rapidly diminishes with the number of categories used by a site and the number of sites

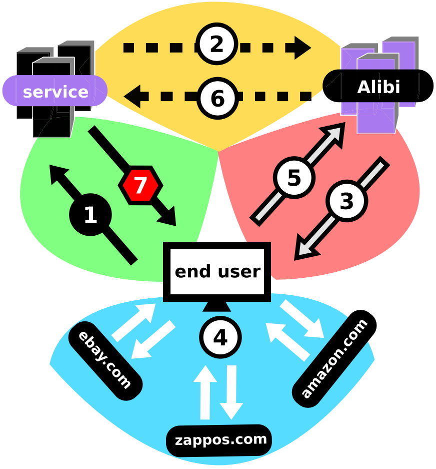


Fig. 1. Overview of the Alibi system.

used by Alibi. We further show that using a moderate number of sites enables Alibi to distinguish among hundreds of thousands of users.

2 PERSONALIZATION AS A SOLUTION

Sybil attacks are known to represent a fundamental problem in the design of distributed systems [13]. In a Sybil attack, a malicious user creates multiple identities and influences the working of systems that rely upon open membership. Traditional defenses against Sybil attacks rely on either trusted identities provided by a certification authority—an approach often rejected by both sites and users—or assumptions on the structure of social network relationships between users [26, 39, 42, 50]. Contrary to existing approaches based the social networks, Alibi requires *no* social network information, thereby offering the potential for both improved user privacy tradeoffs as well as broader applicability.

Instead, Alibi utilizes *user personalization* as a means to counter Sybil attacks at the Internet scale. Indeed, the study of how to provide users with meaningful suggestions has become the foundation for multiple academic conferences and private research competitions [31]. Although a central component to behavioral advertising [16], providing personalized suggestions has also been implemented on virtually all popular shopping and content viewing sites, e.g., Amazon, YouTube, Netflix, and eBay, providing item recommendations. Therefore, Alibi utilizes such tracking services as implicit “certification authorities,” where the “certificate” is personalized content. As a result, any attack that wishes to trick Alibi must also trick all of its constituent sites.

The work that per-site trackers do forms a representation of the user that Alibi utilizes. Since these recommendation engines have become so prevalent, many users are exposed to a large number of them, providing numerous vantage points. Each site further represents different aspects of the user’s online activity: Amazon recommendations reflect a user’s shopping trends, whereas YouTube recommendations reveal the user’s favorite content and musical genres. The set captured by each engine is different, and considering them together provides a diverse set of viewpoints. We show that this diversity is the key that enables Alibi to effectively scale.

System overview. Figure 1 presents the layout of a typical application. The system consists of four main components (shown as colored groupings in the figure). First is the *service* that is using Alibi. The service is generic: it could be any service that has a need to verify users’ uniqueness. Next is the *Alibi server*, which contains a database of previously observed user meta-profiles.

The Alibi server may be operated directly by the service. Next is the *end user* who is attempting to verify herself to the service with the installed Alibi browser extension, as well as the *tracking and recommendation engines* that hold the profile for the user and generate personalized content.

The interaction begins when the user submits a request for the service to validate (① in the diagram). For example, this may be a user attempting to rate content and the service may wish to verify it is the user's first vote. The service then submits a request to the Alibi server ②. The Alibi server then requests a number of pages directly from the end user ③. In particular, it sends a list of sites to the user, who then visits each site ④ and submits the results to the Alibi server ⑤. Next, the Alibi server computes the profile that results from the recommendations in each of these pages and compares it to others. If it matches the user within a statistical guarantee as determined by the service, Alibi sends a message to the service, indicating that the user has been seen before ⑥. Otherwise, it indicates the negative. Finally, the service either allows or disallows the user's request ⑦.

In scenarios where a user does not cooperate with Alibi, the site operator may face such a user with other (heavier) user verification services, such as social network type verification or requiring other trusted identities. Typical examples that are used today are phone verification (a user must prove that he or she has access to a given phone number) or CAPTCHA-like services. Because such other systems either lack privacy guarantees and are more complicated to use, we believe that users are more likely to opt for Alibi.

Threat model. In designing Alibi, we assume that the attacker can freely create identities on recommendation sites and can do only limited browsing with these identities. In other words, the attacker may be able to browse with a small number of identities but cannot do large-scale browsing with a large number of identities (as sites like Google and Amazon already have sophisticated, automated means to detect large-scale, coordinated browsing/scraping). We assume the security of TLS, and that the private keys of recommendation sites' SSL certificates stay private. We assume that the attacker may submit false category distributions.

3 ALIBI-FRIENDLY SERVICES

Here we explore the current state of tracking implemented by recommendation engines, as seen from the end user's perspective. Our goal is to understand the building blocks upon which the Alibi is built. Specifically, we examine the types of interactions necessary to change the output of recommendation engines on different Web sites.

We first select a subset of sites from the Alexa Top 500 [7] that provide user recommendations. For each considered site, we start with a fresh browser profile and manually interact with the site. We record the types of behavior that result in recommendation changes and how changes in the browser state, e.g., cookies affect these recommendations. For sites requiring money, e.g., purchases, we use existing accounts.

Table 1 presents our findings. It reveals a wide spectrum of interactions that influence recommendations seen by the end user. The first three rows of the chart indicate the breakdown of our tested sites ("Influences"). We find that, initially, many sites will start users with a "blank slate" and avoid presenting any user-specific recommendations prior to interaction. These sites begin to generate custom recommendations as the user interacts with the site via clicking items, consuming media and so on. Several of these sites have a higher interaction threshold than others: on some sites, recommendations will react to simple triggers, such as clicking on an item or adding items to the shopping cart (Amazon, HomeDepot). Other sites delay their reaction, e.g., until multiple items have been clicked (Zappos), until a video has been watched (YouTube), or until some other activity has been completed (Netflix). We refer to those that require more than simple browsing as *Special Activity*. Other sites combine both, offering recommendations with browsing and more detailed suggestions with purchases (Amazon).

Table 1. Recommendation Behavior on Amazon (AMZ), YouTube (YT), Netflix (NFLX), NewEgg (EGG), TripAdvisor (TRIP), Zappos (ZPS), TicketMaster (TMR), eBay (EBY), NYTimes (NYT), BestBuy (BBY), and HomeDepot (HD)

		AMZ	YT	NFLX	EGG	TRIP	ZPS	TMR	EBY	NYT	BBY	HD
Infl.	**Browse/Cart**	✓			✓	✓	✓		✓	✓	✓	✓
	Special	✓	✓	✓								
	Time/Loc.					✓		✓				
Store	**Cookie**	✓	✓		✓	✓	✓		✓	✓	✓	✓
	Login	✓	✓	✓					✓	✓		
Recs.	**# of Cat.**	100+	30+	400+	500+	200+	50+	40+	400+	20+	100+	300+
	# of Recs.	120+	30+	30+	5	4	6	10+	15+	20+	9+	4

Note: The first set of rows (*Infl.*) indicates what user actions impact recommendations: regular browsing and carting objects, special behaviors, or the browsing time and location. The second set (*Storage*) indicates what type of storage the site uses to recall users. The final set (*Recs.*) shows how many object *Categories* are found globally on the service and the number of explicit object recommendations found on a recommendations page.

In the case of sites with behavioral tracking, each site employs a mechanism for tracking users. We find two non-mutually exclusive groups, which are shown in the middle two rows of the chart (Storage). The first are sites that connect users to their profile via cookies. In such cases, users simply store a cookie that corresponds with a profile on the site's server. Recommendations for such systems do not persist beyond the life of the cookie; when a user removes his cookie, the site essentially forgets the user. Cookies do not require a login and tend to be created on the first visit. The second group is tied to an account: if the user logs into his account, history-related recommendations will be shown, potentially replacing cookie-based recommendations.

Cookies and logins are not the only criteria by which online services track or identify their users. After deleting cookies from the browser, the online service may consider the installed software components, timing information, IP addresses, browser/OS versions, and so on, to detect a returning user [48]. This means that even private browsing mode does not fully prevent trackers. Alibi is independent from the method in which the service obtains its user information; as long as personalization exists, Alibi can utilize it.

Finally, we examine the nature of the recommendations served to users. First, we estimate the number of categories available on each site based on observation of recommendations and explicit information on each page. In particular, on all of the examined sites, recommend items are grouped into *categories*. In the case of a shopping site, these correspond to "departments" in a traditional store, and for streaming video sites, these often correspond to genres. On many sites, these categories are further subdivided into a tree-like structure. For the purposes of Alibi, we take categories to mean no more than two layers deep, i.e., subcategories, as taking greater subdivisions often introduces inconsistencies across categories, e.g., certain categories may be shallow. Ultimately, Alibi requires only that objects are categorized and allows each site to be configured individually.

When enumerating categories, we see that these values vary across sites, based on the design and purpose of the site, but that numerous sites in our sample provide more than 100 categories each. We next estimate the number of recommendations provided to users on a single page load, finding that the majority of sites give us more than 15 item recommendations, and some as many as 120. The richness of this recommendation data indicates that even with a relatively small set of sites, Alibi is given significant information.

We note that this analysis is necessarily not comprehensive: the specific recommendation engines are entirely determined by the sites that operate them. They are therefore subject to change in both mechanism and character. However, we emphasize that such systems are prevalent on a variety of sites, and are becoming more so. The recommendation engines seen in these sites capture a broad view of user activity and therefore provide a sound foundation for Alibi. Finally, Alibi does *not* need to reverse-engineer nor understand the internals of various recommendation engines but relies on the fact that user recommendations are largely static category-wise, as we demonstrate in the following.

3.1 Assumptions and Limitations

First, Alibi necessarily assumes that users have an established profile. The lack of such profiles on any of Alibi's sites for some users will necessarily make them outliers in Alibis' matching abilities. We further note that some of the most effective sites we consider for Alibi require significant user effort, such as the creation of accounts or the purchasing of subscriptions. Here we focus on a popular set of sites, increasing the likelihood that a user will have at least some established profile.

Additionally, users must necessarily trust the system enough to share the results of these profiles. We explore the nature of this trust and how it compares to other systems in more detail in Section 5. Users who employ obfuscation techniques to trackers will reduce the signal that can be collected from the resulting profiles.

Finally, since Alibi fundamentally relies on information and structures from other services, it may be impacted by changes and updates on each of these services. Site redesigns, account policy changes, and business model shifts may all alter the signal it gets from a site. However, by including many sites, Alibi avoids relying on a single source of information.

4 DESIGN

Alibi takes user recommendations and converts them into a measurable meta-profile. Alibi therefore consists of two components: the first converts recommendations into quantitative vectors, and the second combines these vectors into meta-profiles that can then be compared.

Meta-profile construction. First we consider the recommendations presented to users as a set of *objects*. For each object, we record a title, e.g., the name of a video or the name of an item for sale, the URL of the recommended object, and the site-specific categorization of the recommendation.

To construct a meta-profile from this data, we consider the objects seen by the user on a site-by-site basis. For each site, we consider the associated *categories* for each recommended object. As discussed earlier, the meaning of category depends on the site itself: on a shopping site, it may represent a department, whereas on a music service, it might represent a genre of music. We scrape such information directly from the site and count the number of occurrences of each *category* among the recommended objects, allowing each object to be in multiple categories. We then take the resulting list of category occurrence counts and consider it as a vector, where each dimension is a category from the current site. The user meta-profile is then defined to be the set of these per-site vectors.

Profile space. Next we explore the richness of the profile space. Specifically, we show the size of the profile space, demonstrating its ability to cover many users.

Suppose that we have collected recommendations from a set of S sites. For each $i \in S$, we have collected l_i recommendations that come from a set of potential categories of size c_i. However, recommendations are not simply uniform random selections of categories. Indeed, pages are often composed of groups of items that select from the same set of categories. We therefore consider the following constraints: let p_i be the number of categories that appear in a set of recommendations for site $i \in S$, where $p_i < l_i$. We therefore compute the total number of profiles, N, of this form to be $N = \sum_{i \in S} \binom{c_i}{p_i}$.

Taking S to be the 11 sites in Table 1, and p_i to be 4, a common value based on our observations of their groupings, we see that Alibi is able to observe more than 5 billion profiles, indicating that it can comfortably represent significant sets of users. Necessarily, this represents an *upper bound* for the number of supported users, given that interests are likely to be common across sites and certain interests are likely (or unlikely) to be paired together. We evaluate this in Section 6.

Meta-profile behavior. Since these meta-profiles are essential to the working of the Alibi system, we examine their behavior in more detail. For these meta-profiles to prevent Sybil attacks, we require that they represent a user for a sufficient time, and that this representation is subject to only limited changes. To this end, we perform a number of controlled experiments using a subset of sites to demonstrate some of the representative behaviors we observe. Specifically, we consider recommendation profiles generated on Amazon, YouTube, and Zappos. Based on Table 1, we see that Amazon provides an example of a site that responds to direct browsing behavior, and YouTube responds to a special activity (watching videos). We further considered Zappos, as it shows fewer recommendations. Later we explore the behavior of the system using real user data on a larger set of sites.

Our methodology is as follows: for each experiment, we consider 10 simulated users. Each consists of a single browser session, including cookies, which are stored for the duration of the experiment. Each of the users is further provided with YouTube and Amazon accounts to enable server-side profile state, as discussed in Section 3. Each user selects a list of pages to visit from categorized department overviews; we are thus able to control the categories a user selects from.

Each user then randomly selects a page from the above list and loads the URL and dwells on the item page to make an impression on the browsing profile. The dwell times are selected so as to avoid appearing as a scraper or other unwanted-automated collector. For Amazon and Zappos, a 5-minute wait time was used, whereas on YouTube, 60 minutes was used to ensure that most videos would complete playing. After each user has browsed five pages from each category for each site, the high-level categories for the recommendations for each site are collected to observe the effect of the browsing to that point. Each user then goes to sleep for the remainder of the day. On the subsequent days, the process is repeated.

Since the users begin the experiment with a blank recommendation history, we allow the browsing process to run for 3 days before collecting recommendations, allowing the sites an opportunity to generate personalized recommendations. We emulate the following behavior.

Consistent browsing. In our first experiment, the 10 users randomly select three categories to view per site, then five items per category per site per day for 17 days. Given the waiting times, three categories and five items per site is close to the maximum possible in a day. The goal of this experiment is to demonstrate that profiles experiencing consistent behavior will result in a consistent set of recommendations from sites in a measurable way.

Browsing with quiet period. We next consider another likely user behavior: when the user has used the site in the past but then does not visit the site for a period of time. To simulate this, we consider a browsing pattern that uses three categories for 10 days, then performs no browsing for 7 days. We expect that this represents a normal mode of user behavior, as in general, most users do not visit sites every day. Otherwise, the conditions remain the same.

Browsing with changing behavior. Next we consider the case when a user with a developed profile changes behavior and begins browsing a new topic. We consider a user who browses three

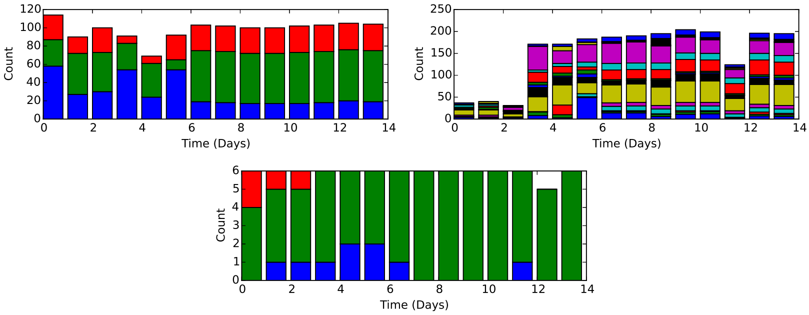


Fig. 2. The distribution of recommendations from the main categories for Amazon (upper left), YouTube (upper right), and Zappos (lower center) for a single user over the 2-week period with a quiet period after day 6. Each color represents a different category. The initial 3 days of browsing are not shown.

categories for 10 days, then browses three categories for 7 days, two of which are the same and one of which is new. Otherwise, the conditions remain the same.

Findings. Figure 2 presents the resulting observed high-level categories as stacked bar graphs from a single user for the browsing with quiet period. Other users demonstrated similar behavior, and our aggregated results confirm these insights. For each day, the components of the bar represent the category vectors, with the height indicating the magnitude, i.e., the number of objects. For clarity, we filter out categories that did not contribute significantly to the meta-profile (<50 for Amazon, <30 for YouTube, and 0 for Zappos). We note that although only a single category was browsed, each site provided recommendations for more than one category, possibly due to objects belonging to multiple categories. This effect is particularly pronounced on YouTube.

We found that the behavior for the Consistent Browsing experiment matched that seen in the first 7 days of Figure 2. Here we see that the primary categories persist for the duration of the regular browsing period. We note, however, that they are subject to high amounts of variation in the number of objects from each category, with the most prominent category shifting nearly every day. We suspect that this variation is the result of the high number of items browsed daily. Figure 2 shows that after a user stops browsing, the profile often becomes static. We note that in the case of Zappos, a single category dominates (green). Although the exact cause is unknown, we suspect that this is the result of the smaller number of categories and the inherent underlying popularity and importance of some categories, e.g., best sellers. Our results for the browsing with changing behavior experiment (figure omitted for space constraints) demonstrate consistency when a new category is introduced: the original categories are retained while the new category is added into the profile on the first day for which it appears.

Comparison methodology. We now develop a comparison for the resulting meta-profiles such that we can identify consistency over time and therefore recognize users over time. First, suppose that we have a user meta-profile from a set of recommendations as described in Section 4, i.e., a set of vectors that reflect the observed recommendation categories. We denote this meta-profile $U = \{u_1, \ldots, u_n\}$, where u_i is the vector from site i in our set of sites S. Next, suppose that we have a second meta-profile $U' = \{u'_1, \ldots, u'_n\}$. To compare these, we will consider a combination of their *per-site* similarity. In particular, we will compute the *cosine similarity* of each pair of vectors u_i and u'_i, which we denote $cos\_sim(u_i, u'_i)$. Cosine similarity measures the cosine of the angle between two vectors, resulting in a measurement that varies from 0 (totally dissimilar) to 1 (totally similar).

Two entire meta-profiles can then be compared by taking a weighted sum of these cosine similarities. Let w_i be a weight given to site i such that $\sum_{i \in S} w_i = 1$. We then take the final meta-profile similarity to be $\text{similarity}(U, U') = \sum_{i \in S} w_i \cdot \text{cos\_sim}(u_i, u'_i)$.

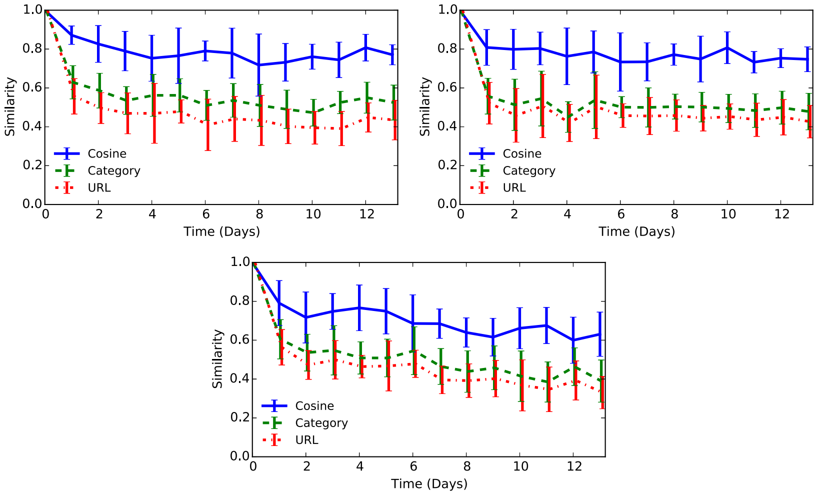


Fig. 3. Median similarity for our three schemes to the first day of collection after browsing with the following behavior: same topics for 14 days (upper left), same topics for 7 days and then no browsing for 7 days (upper right), and same topics for 7 days and then changing a topic and browsing for 7 more days (lower center). Error bars indicate the 10th and 90th percentiles.

The weights w_i can be adjusted to account for differences in the nature of a site's recommendations. For example, sites that use account-based recommendations could be given higher weights than sites that use simple cookie-based recommendations, as discussed in Section 3. In our implementation, we apply double weight to sites that measure special activity, i.e., Amazon, YouTube, and Netflix. Finally, to determine if a user has been seen previously, a simple threshold can then be applied: users whose similarity scores are above a certain value are declared to be the same.

To understand the effectiveness of our metric, we further consider two additional schemes. The first is a simple category match, which indicates the fraction of categories that occurred in both vectors. The second is a URL match, which counts the fraction of object URLs occurring in both.

Figure 3 (upper left) shows the median user similarity to the first day across the three sites for the duration of the 2 weeks. We see that overall, the meta-profiles retain high similarity, with the cosine similarity scores generally staying above 0.8, and remaining high for the entire 14 days. We see similar results in Figure 3 (upper right) as the cosine similarity stays consistently high. Importantly, the cosine similarity remains nearly steady after day 7 when no browsing is occurring, suggesting extremely high cosine similarity in the no browsing case. We see in the third experiment (Figure 3 (lower center)) that the cosine similarity decreases after day 7, as expected when behavior changes, and remains consistent afterward. In all cases, the cosine similarity outperforms the alternative methods.

5 SECURITY AND PRIVACY

We now discuss both how Alibi defends against different security attacks and respects users' privacy.

Fresh account attack. The first attack that may come to mind is whether the attacker can use *fresh* accounts on various services to submit content to Alibi. For example, some services provide recommendations to even new users who show up at the site; the worry is that the attacker would be able to easily fetch personalized content. However, we note that this content cannot be truly personalized to the account, as the site has no information on the account by definition. Thus, the site must provide a recommendation to a blank profile. Moreover, in Section 6, we show that the set of content provided to such accounts is typically small, and Alibi can detect the meta-profiles derived from content provided to them.

Profile creation attack. In this attack, an attacker mimics real user behavior by browsing particular items on sites used by Alibi. Alibi requires a user to present recommendations for at least $|S|$ sites. Hence, the attacker must browse items on each of these sites such that it creates a unique meta-profile. The cost of this browsing varies by site: for Netflix or YouTube, it requires streaming videos (taking significant time and bandwidth), whereas for Amazon, it may only require viewing item pages. In the following, we first analyze a manual approach, then an automatic one.

In a manual approach, the limiting factor lies in the manual work needed to make farmed users appear unique. Farming requires the attacker to select a unique set of sufficiently different items to browse for each site for each fake user. We assume that it takes at least 10 seconds to manually select a new unique item on any of the sites in Alibi. This involves multiple page loads for navigation, reading portions of the page, and determining an item which will be unique, and so on. If the attacker has to view at least five items to generate recommendations, it would require 4 minutes per account, or approximately 35 hours to create 500 accounts. Still, many sites require more complex actions, e.g., streaming on YouTube or Netflix, making it difficult to imprint with a single page in 10 seconds, instead potentially requiring *minutes* per page. If each imprint took 5 minutes, the preceding forgery would take more than 43 days.

In an automatic approach, additional challenges arise: rate limiting or blacklisting by the third-party sites, the need to create a fully automated system to perform browsing with sufficient computing power and network capacity, and so on. Moreover, an automated system is not guaranteed to be able to create realistic user profiles, which are necessarily biased toward given categories, as we show in the following.

The absence of such real-world profile signatures could be utilized by Alibi to prevent any such artificial profile farming attempts via anomaly detection techniques like **Principal Component Analysis (PCA)** [22]. Prior work has demonstrated that anomaly detection via PCA can be used to significantly raise the bar for Sybil attackers on sites like Facebook [41]. In the context of Alibi, PCA can be used to identify users with unusual patterns of behavior that can be learned entirely from the (unlabeled) data itself, i.e., the number of occurrences of each category over time. Anomaly detection could be applied across meta-profiles as a mechanism to identify suspicious meta-profiles. Because it is difficult to accurately generate large amounts of distinct human behavior, this approach enables the Alibi provider to distinguish the fake, groomed accounts from the real ones without requiring labeled training data. In essence, to avoid detection, the attacker would have to correctly generate a unique trace human browsing behavior along each dimension we consider; to detect the attacker, we would only be required to find a single dimension where the personalized content is anomalous.

Content forging attack. In the Alibi system described thus far, a powerful attacker may be able to *forge* content from personalized sites for the purpose of making their identities look distinct. This attack is challenging for the attacker to conduct successfully, as the attacker would need to forge content in the manner that the sites would that is distinct enough that Alibi would believe the user is distinct, without appearing as a "outliner" via anomaly detection mechanisms that the Alibi operator could employ, such as PCA [22].

Privacy. Alibi provides an online identification mechanism that could be beneficial for numerous online services. Contrary to single sign-on systems associated with popular services such as Facebook, which bring a wide-range erosion of user privacy [37], Alibi requires no information about users beyond the content generated by online tracking systems, i.e., ads and other recommendations. Similarly, contrary to social network–based Sybil defense systems, e.g., [26, 39, 42, 50], Alibi requires no knowledge about a user's social graph. As a result, (i) Alibi provides strictly stronger privacy guarantees than any of these systems, and (ii) it operates at the Internet scale, i.e., it is not tied to any one particular system. Furthermore, even if an Alibi server is compromised, it only contains distilled user vector representation, i.e., it does not contain mapping between actual user categories and its numerical representation.

However, Alibi does require that users (i) regularly upload the categories of recommended content and (ii) occasionally enable Alibi to view the content containing the recommendations themselves. Although such information necessarily reveals user habits to an extent, it does not expose any PII, and we believe that it does not reveal a user's identity.

Another privacy concern is that a rogue Alibi provider might try to correlate category information collected from clients with other public datasets in an attempt to de-anonymize users, e.g., like in the Netflix case [30]. A typical reason to publish anonymized micro-data is "collaborative filtering," i.e., predicting a consumer's future choices from his past behavior using the knowledge of what similar consumers did. Contrary to such de-anonymization scenarios, Alibi users do not provide any historic information about their behavior, i.e., neither their browsing patterns nor their actual product or service selections or ratings. Most importantly, an Alibi user provides information about categories (and about objects when challenged) that are *recommended* by the tracker, *not* actually selected by the user. This discrepancy between the actual user behavior (hidden from the Alibi server) and users' expected future choices (represented by the tracker-generated recommendations) fundamentally limits the cross-correlation de-anonymization. Ultimately, however, the risk of such an attack does exist, and the user is required to place some trust in the Alibi server.

6 EVALUATION

Methodology. To obtain real user meta-profiles, we collected recommendations served to a group of 91 users at our local institution. Specifically, we developed browser plugins for Firefox and Chrome. The plugin periodically collect recommendations from a pre-defined list of sites. Then, in the background, the plugin loads each page and sends the observed recommendations to a server via HTTPS. To provide ground truth, the plugin selects a GUID at install time. This GUID is included when submitting recommendations so that a single user can be recognized over time. Once the data is submitted, the server uses the recommendations to determine a meta-profile for the user at that time. Although the plugin collects organic recommendations presented to users on their own accounts, we refrain from collecting any information on if or how often each user visits the measured sites. (The Institutional Review Board office at our institution provided the study a determination of Not Human Research.) We collected data during the period of 45 days by using the 11 sites listed in Table 1.

For each of the sites collected in the study, we authored a site-specific *harvester*. Each harvester was written to examine the recommended items and extract their categories. For each of these sites, we rely on the inherent structure of the object presentation to reveal the categories. For some sites, this is as simple as parsing the URL of the recommendation, and for others, this involves parsing other portions of text in the recommendation, e.g., "Consider other Items in Home Electronics..."

The collection of the data necessarily comes with greater overhead than some existing systems, e.g., CAPTCHA. Since recommendations are generated on a per-user basis, collecting them requires users to load a customized page, i.e., execute javascript, but needs only transmit small portions of the resulting data to Alibi, such as the set of recommendation URLs and some HTML, avoiding the need to transmit images or video. Potential optimizations in the harvesters could

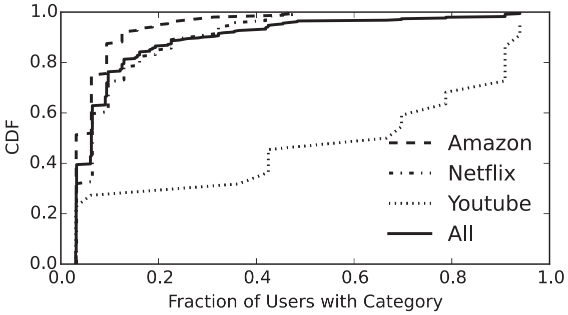


Fig. 4. CDF of categories showing the fraction of users that share each category. Most categories are lightly shared.

reduce this overhead further. Future iterations could potentially also consider site cooperation and uniform APIs for accessing recommendation data, further simplifying the process.

Category popularity. Here we consider the nature of the popularity of the categories seen by users. We would like the majority of categories seen by the users to occur rarely, making them strong identifiers. To measure this, we consider the fraction of users of each site that have a given category. We consider this value on a per-site basis for our three most popular sites, Amazon, Netflix, and YouTube, as well as the overall performance.

Figure 4 presents a CDF of these values. We see that Amazon and Netflix both feature low share rates, with 80% of categories occurring with only 10% and 16% of users, respectively. In the case of YouTube, however, categories are much more popular, with the median category occurring for half of users. This is partially due to the frequency with which YouTube videos occur in multiple categories and the relatively small number of total categories. When we consider categories across all sites, the overall behavior is closer to Netflix, with 80% of categories occurring with 16% of users.

These frequency rates suggest that Amazon and Netflix are providing us with a large collection of categories that reflect user interest, not just popularity. Such categories provide Alibi with a mechanism to discriminate between users. Although YouTube's categories do not provide the same variation, these profiles still provide important information, as a profile is more than just occurrence of categories: it reflects the frequency of each categories, as well as the entire set S. Finally, when combining all sites and considering the full profile, our set of categories is again diverse.

User behavior. Next we show that real user profiles maintain consistent recommendations over time. We measure the change in site vectors for the users after 5 days. We consider the *difference* in cosine similarity, i.e., 1 less the cosine similarity to the first day of collection, for each site and for the meta-profile for all users.

Figure 5 shows the similarity differences measured after 5 days. Recall from Section 4 that the best case cosine similarity for our study was approximately 0.8, i.e., a difference of 0.2, for the median user and 0.65 (differences of 0.35) in the case of changing browsing behavior. We see that Netflix and YouTube strongly outperform these, with nearly 80% of users having a difference less than 0.2 after 5 days. The median case for Amazon is within the target range as well, although we see that it has some interesting properties, as 20% of users have almost exactly the same profile after the 5 days and the remainder of users have profiles that may be anywhere from 0.05 to 1.0 in difference. This is consistent with our measurements, as we speculate that 20% of users did not browse Amazon during this 5-day period, which shows extremely high consistency, whereas the

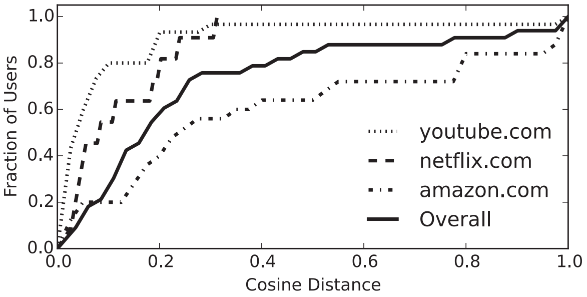
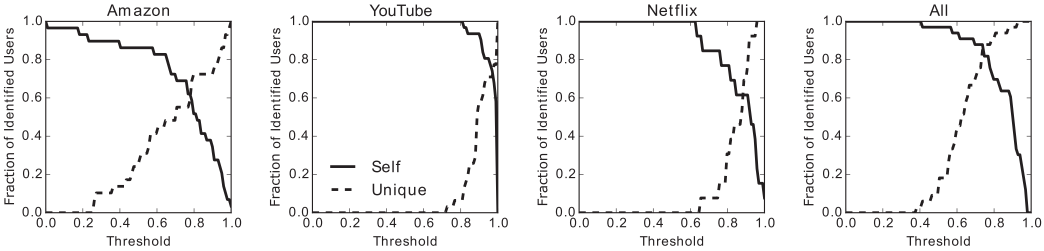


Fig. 5. Cosine distance of user vectors between their last collection and 5 days prior for three popular sites, as well as the overall meta-profile total. Most users show a relatively small difference between their own profile from 5 days prior.



(a) (b) (c) (d)

Fig. 6. The fraction of users who are unique and match themselves (“Self” in the figure). Combining all sites and taking a threshold of .74 results in the best balance between uniqueness and the ability to recall a user.

rest of the users did so to varying degrees. The overall profile consistency matches near our target (0.2) in the median case. As discussed earlier, we believe that this is because users are not normally browsing each site every day. These observations indicate that although profiles are subject to change as time passes, these changes are no worse than those we observed in our measurements.

User uniqueness. Next we explore the trade-off between identifying returning users and distinguishing between users. Let U be our set of users. For each user $i \in U$, we have a set of meta-profiles $\{f_{i,1}, \ldots, f_{i,n}\}$, one for each day of collection: 1 through n. We take $f_{i,n}$ for each i in U, i.e., the final collection from each user. We compare the most recent collection from each user to the second most recent collection: $\gamma_{i,n-1} = \text{similarity}(f_{i,n}, f_{j,n-1})$, for all j in U. For each threshold t_k between 0 and 1, we then compute two values: the fraction of users for which $\gamma_{i,n-1} < t_k$ for all $i \neq j$, i.e., the fraction of users who appeared unique, and the fraction of users for which $\gamma_{i,n-1} \geq t_k$ or all $i = j$, i.e., the fraction of users who matched themselves.

Figure 6 presents the “Unique” and “Matches self” values for each threshold for Amazon, YouTube, Netflix, and our aggregate set S. As the threshold increases, fewer users match themselves, but more users are identified uniquely. Each site obtains a balance between matching itself and uniqueness, although the point of balance depends on the particular site. For example, YouTube achieves its balance at a threshold of .97, as compared to .78 and .87 for Amazon and Netflix, respectively. This relatively high balance point for YouTube is likely the result of the popularity of YouTube categories, which requires a higher threshold to differentiate users. In aggregate, a

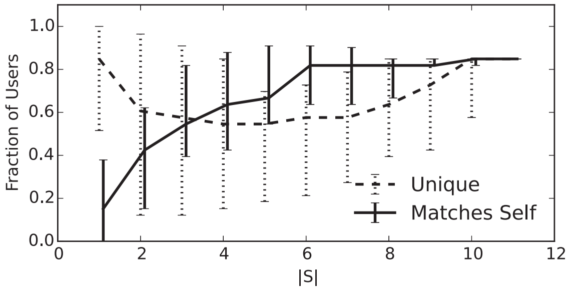


Fig. 7. Median fraction of unique and self-matching users for varying sizes of *S* over all permutations.

balance is achieved at 0.74, where 86% of users match themselves uniquely (more than any site). Therefore, in this configuration, Alibi would achieve a false-positive rate of 14%.[1]

Size of the set S. Next we examine the effect of the size of the set of sites S used to generate the user profiles. In particular, we wish to demonstrate that increasing the size of S has a positive effect on the performance of the system. To this end, we consider restricting the number of sites used in measuring profiles. Here we consider all permutations for each size: for $|S| = 1$, we consider using each site individually. For $|S| = 2$, we consider all pairs of sites, and so on. For each value, we compute the fraction of users who appear unique for a threshold of .74 and the fraction of users who match their previous collection.

Figure 7 presents the results of this experiment. The lines indicate median values, and the error bars indicate 25th and 75th percentiles. We see that with very few sites, the median "Matches Self" performs quite poorly but steadily increases as we increase the size of S. The "Unique" value initially decreases, revealing a similar trade-off in matching versus uniqueness, as seen in the previous analysis, but begins to increase again after S is greater than 6, as additional profiles provide a more complete view of the user. Finally, with all sites, we achieve a balance, as desired from our choice of threshold. This again emphasizes the importance of using a large set S, giving Alibi a broad view of a user's activity.

Large user base. To demonstrate Alibi's scalability, we consider the following simulation. For a user base size N, we consider N randomly generated users. Each user is assigned a profile constructed from our observations of profile behaviors. Each profile is made of a random selection of categories based on the variety of categories and recommendations for each site indicated in Table 1. We assume that each site delivers four categories divided among the recommendations per page. We use a spectrum of possible distributions of categories to assign profiles: (i) is a uniform random distribution, which represents an *upper bound* in terms of the number of supported users. Next we aim to emulate the empirical distribution shown in Figure 4. To that end, we introduce two exponential distributions, which represent lower and upper bounds for the distribution from Figure 4. The lower-bound distribution is (ii) "Exp-25," which is an exponential distribution with scale parameter $\frac{1}{\lambda} = (25\% * \text{num. of categories})$. The upper-bound distribution is (iii) "Exp-50," an exponential distribution with a corresponding scale parameter. We use $|S| = 4$, considering profiles in the forms of Amazon, YouTube, Netflix, and Zappos. We then "age" each profile by replacing a single category with a fresh selection 80% of the time, simulating a change in user behavior. We

[1] Notably, this calculation excludes users who would be labeled true positives because they matched another user, as these would be erroneous.

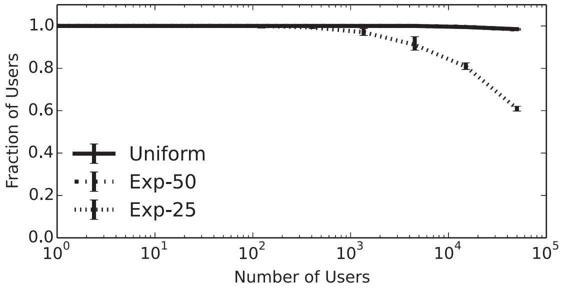


Fig. 8. Simulated performance for large user bases. Even for a relatively small $|S| = 4$, Alibi only begins to degrade with tens of thousands of users.

then consider the similarity of each profile to the aged versions of all profiles, computing uniqueness as before.

Figure 8 presents the results of our simulation. The line represents the median of 15 trials and the error bars the 10th and 90th percentiles. The uniform distribution presents an upper bound of performance with these four sites. The figure further shows that "Exp-50" completely overlaps with the uniform distribution, i.e., demonstrating that user uniqueness remains near perfect for the evaluated user base, shown on the *x*-axis. When drawing from the "Exp-25" distribution, uniqueness starts to degrade as the user base grows to tens of thousands of users. Hence, with $|S|$ as small as 4, Alibi still has the means to distinguish nearly perfectly among nearly 50,000 users, a number already applicable in many Web site–based scenarios. This is because Alibi's performance is upper bounded by "Exp-50" and lower bounded by "Exp-25." However, simply increasing the number of sites significantly increases the uniqueness of each user (not shown), resulting in performance indistinguishable from the uniform distribution in the figure. The design of Alibi's profile system and the use of only a handful of sites allows us to scale to extremely high numbers of users, making it widely applicable.

Detecting fresh accounts. As a final evaluation, we explore how "different" recommendations provided to fresh accounts are to those provided to established accounts. To do so, we focus on Amazon and compare the recommendations provided to four accounts: (i) a fresh account with no cookie (*Fresh*), (ii) an account that browsed items in a single category (*1 Cat*), (iii) an account that browsed items in two categories (*2 Cat*), and (iv) the long-established account of one of the authors (*Developed*). We collected the set of recommended items on the landing page for a week, counting the total number of item URLs on the page.

Figure 9 shows the percentage of recommendations, i.e., item URLs, that were new as compared to the previous day. There is a massive difference among the recommendations provided to a fresh account (no recommendations, hence no variability), those provided to our synthetic accounts (a near-stable set of recommendations), and those provided to our established account (a highly dynamic set of recommendations). Thus, malicious users must do significant work on sites like Amazon to be able to generate sufficient amounts of meaningful recommended content while avoiding anomaly detection.

7 RELATED WORK

Prior work has dealt with measuring user personalization, including in web search engines [20, 47] and ad networks [17, 45]. It has been found that different user behavioral properties lead to different user characterization by trackers and to different personalized content. Unlike this work, we aim

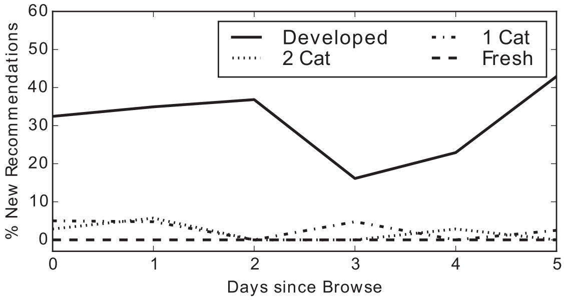


Fig. 9. Percentage of new items on a recommendation page based on account age. Fresh accounts return no personalized items, whereas fully developed accounts show significant personalization.

to *use* personalization for the benefit of users and numerous distributed systems. To achieve this goal, we do not need to reverse-engineer or understand the inner workings of tracking systems. If personalization exists, Alibi can use it.

Understanding [37] and defending against [6, 8, 15, 18, 38] third-party tracking has been a goal of several recent systems. Typically, such systems sit between advertisers and users, preventing information from flowing freely between them. The key distinction is that Alibi's aim is not to monitor or defend against personalization but to utilize it. In that context, Alibi is closest to the work of Riederer et al. [36], which argues that data should be monetized by users by selling their profiles to advertisers.

Personal user information can often leak into the network from numerous online services, e.g., [19, 23, 24, 25]. Such leaks, as well as information on user behavior or traffic characteristics, can lead to effective user fingerprinting in various online contexts, e.g., [11, 33, 46, 48], raising additional privacy concerns. Contrary to scenarios where user fingerprinting is utilized for collective or individual surveillance, we have a different agenda.

Other services have focused on allowing users to hide their behavior on various Internet services, such as TrackMeNot [32]. A number of other systems have sought to provide user protection and privacy by adding additional browsing and search queries to obfuscate a user's regular behavior [9, 12, 29, 34, 35]. Users who employ such systems would then generate non-realistic queries to the sites on which Alibi relies. However, as long as there are still relatively unique recommendations on each site due to this behavior, this would not affect Alibi. But necessarily, uniform profiles and non-personalized content could negatively impact Alibi, i.e., cause users to appear illegitimate.

Finally, there has been significant recent research focused on applying supervised [10, 27, 43, 44] or unsupervised [41, 43] machine learning techniques to identify Sybils. Unfortunately, many of these approaches result in a cat-and-mouse game, where the sites develop techniques to detect ever strong attackers, whereas the attackers develop new attacks. Alibi aims to counter such adaptive behavior.

8 CONCLUSION

We presented Alibi, a system that uses the behavioral tracking performed by numerous Web sites across the Internet to tame Sybil attacks. Contrary to social network–based counter-Sybil systems, Alibi requires no knowledge about a user's identity or social graph. As a result, it provides strictly stronger privacy guarantees, and it is not tied to any one particular system, i.e., it operates at the Internet scale. Alibi works by constructing profiles from recommendations presented to users on such Web sites, coupled with a comparison methodology that enables a service to determine if two users are the same. We validated this design using measurements from real-world recommendation engines. We also conducted a user study based on everyday web browsing behavior; we found that Alibi is able to distinguish among users using our recommendation profiles. We revealed that these profiles are robust, that they effectively scale, that they are capable of correctly identifying users using profiles over several weeks old, and that Alibi is resilient to a range of system manipulations and attacks. Finally, we showed that Alibi can successfully mitigate Sybil attacks, demonstrating the power of harnessing the user-tracking work performed by third parties.

REFERENCES

[1] Black Enterprise. 2012. New Facebook Privacy Policy Ruffles Feathers. Retrieved February 22, 2021 from http://www.blackenterprise.com/technology/new-facebook-privacy-policy/.

[2] Associated Press. 2012. New Google privacy policy allows even more access to personal information. *Fox News*. Retrieved February 22, 2021 from https://www.foxnews.com/tech/new-google-privacy-policy-allows-even-more-access-to-personal-information.

[3] Robert J. Mullins. 2012. New Microsoft privacy policy expands its user data mining rights. *eWeek*. Retrieved February 22, 2021 from http://www.eweek.com/enterprise-apps/new-microsoft-privacy-policy-expands-its-user-data-mining-rights/.

[4] BBC News. 2015. Amazon Targets 1,114 Fake Reviewers in Seattle Lawsuit. Retrieved February 22, 2021 from http://www.bbc.com/news/technology-34565631.

[5] Maeve Shearlaw. 2015. From Britain to Beijing: How governments manipulate the Internet. *The Guardian*. Retrieved February 22, 2021 from http://www.theguardian.com/world/2015/apr/02/russia-troll-factory-kremlin-cyber-army-comparisons.

[6] ABP. 2016. AdBlock. Retrieved February 22, 2021 from http://adblockplus.org/en/.

[7] Alexa. 2017. Home Page. Retrieved February 22, 2021 from http://www.alexa.com/.

[8] M. Backes, A. Kate, M. Maffei, and K. Pecina. 2012. ObliviAd: Provably secure and practical online behavioral advertising. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP'12)*. 257–271. DOI: https://doi.org/10.1109/SP.2012.25

[9] E. Balsa, C. Troncoso, and C. Diaz. 2012. OB-PWS: Obfuscation-based private web search. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP'12)*. 491–505.

[10] Fabrício Benevenuto, Gabriel Magno, Tiago Rodrigues, and Virgílio Almeida. 2010. Detecting spammers on Twitter. In *Proceedings of the 7th Annual Collaboration, Electronic Messaging, Anti-Abuse, and Spam Conference (CEAS'10)*.

[11] S. Chen, R. Wang, X. Wang, and K. Zhang. 2010. Side-channel leaks in web applications: A reality today, a challenge tomorrow. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP'10)*.

[12] Josep Domingo-Ferrer, Agusti Solanas, and Jordi Castella-Roca. 2009. h(k)-Private information retrieval from privacy-uncooperative queryable databases. *Online Information Review* 33, 4 (2009), 720–744.

[13] J. Douceur. 2002. The Sybil attack. In *Proceedings of the International Workshop on Peer-to-Peer Systems (IPTPS'02)*.

[14] Federal Trade Commission. 2012. *Protecting Consumer Privacy in an Era of Rapid Change*. Federal Trade Commission.

[15] M. Fredrikson and B. Livshits. 2011. RePriv: Re-imagining content personalization and in-browser privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy (SP'11)*.

[16] Google. 2012. How It Works: Ads Help. Retrieved February 22, 2021 from http://support.google.com/ads/bin/answer.py?hl=en&answer=2662749.

[17] S. Guha, B. Cheng, and P. Francis. 2010. Challenges in measuring online advertising systems. In *Proceedings of the Internet Measurement Conference IMC'10)*.

[18] S. Guha, B. Cheng, and P. Francis. 2011. Privad: Practical privacy in online advertising. In *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI'11)*. 169–182.

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

[20] A. Hannak, P. Sapiezynski, A. Kakhki, B. Krishnamurthy, D. Lazer, A. Mislove, and C. Wilson. 2013. Measuring personalization of Web search. In *Proceedings of the 22nd International Conference on World Wide Web (WWW'13)*.

[21] Networking Advertising Initiative. 2017. Home Page. Retrieved February 22, 2021 from http://www.networkadvertising.org.

[22] Ian Jolliffe. 2002. *Principal Component Analysis*. Wiley Online Library.

[23] B. Krishnamurthy and C. Wills.2008. Characterizing privacy in online social networks. In *Proceedings of the 1st Workshop on Online Social Networks (WOSN'08).*

[24] B. Krishnamurthy and C. Wills. 2009. Privacy diffusion on the web: A longitudinal perspective. In *Proceedings of the 18th International Conference on World Wide Web (WWW'09).*

[25] B. Krishnamurthy and C. Wills.2010. Privacy leakage in mobile online social networks. In *Proceedings of the 3rd Conference on Online Social Networks (WOSN'10).*

[26] C. Lesniewski-Laas and F. Kaashoek. 2010. Whanau: A Sybil-proof distributed hash table. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10).*

[27] Ee-Peng Lim, Viet-An Nguyen, Nitin Jindal, Bing Liu, and Hady Wirawan Lauw. 2010. Detecting product review spammers using rating behaviors. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10).*

[28] Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. 2010. Measuring the mixing time of social graphs. In *Proceedings of the 2010 ACM/USENIX Internet Measurement Conference.*

[29] Mummoorthy Murugesan and Chris Clifton. 2009. Providing privacy through plausibly deniable search. In *Proceedings of the SIAM International Conference on Data Mining (SDM'09)*, 768–779.

[30] A. Narayanan and V. Shmatikov. 2008. Robust de-anonymization of large sparse datasets, or how to break anonymity of the Netflix Prize dataset. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP'08).*

[31] Netflix. 2009. Netflix Prize. Retrieved February 22, 2021 from http://www.netflixprize.com/.

[32] Helen F. Nissenbaum and Howe Daniel. 2009. TrackMeNot: Resisting surveillance in web search. In *Lessons fromthe Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, I. Kerr, C. Lucock, and V. Steeves (Eds.). Oxford University Press, Oxford, UK, 1–23.

[33] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 2007. 802.11 user fingerprinting. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom'07).* 99–110

[34] Panagiotis Papadopoulos, Antonis Papadogiannakis, Michalis Polychronakis, Apostolis Zarras, Thorsten Holz, and Evangelos Markatos. 2013. k-Subscription: Privacy-preserving microblogging browsing through obfuscation. In *Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC'13).* 49–58.

[35] A. Petit, T. Cerqueus, S. B. Mokhtar, L. Brunie, and H. Kosch. 2015. PEAS: Private, efficient and accurate web search. In *Proceedings of 2015 IEEE International Conference on Trust, Security, and Privacy in Computing and Communications (TrustCom'15).* 571–580.

[36] C. Riederer, V. Erramilli, A. Chaintreau, B. Krishnamurthy, and P. Rodriguez. 2011. For sale : Your data: By : You. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks (HotNets'11).*

[37] F. Roesner, T. Kohno, and D. Wetherall. 2012. Detecting and defending against third-party tracking on the web. In *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation (NSDI'12).*

[38] V. Toubiana, A. Narayanan, D. Boneh, H. Nissenbaum, and S. Barocas. 2010. Adnostic: Privacy preserving targeted advertising. In *Proceedings of the 2010 17th Annual Network and Distributed System Security Symposium (NDSS'10).*

[39] N. Tran, B. Min, J. Li, and L. Subramanian. 2009. Sybil-resilient online content voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI'09).*

[40] TRUSTe. 2017. Home Page. Retrieved February 22, 2021 from http://www.truste.com.

[41] Bimal Viswanath, Muhammad Ahmad Bashir, Mark Crovella, Saikat Guha, Krishna P. Gummadi, Balachander Krishnamurthy, and Alan Mislove. 2014. Towards detecting anomalous user behavior in online social networks. In *Proceedings of USENIX Security Symposium (USENIX Security'14).*

[42] B. Viswanath, A. Post, K. Gummadi, and A. Misolve. 2010. An analysis of social network-based Sybil defenses. In *Proceedings of the 2010 Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'10).*

[43] Gang Wang, Tristan Konolige, Christo Wilson, Xiao Wang, Haitao Zheng, and Ben Y. Zhao. 2013. You are how you click: Clickstream analysis for Sybil detection. In *Proceedings of the 22nd USENIX Security Symposium (Usenix Security'13).*

[44] Gang Wang, Tianyi Wang, Haitao Zheng, and Ben Y. Zhao. 2014. Man vs. machine: Practical adversarial detection of malicious crowdsourcing workers. In *Proceedings of the 23rd USENIX Security Symposium (Usenix Security'14).*

[45] Y. Wang, D. Burgener, A. Kuzmanovic, and G. Macia. 2011. Understanding the network and user-targeting properties of web advertising networks. In *Proceedings of the 2011 31st International Conference on Distributed Computing Systems (ICDCS'11).*

[46] N. Xia, H. Song, Y. Liao, M. Iliofotou, A. Nucci, Z. Zhang, and A. Kuzmanovic. 2013. Mosaic: Quantifying privacy leakage in mobile networks. In *Proceedings of the 2013 Annual Conference of theACM Special Interest Group on Data Communication (SIGCOMM'13).*

[47] X. Xing, W. Meng, D. Doozan, N. Feamster, W. Lee, and A. Snoeron. 2014. Exposing inconsistent Web search results with Bobble. In *Proceedings of the 2014 Passive and Active Measurement Conference.*

[48] T.-F. Yen, Y. Xie, F. Yu, R. Yu, and M. Abadi. 2012. Host fingerprinting and tracking on the web: Privacy and security implications. In *Proceedings of the 2012 19th Annual Network and Distributed System Security Symposium (NDSS'12)*.

[49] Haifeng Yu, Phillip B. Gibbons, Michael Kaminsky, and Feng Xiao. 2008. SybilLimit: A near-optimal social network defense against Sybil attacks. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy (SP'08)*.

[50] H. Yu, M. Kaminsky, P. Gibbons, and A. Flaxman. 2006. SybilGuard: Defending against Sybil attacks via social networks. In *Proceedings of the 2006 Annual Conference of the ACM Special Interest Group on Data Communication (SIGCOMM'06)*.

Received April 2019; revised September 2020; accepted October 2020