Host Fingerprinting and Tracking on the Web: Privacy and Security Implications

Ting-Fang Yen[1]*, Yinglian Xie[2], Fang Yu[2], Roger Peng Yu[3], Martín Abadi[2]†

[1]RSA Laboratories
[2]Microsoft Research Silicon Valley
[3]Microsoft Corporation
tingfang.yen@rsa.com, {yxie, fangyu, rogeryu, abadi@microsoft.com}

Abstract

Many web services aim to track clients as a basis for analyzing their behavior and providing personalized services. Despite much debate regarding the collection of client information, there have been few quantitative studies that analyze the effectiveness of host-tracking and the associated privacy risks.

In this paper, we perform a large-scale study to quantify the amount of information revealed by common host identifiers. We analyze month-long anonymized datasets collected by the Hotmail web-mail service and the Bing search engine, which include millions of hosts across the global IP address space. In this setting, we compare the use of multiple identifiers, including browser information, IP addresses, cookies, and user login IDs.

We further demonstrate the privacy and security implications of host-tracking in two contexts. In the first, we study the causes of cookie churn in web services, and show that many returning users can still be tracked even if they clear cookies or utilize private browsing. In the second, we show that host-tracking can be leveraged to improve security. Specifically, by aggregating information across hosts, we uncover a stealthy malicious attack associated with over 75,000 bot accounts that forward cookies to distributed locations.

1 Introduction

It is in the interest of web services and ISPs to track the mobility and usage patterns of client hosts. This tracking allows them to understand user behavior for supporting applications such as product suggestions, targeted advertising, and online fraud detection. However, clients may not wish that their activities be tracked, and can intentionally remove stored browser cookies or choose not to perform user logins. The growing awareness of privacy concerns is exemplified by the recent "do-not-track" initiative from the Federal Trade Commission [15], which outlines guidelines to which service providers must adhere in the collection and distribution of client information.

Several works aim to improve the accuracy of host-tracking by collecting detailed host information, such as installed browser plug-ins and system fonts [20, 31] or packet-level information that reveals subtle hardware differences [28]. By comparison, few studies exist on the effectiveness and privacy implications of host-tracking. Previous work tends to be qualitative in nature [29, 30] or limited to a single identifier [20].

In this paper, we attempt to facilitate the debate regarding host-tracking by performing a large-scale study to quantify the amount of identifying information revealed by common identifiers. Such analysis is critical to both service providers and end users. For example, service providers can determine where existing identifiers are insufficient and more sophisticated methods may be preferred. Users who do not wish to be tracked can learn the circumstances in which they can be identified accurately, so that they can take effective measures to protect privacy. Our analysis is based on

*This work was done while Ting-Fang was an intern at Microsoft Research.

†Martín Abadi is also affiliated with the University of California, Santa Cruz.

month-long anonymized datasets from the Hotmail webmail service and the Bing search engine, including hundreds of millions of users across the global Internet IP address space. By characterizing hosts' activities across time using "binding windows", we show that common identifiers allow us to track hosts with high accuracy.

We further consider cases where users take initiatives to preserve privacy, e.g., by clearing cookies or switching to private browsing mode. Specifically, we analyze "one-time" cookies that do not return again in subsequent web requests, a phenomenon known as cookie churn. These cookies appear to be anonymous. However, by applying our host-tracking results, we show that a surprisingly large fraction can be recognized as belonging to returning users.

In addition to its privacy implications, we demonstrate that host-tracking can also be applied to improve security. We examine the mobility patterns of hosts traveling across multiple IP ranges, and establish normal user mobility profiles from aggregate host activities. In doing so, we are able to analyze unusual activities, e.g., the use of anonymous routing networks, and develop methods to detect attacks. In particular, our study uncovers previously unknown suspicious cookie-forwarding activities, which may have been adopted by attackers to evade spamming detection.

The key findings of this paper include:

- We show that 60%-70% of HTTP user-agent strings can accurately identify hosts in our datasets. When augmented with coarse-grained IP prefix information, the accuracy can be improved to 80%, similar to that obtained with cookies. User-agent strings combined with IP addresses have an entropy of 20.29 bits—higher than that of browser plug-ins, screen resolution, timezone, and system fonts combined [20].

- Applying our results to study cookie churn, we find that a service provider can recognize and track 88% of the "one-time" cookies as corresponding to users who later returned to the service. Among these users, 33% made an effort to preserve their privacy, either by clearing cookies through browser options or utilizing private browsing mode.

- Employing general mobility patterns derived by tracking hosts across network domains, we uncover malicious behaviors where cookies are forwarded from one IP address to distributed locations. In total, we identify over 75,000 bot Hotmail accounts in this relatively stealthy attack that has not been detected before.

Although our research relies on anonymized datasets from Hotmail and Bing, the analyses that we describe are a research effort only. Our goal is not to identify or study specific individual activities, but rather to understand the patterns of the aggregated activities and to explore their implications.

In the following, we first describe the identifiers that we study and our host-tracking methodology in Section 2, and present the evaluation of those identifiers in Section 3. We investigate the privacy and security implications of host-tracking in the context of cookie churn in Section 4 and of host mobility in Section 5. Finally, we describe related work in Section 6 and conclude in Section 7.

2 Exploring Common Identifiers

Given a log of application-level events collected over time, such as requests directed to a web server or user logins to a service, our goal is to quantify the amount of host-identifying information that is captured in identifiers within the log. Specifically, for an identifier I, which may take on a finite set $F_I = \{f_1, f_2, \ldots, f_n\}$ of possible values (called *fingerprints*), we are interested in whether a fingerprint f_i uniquely corresponds to a single host, among all hosts involved in the log. As we consider only client hosts in our scenario, we use *clients* or *hosts* interchangeably throughout the paper.

We assume the perspective of a passive observer of identifiers within application-level events. The common identifiers explored in this work include 1) user-agent string (UA), 2) IP address, 3) browser cookie, and 4) user login ID. We choose these identifiers because they are not particular to our datasets, and are available in a wide variety of service logs.

2.1 Host-tracking Graph

Our host-tracking approach attempts to infer the presence of a host at an IP address during a certain time interval. Upon observing a fingerprint f (and only f) that appears at an IP address A over a time interval Δt, we can infer a "binding window" for f. Events occurring within Δt at A can then be attributed to the host corresponding to f. (Hosts behind NATs/proxies can complicate matters; we quantify the occurrence of such hosts in our data in Section 3.3.)

Figure 1 illustrates how we infer the binding windows. In this example, user-agent strings (UA) are the identifiers, and the events are queries to a web search engine. A fingerprint UA_1 appears in two consecutive

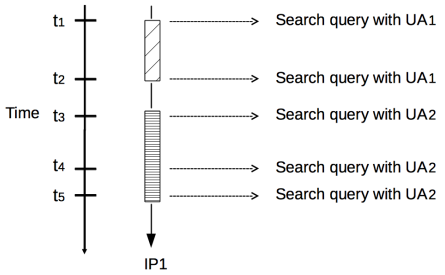


Figure 1. Binding windows identified on one IP.

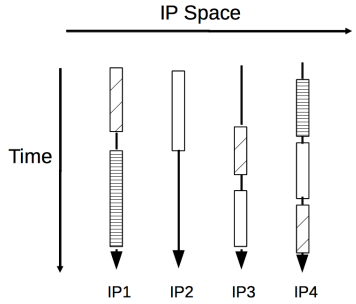


Figure 2. Example of a host-tracking graph. Bars with different patterns denote binding windows corresponding to different fingerprints.

search queries at time t_1 and t_2, followed by queries at time t_3, t_4, and t_5 with a different fingerprint UA_2. Thus we can identify binding windows corresponding to two different "hosts" on this IP: one spanning the time range $[t_1, t_2]$, and another spanning $[t_3, t_5]$. Having examined all search query events, we can construct a host-tracking graph as in Figure 2. Note that a fingerprint may be associated with multiple binding windows (since the host may not be up all the time) and across different IP addresses (e.g., because of DHCP). We refer to the host-tracking graph that represents hosts by identifier I as G_I.

A similar concept of host-tracking graph was also used by HostTracker [38] to support Internet accountability. HostTracker groups together user login IDs that are likely to be associated with the same host, e.g., family members that share a computer at home. It also filters events related to bots and large proxies. In contrast to this previous work, we make a broader use of the host-tracking graph (with a variety of common identifiers), and we apply host-tracking to the cookie-churn study (in Section 4) and the host-mobility analysis (in Section 5).

2.2 Datasets

The data for our study includes a month-long user login trace collected by the Hotmail web-mail service in August 2010. The trace contains coarse-grained information about the OS and browser type (e.g., Windows, Mozilla), the IP address from which the login was made, the time of the login event, and the anonymized user ID. In the following, we refer to this as the Webmail dataset.

We also obtained a month-long dataset consisting of search query events directed to the Bing search engine in August 2010. This data includes the fine-grained user-agent string from the HTTP header (anonymized via hashing), the IP address from which the query was issued, the time of the query, the anonymized cookie ID assigned by the search engine, and the date that the cookie ID was created. Specifically, the anonymized cookie ID is a persistent identifier that does not change over time, if users do not clear cookies or use private browsing. We refer to this as the Search dataset. As part of the processing performed by the Bing search engine, events generated by known bots are filtered in advance.

To validate our client-tracking approach, we leveraged a month-long sampled log of Windows Update events, also from August 2010. This data contains the time at which the update was performed, the IP address, and the anonymized hardware ID that is unique to the host. This is the Validation dataset.

Table 1 shows the fields and the total number of unique IPs observed in each dataset. All three datasets include tens to hundreds of millions of IP addresses, spanning a large IP address space.

The published privacy policies for Hotmail, Bing, and Windows Update address the storage, use, sharing, and retention of data collected in the course of the operation of these services. In particular, they indicate that Microsoft may employ this data for analyzing trends and for operating and improving its products and services, as we aim to do with this work. Since the datasets are sensitive, they are not publically available for further research.

2.3 Validation and Metrics

Without ground truth for the host-IP mappings, we evaluate a host-tracking graph G_I by overlapping it with the Validation dataset. If a fingerprint is able to correctly

Dataset	User-agent information	IP address	Timestamp	ID	Unique IP addresses
Webmail	OS and browser type	Yes	Yes	User ID	308 million
Search	User-agent string (UA)	Yes	Yes	Cookie ID	131 million
Validation	N/A	Yes	Yes	Hardware ID	74 million

Table 1. Fields in each dataset.

track a host, its bindings should overlap only with Windows Update events associated with a single hardware ID. Conversely, a hardware ID is also expected to overlap with bindings associated with only one fingerprint.

We quantify the accuracy of an identifier using *precision* and *recall*. Let $\mathsf{hidcount}(f)$ denote the number of hardware IDs to which a fingerprint f corresponds, and $\mathsf{fpcount}(m)$ the number of fingerprints to which a hardware ID m corresponds. Precision is defined as the percentage of fingerprints that correspond to one host (i.e., one hardware ID), while recall is the percentage of hosts that correspond to one fingerprint.

$$\mathsf{Precision}_I = \frac{\mid \{f : \mathsf{hidcount}(f) = 1, f \in F_I\} \mid}{\mid F_I \mid}$$

$$\mathsf{Recall}_I = \frac{\mid \{m : \mathsf{fpcount}(m) = 1, m \in M_I\} \mid}{\mid M_I \mid}$$

F_I is the finite set of values that identifier I takes in our dataset, i.e., the fingerprints (after some initial filtering, as described below). M_I is the set of hardware IDs that overlap with the host-tracking graph G_I. Roughly speaking, precision quantifies how accurate an identifier is at representing a host. Recall quantifies how well an identifier is able to track the events associated with the corresponding host in a log.

We also measure the *entropy* of an identifier, H_I, which is the amount of information identifier I contains that can distinguish hosts. The entropy is defined as

$$\mathsf{H}_I = -\sum_{f \in F_I} \mathsf{Pr}(f) \log_2(\mathsf{Pr}(f))$$

where $\mathsf{Pr}(f)$ is the probability of observing fingerprint f in the application log. A higher entropy indicates a smaller probability that any two clients are associated with the same fingerprint.

In our validation, we consider only those fingerprints that overlap with more than one Windows Update event, and only those hardware IDs that overlap with more than one application-level event pertaining to our identifiers. These restrictions allow us to focus on the portion of data that we can validate, though they can be biased to those clients that access the services consistently (i.e., multiple times and with the same identifiers). Similarly, because of the datasets available to us, our study is based on clients of Microsoft services. We acknowledge that any dataset will be incomplete and possibly biased.

3 Client-Tracking Results

In this section, we construct host-tracking graphs using the common identifiers user-agent string (UA), IP address, cookie ID, and user login ID, and evaluate their precision and recall. In particular, we explore the distinguishing power of UA by examining the browser anonymity sets. We also measure the impact of proxies and NATs in our study in Section 3.3, and describe the increased accuracy and confidence of tracking stable hosts in Appendix A.

Our analysis focuses on host-tracking within each network domain, derived using the BGP prefix entries obtained from RouteViews [9]. We investigate the occurrences of identifiers at multiple network locations in Section 5, in which we also study the security implications of host-tracking.

3.1 Precision and Recall

Table 2 presents our results on host-tracking. After overlapping the Validation dataset with the host-tracking graphs, the number of unique fingerprints and hardware IDs included in our evaluation is still large—on the order of millions.

Several observations are evident from Table 2. First, browser information (UA) alone can identify hosts quite well. Its 62.01% precision is perhaps surprising, as UA strings are commonly regarded as providing insufficient information to reveal host identities. Second, a combination of UA with the IP address (i.e., fingerprinting hosts by distinct (UA, IP) pairs) can boost the precision up to 80.62%. In fact, combining UA with only the IP prefix is sufficient to achieve approximately the same result as with UA+IP. This suggests that anonymization techniques that store the IP prefix may still retain distinguishing information. Third, cookie IDs offer only slightly better precision and recall than UA+IP. The inaccuracies of cookie IDs can be partly attributed to cookie churn, a phenomenon we study in more detail in Section 4.

Identifier I	Precision (%)	Recall (%)	Fingerprint count	Hardware ID count
UA	62.01%	72.11%	254,762	3,073,690
UA, IP address	80.62%	68.84%	1,685,416	1,771,907
UA, /24 IP prefix	79.33%	69.43%	1,652,546	1,772,104
Cookie ID	82.35%	68.64%	1,340,635	1,375,074
Cookie ID (with HostTracker)	79.74%	99.13%	713,110	1,001,450
User ID (with HostTracker)	92.82%	93.51%	4,608,980	4,820,116

Table 2. Common identifiers in host-tracking, evaluated using the Validation dataset.

As another method to make use of the identifiers, we also apply HostTracker [38] to the cookie IDs and user IDs from our Search and Webmail datasets, respectively. In the former case, the clients are now tracked by a group of correlated cookies, e.g., those belonging to two browsers running on a machine in parallel. In the latter case, user login IDs that frequently appear together, e.g., family members that share a computer at home, are used to track clients. We find user IDs achieving high precision and recall (over 92%), demonstrating that they are strongly tied to individual hosts.

Since HostTracker yields relatively high precision and recall with user IDs, we have also evaluated the other identifiers against user IDs (instead of hardware IDs). Even though hardware IDs and user IDs overlap with different portions of the datasets, we obtain results consistent with those of Table 2.

To summarize, we show that common identifiers can track hosts reasonably well, particularly when they are used in combination.

3.2 Browser Anonymity Set

Our evaluation suggests that a large fraction of browsers provide enough information to fingerprint hosts within each network domain. In this section, we examine in detail the anonymity set of browser fingerprints, defined as the set of hardware IDs that share the same fingerprint. Even though 62% of UAs map to unique hosts, popular UA strings still have large anonymity sets, i.e., additional examination shows that the most common fingerprint, `Mozilla/4.0(compatible;MSIE6.0;WindowsNT5.1;SV1)`, corresponds to 124,355 (4.05%) of the hardware IDs that overlap with the UA host-tracking graph.

Figure 3 compares the size of the anonymity sets for UA and UA+IP. We find 98.92% of the UA+IP fingerprints to be relatively rare, with fewer than five hardware IDs, while this holds for only 89.69% of the UA fingerprints.

To quantify the amount of identifying information provided by browser fingerprints, we calculate their entropy. In our data, UA has an entropy of 11.59 bits, while the entropy of UA+IP is 20.29 bits. A study performed by Eckersley et al. [20] probed the remote client for installed plug-ins, screen resolution, timezone, system fonts, and user-agent strings, which altogether yielded an entropy of 18.1 bits. While this suggests that their detailed information provides more distinguishing power than UA alone, it is interesting to observe that such information may be less distinguishing than simply combining UA and IP address.

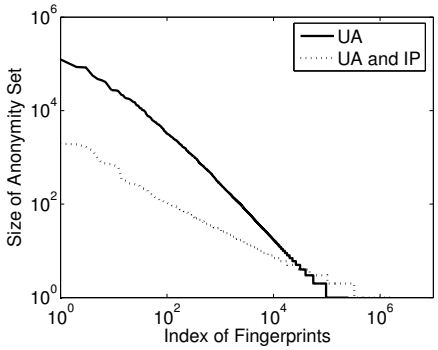


Figure 3. The distribution of the size of the browser anonymity sets, plotted in log-scale.

These results confirm our finding that UA strings augmented with IP addresses can identify hosts well. However, popular UA strings still have large anonymity sets. Changing the default UA string to one that corresponds to a popular browser version may hence allow a client to become less distinguishable.

3.3 Impacts of Proxies and NATs

Among the common identifiers we explored, none of them performs perfectly. Aside from their inherent ambiguity (e.g., some UAs are more common than others, cookies can be removed), proxies and NATs introduce

fundamental difficulties in tracking hosts. The ability to detect and measure them allows us to understand precisely where such practical limitations will apply.

We first quantify the prevalence of large proxies and NATs that are typically configured by ISPs or enterprises. To do so, we examine "hosts" that correspond to a large number of user login IDs or cookie IDs. A small fraction of IP addresses in our datasets—31,874 and 2,151 from the Webmail and Search dataset, respectively—is each associated with more than 5,000 unique login IDs and cookie IDs. These are likely large proxies and we filter them in our evaluation.

Next, we examine small NAT devices that are often used by home networks. In this case, since it is relatively rare for a client to be running multiple operating systems in parallel, we leverage the coarse-grained OS type and IP address recorded for each user login event in the Webmail dataset. The majority (80.31%) of our login ID fingerprints are associated with only one unique user ID. When we observe multiple OS types, all from the same IP address, it indicates that the "host" may actually be a NAT device that masks multiple clients.

From this experiment, we find 10.60% hosts likely to be NATs. This number is a lower bound, since we cannot distinguish clients that are running the same OS behind a NAT device. Table 3 shows that while the large majority of NATed hosts include multiple Microsoft Windows OSes, hand-held devices also comprise a large fraction (about 16%). With the increasing popularity of multiple home devices and smart phones, we expect the percentage of NATs to grow further.

OS Types	NAT hosts (%)
Multiple Windows	81.32%
Windows and Hand-held device	15.62%
Windows and Mac OS/Unix	2.19%
Hand-held and Mac OS/Unix	0.55%
Windows, Hand-held, and Mac OS/Unix	0.31%
Multiple Mac OS/Unix	0.01%
Multiple Hand-held devices	0.01%

Table 3. Breakdown of the OS types found to be associated with hosts behind NATs.

4 Application: Cookie Churn Study

As the primary method for web sites to track returning users without requiring login-based authentication, browser cookies play an important role in customizing web services and maintaining user statistics. However, as shown in Section 3, using cookie IDs as client fingerprints can be unreliable. In particular, they have a relatively low recall rate—32% of the hardware IDs in our evaluation cannot be completely tracked by cookies.

A main source of the low recall rate is *cookie churn*, which we define as the phenomenon of cookies appearing at least once but *not* appearing again in subsequent web requests received by a server (within some observation time window). For service providers, being able to track hosts will allow them to quantify the underlying causes behind the cookie-churn phenomenon. In this section, we measure and analyze cookie churn in the Search dataset. (Among the datasets available to us, it is the only one that contains cookie IDs.) By applying our host-tracking methodology, we show that some client users may still be identified despite cookie churn.

4.1 Cookie Churn Measurement

Among cookie IDs that appear on the first day of our Search dataset, the rate of cookie churn, i.e., the fraction of cookie IDs that never returned again within our month-long observation, is 47.86%. On average, the daily cookie churn rate is around 45% across month-long sliding windows.

Furthermore, 81.98% of the new cookie IDs that are born on the first day of the Search dataset never returned within the month. For all cookie IDs observed on the first day of the month, Figure 4 shows cumulative distributions of the date that old and new cookies appear a second time. The churn rate of new cookies is significantly higher than that of old cookies—a difference of more than 40%.

4.2 Possible Reasons for Churn

Clearly, cookie churn can result from users quitting the service. As shown in Figure 4, engaged users that access the service multiple times (with old cookies) are more likely to return than new users.

Another reason for cookie churn is the removal of cookies from the client browser. This removal can happen in several cases, including when users manually clear cookies, when they set their browsers to automatically clear cookies on exit, or when users switch into or out of private browsing mode. Supported by all major web browsers today, private browsing takes a user's activities off records by removing caches, history, and in particular, cookies that are set during private mode.

To study how private browsing mode affects the cookie events observed by web services, we examine

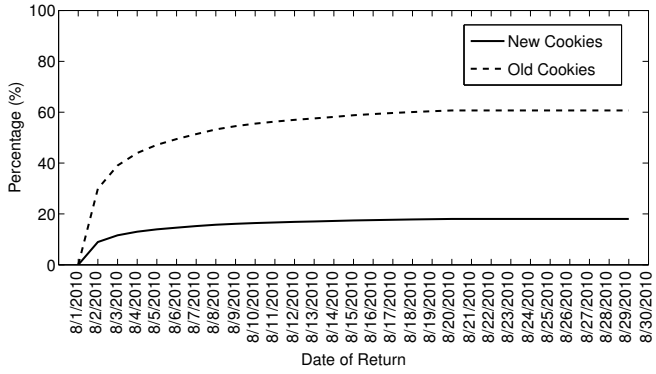


Figure 4. For cookie IDs observed on the first day of the month, the cumulative distribution of the date that old and new cookies appear again in our dataset.

Cookie Set	Cookie Accessed	Firefox	Safari	Chrome	IE
Public	Private	No	Yes	No	No
Private	Same private session	Yes	Yes	Yes	Yes
Private	Different private session	No	No	No	No
Private	Public	No	No	No	No

Table 4. Accessibility of cookies in different browsing modes.

four of the most popular browsers in use today: Firefox (version 3.6.11), Safari (version 5.0.2), Chrome (version 7.0.517.41), and Internet Explorer (version 8.0). Table 4 shows, for the browsing mode under which a cookie is set (the first column), whether the same cookie can be accessed under another browsing mode (the second column). In all cases, a cookie set in private mode can be accessed repeatedly in the same private browsing session, but not across different private browsing sessions. No cookies set in private mode can be accessed in public mode. Safari is the only browser that allows private mode to access cookies set in public mode.

In the next subsection, we perform fine-grained classification to quantify the above possible causes of cookie churn and characterize the corresponding users.

4.3 Understanding Cookie Churn

Applying the host-tracking results, we analyze cookie churn by identifying cookies that are associated with the same client host. In Section 3.1, we show that the host-tracking graph G_{UID} derived from user login IDs (with HostTracker) achieved over 92% precision and recall in tracking clients, which are represented by hardware IDs from the Validation dataset. Thus we use the hosts defined in G_{UID} to serve as ground truth for studying cookie churn. By overlapping G_{UID} with the Search dataset, we consider cookies whose query events fall into binding windows associated with the same host as corresponding to the same user (since user activity roughly approximates host activity).

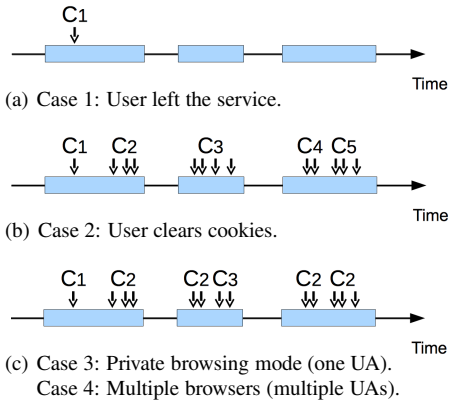
We focus on studying new cookie churn, as it is more significant than that of old cookies (see Figure 4). We refer to the set of "one-time" cookie IDs (CIDs) that are born on the first day but do not return again in our dataset as the *churned new cookie IDs*. In total, there are 437,914 users (or hosts) that overlap with 847,196 churned new CIDs in the Search data. The number of hosts is only about half of the number of churned cookies IDs. We investigate the four cases that result in new cookie churn, as illustrated in Figure 5, where the breakdown of users belonging to each category is shown in Table 5. We elaborate on each of these cases separately below.

4.3.1 Case 1: Non-Returning Users

If a CID overlaps with one of host h's binding windows at time t, but no other CIDs overlap h's bindings from time t onwards, we consider this as corresponding to a user who does not return to the service (Figure 5(a)).

	Case 1	Case 2	Case 3	Case 4
Number of users	101,427	77,120	67,310	192,057
Percentage of users (%)	23.16%	17.61%	15.37%	43.86%
Number of churned new CIDs	101,427	77,147	123,757	544,865
Percentage of churned new CIDs (%)	11.97%	9.12%	14.60%	64.31%

Table 5. Breakdown of the churned new cookie IDs into four categories of users.



(a) Case 1: User left the service.

(b) Case 2: User clears cookies.

(c) Case 3: Private browsing mode (one UA).
Case 4: Multiple browsers (multiple UAs).

Figure 5. Four cases of cookie churn. C_1 is the churned new cookie ID. Horizontal bars denote binding windows for a "host" defined by user IDs.

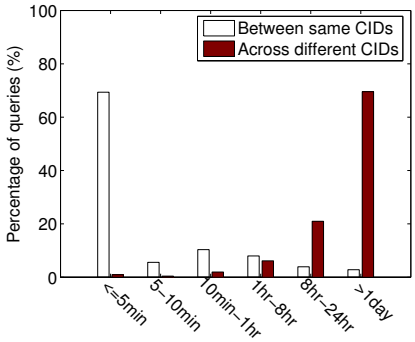


Figure 6. Distributions of query intervals.

We find that this case accounts for only 11.97% of the churned new CIDs. Thus, despite the high cookie churn rate, the majority (88.03%) of the churned new cookie IDs correspond to returning users who might still be tracked. The behaviors of the non-returning users are examined in detail in Appendix B.

4.3.2 Case 2: Users that Clear Cookies

Cookie churn can also result from users intentionally removing cookies. In this case, a host h's bindings should overlap with CIDs generated consecutively in time (Figure 5(b)). Each CID may be associated with multiple queries that typically belong to a session. Among hosts with new cookie churn, we find 77,120 (17.61%) in this category. Since we observe only cookies issued by the Bing search engine, we cannot distinguish between users who clear all cookies and those who selectively clear cookies from certain domains.

To find whether users clear cookies on exiting browsers, we examine the time intervals between consecutive queries associated with the same CID, and compare with those between consecutive queries associated with different CIDs. Figure 6 shows that the former is distinctly smaller, with 75% of them below 10 minutes and hence likely to belong to one session. By contrast, 90% of the query intervals between different CIDs are larger than 8 hours. This suggests that most users clear cookies *per session*, e.g., when they close the browser window.

We also find a small fraction (3.85%) of users whose cookies are cleared *per query*, i.e., each of their queries is associated with a different CID. These might be users who take extreme measures to clear cookies for each query to preserve privacy. However, such patterns can become a distinctive feature that makes tracking easier, despite the user's intention of remaining anonymous.

4.3.3 Case 3: Users with In-Private Browsing Mode

Another reason for cookie churn is the use of the browser's private browsing mode. As illustrated in Figure 5(c), upon entering private mode, the old cookie (C_2) set under public mode is replaced by a new cookie (C_3). Upon exiting the private mode, the old cookie (C_2) will continue to be used by the same user.

We focus on the hosts whose cookies appear interleaved in their binding windows, where an old cookie continues to appear after the user submit queries with a newer cookie. Since entering private browsing mode does not change the browser used by the host, we identify those associated with a single UA string as users

who utilize private browsing mode, and 15.37% of users belong to this category.

Together with case 2, there are in total around 33%—a non-trivial fraction—of users who would like to preserve privacy by either clearing cookies or entering private browsing mode. These users may still be tracked when service providers combine the host-tracking results from other identifiers (e.g., login IDs) with cookie data.

4.3.4 Case 4: Users with Multiple Browsers

For the remaining users, we observe multiple cookies co-existing (as in Section 4.3.3), though they are associated with different UA strings. Upon examining these users more closely, we find around 67% associated with only two or three UAs. This observation suggests that these cases correspond to single hosts with multiple browsers or small home NATs. While it is more difficult to track hosts behind NATs, we note that the anonymity sets tend to be too small in such cases to protect user privacy.

A small fraction of these cases (3%) are associated with a large number of UA strings, which suggests that they are large proxies or NATs. Routing traffic through proxies thus provides better means for users who do not wish to be tracked.

Summary We study the cookie-churn phenomenon where privacy-aware users may clear cookies or switch to private browsing. We show that by applying host-tracking results with other identifiers, service providers may still be able to identify a large fraction (88%) of the "one-time", churned new cookie IDs as corresponding to users who return to the service.

5 Application: Host Mobility Study

In addition to switching between IP addresses within the same network (for instance, because of DHCP), a host may also travel across different IP ranges. This can occur if the host is a mobile device, or when a virtual private network (VPN) is used. Above, we track hosts within each IP prefix range separately, though it is also desirable to study clients that travel across domains, e.g., for traffic engineering or network management. More importantly, host mobility patterns can benefit security as well. We demonstrate this point by applying our host-tracking results to detect abnormal and malicious activities.

To understand the mobile behavior of hosts at a large scale, we make use of cookie IDs, since they are more closely tied to specific devices than other identifiers we studied in Section 3. We use the Search dataset for our study. Among cookie IDs in this dataset, 7.9 million appeared at more than one domain. While the majority of these cross-domain activities are associated with normal user travel patterns, there also exist unusual or suspicious activities, for example, cookie forwarding of the kind supported by CookieCooker [1].

In this section, we focus on detecting the following two abnormal host mobility patterns:

- Some cookie IDs move quickly between multiple domains, suggesting that they may not correspond to hosts who travel physically. In particular, we study those cookies that may be associated with anonymous routing, such as Tor routing [37].

- During an investigation into suspicious user email traffic that do not conform to the general host mobility profile, we uncover a stealthy type of malicious cookie-forwarding activity.

In the following, we first study patterns corresponding to users traveling across domains in general. We then use those patterns as baseline to identify abnormal activities.

5.1 Host Mobility Patterns

Our analysis yields a few key observations on general host mobility patterns. First, as shown in Table 6, ASes associated with cellular networks, i.e., Verizon Wireless and Carphone Warehouse Broadband Services, are ranked among the top domains with the largest number of traveling cookies. This fact reflects the proliferation of smart phones with mobile Internet access. In total, we find around 20% of the cookies among the top 500 AS pairs to be associated with cellular networks (Verizon Wireless, AT&T Wireless, Vodafone, Sprint, etc.).

AS pair	# Cookies	Affiliations
AS 17557, 45595	152871	Pakistan Telecom (PK)
AS 6167, 22394	70941	Verizon Wireless (US)
AS 13285, 43234	56600	Opal Telecom, Carphone Warehouse Broadband (GB)
AS 4134, 4837	52520	ChinaNet (CN)
AS 8228, 15557	36812	Neuf Cegetel (FR)

Table 6. Top five AS pairs associated with traveling cookies.

Second, we find traveling hosts to exhibit strong geographic locality. 83% of the cookies move between networks within the same country, and this number is even higher for the U.S. (95.44%). The strong geographic locality pattern can also be observed among cookies that travel across countries. Figure 7 shows the topology of international host travel, also drawn from the top 500 AS pairs. The node "EU" in the figure represents multi-regional networks in the European Union, which are not exclusively part of any European country. The size of each node in the figure is proportional to the number of cookies that originated from that country or region. The edges indicate the direction of travel. The figure shows that host mobility is largely bi-directional, and is commonly localized within the same general region (e.g., Europe).

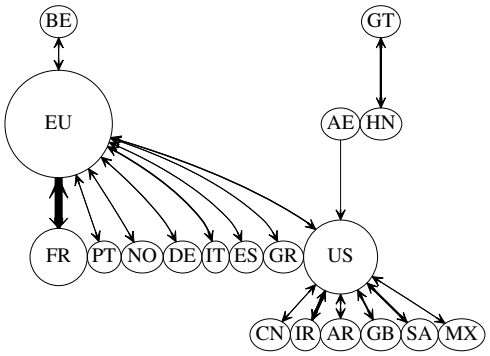


Figure 7. The topology of host mobility by country or region (e.g., "EU"), for top 500 AS pairs.

Third, a closer look at the AS topology of host mobility in the U.S. shows the existence of "hub" ASes that are connected to many smaller "leaf" ASes. The former are commonly associated with DSL broadband Internet services, while the latter include institutional and corporate networks. This star topology could result from clients' commuting patterns between home and work.

Finally, in addition to the source and sink domains, we are also interested in how far the hosts roam, i.e., how many ASes they travel through. Figure 8 plots the distribution of the number of ASes traveled by each host, with the Y-axis in log scale. The large majority (90%) of cookies are associated with only two domains.

These observations, based on aggregate information across the 7.9 million traveling hosts in the Search dataset, reflect general mobility patterns at a large scale. In the following, we investigate specific activities that fall outside this norm, including those that may involve suspicious behavior.

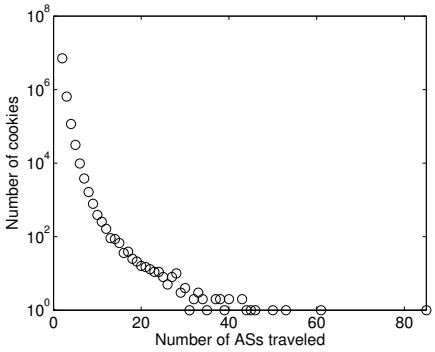


Figure 8. The distribution of the number of ASes traveled by each CID. Y-axis is in log scale.

5.2 Identifying Virtual Client Travel

Although the majority of traveling cookies correspond to physical host mobility, such as those associated with cellular networks, some switch between domains faster than seemingly possible for physical travel. Consecutively appearing from different ASes within a matter of minutes, the rapid movement of these cookies suggests the presence of some form of virtual client travel.

5.2.1 VPN Traffic Patterns

For the large majority of hosts that travel rapidly across only two or three domains, they likely have used VPNs or proxies. Virtual private networks (VPNs) allow traffic to be privately tunneled between two machines that are not in the same subnet. Creating an overlay network of clients that belong to the same organization, they are commonly used to provide corporate resources to remote employees. From the perspective of a web server, a user connecting to her company network from a DSL line at home can generate multiple requests with the same cookie, though they appear from two domains.

Specifically, we find a total 960,885 (12%) mobile cookies that travel between only two ASes, and that appear at the ASes consecutively within a short interval (i.e., 10 minutes). We call such cookies VPN-style cookies. Table 7 lists the top five AS pairs with the highest number of these cookies, which include institutional and corporate networks, e.g., City University of New York, NTT, and KDDI Corporation. VPN-style cookies com-

prise around 60% of all traveling cookies between a corporate network and a DSL broadband service provider. This observation indicates that VPNs can be a major explanation for host mobility.

AS pair	# Cookies	Affiliations
AS 6389, 35985	13249	BellSouth, One Ring Net. (US)
AS 702, 2856	8977	Verizon (US), BTnet UK Reg. Net. (GB)
AS 7018, 31822	7878	AT&T, City Univ. N.Y. (US)
AS 174, 701	6630	Cogent, MCI Comm.(US)
AS 4713, 4716	5770	NTT Comm., KDDI Corp. (JP)

Table 7. Top AS pairs associated with VPN cookies.

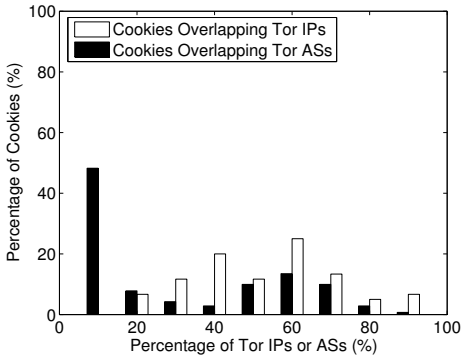


Figure 9. The percentage of Tor IPs or ASes on the path of wandering cookies.

5.2.2 The Use of Anonymous Routing

Examining the tail of the distribution in Figure 8, we also find a small fraction (0.02%) of cookies that migrate across more than 10 different domains. Stopping in each AS only for short durations, they do not return to a previously visited domain. Focusing on this behavior, we identify 309 cookies that travel across more than 10 ASes, and where the time between consecutive "jumps" to different ASes is less than 10 minutes (which is the default time to use a Tor circuit for new application connections). Compared to the AS peering relationship in Section 5.1, there does not appear to be any clear delineation of geographical regions.

The top ASes in this case are dominated by cable networks, with the previously top cellular networks disappearing completely from the list. Some university networks ranked significantly higher than before (AS 111, associated with Boston University, is on the path of 9% of these cookies). One explanation for the behavior of these cookies is the use of anonymous routing systems, such as Tor [37]. For a user that routes her traffic in this manner, if her traffic exits from different nodes in the mixing network, the same cookie may appear at different domains.

We obtained a list of active Tor nodes [8], including, for each node, its IP address, country, ISP, and whether it is an exit node. Among the 309 wandering cookies, 60 of them traverse through at least one Tor node, and 142 of them traverse through at least one AS that is also shared by a Tor node. We also examine ASes since some Tor nodes may already be assigned different IPs at the time of our lookup. Figure 9 plots the distribution of the percentage of Tor IP addresses or ASes that a cookie traverses. All of the cookies spend at least 12% of their time at a Tor IP address, with the maximum being 83%.

Using the Tor network hides the network origin of a user, addressing one aspect of online anonymity. However, the use of cookies may still reveal user activity patterns and potentially user network origins, e.g., if a user does not clear cookies prior to using the Tor network. To mitigate such privacy threats, users can install Torbutton [7] to manage their identifying information, for example.

5.3 Detecting Cookie-Forwarding Attacks

Based on the host mobility patterns derived from our analysis, we launch an investigation into abnormal user activities that include 28,208 unique user accounts, provided by the Hotmail web-mail service. These events are sampled over a 24-hour window in November 2010. In each event, a user submitted requests (e.g., checking new emails, listing contacts) from an IP range that was different from the one she used to log into her Hotmail account. One would imagine that this behavior can be attributed to the use of cellular networks, VPNs, or proxies. Surprisingly, we find many users exhibit quite different traveling patterns than those we learned in Sections 5.1 and 5.2.

5.3.1 Detection Methodology

We find two distinct patterns in these events that differ from those of general mobile hosts:

- One-third of the ASes associated with these events are exclusively sinks or sources. This is in contrast to normal host mobility, where the direction

of travel is largely bi-directional. Table 8 lists the dominant sink ASes.

- Among the AS pairs with the largest number of these abnormal events, seven out of the top ten do not appear at all among those associated with normal hosts. These AS pairs are listed in Table 9.

Sink AS	# Cookies	Location
AS 34285	308	Seville, Spain
AS 40430	201	Miami, FL, USA
AS 14141	192	Atlanta, GA, USA
AS 19318	189	Jersey City, NJ, USA
AS 19194	174	Unknown (Satellite provider)

Table 8. Top ASes that are exclusively sinks in the abnormal events.

AS Pair	# Cookies	Affiliations
AS 766, 34285	308	RedIRIS AS (EU), SANDETEL (ES)
AS 30736, 25761	235	Easyspeedy Net. (DK), Staminus Comm. (US)
AS 30736, 40430	201	Colo4jax (US)
AS 30736, 1421	198	WANSecurity (US)
AS 30736, 14141	192	WireSix (US)
AS 30736, 29761	192	OC3 Net. & Web Solutions (US)
AS 30736, 19318	188	New Jersey Intl. Internet Exchange (US)

Table 9. Top AS pairs related to abnormal events.

Combining these two observations, we find that the dominant sinks in Table 8 significantly overlap with the sink ASes in Table 9. They share the common source AS 30736, located in Denmark. Upon examination, we find that there is a single IP address generating login events for a large number of users, who then submit subsequent requests from multiple ASes in the U.S., violating the geo-locality travel pattern observed in Figure 7 as well.

We find that the user login IDs associated with this particular source IP address contain more suspicious patterns. In particular, they are groups of bot-user accounts all registered on the same day in November 2010, with the same user age, location information (country, state), and scripted naming patterns. Among the top five dominantly sink ASes, four of them are used by these bot groups to submit requests.

Sink AS	# IP	# Req.	# Acct.	Location
AS 14141	12	262	192	Atlanta, GA
AS 19194	10	225	174	Unknown
AS 19318	11	242	189	Jersey City, NJ
AS 40430	12	269	201	Miami, FL
AS 25761	14	324	235	Fullerton, CA
AS 1421	10	265	198	Bordentown, NJ
AS 29761	10	244	192	Los Angeles, CA
AS 30058	10	261	180	Woodstock, IL
AS 18779	10	246	180	San Francisco, CA

Table 10. Statistics for detected bot-user groups.

By examining all the sink ASes with source AS 30736 in these events, we find a total of 9 bot-user groups, corresponding to 9 sink ASes geographically distributed over the U.S. The activities between some of these ASes are subtle, and would not have been detected without leveraging the normal host mobility patterns described in Section 5.1.

5.3.2 Cookie-Forwarding Bot Users

Table 10 lists the statistics for the 9 detected bot-user groups. Each of these groups includes around 190 users. A different /24 subnet is associated with each user group that submit requests without explicit login activities from the same subnet. For each /24, the sink IP rotates among 10 to 14 addresses.

From a more recent user login dataset collected by Hotmail in January 2011, we find over 75,000 email accounts associated with the suspicious source IP address in Denmark, all exhibiting similar patterns to the 9 groups we discovered. Manual investigation by Hotmail shows that these accounts were used by attackers for the purpose of receiving and testing spam. After these accounts are logged into from one machine (i.e., one IP address), their cookies are forwarded to multiple locations so that further requests can be submitted in a distributed fashion during the validity period of the cookies, which is 24 hours in our case.

There are at least two possible explanations for such malicious cookie-forwarding activities. First, some web-mail providers identify an account as suspicious if it performs logins from multiple geographic locations within a short time interval. By forwarding cookies to other locations through a private communication channel, attackers can successfully offload the requests to distributed hosts without them performing explicit user logins, hence reducing the likelihood of detection. Sec-

ond, as a preparation step in launching session-hijacking attacks on real user accounts (e.g., [6]), attackers may be testing the effectiveness of forwarding cookies via stealthy communication channels.

Although the user accounts we identified were all newly created, it is possible that attackers can employ hijacked cookies stolen from actual users and forward them to botnet hosts in the future. Understanding normal host mobility patterns can help detect such stealthy attacks.

6 Related Work

Many efforts on tracking hosts focus on identifying specific hardware characteristics, such as radio frequency [23, 34, 18] or driver [21]. Identifiers such as network names or the IP addresses of frequently accessed services also enable host fingerprinting [32]. However, these approaches require the observer to be in close physical proximity to the target host.

Remote host fingerprinting can leverage packet-level information to identify the differences in software systems [2, 4, 5] or hardware devices [28]. Other works on tracking web clients require probing hosts' system configurations [20] or the installation order of browser plug-ins [31]. Persistent browser cookies [3, 36] have also been proposed; these systems store several copies of a cookie in different locations and formats, so that they cannot be removed by standard methods.

Compared with these efforts, our work focuses on studying the effectiveness and implications of tracking hosts using existing identifiers, without requiring new information or probes. Although the issue of privacy leakage has been repeatedly discussed, e.g., personally identifiable information in online social networks [29, 30], there has been limited study using large-scale datasets. Our work uses month-long datasets from a large search engine and a popular email provider to quantify the amount of host-identifying information revealed by a variety of common identifiers. To the best of our knowledge, we are also the first to demonstrate applications of host tracking to analyze cookie churn in web services and to detect suspicious cookie-forwarding activities.

Apart from its privacy implications, understanding cookie churn is an important topic for estimating web user population and personalization. Previous studies mostly rely on user surveys or active user participation (e.g., by installing a software on user machines) [12, 11, 16, 14]. Their findings show that 30% to 40% of users clear cookies monthly. A separate study by Yahoo! [13] find that 40% and 60% of users have empty browser caches, so they probably have cleared cookies as well. While our results are consistent with previous findings, the approach we take requires neither user cooperation nor special content setup.

Host mobility studies have been performed in the context of wireless [17, 27, 22, 25], ad hoc [24, 26], and cellular networks [19] to obtain more accurate device moving models or to predict user locations. Simler et al. [35] studied user mobility in terms of session characteristics based on login events to a university email server in order to generate synthetic traces. Recent work [33] proposed a technique for classifying IP addresses into home and travel categories to study host travel and relocation patterns in the U.S. By studying cross-domain cookies, our work focuses on normal host mobility patterns that enable us to observe uncommon phenomena and detect malicious activities.

7 Discussion and Conclusion

In this paper, we perform a large-scale exploration of common identifiers and quantify the amount of host-identifying information that they reveal. Using month-long datasets from Hotmail and Bing, we show that common identifiers can help track hosts with high precision and recall.

Our study also informs service providers of the potential information leakage when they anonymize datasets (e.g., replacing IP addresses with IP prefixes) and release data to third-party collaborators or to the public. For example, we show that hashes of browser information (i.e., the anonymized UA strings) alone can be quite revealing when examined in one network domain. Furthermore, coarse-grained IP prefixes achieve similar host-tracking accuracy to that of precise IP address information when they are combined with hashed UA strings.

Our analysis suggests that users who do not wish to be tracked should do much more than clear cookies. Uncommon behaviors such as clearing cookies for each request may instead distinguish a host from others who do not do so. Users should take notice of their user-agent strings (e.g., modify the default setting [10]), consider the use of proxies, and possibly resort to sophisticated techniques such as anonymous routing [37]. In some cases, several of these techniques should be combined to be effective, e.g., clearing cookies in addition to the use of proxies or Tor.

Finally, despite its privacy implications, we demonstrate the security benefit of host-tracking. Given the

growing concerns over account hijacking and session hijacking, we expect host fingerprinting and tracking techniques can help defend against such attacks in the future.

Acknowledgments

We are grateful to Hotmail, Bing, and Windows Update for providing us with data access that makes this study possible. We thank Zijian Zheng for his guidance and insight on cookie-churn analysis. We thank Keiji Oenoki and Hersh Dangayach for providing us with data related with cookie-forwarding attacks and for the help in the subsequent investigation. We thank the reviewers, and in particular Paul Syverson, for their suggestions of improvements to this paper.

References

[1] CookieCooker. http://www.cookiecooker.de/.

[2] Nmap free security scanner. http://nmap.org.

[3] Project details for evercookie. http://samy.pl/evercookie/.

[4] Project details for p0f. http://lcamtuf.coredump.cx/p0f.shtml.

[5] Project details for xprobe. http://sourceforge.net/projects/xprobe/.

[6] Secure your PC and website from Firesheep session hijacking. http://www.pcworld.com/businesscenter/article/210028/secure_your_pc_and_website_from_firesheep_session_hijacking.html.

[7] Tor Project: Torbutton. https://www.torproject.org/torbutton/.

[8] Tor Proxy List. http://proxy.org/tor.shtml.

[9] U. Oregon Route Views Project. http://www.routeviews.org/.

[10] User-agent switcher. https://addons.mozilla.org/en-US/firefox/addon/59/?id=59.

[11] 40% of consumers zap cookies weekly. http://www.marketingsherpa.com/!newsletters/bestofweekly-4-22-04.htm#topic1, 2004.

[12] Measuring unique visitors: Addressing the dramatic decline in accuracy of cookie-based measurement. White paper, Jupiter Research, 2005.

[13] Yahoo! user interface blog: Performance research, part 2: Browser cache usage exposed! http://yuiblog.com/blog/2007/01/04/performance-research-part-2/, 2007.

[14] Cookie corrected audience data. White paper, Quantcast Corp., 2008.

[15] Protecting consumer privacy in an era of rapid change. Federal Trade Commission Staff Report, 2010.

[16] M. Abraham, C. Meierhoefer, and A. Lipsman. The impact of cookie deletion on the accuracy of site-server and ad-server metrics: an empirical comScore study. White paper, comScore, Inc., 2007.

[17] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *Intl. Conf. Mobile Systems, Applications, Services*, 2003.

[18] V. Brik, S. Banerjee, M. Gruteser, and S. Oh. Wireless device identification with radiometric signatures. In *Intl. Conf. Mobile Computing and Networking*, 2006.

[19] I. Constandache, S. Gaonkar, M. Sayler, R. Choudhury, and L. Cox. Energy-efficient localization via personal mobility profiling. In *Intl. Conf. Mobile Computing, Applications, and Services*, 2009.

[20] P. Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies Symp.*, 2010.

[21] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. V. Randwyk, and D. Sicker. Passive data link layer 802.11 wireless device driver fingerprinting. In *USENIX Security Symp.*, 2006.

[22] J. Ghosh, M. Beal, H. Ngo, and C. Qiao. On profiling mobility and predicting locations of wireless users. In *Intl. Workshop on Multi-hop ad hoc networks*, 2006.

[23] J. Hall, M. Barbeau, and E. Kranakis. Detection of transient in radio frequency fingerprinting using signal phase. In *Intl. Conf. Wireless and Optical Communications*, 2003.

[24] X. Hong, M. Gerla, G. Pei, and C. Chiang. A group mobility model for ad hoc wireless networks. In *ACM Intl. Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, 1999.

[25] N. Husted and S. Myers. Mobile location tracking in metro areas: Malnets and others. In *ACM Conf. Computer and Communication Security*, 2010.

[26] A. Jardosh, E. Belding-Royer, K. Almeroth, and S. Suri. Towards realistic mobility models for mobile ad hoc networks. In *Intl. Conf. Mobile Computing and Networking*, 2003.

[27] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *IEEE Infocom*, 2006.

[28] T. Kohno, A. Broido, and K. Claffy. Remote physical device fingerprinting. In *IEEE Symp. Security and Privacy*, 2005.

[29] B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *ACM Workshop on Online Social Networks*, 2008.

[30] B. Krishnamurthy and C. E. Wills. Privacy leakage in mobile online social networks. In *USENIX Conf. Online Social Networks*, 2010.

[31] J. R. Mayer. "Any person... a pamphleteer": Internet anonymity in the age of Web 2.0. Senior Thesis, Stanford University, 2009.

[32] J. Pang, B. Greenstein, R. Gummadi, S. Seshan, and D. Wetherall. 802.11 user fingerprinting. In *Intl. Conf. Mobile Computing and Networking*, 2007.

[33] A. Pitsillidis, Y. Xie, F. Yu, M. Abadi, G. Voelker, and S. Savage. How to tell an airport from a home: Techniques and applications. In *ACM Workshop on Hot Topics in Networks*, 2010.

[34] K. Rasmussen and S. Capkun. Implications of radio fingerprinting on the security of sensor networks. In *Intl. Conf. Security and Privacy in Comm. Networks*, 2007.

[35] K. Simler, S. Czerwinski, and A. Joseph. Analysis ofwide area user mobility patterns. In *IEEE Workshop on Mobile Computing Systems and Applications*, 2004.

[36] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. Hoofnagle. Flash cookies and privacy. *SSRN preprint*, 2009.

[37] P. Syversion, D. Goldschlag, and M. Reed. Anonymous connections and onion routing. In *IEEE Symp. Security and Privacy*, 1997.

[38] Y. Xie, F. Yu, and M. Abadi. De-anonymizing the internet using unreliable IDs. In *ACM SIGCOMM*, 2009.

Appendix

A Tracking Stable Hosts

In the presence of NATs, proxies, and dynamic IP addresses, the mapping between a host and an IP address can be extremely volatile. Service providers that are interested in fingerprinting *stable* hosts may trade coverage for accuracy. We show that the binding window length can serve as a confidence measure for this purpose.

Intuitively, stable and active hosts should have longer binding windows that make them easier to track than hosts that appear infrequently or that change IP addresses often. Indeed, using UA+IP as an example, Figure 10(a) shows the increase in precision and recall with longer binding windows.

However, as we impose increasingly strict requirements on the binding window length, the percentage of fingerprints remaining decreases roughly proportionally, as shown in Figure 10(b). Half of the fingerprints have binding windows no longer than one week. We can thus explore a tradeoff between accuracy and coverage of

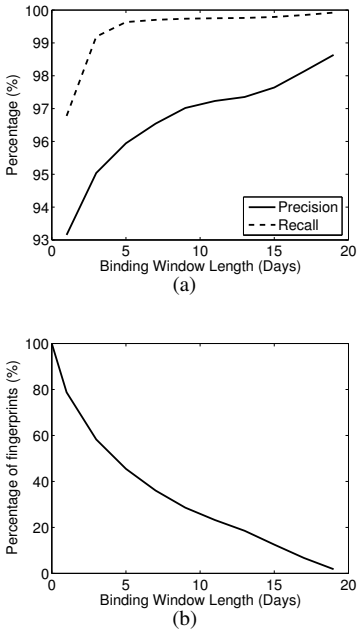


Figure 10. Binding length and accuracy tradeoff.

tracking hosts using the binding window length as an adjustable parameter. In particular, Figure 10 suggests that using a binding window length of five days in practice can achieve both high precision and recall without losing significant coverage.

B Non-returning Users

For those 101,427 "one-time" non-returning users that were observed only on the first day of the Search dataset we are interested in whether this is because they stopped using the service or because they cannot be tracked. We expect users who leave the service to be less engaged than returning users. To test this hypothesis, we examine the average number of queries submitted by each CID and the percentage of CIDs that have clicked on the query results. We compare these two statistics between the set of returning users and the set of non-returning users. We consider only churned *new* CIDs in this comparison. For example, if a returning user has queries associated first with CID_1 and later with CID_2, we consider the subset of queries that correspond to CID_1 only, as they represent first-time user experience.

Table 11 shows that returning users indeed appear to

	Non-returning users	Returning users
Average number of queries per CID	4.7	7.0
Percentage of CIDs with clicks	60.73%	77.85%

Table 11. The query and click behaviors of returning and non-returning users from the first day of the log.

be more engaged in the service, generating more queries on average and are also more likely to make clicks. Overall, 77.85% of the churned new CIDs that belong to returning users have clicks, while only 60.73% of the churned new CIDs from non-returning users have clicks.

We further examine, for each CID, the percentage of search queries that resulted in clicks. For CIDs that belong to returning users, Figure 11 shows a larger percentage of queries have clicks than CIDs that belong to non-returning users. Half of the CIDs associated with returning users have clicks on 80% of their queries, while half of those associated with non-returning users have them on less than 50%.

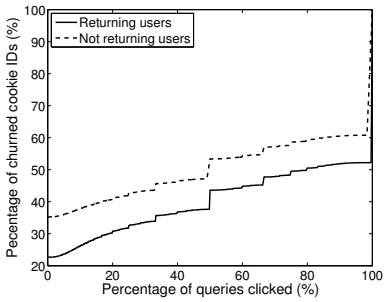


Figure 11. Cumulative distribution of the fraction of queries per CID that resulted in clicks.

Another question of interest is whether users stop using the service because they are less active and have infrequent online activities. To quantify the degree of activity of the non-returning users, we measure the time interval and the number of login events between the last Bing search query and the last Hotmail login event that fall within the host's binding windows, shown in Figure 12. We find that though users in our data may have left the search service, many of them have continued online activities. More than 80% of these users are active even after 25 days (Figure 12(a)), and around 60% of them logged in more than 40 times (Figure 12(b)).

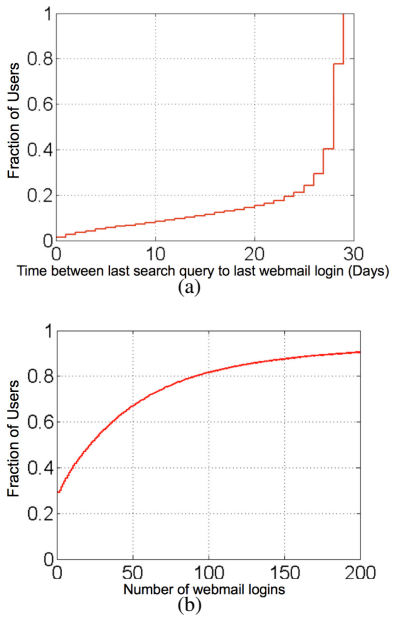


Figure 12. (a) The time between the last Bing search query and last Hotmail login. (b) The number of Hotmail logins after the last Bing search query.