Alternative to third-party cookies: Investigating persistent PII leakage-based web tracking

Ha Dao
The Graduate University for Advanced Studies (Sokendai)
Tokyo, Japan

Kensuke Fukuda
NII / Sokendai
Tokyo, Japan

ABSTRACT

Many popular websites give users the ability to sign up for their services, which requires personally identifiable information (PII). However, these websites embed third-party tracking and advertising resources, and as a consequence, the authentication flow can intentionally or unintentionally leak PII to these services. Since a user can be identified with PII, trackers can use it for tracking purposes, leading to further privacy leaks when cross-site, cross-browser, and cross-device tracking occur.

In this paper, we document a persistent web tracking mechanism that relies on manipulating PII leakage after a user completes the sign-up and sign-in flow (authentication flows) on first-party sites. To the best of our knowledge, this is the first in-depth analysis of leaked PII in the authentication flows. By investigating the authentication flows for 307 popular shopping sites from the Tranco top 10,000 sites, we first discover that 42.3% of sites leak the PII to third-party services. Then, we present a previously unknown persistent web tracking technique based on PII leakage that enables tracking providers to generate and store a unique persistent identifier for a user with his/her browsing history on their tracking servers. By analyzing 130 first-party senders along with 100 third-party receiver domains, we show that PII leakage is a potentially important vector for online tracking for at least 20 providers. In addition, we check the privacy policy of the 130 first-party senders and observe that they are not clear about PII exchange with third parties. Finally, to provide a wider picture of current in-browser privacy protection techniques, we evaluate the effect of browsers and well-known blocklists against PII leakage. We point out that browsers are unable to deal with PII leakage except for Brave with its privacy-improving features, whereas blocklists reduce the number of leaked PII resources but do not fix this problem in general.

CCS CONCEPTS

• **Security and privacy → Privacy protections**; • **Networks → Network measurement**.

KEYWORDS

PII, PII Leakage, Persistent Web Tracking, Privacy, Transparency, Web Measurements

CoNEXT '21, December 7–10, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-9098-9/21/12…$15.00
https://doi.org/10.1145/3485983.3494860

ACM Reference Format:
Ha Dao and Kensuke Fukuda. 2021. Alternative to third-party cookies: Investigating persistent PII leakage-based web tracking. In *The 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '21), December 7–10, 2021, Virtual Event, Germany.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3485983.3494860

1 INTRODUCTION

The advertisement and tracking ecosystem has substantially grown in the last 10 years, building on standard and more advanced web tracking technologies [18]. Tracking providers usually try to attach a virtual identity to a unique user by collecting and correlating the user's browsing behavior.

Third-party cookies refer to cookies that are created by domains differently from a website domain directly visited by the user, and they are mainly used for online tracking purposes. They are accessible on any website that loads third-party domain resources. To protect user privacy, there have been various proposes to mitigate the usage of third-party cookies [11, 13]. This has resulted in third-party web tracking getting more sophisticated, making it complicated to overcome the countermeasures. One potential tracking mechanism is the usage of personally identifiable information (PII) leakage, which refers to when a user's PII (i.e., email address, name, address) leaks to third-party domains (different from visited domains), to track user activities. For instance, a first-party website, *site.com*, embeds a third-party tracking provider, *tracker.net*. After a user completes the sign-in flow on *site.com* by inputting his/her PII, a tracking script reads this PII and sends it to the tracking provider server, *tracker.net*. Because PII is a unique identifier, it allows the *tracker.net* to match the user's browsing history across sites, browsers, and devices without using third-party cookies.

This work not only focuses on PII leakage but also presents a persistent web tracking technique based on this data transfer. We first collect data on sign-up and sign-in flows completed like a typical user (§3). We then detect PII leakage to third-party domains (including CNAME cloaking) through four distinct methods even when that information is obfuscated (§4.1). Our results show that 42.3% of popular shopping-related first-party services leak PII to third-party receivers, the most common being Facebook (§4.2). Furthermore, we identify a tracking mechanism based on PII leakage that allows tracking providers to persistently identify users through cross-site, cross-browser, and cross-device tracking (§5.1). We then analyze the presence of PII leakage-based tracking and demonstrate that 20 tracking providers leverage PII so as to be able to tenaciously track user activities (§5.2). In addition, we look at the privacy policies of those 130 first parties to evaluate what disclosures are made to users about sharing PII to third parties (§6). Finally, we evaluate browsers and blocklists against leaked PII resources and confirm

that there is still room for PII leakage countermeasures to improve (§7).

2 BACKGROUND AND RELATED WORK

2.1 Third-party web tracking

Third-party web tracking refers to the practice of an entity, other than the domain directly visited by the user, that identifies and collects information about web users. It is useful for a variety of purposes, such as online behavioral advertising that targets users with ads based on their profiles or interests. It considers to show a greater privacy risk to the user.

Third-party web tracking is complicated in that it relies on a wide variety of web tracking technologies, ranging from stateful tracking mechanisms that recognize users by retrieving information stored on users' machines, and stateless tracking mechanisms that recognize users without storing any information. Undoubtedly, many tracking techniques have been developed to maximize the benefits of tracing user browsing behavior. They have been studied considerably in order to get a better understanding of the risks involved with these techniques. Some approaches have detected the main mechanisms behind specific web tracking techniques [16, 23, 25, 27], analyzed cookie synchronization that bypasses same-origin policies on the web [17, 33, 39], and explored the unique identifier stored in a cookie or embedded as a parameter in a URL [26]. These techniques allow trackers to compile unique browsing histories, but they do not link PII to track user activities.

2.2 PII leakage

PII is a piece of information that can be used to distinguish or trace an individual's identity either alone or when combined with other information that is associable to a specific individual [31].

PII leakage occurs when PII information leaks from a first party that gives it to a third-party. Recent work includes the detection of PII leakage to third parties in smartphone apps [35], data leakage due to browser extensions [37], and data leakage due to several web forms, including PII leakage from contact forms [36], mailing list subscription forms [24], registration forms [20], and authentication flows [30, 32]. Obviously, PII leakage on the web has been studied considerably. However, these studies only concentrated on the PII leakage to third parties.

Here, we roughly follow the same methodology as in Refs. [20, 30, 32] to detect PII leakage to third parties, but we consider more detection methods (four and an additional combination with CNAME cloaking [21, 22]) even when that information is obfuscated. We also present a tracking technique that relies on manipulating PII leakage and show that tracking providers can use this transfer of data for cross-site, cross-browser, and even cross-device user tracking. Furthermore, we explore the transparency of first-party privacy policies and evaluate modern browsers' protections against PII leakage that have not been investigated before.

3 DATA COLLECTION

3.1 Persona making

To know in advance the strings that constitute PII, we first create an account as a persona to fill in the sign-up form. This account contains the following information: username, name, phone, email address, date of birth, gender, job title, and postal address. We consider any information input by the user to be PII. We take inspiration from Refs. [19, 24, 37] to detect not only the plaintext form but also the encoding/hashing form of leaked PII. We pre-compute a candidate set of tokens by applying all supported encodings, hashes, and checksums for each PII. Note that the encoding or hashing could be applied multiple times. Here we encode/hash each PII at most three time according to the previous work [24] to obtain a set of PII strings. We thus consider two different inputs that are used to build the PII-related strings that are detected in the next steps:

(1) Any plaintext PII persona information.
(2) A set of encodings, hashes, and their combination based on PII. The full list of supported hashes and encodings is given in Appendix.

3.2 Data acquisition

We first select websites that would be most appropriate for our work. We use the Tranco popular site list, which improves upon the shortcomings of the existing popular site lists: being unstable, having unreachable domain presence, and containing domains that are easily altered by an adversary [34]. From that, we obtain 404 shopping sites from the Tranco top 10,000 sites [1] since we found that 95.0% of sites in this category contains authentication flows.

Next, to avoid the bias introduced by bots [28, 29, 40, 41] and capture data precisely even when security mechanisms (such as bot detection and multi-factor authentication) are in place, we decide to collect our dataset by using a manual approach like that of a natural user as opposed to automated control. In detail, we use a vanilla setting for Firefox 88 (Enhanced Tracking Protection was turned off) from an IP address located in Japan in May 2021. We always accept the default cookie settings for pop-ups, but refuse all other types of solicitations, such as geolocation and notifications. We then browse the selected websites, fill all input fields in with the created persona information, press the submit button. After that, we open another browser and got the email confirmation link/code to activate the account successfully (if necessary), sign in with the created account, and reload the site with a logged account. We also click a link (usually to a specific product) from the same domain to observe the event using leakage behavior on subpages compared with the homepage. During these processes, we collect HTTP requests (URLs, headers, and payload body - if any), HTTP responses (URLs and headers), and cookies (both those set/sent and a copy of stored browser cookies). Overall, after removing sites that were unreachable (22 sites), did not contain authentication flows (19 sites), and could not be signed up to due to the website's policy (56 sites[2]), we obtained successful authentication flows on 307 shopping sites, includes 68 sites that require email confirmation and 43 sites that use bot detection. These sites can not be crawled automatically even when effort has been made to match all complicated fields with the right information [20].

[1] We use the FortiGuard Web Filtering [5] dataset from May 2021 for the website category classification.
[2] Forty-seven sites required phone verification, 6 sites required special pieces of personal information such as identity documents, and 3 sites blocked the creation of new accounts for global customers.

Regarding the ethical aspects of our research design and process, although we completed the authentication flows on target websites with a simulated persona, we believe that our procedure did not have any impact on their website operations.

4 PII LEAKAGE DETECTION AND ANALYSIS

4.1 PII leakage detection methods

First of all, we separate the generic Top-Level Domain (gTLD) and country-code top-level domain (ccTLD) from the visited websites for all HTTP requests using the Public Suffix List [7] to separate first-party and *third-party* resources. In addition, to discover hidden third parties using *CNAME cloaking*[3], we check CNAME records for each subdomain of the visited sites. We then match these CNAME answer sets with the CNAME cloaking blocklist by [12, 14, 21]. Finally, we combine these *CNAME cloaking* and *third-party* resources into the third parties and identify the following four methods that allow a first-party to send PII to a third-party.

Via referer header: A common method for PII leakage is based on the poorly coded sign-up and sign-in forms that expose sensitive information in URLs. More precisely, these forms submit user information to a first-party web server using the GET method. For instance, if *site.com* has a sign-up form that uses the GET method to submit a user's email address, the user's browser creates a request with the form parameters containing the email address http://site.com/signup?&email=*foo@mydom.com*. In the case that *site.com* includes an external third-party, *tracker.net*, in its authentication flow, a request is sent to this third party. Because the *referer header* is sent with any request for a remote resource, this request will link to http://site.com/signup?&email=*foo@mydom.com* as the *referer header*. The PII provided by the user on the sign-up/sign-in form is thus sent to third-party domains (see Figure 1.a). We consider this method to be a form of unintentional leakage as in [36].

Via request URI: A request is sent to a server can be sent together with a set of parameters attached to the end of a URI, which structures additional information like PII for a given URI. For instance, the third-party *tracker.com* requests an email address and captures *foo@mydom.com* in the authentication flows. This URI is sent to *tracker.com* along with this information in the request URI (see Figure 1.b).

Via cookie: PII also can be leaked when third-party resources use PII to create cookie values. For instance, when *tracker.com* creates a cookie *id=foo@mydom.com*, this PII containing resource is sent to the third-party domain *tracker.com* (see Figure 1.c).

Via payload body: The last method involves PII being added to the payload body of an HTTP request and sent to a third-party domain. For instance, *tracker.com* will receive a request body from a first party with a payload that contains the parameter email address *foo@mydom.com* (see Figure 1.d).

Finally, we look for specific substrings as plaintext, encoded, hashed values and their combinations (§3.1), inside all third-party-related requests to detect any leaked PII.

[3]CNAME cloaking is a technique that circumvents the third-party targeting privacy protections by using the Canonical Name Record or Alias (CNAME) record in Domain Name System (DNS) to hide usual tracking domains.

Table 1: Breakdown of PII leakage to third parties. Percentages are given out of total of 130 first-party senders and 100 third-party leak receivers.

(a) By method.

Method	#of Senders	# of Receivers
Referer header	3/2.3%	7/7.0%
URI	118/90.8%	78/78.0%
Payload body	43/33.1%	17/17.0%
Cookie	5/3.8%	1/1.0%
Combined	27/20.8%	8/8.0%

(b) By encoding/hashing

Encoding/hashing	#of Senders	# of Receivers
Plaintext	42/32.3%	56/56.0%
BASE64	19/14.6%	20/20.0%
MD5	35/26.9%	24/24.0%
SHA1	9/6.9%	6/6.0%
SHA256	91/70.0%	30/30.0%
SHA256 of MD5	2/1.5%	1/1.0%
Combined	21/16.2%	14/14.0%

(c) By PII type.

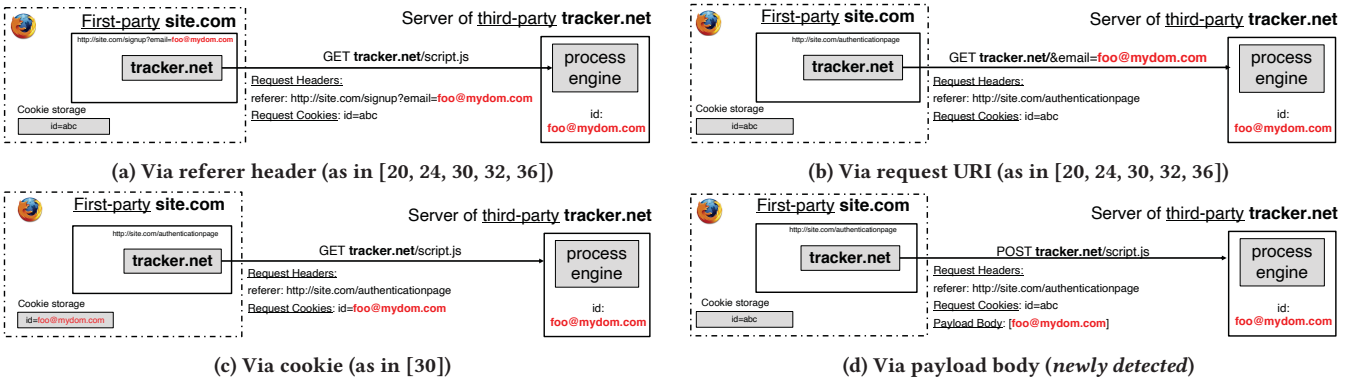
PII type	#of Senders	# of Receivers
Email	116/89.2%	94/94.0%
Username	1/0.8%	1/1.0%
Email,username	3/2.3%	6/6.0%
Email,name	29/22.3%	12/12.0%

4.2 PII leakage analysis

We first focus on the PII leakage to third parties observed in the authentication flow of top shopping sites.

Overall, we detect the 130 first parties that leak PII among the 307 original authentication flows (through 1,522 requests that contain leaked PII). These first parties' PII leaks are sent to 100 third parties. Among these first parties, 46.15% send PII to at least three third-party domains, and the average number of third parties receiving PII was 2.97 per first-party site. We also observe 16 third-party receivers as the maximum from *loccitane.com*). We provide a breakdown of the top 15 third-party domains that received PII from the 130 first-party senders in the Tranco top shopping sites in Figure 2. We find that the domain *facebook.com* gets PII from 60.0% of the first-party senders. Regarding domain usage, while *Facebook*, *Criteo*, *Pinterest*, and *Snapchat* use a single domain to receive PII, *Google* and *Adobe* use multiple domains to receive PII from first-party senders.

4.2.1 PII leakage method. We break down the PII leakage to third parties according to the four methods in Table 1a. We observe seven accidental leakage cases in which three sites have a sign-up form that uses the GET method to submit a user's information data, and as a result, they leak PII to third parties *via referer header* (see Figure 1.a). Also, there are five cases in which first-party senders use PII to create a cookie value that is automatically sent to the tracking provider Adobe via CNAME cloaking (see Figure 1.d). In



(a) Via referer header (as in [20, 24, 30, 32, 36])

(b) Via request URI (as in [20, 24, 30, 32, 36])

(c) Via cookie (as in [30])

(d) Via payload body (*newly detected*)

Figure 1: Four PII methods of leakage to third parties. This includes additional use of CNAME cloaking that was not considered before. PII is displayed in red.

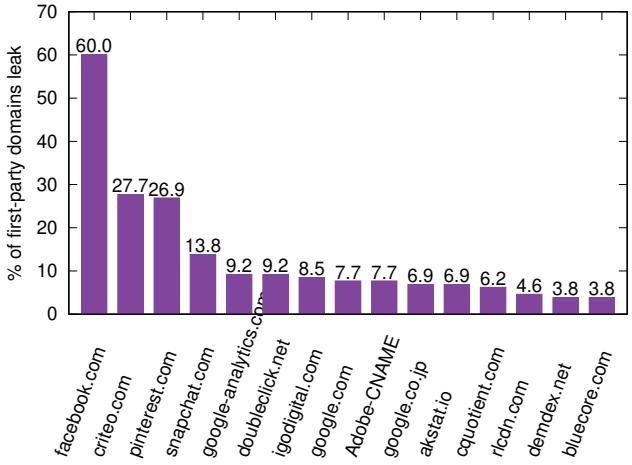


Figure 2: Top 15 third-party receiver domains involved in PII leakage included in 130 first-party senders from Tranco top shopping sites.

the meantime, we discover that 126 first-party senders intentionally leak user' data to third-party receivers. In addition, we find that 17 third-party domains receive PII via the payload body from 43 first-party senders that were not detected by observing the URL strings in previous work [20, 24, 30, 32]. Furthermore, there are 27 first-party senders that leak PII to 8 third parties by using combined methods (i.e., , via request URI and cookie, via request URI and payload body). We hypothesize that the goal of this combination is to maximize the possibility of obtaining PII from first-party sites in the context of third-party cookie restrictions and ad blocker utilization for privacy.

4.2.2 PII leakage by encoding/hashing. Next, we break down PII leakage by encoding/hashing to third parties in Table 1b. We first check whether any PII is sent as plaintext directly to the third-party domains. We find that 32.3% of the first-party senders directly send PII to 56% third-party receivers without any encoding or hashing function. In addition, 83.1% of first parties send PII as a hashed string to 49.0% of third-party domains. The most common one was SHA256 (70.0%). We also find two sites that use SHA56 of the MD5 hash string to send PII to the third-party domain *criteo.com*. In addition, we observed 21 first-party senders that leaked PII to third-party receivers by combining multiple encoding/hashing forms (i.e., plaintext and SHA256; BASE64, SHA1, and SHA256). PII leakage is much more complicated than previously reported [20, 24, 30, 32].

4.2.3 PII leakage type. Finally, we analyze the nature of the leaked PII in Table 1c. The most common values that leaked were: email address (88.5%), username (0.8%), both email and username (1.5%), and email and name (22.3%). These PIIs are sufficient for correlating any user along with his or her browsing activity. Considering that third parties collect large amounts of data for a user, these types of information are valuable for the tracking ecosystem.

When we conducted our experiments, the test email account started to receive email notifications from the visited sites. They mainly offered discounts or introduced new products. In total, we received 2,172 emails in the inbox and 141 emails in the spam folder. Notably, we have not yet received any emails belonging to any third-party domains involved in the PII leakage that were included in 130 first-party senders. Apparently, We see that leaked PII is not used for third-party email marketing, but it could be used as an identifier to track user activity on the Internet.

5 PERSISTENT WEB TRACKING BASED ON PII LEAKAGE

Here, we discuss the case where some third parties use PII to generate and store a unique persistent identifier of the user along with his/her browsing history for tracking purposes.

5.1 PII leakage-based tracking presumption

We first presume that PII can be a potential vector for persistent web tracking. A typical example of an HTTP request for persistent web tracking based on PII leakage is shown in Figure 3. The tracking provider *tracker.net* uses a PII identifier parameter (*trackid*) to capture information from first-party senders. After a user proceeds the sign-in flows on this site by submitting his/her information, a tracking script reads and pushes the user's email address

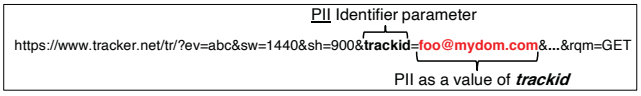


Figure 3: Example of HTTP traffic for persistent web tracking based on PII leakage. PII identifier parameter is different for each tracking provider, and PII value can be in hashing/encoding form.

Table 2: Breakdown of third-party receiver domains for persistent tracking based on PII leakage on popular shopping sites. All hashes are of full email address.

#	Receivers	# of Sender	Method	Encoding form	*trackid* Parameter
1	facebook.com	72	URI /Payload	SHA256	udff[em]/ud[em]
		2	URI	MD5	ud[em]
2	criteo.com	26	URI	MD5	p0/p1
		4	URI	SHA256	p0
		5	URI	plaintext	p0/p1
		2	URI	SHA256ofMD5	p0/p1
3	pinterest.com	25	URI	SHA256	pd
		8	URI	MD5	pd
4	snapchat.com	18	URI/Payload	SHA256	u_hem
		2	Payload	MD5	u_hem
5	cquotient.com	7	URI	SHA256	emailId
6	bluecore.com	5	Payload	BASE64	data
7	klaviyo.com	4	URI	BASE64	data
8	oracleinfinity.io	4	URI	SHA256	email_hash/ora*
9	rlcdn.com	4	URI	SHA1	s
10	adobe_cname	3	URI	SHA256	v*
11	castle.io	2	URI	plaintext	up
12	custora.com	2	URI/cookie	SHA1	uid/_custrack1_identified*
13	dotomi.com	2	URI	SHA256	dtm_email_hash
14	inside-graph.com	2	Payload	plaintext	md
15	krxd.net	2	URI	SHA256	_kua_email_sha256
16	pxf.io	2	Payload	SHA1	custemail
17	taboola.com	2	URI	SHA256	eflp
18	thebrighttag.com	2	URI	SHA256	_cb_bt_data
19	yahoo.com	2	URI	SHA256	he
20	zendesk.com	2	URI	BASE64	data

foo@mydom.com as a value into the *trackid* parameter. This request is sent to the server of the tracking provider *tracker.net* along with this information. From there, the assumption is that *tracker.net* uses the *trackid* value *foo@mydom.com* as an identifier (*ID*) for user tracking. In case that a user uses this *ID* to complete the authentication flows in many sites, this method could be more effective than the traditional cookie-based approach because it can be used to identify user information on multiple sites, multiple browsers, and even multiple devices.

5.2 PII leakage-based web tracking cues

To confirm such a possibility, we first group the 130 first-party senders involved in the PII leakage together with the third-party receivers. Next, for each third-party receiver, we look for a PII identifier parameter (*trackid*) by checking the *parameter name* that assigns PII information as a *parameter value* in the URI parameters. We then list the PII identifier parameters per tracking provider. Then, to make sure that PII leakage is used for cross-site tracking, we focus on 34 third-party receivers involved in PII leakage that get the same ID from more than one first-party sender. Finally, to indicate the storage of a unique persistent identifier behavior, we consider 20 third-party receivers as tracking providers using PII leakage-based tracking when the IDs appears not only in the authentication flows but also on the subpage of the first-party senders. This is an indisputable tracking behavior because these tracking providers appear on any subpages of the first-party senders along with the ID (PII), which can replace the usage of third-party cookies for tracking. Note that our experimental evaluation could have missed some tracking providers that appears only one time in the dataset (58 third-party receivers). We intend to expand our dataset in future work by using crowdsourced data collection to overcome this drawback.

We show a breakdown of third-party receivers for persistent tracking based on PII leakage in Table 2. First, we notice that all tracking providers consistently use a PII type (email address) for tracking purposes. Obviously, the email address is an effective identifier that can reveal a user's identity. Second, we confirm that each tracking provider has at least one specific PII identifier parameter for multiple first-party senders. Third, the leaked PII (email address) is hashed in six encoding/hashing forms, mainly in SHA256. This raises a significant privacy concern for users when this ID can be used to share data among many tracking providers. Finally, we observed that some third parties received leaked PII in different encoding/hashing forms. In summary, PII can be used by third parties for user tracking not only on a single site but also for multiple sites, browsers, and devices when the user performs sign-up/sign-in in many websites by the same PII.

6 TRANSPARENCY

In this section, we look at the privacy policies of the 130 first-party sites to evaluate their disclosure regarding sharing PII to third parties.

First, the 130 first-party sites that leak PII to third parties obtain user consent in the authentication flow as a mandatory requirement. However, they all give users a form to sign in order to indicate agreement without showing them the policy. As a result, *most end-users skip the policy altogether* [38]. This is a type of dark pattern that manipulates the user to agree to a privacy policy without user awareness. On the other hand, as shown in Table 3, we observe all of the sites disclose that they collect personal information from users, and 111 sites state that they share or express degrees of the possibility of sharing personal information to third-party. However, *only nine sites* provide a list of third parties that received this information. Finally, some policies are very ambiguous; they do not mention PII sharing or even state that PII is used. In particular, 15 sites do not even mention PII sharing, and four sites declare that

Table 3: Privacy policy disclosures of 130 first-party senders that leak PII to third parties.

Disclosure		Number/percentage
Disclose PII sharing	Not specific	102/78.5%
	Specific	9/6.9%
No description of PII sharing		15/11.5%
Explicitly disclose PII NOT shared		4/3.1%
Total		**130/100%**

they do not share users' personal information with third parties for their marketing purposes.

7 IN-BROWSER PROTECTION TECHNIQUES EFFECTIVE AGAINST PII LEAKAGE?

7.1 Evaluating Browser Countermeasures

Some browsers focus on security and privacy by blocking trackers. Brave provides a Shields feature that blocks third-party cookies, third-party storage, and blocks or replaces requests for tracking-related JavaScript. Safari has an Intelligent Tracking Prevention (ITP) feature that blocks third-party cookies, and partitions third-party storage by default. Also, Firefox restricts third-party cookies and storage for known trackers by default by introducing Enhanced Tracking Protection (ETP). We evaluate the ability of these browsers (vanilla/bare setting) to block PII leakage. We choose the following popular browsers [2]: Chrome 93 [8], Opera 79.0 [1], Safari 14.03, and Firefox 73.0 [3]. Brave V1.29.81 [10]. We then obtain data (see §3.2) on the 130 first-party sites that leak PII to third parties. Finally, we apply the same method (see §4.1) to detect PII leakage among these profiles.

Overall, we confirm that only the Brave browser has an impact on PII leakage detection. This browser accounts for a small percentage of the browser market share [2, 15]. The remaining browsers do not solve this problem at all, even the ITP of Safari and the ETP of Firefox. For the Brave browser, the leakage of PII by first-party senders is reduced by 93.1%, and the third-party receivers are reduced by 92%. It miss eight third-party domains involved in this PII leakage[4]. Also, we observe that there is one site on which we could not complete the sign-up flow by using Brave due to Shields block CAPTCHA verification (*nykaa.com*). These results show that this feature of Brave is still not able to completely block PII leakage and that there is thus room to improve protection.

7.2 Evaluating Blocklist-based Countermeasures

Existing blocklists target advertisement and tracking resources and are utilized by browser extensions that intend to protect user privacy online. These lists may thus be able to block PII leakage resources provided by third parties. Here we assess their efficiency as countermeasures. We consider Easylist [4] and EasyPrivacy list [6]. EasyList is a blocklist that removes ads from web pages and it is used by popular browser plugins. EasyPrivacy is an optional supplementary blocklist that removes trackers from websites. We use *Adblockparser* [9] to evaluate the effectiveness of these two major blocklists by matching URLs against them. To determine if a request would have been blocked by an extension utilizing these lists, we directly match the block list rules quoted above with 1,522 HTTP requests that contained leaked PII and all requests in their request initiator chains. We inspect individual URLs containing leaked PII using these blocklists from June 2021.

The results of this experiment are shown in Table 4. First, the performance of the combined blocklist varies widely depending on the PII leakage method. It is the most effective against the leakage via cookie. It blocks all senders and receivers of this data leak method. Secondly, the EasyList blocks only 8.0% of PII leakage, and it even seems to not impact first-party senders. These are not surprising results because this list is used for ad-blocking protection. Finally, 78.5% (resp. 72.0%) of first-party (resp. third-party) senders (receivers) could be blocked. Considering PII leakage-based tracking (see Table 2), we observe that they block almost all resources belonging to these third-party domains. However, we also observe that there are still three missed third-party domains that received leaked PII according to our presumption: *custora.com*, *taboola.com*, and *zendesk.com*. In summary, even though the blocklist-based countermeasures can only partially deal with PII leakage, they can help reduce the number of leaked PII resources.

Table 4: Detection performance of well-known filters.

Metric		EasyList	EasyPrivacy	Combined
Senders	Referer	0/0%	2/66.7%	2/66.7%
	URI	1/0.8%	89/75.4%	97/82.2%
	Payload	0/0%	38/88.4%	38/88.4%
	Cookie	0/0%	5/100.0%	5/100.0%
	Combined	0/0%	24/88.9%	24/88.9%
	Total	**1/0.8%**	**95/73.1%**	**102/78.5%**
Receivers	Referer	1/14.3%	6/85.7%	6/85.7%
	URI	7/9.0%	51/65.4%	58/74.4%
	Payload	0/0%	12/70.6%	12/70.6%
	Cookie	0/0%	1/100.0%	1/100.0%
	Combined	0/0%	6/75.0%	6/75.0%
	Total	**8/8%**	**65/65.0%**	**72/72.0%**

8 CONCLUSION

In this paper, we quantified the leakage of PII from the authentication flows in the Tranco top shopping sites and revealed a tracking technique that relies on this authentication. We showed that the PII leakage is more intricate than previously thought. In addition, we conducted experiments to assess the possibility of using leaked PII for persistent web tracking and evaluated this tracking technique in our dataset. We also evaluated the ability of common browsers and well-known filter lists to detect PII leakage. We believe that the site's publishers should take a more proactive approach to terminating this type of data transfer. Our observations shed light on a little-studied, yet important, aspect of third-party web tracking and can help increase user awareness regarding privacy on the Internet.

Dataset availability: The lists of PII leakage URLs, first-party senders, and third-party receivers for popular shopping sites from the Tranco Top 10,000 sites (May 2021) are available at https://github.com/fukuda-lab/PII_leakage.

ACKNOWLEDGMENTS

We thank our shepherd, Marco Mellia, the anonymous reviewers, Satoru Kobayashi, and Johan Mazel, for their feedback, which helped improve our paper.

[4]List of eight third-party receivers missed by Brave version V1.29.81: *aliyun.com*, *cartsync.io*, *gravatar.com*, *herokuapp.com*, *intercom.io*, *lmcdn.ru okta-emea.com*, *zendesk.com*

REFERENCES

[1] 1995. *Opera browser.* Retrieved September 30, 2021 from https://www.opera.com/
[2] 1998. *Browser Statistics.* Retrieved September 30, 2021 from https://www.w3schools.com/browsers/
[3] 2004. *Firefox browser.* Retrieved September 30, 2021 from https://www.mozilla.org/en-US/exp/firefox/
[4] 2005. *Easy list.* Retrieved June 30, 2021 from https://easylist.to/easylist/easylist.txt
[5] 2005. *FortiGuard Web Filtering.* Retrieved June 30, 2021 from https://fortiguard.com/webfilter
[6] 2006. *EasyPrivacy.* Retrieved June 30, 2021 from https://easylist.to/easylist/easyprivacy.txt
[7] 2007. *Public suffix list.* Retrieved June 30, 2021 from https://publicsuffix.org/list/
[8] 2008. *Chrome browser.* Retrieved September 30, 2021 from https://www.google.com/chrome/
[9] 2016. *Python parser for Adblock Plus filters.* Retrieved June 30, 2021 from https://github.com/scrapinghub/adblockparser
[10] 2019. *Brave browser.* Retrieved September 30, 2021 from https://brave.com/
[11] 2019. *Today's Firefox Blocks Third-Party Tracking Cookies and Cryptomining by Default.* Retrieved September 30, 2021 from https://blog.mozilla.org/blog/2019/09/03/todays-firefox-blocks-third-party-tracking-cookies-and-cryptomining-by-default/
[12] 2020. *CNAME-cloaked trackers.* Retrieved June 30, 2021 from https://github.com/AdguardTeam/cname-trackers
[13] 2020. *Full Third-Party Cookie Blocking and More.* Retrieved September 30, 2021 from https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more/
[14] 2020. *NextDNS CNAME Cloaking Blocklist.* Retrieved June 30, 2021 from https://github.com/nextdns/cname-cloaking-blocklist
[15] 2020. *What is Brave browser's market share.* Retrieved September 30, 2021 from https://www.ctrl.blog/entry/brave-market-share.html
[16] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The web never forgets: Persistent tracking mechanisms in the wild. In *Proceedings of ACM SAC*. 674–689.
[17] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. 2013. FPDetective: dusting the web for fingerprinters. In *Proceedings of ACM CCS*. 1129–1140.
[18] Nataliia Bielova. 2017. Web tracking technologies and protection mechanisms. In *Proceedings of ACM CCS*. 2607–2609.
[19] Justin Brookman, Phoebe Rouge, Aaron Alva, and Christina Yeung. 2017. Cross-device tracking: Measurement and disclosures. *Proceedings on PET* 2017, 2 (2017), 133–148.
[20] Manolis Chatzimpyrros, Konstantinos Solomos, and Sotiris Ioannidis. 2019. You shall not register! detecting privacy leaks across registration forms. In *Computer Security*. Springer, 91–104.
[21] Ha Dao, Johan Mazel, and Kensuke Fukuda. 2021. CNAME Cloaking-based Tracking on the Web: Characterization, Detection, and Protection. *IEEE TNSM* 18, 3 (2021), 3873–3888.
[22] Yana Dimova, Gunes Acar, Lukasz Olejnik, Wouter Joosen, and Tom Van Goethem. 2021. The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion. *Proceedings on PET* 2021, 3 (2021), 393–411.
[23] Valery Dudykevych and Vitalii Nechypor. 2016. Detecting third-party user trackers with cookie files. In *Proceedings of IEEE PIC S&T*. 78–80.
[24] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. 2018. I never signed up for this! Privacy implications of email tracking. *Proceedings on PET* 2018, 1 (2018), 109–126.
[25] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *Proceedings of ACM CCS*. 1388–1401.
[26] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. 2016. Tracking personal identifiers across the web. In *Proceedings of PAM*. Springer, 30–41.
[27] Imane Fouad, Nataliia Bielova, Arnaud Legout, and Natasa Sarafijanovic-Djukic. 2020. Missed by Filter Lists: Detecting Unknown Third-Party Trackers with Invisible Pixels. *Proceedings on PET* 2020, 2 (2020), 499–518.
[28] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. 2016. Cloak of visibility: Detecting when machines browse a different web. In *Proceedings of IEEE S&P*. 743–758.
[29] Hugo Jonker, Benjamin Krumnow, and Gabry Vlot. 2019. Fingerprint surface-based detection of web bot detectors. In *Proceedings of ESORICS*. Springer, 586–605.
[30] Balachander Krishnamurthy, Konstantin Naryshkin, and Craig Wills. 2011. Privacy leakage vs. protection measures: the growing disconnect. In *Proceedings of WWW*, Vol. 2. 1–10.
[31] Balachander Krishnamurthy and Craig Wills. 2009. Privacy diffusion on the web: a longitudinal perspective. In *Proceedings of WWW*. 541–550.
[32] Jonathan Mayer. 2011. *Tracking the trackers: Where everybody knows your username.* Retrieved May 15, 2021 from http://cyberlaw.stanford.edu/node/6740
[33] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. 2019. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *Proceedings of WWW*. 1432–1442.
[34] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A research-oriented top sites ranking hardened against manipulation. *Proceedings of NDSS* (2019).
[35] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. 2016. Recon: Revealing and controlling pii leaks in mobile network traffic. In *Proceedings of ACM MobiSys*. 361–374.
[36] Oleksii Starov, Phillipa Gill, and Nick Nikiforakis. 2016. Are you sure you want to contact us? quantifying the leakage of pii via website contact forms. *Proceedings on PET* 2016, 1 (2016), 20–33.
[37] Oleksii Starov and Nick Nikiforakis. 2017. Extended tracking powers: Measuring the privacy diffusion enabled by browser extensions. In *Proceedings of WWW*. 1481–1490.
[38] Nili Steinfeld. 2016. "I agree to the terms and conditions":(How) do users read privacy policies online? An eye-tracking experiment. *Computers in human behavior* 55 (2016), 992–1000.
[39] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Measuring the impact of the gdpr on data sharing in ad networks. In *Proceedings of ACM CCS*. 222–235.
[40] Antoine Vastel, Walter Rudametkin, Romain Rouvoy, and Xavier Blanc. 2020. FP-Crawlers: studying the resilience of browser fingerprinting to block crawlers. In *MADWeb'20-NDSS*.
[41] David Zeber, Sarah Bird, Camila Oliveira, Walter Rudametkin, Ilana Segall, Fredrik Wollsén, and Martin Lopatka. 2020. The representativeness of automated web crawls as a surrogate for human browsing. In *Proceedings of WWW*. 167–178.

SUPPORTED HASH FUNCTIONS AND ENCODINGS FOR LEAK DETECTION

base16, base32, base32hex, base58, base64, gz, bzip2, deflate; and md2, md4, md5, sha1, sha224, sha256, sha384, sha512, crc16, crc32, sha3_224, sh3_256, sh3_384, sh3_512, ripemd_128, ripemd_169, ripemd_256, ripemd_320, whirlpool, rot13, snefru128, snefru256, adler32, blake2s, blake2b.