Think Before You Type:
A Study of Email Exfiltration Before Form Submission

Asuman Senol
imec-COSIC, KU Leuven
asenol@esat.kuleuven.be

Gunes Acar
imec-COSIC, KU Leuven
gunes.acar@esat.kuleuven.be

Mathias Humbert
Cyber-Defence Campus, armasuisse S+T
mathias.humbert@armasuisse.ch

***Abstract*—Online tracking enables companies to build behavioral profiles of users to effectively target them with ads. Since most companies provide their services on multiple platforms such as web and mobile, tracking users across platforms is crucial for effective profiling. As an increasing number of browsers effectively block third-party cookies, email addresses emerge as an alternative tracking mechanism that supports persistent cross-site, cross-platform tracking and marketing. While prior work investigated the dissemination of email addresses submitted on web forms, we focus on the collection before form submission. In particular, we present a measurement of email and password exfiltration that occurs without any form submission on top $100,000$ websites from two vantage points (EU & US). We also crawl the same websites with a crawler that emulates a mobile browser to compare the results across desktop and mobile websites. For data collection, we extend DuckDuckGo's Tracker Radar Collector software with an ML classifier to build an interactive crawler that finds and fills email and password fields in an automated manner. Our crawler features in-depth instrumentation to record script behavior and network traffic, which are then analyzed to identify tracking-related exfiltration of credentials.**

In addition, we study the effect of user consent on exfiltration by repeating our crawls while rejecting or accepting all personal data processing through automated means. Our findings show that users' emails are sent to trackers before form submission on thousands of sites in both EU and US crawls. Limiting our analysis to emails sent to known trackers, we find that on 44% more US websites (compared to the EU crawl) users' emails are collected before submission by one or more trackers. Finally, we identify incidental password collection by third-party trackers on 44 websites.

***Index Terms*—privacy, email leakage, web tracking, online tracking, password leakage**

1. Introduction

Websites commonly use third-party advertising and marketing services to monetize their content. While tracking and advertising enable websites to offer their services for free, they heavily depend on the collection of users' online activities, at times without their knowledge and consent.

As users' online activities are spread over a number of devices, tracking users only on websites may not be enough to get a complete view of their profile. As of April 2021, 56.16% percent of all web traffic is attributed to mobile phones [48], and 84% of the time on mobile is spent on applications [13]. Traditional tracking mechanisms such as cookies are limited to origins and platforms, and thus cannot be used to track users across platforms. On the contrary, email addresses are perfect identifiers to track users across different platforms since these identifiers are unique, persistent, and can even be available in the offline realm—e.g. when a user signs up for a loyalty card. Many data brokers already use email hashes to identify users [58, 11].

The demand for an alternative mechanism to track users across websites and devices has also increased since major browser vendors such as Safari and Firefox have started blocking cookies and trackers. Compared to other personal information such as name or address, email address is more effective for tracking users across platforms since it is long-term, unique and available on many websites and applications to facilitate account login, registration and newsletter subscriptions. In a recent study, Chatzimpyrros et al. found that email addresses and usernames are commonly collected by third-party scripts from registration forms [9]. Similarly, a recent news article by Surya Mattu and Kashmir Hill showed how a third party called Navistone was collecting personal information from mortgage calculator forms before the user submitted the form [37].

In this study, we investigate third-party trackers that collect email addresses, and (incidentally) passwords even if the user does not submit any form. Unlike prior work, we analyze the effect of location, user consent and mobile/desktop websites on data exfiltration by running multiple crawls of the same websites. In particular, we ran crawls from two vantage points (EU vs US), with desktop and mobile emulation; in addition, we use three different consent settings: accept all, reject all, and no action. Our contributions include the following:

- We develop an interactive and instrumented crawler based on DuckDuckGo's Tracker Radar Collector [16] to measure email and password exfiltration on Tranco top 100K sites. We fit the crawler with an ML classifier that can robustly detect email fields.
- We analyze the effect of location (the EU vs. US) and user consent on email and password exfiltration considering three different consent modes: (*accept all* / *reject all* / *no action*). This analysis

is enabled by a crawler module that automatically interacts with consent management platforms.

- Our findings show that trackers collect email addresses even before the user submits any form on thousands of websites—including popular sites with potentially sensitive content such as `webmd.com`.
- Comparing findings from two vantage points, we find that 44% more sites in the US crawl (compared to the EU) send the users' email to one or more trackers.
- While there is a slight but consistent drop-in the number of email exfiltrations in the `reject all` crawls, we find the effect of user consent to be minimal.
- Using a mobile-emulated browser, we repeat the email and password exfiltration measurements on the mobile web, finding similar results to desktop web.
- We uncover incidental password collection by session replay scripts on 44 websites.

2. Background and Related Work

2.1. Background

Web tracking is the process of identifying users' activity across websites. The personal information that can be collected or inferred by the trackers may include personal and sensitive information such as sexual orientation, political and religious beliefs. Tracking may be performed for various purposes including analytics, personalization, and to build a behavioral profile for marketing and targeted advertisements.

The most traditional way to track users across websites is to store a unique identifier in users' cookies. However, in the last decade more intrusive and persistent tracking mechanisms have emerged. Browser fingerprinting [21], evercookies [56] and cookie syncing [49] are such mechanisms that are harder to control and detect than the traditional cookies. As a reaction to these emergent tracking mechanisms, tracking protection countermeasures such as browser extensions and built-in browser defenses were developed. For instance, Safari's Intelligent Tracking Prevention, and Firefox's Enhanced Tracking Protection can prevent third-party tracking by identifying trackers and blocking cookies that are used for cross-site tracking [71, 41]. The countermeasures against traditional tracking mechanisms made alternatives such as PII-based tracking or "people-based marketing" [12] even more necessary.

2.2. Related Work

Web Tracking Several web measurement studies quantified and categorized different ways third-party trackers collect personal information across websites [32, 55, 38]. Multiple studies investigated stateful [73, 35, 25, 60] and stateless [2, 46] tracking techniques and their evolution over time. Taking an offensive approach, other studies proposed or uncovered new tracking techniques that are difficult to detect such as Flash cookies [61], canvas fingerprinting [40], and fingerprinting browser extensions via their style changes [33].

PII Leaks Krishnamurthy and Wills showed that it is possible to link PII leaked via online social networks and the data which is leaked elsewhere [31]. Since PII leaks enable cross-device tracking, some prior work investigated PII leaks on mobile devices [54, 53], or compared tracking on mobile and desktop devices [72]. Other recent work includes PII leaks due to browser extensions [64].

In a recent paper, Lin et al. presented the first comprehensive study of the privacy threats emanating from browsers' auto-fill functionality [36]. Their large-scale study showed that browsers' auto-fill functionality leads to sending sensitive personal information to tracker domains, either in hidden form fields or due to autofill preview functionality—even if users choose not to use it.

Englehardt et al. built a corpus of emails by signing up to mailing lists, and found that 30% of emails they received leaked the recipient's email address to one or more third-party servers when viewed [24]. Similar to our study, Englehardt et al. also searched and filled email fields, but their method aimed to identify leaks that happen when viewing the emails—not when typing them on the page.

Chandramouli et al. measured the prevalence of email header injection vulnerabilities, which can be used for phishing, spoofing and other attacks [8]. In an email header injection attack, the attacker provides a malicious input to a web or contact form that adds additional headers (such as CC, or BCC) to the email sent by the form or the associated web application. This may instruct the email server to, for instance, CC an email to an arbitrary address. Chandramouli et al. developed a crawler that can detect such vulnerabilities in web forms, and found 994 vulnerable pages on 414 domains by testing $23,5$M websites.

Starov et al. studied PII leakages on contact pages of the 100,000 most popular sites on the web [63]. They populated contact forms with the name, surname, email address and a sample contact message. Their results showed that after removing accidental leakages, 6.1% (1,035) of all contact forms leaked PIIs to third parties after form submission. They also found that PIIs were leaked to third parties before submitting the contact form 13 websites. While not directly comparable, our results indicate much higher number of leaks than theirs. Chatzimpyrros et al. [9] measured PII leakage on registration pages of top 200K websites and found that 6% of websites leak PII to third parties.

Our study differs from these works by focusing on email and password exfiltration during filling of the forms, and running crawls from multiple vantage points, with different consent modes to evaluate their effect on data exfiltration. Further, we compare email and password collection on mobile and desktop crawls.

Web measurement studies Many researchers developed their own tools to study web tracking techniques in the wild. Mayer and Mitchell implemented FourthParty, a Firefox extension that instrumented browser APIs, HTTP traffic and cookies [38]. Using FourthParty, they examined web tracking techniques on more than 500 websites. FPDetective is based on a modified PhantomJS and Chromium, and was used to measure browser fingerprinting on top million pages [2]. Englehardt and Narayanan developed OpenWPM, which is consisted of an instrumentation extension and automation code that drives a full-fledged Firefox browser [25]. Recently, DuckDuckGo developed Tracker Radar Collector [16], an instrumented Puppeteer-based crawler that is used to detect trackers through large-scale crawls. We chose to build our crawler by extending Tracker Radar Collector for its simplicity and scalability. We explain the details of this process in the following section.

3. Methods

3.1. Extending Tracker Radar Collector

Tracker Radar Collector (TRC) is a modular, multi-threaded, crawler that is tailored for large-scale web measurements. Using Puppeteer under the hood, TRC takes advantage of all the capabilities of the Chrome DevTools Protocol. TRC uses *collectors*—modules in charge of instrumenting tracking-related behavior—for instrumenting browser API accesses, cookies, requests and other metadata. Unlike OpenWPM's inline instrumentation [30] that wraps functions and objects with getters, TRC uses Chrome DevTools Protocol to set conditional breakpoints that are evaluated when a certain function is called or a property is accessed. When the debugger hits a breakpoint, the condition script collects the JavaScript stack trace and other metadata about the property access or function invocation.

In order to detect email and password exfiltration, we extended TRC by adding a *collector* that finds and fills email and password fields. Besides, we extended TRC's network instrumentation to capture WebSocket traffic and HTTP POST payloads, in addition to GET requests, which are already being intercepted. We also added instrumentation to intercept JavaScript access to input fields, capturing the access time, input value, and attributes of the accessed fields. A high-level overview of our crawler is shown in Figure 1.

3.2. Discovering Inner Pages

Our crawler starts to search email and password fields on the landing pages. If no field can be found, it tries to follow links to discover fields in the inner pages. To find links that are more likely to yield email and password fields we use a combined regular expression pattern that we extract from Firefox's Password Manager module [26]. The pattern contains several translations of words related to "sign in", "sign up" and "register". We search for this pattern in the following attributes of a, $button$, div, $span$ elements: `innerText`, `title`, `href`, `placeholder`, `id`, `name` and `className`. We limit ourselves to these four elements since they can be used to create links on the page. We prioritize elements that exactly match the regular expression pattern over elements that partially match the pattern. As a final fallback, we search for links (this time only considering a, $button$ elements) according to their page coordinates (i.e. distance from the top left corner). Based on a pilot crawl of 100K websites, we calculated the median position of the links that led to pages with email or password fields. The median X and Y coordinates turned out to be 1113px and 64.5px, respectively. Note that, since we used a 1440px-wide viewport in the desktop crawls, this point is very close to the viewport's top right corner, where sign-in/sign-up links are commonly found. This coordinate-based link detection method increased the number of detected email fields by around 10%. Within each link category (exact match, loose match, coordinate-based match) we prioritize 1) a and $button$ links, 2) links that are in the viewport, 3) links that are on top of other elements (computed via `Document.elementFromPoint()`). We arrived at these prioritization steps by comparing email and password yields using different methods in the pilot crawls.

While clicking the links, we keep a record of the URLs we have visited and we skip links to already visited pages. We continue to click these sorted links until we find an email field, or until we clicked ten links. We choose ten as the maximum number of links to click, since pilot crawls showed diminishing returns after ten. The complete detection process is shown in Figure 2.

3.3. Identifying Email and Password Fields

After clicking each link, we search for email and password fields on the new page and on all of its iframes. In a pilot crawl, we found that 3% of email fields are located in iframes. For detecting password fields, we search for input fields with type `password` (i.e. $input[type = password]$). However, web developers may use different types for email input fields such as $input[type = text]$ and $input[type = email]$. In fact, through pilot crawls we found that many websites, including very popular ones such as facebook.com use $input[type = text]$ elements to accommodate login with phone numbers or other username formats. To address this challenge, we adopted a classifier based on Mozilla Fathom—a supervised learning framework that can be trained to detect webpage parts such as popups [44]. Fathom works by applying *rules* that take a DOM node and score, type, or note it to express future rules. For example, for email detection, if the input element's label matches certain keywords or regular expressions, it will be scored higher than elements that do not match those keywords. These rules can be chained to create even more complex rules. Through supervised learning, Fathom determines each rule's weight, which can then be used to classify DOM elements.

In this study, we used the Fathom-based email field detector model used in Firefox Relay add-on [42]. Firefox Relay is a privacy-focused service from Mozilla that offers free email aliases. The Firefox Relay add-on automatically detects email fields on web pages to facilitate the use of email aliases. Using the Fathom-based detector allowed us to identify 76% more email fields than we would detect by simply searching for input fields with type `email`. This substantial increase justifies our use of Fathom, and shows that earlier studies that relied on `email` input type could have a missed a significant number of email fields.

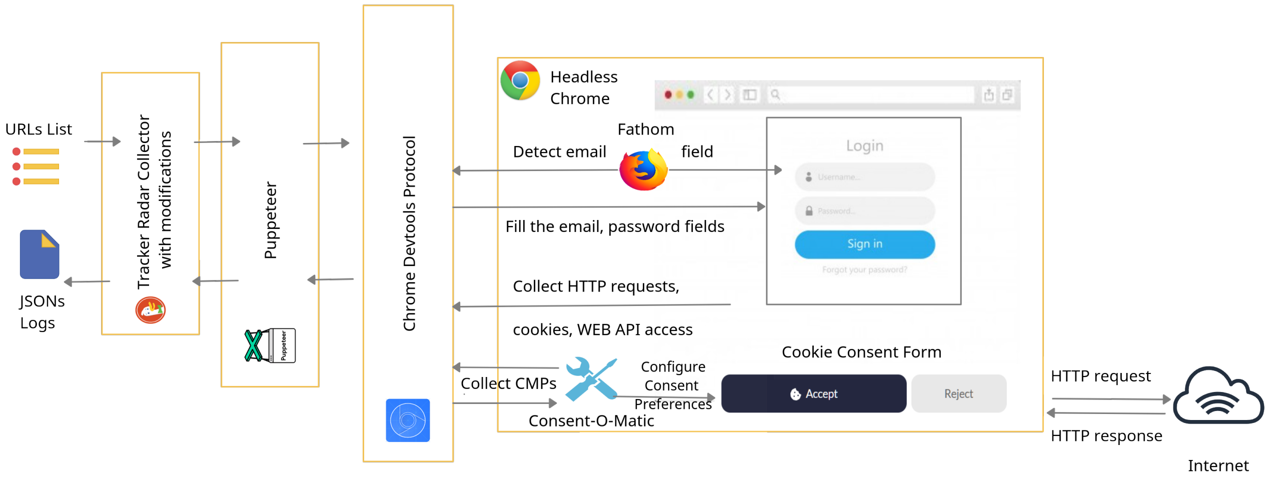

Figure 1. Components of our crawler

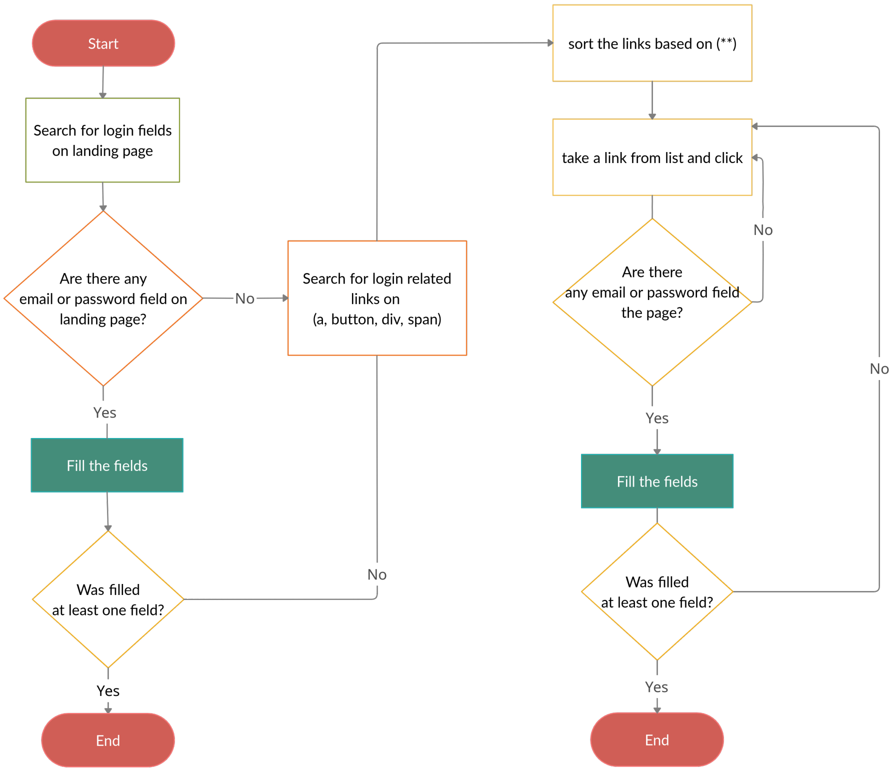
Start
Search for login fields on landing page
Are there any email or password field on landing page?
No
Search for login related links on (a, button, div, span)
Yes
Fill the fields
Was filled at least one field?
No
Yes
End
sort the links based on (**)
take a link from list and click
No
Are there any email or password field the page?
No
Yes
Fill the fields
Was filled at least one field?
Yes
End

Figure 2. Steps of detecting and filling password and email fields

3.4. Filling Email and Password Fields

We use a unique email address on each page by adding the site domain to the email address after a plus(+) character. This allowed us to uniquely attribute received emails to the websites they are collected. To address potential bot detection measures, we simulate user typing behavior by using randomized intervals for each key press and dwell times, as well as the delay times between each press.

Englehardt et al. found that the “Show password” feature, which was implemented by changing the `type` of the password input field from `password` to `text`, was the cause of password leaks on several websites [22]. To identify such leaks, before filling a password field, we change the input element type to `text` to simulate the effect of browser extensions (e.g. ShowPassword [59]) that displays passwords in cleartext. We then run a follow-up crawl without changing the password input type on websites that we find to leak password. Overall, our password exfiltration measurements aim to identify incidental collection, rather than malicious password theft.

3.5. Interaction with Consent Management Dialogs

With the introduction of the GDPR, an increasing number of websites show consent dialogs to get users’ consent for personal data processing. The acceptance or refusal to give consent may have an effect on how the website and the third parties may collect, process and share users' personal data. While one would naturally expects less tracking and data collection when refusing to give consent, prior research showed that the opposite may be true in some cases. In fact, Papadogiannakis et al. found that websites are more likely to use sophisticated forms of tracking such as ID syncing and fingerprinting when users reject cookies [50]. Regardless, web privacy studies such as ours should take consent dialog interaction into account since it may affect how websites and third parties behave.

In order to investigate the effect of users' consent preferences, we integrate Consent-O-Matic [10, 47] to our crawler. Developed by Nouwens et al. to study dark patterns in consent dialogs, Consent-O-Matic is a browser extension that can recognize and interact (e.g., accept or reject cookies) with various Consent Management Provider (CMP) pop-ups. We configure Consent-O-Matic to perform the following interactions with the CMPs, and log all the CMP-related data:

- **Accept All**: Allow processing for all purposes.
- **Reject All**: Disallow processing for all purposes.
- **No Action**: Continue without interacting with the CMP dialog.

3.6. Measurement configuration

To detect email and password leakages, we crawled the top $100,000$ Tranco websites [1] [34]. First, we directly used the Tranco domains, but we encountered DNS errors even on some top sites such as windowsupdate.com, eighth most popular site in Tranco. To address this problem, we matched the ranked Tranco domains to URLs listed in the Chrome User Experience Report–top sites visited by Chrome users [2]. When matching domains to URLs, we picked the URL with the lower rank (more popular) if there were multiple alternatives. Using Chrome UX Report URLs increased the successfully visited websites from $94,427$ (100K pilot crawl) to $99,380/99,437$. We used the March 2021 versions of both Tranco and Chrome UX Report lists.

To measure the effect of user location, we run two synchronous crawls from the EU and US—both using cloud-based servers.

We limit the maximum crawl duration to 180 seconds and maximum page load time to 90 seconds. After detecting a CMP on a website, we wait 6 seconds for the CMP interaction (accept or reject) to complete. We determined these timeouts and other crawl parameters based on data from 1K pilot crawls. For instance, we measured how long the CMP operations take and set the extra wait time to P99 of the distribution.

In addition, we run crawls for mobile websites to measure the email and password exfiltration on the mobile web. We emulated a mobile browser by adjusting the viewport dimensions, spoofing touch support, and using a mobile user agent string. The mobile-specific parameters we used are available in the TRC source code [17].

4. Email and Password Leak Detection

Identifying encoded, hashed or obfuscated leaks is a challenge that we need to address to avoid underestimating leaks. This challenge was tackled in different ways in prior work on persondal data exfiltration. Starov et al. compare data from three different crawls to identify PII in HTTP traffic [63]. Since Starov et al.'s method requires more crawls and manual analysis, we prefer Englehardt et al. method [24] which involves searching for different encodings and hashes of search terms, including Base-64, URL encoding, and several hash functions such as SHA-256. Starting with the email and password we filled, we compute a `precomputed pool` that contains all possible sets of tokens by iteratively applying the hashes and encodings. We then search for the leaks in the referrer header, cookies, URL and POST bodies of the requests, by splitting the contents by potential separator characters, such as '='. We apply all possible decodings and check whether the decoded result is in the precomputed pool. We repeat this process until we reach a level of three layers of encodings or decodings. We list the hash and encoding algorithms we used in the Appendix A.

We improve upon the original method by Englehardt et al. in a few different ways. First, in addition to splitting content by separators and decoding the resulting strings, we search for different encodings of the search terms (e.g. email and password values). This enabled us to detect leaks that do not conform to the standard `key=value` structure. Similar to the precomputed pool mentioned above, we iteratively apply the encodings. Further, we identify two new encodings and one hash method that were not covered by Englehardt et al.'s original detector. The newly discovered encoding methods include a simple substitution cipher that replaces each letter with another based on a fixed mapping. We extract this mapping from a third-party script's source code, and incorporated it into the leak detector. We identified such missed leaks by using the received emails as the proof of email collection. We manually analyzed scripts from parties that send emails, but were not found to collect leaked emails. Using this method, we also found a third party that compresses payloads using `lzstring`, and another third party that hashes email addresses with a fixed salt. Note that the latter would preclude this third party to share these hashed emails with other entities such as data brokers.

4.1. Determining Tracker-related Leaks

We exclude all requests that are sent to the first parties from our analysis. In addition, we exclude cases where we filled the email on a page that is on a different domain than the crawled website. Lastly, we only consider requests that are sent to domains flagged as a tracker by one of Disconnect [15], EasyList [19], EasyPrivacy [20], Tracker Radar [18] and Whotracks.me [69] blocklists. For Disconnect list, we also consider domains in the "Content" category, which are only blocked if Firefox is in Private Browsing mode.

4.2. Dataset

Our main dataset consist of eight crawls, all of which were run in May and June of 2021. A total of six desktop

[1] Available at https://tranco-list.eu/list/6WGX/100000

[2] https://developers.google.com/web/updates/2021/03/crux-rank-magnitude

TABLE 1. CRAWL STATISTICS BASED ON SERVERS LOCATED IN EU AND THE US.

	EU				USA			
Crawl Option	**No Action**	**Accept All**	**Reject All**	**Mobile**	**No Action**	**Accept All**	**Reject All**	**Mobile**
Crawled URLs	100K	7720	7720	100K	100K	7720	7720	100K
Successfully loaded websites	99,380	7,716	7,716	99,363	99,437	7,714	7,716	99,409
Crawled pages	625,145	44,752	40,385	597,791	690,396	51,735	49,260	668,848
Websites where we filled email	52,055	5,076	5,115	47,825	53,038	5,071	5,077	49,615
Websites where we filled password	31,002	2,306	2,342	29,422	31,324	2,263	2,283	30,356

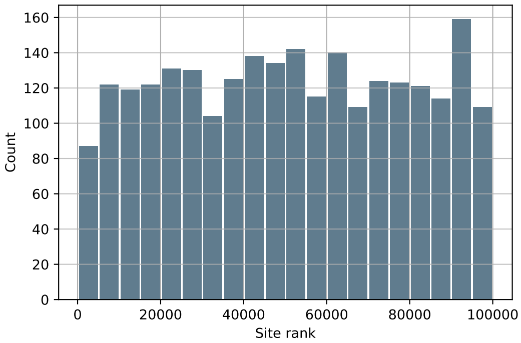

Figure 3. Distribution of tracking related leaks across various Tranco ranks in the EU

TABLE 2. THE NUMBER OF DISTINCT WEBSITES WHERE TRACKER COLLECTS EMAILS/PASSWORDS AND CMP DETECTED.

Options	**EU**	**US**
Accept all	309	295
Reject all	273	258
No action	281	279

crawls were run from the EU and US using three consent modes: *no action*, *accept all*, *reject all*. In addition, two mobile crawls were run using the *no-action* mode from the two locations. In the four, no-action crawls (100K websites) we flag the websites where we detected (but not interacted) the presence of a CMP using Consent-O-Matic. We then use these CMP-detected websites in the accept-all and reject-all crawls. For comparability we use the same $7,720$ CMP-detected websites in the accept-all and reject-all crawls on both locations—the $7,720$ websites were detected in the EU crawl.

While we limit our crawls to the top 100K websites, our dataset contains approximately $2,8$M page visits across all crawls considering the inner pages visited when searching for email and password fields. In addition to the HTTP request and response details, our dataset also contains HTML sources, JavaScript instrumentation logs, and screenshots that we use for debugging crawls when necessary.

5. Measurement Results

5.1. Prevalence of Leaks

First, we check how often emails and passwords are leaked to trackers. Table 3 shows that emails are sent to a third-party domain before form submission on $2,757$ (EU) and $3,930$ (US) websites. Passwords are sent to a third-party domain on around 100 sites in both crawls. If we only consider email leaks to tracker domains, we find a striking difference between the EU and the US crawls: $2,423$ (EU) vs. $3,484$ (US) distinct sites (44% difference).

We also look at the distribution of the website ranks where we detected an email or password leak. Figure 3 and 4 in the Appendices show a roughly uniform distribution across ranks, except a lower count in the top-1K websites.

Table 4 gives a more detailed overview of the most common trackers, including the *prominence* metric developed by Englehardt and Narayanan [25]. Prominence captures both the number and popularity of websites a third party is embedded on. Thus it better represents the scale of a given third party's reach.

The rlcdn.com domain, owned by TowerData [3] tops (by prominence) the tracker domains that received leaked emails in the US list. In their marketing material titled "Website Visitor Identification" and "Know your anonymous users", TowerData boasts about matching anonymous visitors to email hashes, "deliverable email or postal address" [67]. On the other hand, the EU list is dominated by Taboola, a native advertising company that was found to promote clickbait content and fake news [65]. According to their help pages, Taboola reaches over 1.4 billion unique visitors every month [66].

Recall that we record script access to input fields during our crawls. On certain websites, email or password is sent without directly accessing the input fields. Among $4,056$ distinct (grouped by hostname, search type, request domain) requests that leak emails and passwords, 277 were on sites where we have not recorded any access to the input fields we filled. By manually inspecting a sample of these cases, we verified that these were due to the wholesale collection of DOM [1]. In one particular example, a script from bronto.com—an email marketing platform from Oracle—iterates over the entire DOM while searching for an email address; and exfiltrates the email address to their backend.

On certain websites, email addresses (or their encodings or hashes) were sent to more than one tracker domain. For instance, on beeketing.com, 13 tracker domains including facebook.com, doubleclick.net and snapchat.com receive the email leak. The large number of trackers may be explained by cookie synchronization, in which different trackers sync their IDs to enable background data merges [51].

Table 5 shows different encodings and hashes detected in the leaks to tracker domains. A significant number of leaks on both crawls occur without any encoding or hashing. We expect email marketers to prefer unhashed formats so that they can recover the addresses to reach out to the users. On the other hand, email hashes can

[3] Formerly Rapleaf.

TABLE 3. THE NUMBER OF DISTINCT WEBSITES WHERE EMAILS OR PASSWORDS ARE LEAKED.

	EU			US		
	Distinct websites (All)	**Distinct websites (Leaks to 3rd P)**	**Distinct websites (Leaks to trackers)**	**Distinct websites (All)**	**Distinct websites (Leaks to 3rd P)**	**Distinct websites (Leaks to trackers)**
Email	4,395	2,757	**2,423**	5,518	3,930	**3,484**
Password	748	104	**83**	765	97	**79**

TABLE 4. TOP TRACKER DOMAINS THAT RECEIVE LEAKED EMAIL AND PASSWORDS.

	EU				US			
Leak Type	**Tracker Domain**	**Num. sites**	**Prominence**	**Min. Rank**	**Tracker Domain**	**Num. sites**	**Prominence**	**Min. Rank**
Email	taboola.com	327	0.0303	154	rlcdn.com	524	0.0554	217
	hsforms.com	531	0.0223	615	taboola.com	383	0.0500	95
	bizible.com	160	0.0173	242	hsforms.com	539	0.0228	615
	fullstory.com	182	0.0076	1,311	bouncex.net	189	0.0225	191
	zenaps.com	113	0.0049	2,043	bizible.com	191	0.0212	242
	awin1.com	112	0.0048	2,043	zenaps.com	119	0.0111	196
	yandex.com	121	0.0042	1,688	awin1.com	118	0.0110	196
	adroll.com	117	0.0040	3,753	fullstory.com	230	0.0106	1,311
	glassboxdigital.io	6	0.0032	328	listrakbi.com	226	0.0066	1,403
	pardot.com	78	0.0031	1,694	pippio.com	138	0.0065	567
	listrakbi.com	91	0.0025	2,219	smarterhq.io	32	0.0064	556
	bronto.com	90	0.0024	2,332	yahoo.com	255	0.0063	4,281
	rlcdn.com	11	0.0020	567	adroll.com	122	0.0049	2,343
	salecycle.com	35	0.0018	2,577	yandex.ru	141	0.0049	1,648
	gravatar.com	38	0.0017	2,048	criteo.com	134	0.0047	1,403
Password (swapped with text)	yandex.com	53	0.0020	1,688				
	yandex.ru	11	0.0005	8,714				
	logsss.com	1	0.0004	2,501	glassboxdigital.io	2	0.0031	328
	trustedform.com	1	0.0002	4,168	yandex.ru	64	0.0018	4,007
	smartlook.com	2	0.0001	12,420	dynatrace.com	2	0.0002	11,965
	dynatrace.com	1	0.0001	15,147	smartlook.com	2	0.0001	12,420
	zoho.eu	3	0.0001	26,034	inspectlet.com	2	0.0001	26,034
	inspeclet.com	2	0.0001	34,405	baidu.com	1	0.0001	20,187
	rejoiner.com	2	0.0001	42,976	rejoiner.com	2	0.0001	42,976
	baidu.com	1	0.0001	20,187	noibu.com	1	0.0001	46,936
	noibu.com	1	0.0001	29,969	trustedform.com	1	0.0001	59,442
	cactusglobal.io	1	0.0001	46,936	mixpanel.com	1	0.0001	84,547
	lr-ingest.io	1	0.0001	48,512	bigpoint.net	1	0.0001	92,125
	mixpanel.com	1	0.0001	84,547				
	glassboxdigital.io	1	0.0001	86,179				

TABLE 5. OVERVIEW OF ENCODINGS USED IN LEAKS TO TRACKER DOMAINS.

Encodings and hashes	**EU**	**US**
unencoded	1,287	1,465
urlencode	648	262
sha256	390	2,173
urlencode-sha256	360	1,857
urlencode-urlencode	259	1,743
urlencode-sha_salted_1	225	237
sha_salted_1	225	237
md5	210	1,752
urlencode-md5	149	1,666
base64	121	240
urlencode-base64	76	192
sha1	61	1,072
urlencode-sha1	39	1,023
lzstring-urlencode	-	195
urlencode-lzstring-urlencode	-	194
urlencode-custom_map_1	-	74
sha512	1	2
base64-md5	1	3

TABLE 6. BREAKDOWN OF LEAKS BY TYPE.

Leak type	**EU**	**US**
URL	61%	85%
POST body	39%	15%

be enough to track users, which could be preferred by trackers who want to avoid collecting email addresses.

Recall that we change the type of password elements to $text$ before filling them. To better understand why passwords are collected, we manually analyzed a sample of websites, including leaks to non-tracker third parties. We found that in some cases passwords were sent to third parties for checking the password strength. However, we have not found such a use case in leaks to trackers. We found most cases we analyzed to be due to incidental collection by session recording scripts, most prominently by Yandex Metrica.

When considering the results of the initial crawl where we simulated a user with the ShowPassword add-on, we found that glassboxdigital.io (Glassbox) collects users' passwords on marriott.com (328th on Tranco). According to TechCrunch, Glassbox provides session recording services to highly popular mobile applications such as Hotels.com and Singapore Airlines [68].

Password collection without input type swapping Since our primary findings are based on changing the type of the password field, they only apply to a limited number of users or websites. In order to better characterize password leaks at large, we ran an additional crawl where we did not change the input type from $password$ to $text$. We found that passwords are collected by trackers on

TABLE 7. The percentage of email or password exfiltration requests blocked by different blocklists.

	EU					US				
	Disconnect	**EasyList**	**EasyPrivacy**	**Radar**	**Ghostery**	**Disconnect**	**EasyList**	**EasyPrivacy**	**Radar**	**Ghostery**
Requests	50.1%	8.9%	35.3%	98.3%	44.4%	71.8%	20.5%	65.2%	99%	65.6%

TABLE 8. The number of distinct websites where email and passwords are set to cookies.

Configuration / Vantage Points	EU	US
Desktop	103	364
Mobile	108	422

44 distinct websites even users do not use ShowPassword or similar extensions. An overwhelming majority (42/44) of these leaks were due to Yandex Metrica's session recording feature. However, a manual analysis of Yandex Metrica's code showed that it has filters to exclude password fields from the collection. Comparing websites where Yandex collects passwords, to websites where it does not, we found that almost all *leaky* websites were built using the React framework. We have reported this problem to Yandex, and we plan to reach out to the affected first parties too. Note that three of the 44 affected websites are in the Tranco top 10K, and among them are major banks and other highly visible websites such as toyota.ru. Yandex has positively responded to our disclosure and indicated that their security team is working to address the problem.

While the majority of the leaks are sent in the URLs in both the EU and the US crawls, the proportion of URL leaks are much higher in the US as shown in Table 6. In addition, by searching for leaks in JavaScript cookie contents we found that some scripts store the email or its encodings/hashes in the cookies (see, Table 8). In order to detect JavaScript cookies set by trackers, we used stack traces we obtained through our JavaScript instrumentation.

We found that on some websites emails or passwords are sent to a third-party one character at a time, while the user is typing. We plan to investigate these cases in future work.

The leaked emails and passwords are almost always sent over encrypted connections. We only found 27 and 26 websites where emails are leaked over HTTP in the EU and US, respectively. Only on 34 and 54 websites leaks were sent to trackers over the WebSocket protocol in the EU and US.

5.2. Crawl Comparison

In this section, we compare the results from our two crawl vantage points; the EU (Germany) and the US (NYC). The differences in privacy regulations are the main motivation behind this comparison.

In the US, the number of websites where the email and password were leaked to a tracker is 44% higher than that of the EU. Certain trackers only seem to collect email addresses in the US crawls, perhaps due to stricter data protection regulations in the EU. For instance, the most prominent email collecting tracker in the US crawl (rlcdn.com, TowerData), is not even among to top ten trackers in the EU in Table 4. Similarly, some of the most prominent trackers in the US crawl that receive emails such as snapchat.com, yahoo.com and bouncex.net do not collect or receive emails in the EU crawl—or only collect/receive them on a few websites. In certain cases, the same tracking script is served with a different content based on the vantage point. For instance, securedvisit.com, the tracker that uses the substitution cipher (Section 4), is served with a different content in the EU that disables email collection.

The results seem to indicate that certain third parties actively avoiding collecting emails of the EU visitors. While we cannot know the exact reason behind this differential treatment, avoiding hefty GDPR fines could be a potential explanation. In future work, we plan to use GDPR subject access requests to further investigate this discrepancy.

5.3. The Effect of Consent

Table 2 shows the number of pages where we detect CMPs and trackers. When we reject all data processing, the number of sites with leaks to trackers decreases by 9.05%. Although the total numbers are higher in the US crawl, the rate of decrease is almost the same in the case of rejecting all data processing. Recall that, we found consent popups only on $7,720$ (7.7%) sites in the EU and $5,391$ (5.4%) sites in the US (of $100K$ sites). While our results seem to confirm Papadogiannakis et al.'s conclusion that cookie consent choices are not effective in preventing tracking [50], we note the small number of websites where we could detect CMPs as a limitation.

5.4. Mobile

We detected leaks on $2,251$, $3,211$ distinct mobile websites in the EU and US crawls, respectively 9. Although the numbers of sites with exfiltration is lower compared to desktop crawls, the ratio of the sites with leaks to the sites where we could fill email or password is nearly the same in both vantage points.

We also found several tracker domains that only received email leaks on mobile crawls. These include yieldify.com, td3x.com, getdrip.com, idx.lat and savecart.pl. A cursory check on the websites associated with these domains did not suggest that they are only targeting mobile web visitors.

5.5. Received Emails

Recall that we fill a distinct email address for each website by adding the website hostname to the email (Gmail) address after a + character. This allows us to attribute the received emails to distinct websites[4]. In the

[4] A caveat to our method is the following: we did not use separate email addresses for the EU and the US crawls, thus we cannot attribute the received emails to visits from specific locations.

TABLE 9. The number of sites leaking emails or passwords to trackers, compared to the number of sites where we could fill email.

	Sites with exfiltration/ Filled Sites EU	Sites with exfiltration/ Filled Sites US
Desktop	2,444/60,008 (0.041)	3,508/60,999 (0.058)
Mobile	2,251/55,738 (0.040)	3,211/57,715 (0.056)

six-week period following the crawls, we received 290 emails from 88 distinct sites on the email addresses used in the desktop crawls, despite not submitting any form. Most emails offer a discount, or just invite us back to their site. The sender websites seem to vary by topic and theme. Most notable examples include diabetes.org.uk, mypillow.com, and walmart.com.mx. The highest number of emails (36) sent by thecompanystore.com, for which Gmail displayed a suggestion to unsubscribe, saying we have not opened an email from this sender in the last month. On the mobile crawl email address, we received 187 emails from 71 distinct websites following the four-week period after the crawls—mobile crawls were run two weeks after the desktop crawls.

5.6. Exfiltration to First-Party Domains

Although we only consider the leaks to third-party tracker domains before form submission, we also analyzed a sample of exfiltration to first-party domains. The use cases we identified included verifying email addresses as the user is typing, and self-hosted analytics services. For instance, on shift.com, the filled email is sent to a Segment [57] instance hosted on the first-party subdomain (analytics1-api.shift.com). Future work could further investigate exfiltration to first-party domains to uncover such self-hosted analytics services, and CNAME-based trackers that appear as first parties but operated by third parties [14].

6. Limitations

Through an iterative design process, pilot crawls and extensive sanity checking, we built our crawler and analysis processes to be robust and scalable. Where possible we set the parameters of the crawler such as timeout duration and crawl depth by using data from pilot crawls. However, certain limitations apply to our data collection and analysis methods. While we search for an extensive set of encodings and hashes, and we substantially improved the leak detector module we inherited from the prior work, our leak detection method may still miss leaks that are custom encoded, encrypted, or compressed. Future work may improve the leak detector by applying methods including JavaScript execution tracing and information flow tracking [28].

During a 1K website pilot crawl, we identified three CloudFlare CAPTCHA pages that blocked our crawler. However, in larger crawls, our crawler might have been served more CAPTCHA pages, or treated differently due to crawling from cloud IP addresses. While potentially costly, future research may consider proxying the crawler traffic through residential IP addresses.

During our pilot crawls we found that we cannot detect email and password fields if they are in the Shadow DOM [39] of other elements. Since we only found two such cases in a pilot crawl of 1K websites, we believe this is an acceptable limitation. Further, our crawler is limited to crawls of one-click depth for simplicity. Input fields that can only be discovered through multiple subsequent clicks may be missed by our crawler.

We use a combination of blocklists from different providers to flag domains as trackers. These lists vary by quality and compilation method (e.g. crowdsourced vs. maintained by a company such as Disconnect). Further, we flag domains as trackers if they are present in only one of these lists. As such, our results may have both false positives and false negatives due to imperfections in those blocklists.

When presenting leaks to tracker domains, we do not distinguish third-party scripts that *collect* and *exfiltrate* emails to their backend, from third parties that only *receive* these exfiltrated emails. For instance, while emails are sent to snapchat.com on 131 sites in the desktop US crawl, the sending party is always different than Snapchat itself. We plan to study the sender-receiver pairs in more detail in future work.

7. Discussion

In this section, we discuss potential countermeasures against the exfiltration of email addresses and passwords.

7.1. Countermeasures

In recent years, all major browsers except Google Chrome implemented different forms of protection against online tracking. In 2017 Apple introduced Safari Intelligent Tracking Prevention (ITP), which combines machine learning with a rule-based system that prevents cross-site tracking [71]. Since March 2020, Safari blocks all third-party cookies [70]. Mozilla introduced tracking protection in 2018 by stripping cookies from requests to tracker domains, based on a tracker list compiled by Disconnect [45, 15]. Since the beginning of 2021, Firefox partitions network state to prevent Supercookies that abuse obscure client-side storage mechanisms for tracking [23], and blocks all third-party cookies in private browsing mode through a privacy feature called Total Cookie Protection [29].

Since we used existing blocklists to flag exfiltrations to tracker domains, browsers that employ these blocklists (such as Disconnect by Firefox) could also be blocking email and password exfiltrations automatically. In order to check whether different browsers block the exfiltrations we uncovered, we manually analyzed ten different websites containing a distinct tracker that we found to exfiltrate email addresses. We manually filled the email fields on these websites and checked whether the exfiltration occurs by inspecting the HTTP request payloads in the developer tools interface. We found that neither Safari nor Firefox blocked email exfiltrations to tracking endpoints in our small sample. This result may be expected as these browsers try to strike a balance between usability and privacy by minimizing breakage and curtailing cross-site tracking at the same time. To this end, they allow requests to tracker domains, but they strip cookies, partition network state, or block access to storage that may facilitate cross-site tracking. In particular, Firefox in default mode only blocks requests to third parties that use browser fingerprinting for certain purposes such as advertising and analytics [4]. When in Private Browsing mode, however, Firefox does block requests to tracker domains. Similarly, Safari blocks cookies and other storage in third-party context, limits the lifetime of third-party cookies, but does not block requests to tracker domains [4].

Browser vendors may take further steps to protect against scripts that exfiltrate email addresses before any form submission, which effectively bypass their built-in tracking protections. Browsers may block the loading of such scripts, or prevent them from accessing certain form fields, or provide them with fake data—e.g. an empty string similar to how a zero-filled IDFA is returned on iOS devices unless the user has given their consent [3]. Further, Firefox already uses such an exception to block loading of scripts that use fingerprinting for advertisement and analytics [5]. We believe the scale of unconsented data collection uncovered in our study justifies a similar exception for scripts that harvest email addresses.

Browser extensions such as uBlock Origin [27], and browsers such as Brave [7] block requests to tracker domains, which better protects against email exfiltration than countermeasures built-in to Firefox and Safari. Since mobile Chrome does not support extensions, available options for mobile browsers are limited, but users may still opt for browsers that support extensions (e.g. Firefox, Safari), or use a privacy-focused mobile browser that blocks trackers such as Brave [7] and DuckDuckGo [52].

Recently, Firefox [43] and Apple [6] started to offer private email relay services that give users the ability to generate and use pseudonymous (alias) email addresses. These privacy-focused services automatically forward emails received at the alias addresses, and allow users to keep their real email address hidden from untrusted online services.

In their study on data exfiltration from contact forms, Starov et al. developed an extension called FormLock that identifies and highlights “leaky” forms [62]. Formlock also protects the PII from leaking even if the user decides to use a leaky contact form despite the warning. Formlock achieves this protection by blocking requests to parties other than the first party and the form’s intended endpoint.

7.2. Future Work

We plan to reach the websites where email and password leaks occurred, and warn them, especially of the password leaks. We have already contacted Yandex about incidental password collection, who acknowledged the problem and working on solving it. We plan to use GDPR subject access requests to ask the first parties whether they are aware of the email exfiltration to trackers on their websites. Further, we plan to ask the third parties about how they use the collected email addresses, how long they retain them, and whether they share the addresses further with other third parties.

In this study, we also inspected email and password exfiltration on the desktop and mobile web. Similar leaks that occur in mobile applications can be studied in future work.

8. Conclusions

We presented a large-scale study of email and password exfiltration by third-party trackers. In order to address the challenges of finding and filling input fields, we integrated into our crawler an ML classifier that detects email fields. We found thousands of sites where emails are sent to trackers before users submit any form. Further, we found tens of sites where passwords are incidentally collected by third parties providing session replay services. Comparing data collected from the EU and the US vantage points, we found that 44% more websites in the US crawl leaked users’ emails to third-party trackers. Measuring the effect of consent choices on the exfiltration, we found their effect to be minimal. Our findings show that users should assume that the personal information that they enter into web forms–not submit–may be collected by third-party trackers. We believe the problem uncovered in our study deserves the attention of browser vendors, privacy tool developers and data protection agencies.

Acknowledgments

We thank all SecWeb reviewers for their valuable and constructive comments. Gunes Acar holds a Postdoctoral fellowship from the Research Foundation Flanders (FWO). This work was supported by CyberSecurity Research Flanders with reference number VR20192203.

References

[1] G. Acar, S. Englehardt, and A. Narayanan. “No boundaries: data exfiltration by third parties embedded on web pages.” In: *Proc. Priv. Enhancing Technol.* 2020.4 (2020), pp. 220–238.

[2] G. Acar, M. Juárez, N. Nikiforakis, C. Dıéaz, S. F. Gürses, F. Piessens, and B. Preneel. “FPDetective: dusting the web for fingerprinters”. In: *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS’13, Berlin, Germany, November 4-8, 2013*. ACM, 2013, pp. 1129–1140.

[3] *advertisingIdentifier | Apple Developer Documentation*. [Online; accessed 17. Jul. 2021]. July 2021. URL: https : / / developer . apple . com / documentation / adsupport / asidentifiermanager / 1614151-advertisingidentifier.

[4] S. Ahava. *Firefox :: Current status | cookiestatus.com*. [Online; accessed 17. Jul. 2021]. May 2021. URL: https://www.cookiestatus.com.

[5] S. Ahava. *Firefox :: Current status | cookiestatus.com*. [Online; accessed 17. Jul. 2021]. May 2021. URL: https://www.cookiestatus.com/firefox/#other.

[6] Apple. *Hide My Email for Sign in with Apple*. https://support.apple.com/en-us/HT210425. [Online; accessed 17-July-2021]. 2020.

[7] Brave. *Secure, Fast & Private Web Browser with Adblocker — Brave Browser*. https://brave.com/. [Online; accessed 12-July-2021]. 2021.

[8] S. P. Chandramouli, P.-M. Bajan, C. Kruegel, G. Vigna, Z. Zhao, A. Doupé, and G.-J. Ahn. “Measuring E-mail header injections on the world wide web”. In: *Proceedings of the 33rd Annual ACM Symposium on Applied Computing, SAC 2018, Pau, France, April 09-13, 2018*. Ed. by H. M. Haddad, R. L. Wainwright, and R. Chbeir. ACM, 2018, pp. 1647–1656. DOI: 10.1145/3167132.3167308. URL: https://doi.org/10.1145/3167132.3167308.

[9] M. Chatzimpyrros, K. Solomos, and S. Ioannidis. “You Shall Not Register! Detecting Privacy Leaks Across Registration Forms”. In: *Computer Security - ESORICS 2019 International Workshops, IOSec, MSTEC, and FINSEC, Luxembourg City, Luxembourg, September 26-27, 2019, Revised Selected Papers*. Vol. 11981. Lecture Notes in Computer Science. Springer, 2019, pp. 91–104.

[10] *Consent-O-Matic*. https://github.com/cavi-au/Consent-O-Matic. [Online; accessed 30-May-2021].

[11] *Data Services API: Endpoints*. [Online; accessed 16. Jul. 2021]. July 2021. URL: https://developer.myacxiom.com/code/api/endpoints/hashed-entity.

[12] DataQ. *People-Based Marketing In The Cookiepocalypse*. https://dataq.ai/blog/the-rise-of-people-based-marketing/. 2021.

[13] N. Deshdeep. *Mobile App or Website? 10 Reasons Why Apps are Better*. https://vwo.com/blog/10-reasons-mobile-apps-are-better. [Online; accessed 30-May-2021]. 2021.

[14] Y. Dimova, G. Acar, L. Olejnik, W. Joosen, and T. Van Goethem. “The cname of the game: Large-scale analysis of dns-based tracking evasion”. In: *arXiv preprint arXiv:2102.09301* (2021).

[15] Disconnect-Tracking-Protection. *disconnect-tracking-protection*. https://github.com/disconnectme/disconnect-tracking-protection/blob/master/services.json. [Online; accessed 11-Jun-2021].

[16] DuckDuckGo. *Tracker Radar Collector*. https://github.com/duckduckgo/tracker-radar-collector. [Online; accessed 31-May-2021]. 2020.

[17] duckduckgo. *tracker-radar-collector*. [Online; accessed 18. Jul. 2021]. July 2021. URL: https://github.com/duckduckgo/tracker-radar-collector/blob/380e3bfec591001cda35a2436f0c07f85458ec92/crawler.js#L19-L25.

[18] *DuckDuckGo Tracker Radar*. https://github.com/duckduckgo/tracker-radar. [Online; accessed 14-July-2021]. 2021.

[19] EasyList. https://easylist.to/easylist/easylist.txt. [Online; accessed 11-Jun-2021].

[20] EasyPrivacy. https://easylist.to/easylist/easyprivacy.txt. [Online; accessed 11-Jun-2021].

[21] P. Eckersley. “How unique is your web browser?” In: *International Symposium on Privacy Enhancing Technologies Symposium*. Springer. 2010, pp. 1–18.

[22] S. Englehardt, G. Acar, and A. Narayanan. *No boundaries for credentials: New password leaks to Mixpanel and Session Replay Companies*. https://freedom-to-tinker.com/2018/02/26/no-boundaries-for-credentials-password-leaks-to-mixpanel-and-session-replay-companies/. [Online; accessed 12-july-2021]. 2018.

[23] S. Englehardt and A. Edelstein. *Firefox 85 Cracks Down on Supercookies – Mozilla Security Blog*. [Online; accessed 18. Jul. 2021]. July 2021. URL: https://blog.mozilla.org/security/2021/01/26/supercookie-protections.

[24] S. Englehardt, J. Han, and A. Narayanan. “I never signed up for this! Privacy implications of email tracking”. In: *Proc. Priv. Enhancing Technol.* 2018.1 (2018), pp. 109–126.

[25] S. Englehardt and A. Narayanan. “Online tracking: A 1-million-site measurement and analysis”. In: *Proceedings of ACM CCS 2016*. 2016.

[26] Firefox. *Firefox password manager*. https://searchfox.org/mozilla-central/source/toolkit/components/passwordmgr/NewPasswordModel.jsm. [Online; accessed 30-May-2021].

[27] gorhill. *uBlock*. [Online; accessed 17. Jul. 2021]. July 2021. URL: https://github.com/gorhill/uBlock.

[28] D. Hedin, A. Birgisson, L. Bello, and A. Sabelfeld. “JSFlow: Tracking information flow in JavaScript and its APIs”. In: *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. 2014, pp. 1663–1671.

[29] T. Huang, J. Hofmann, and A. Edelstein. *Firefox 86 Introduces Total Cookie Protection – Mozilla Security Blog*. [Online; accessed 16. Jul. 2021]. July 2021. URL: https://blog.mozilla.org/security/2021/02/23/total-cookie-protection.

[30] J. Jueckstock and A. Kapravelos. “VisibleV8: In-browser monitoring of JavaScript in the wild”. In: *Proceedings of the Internet Measurement Conference*. 2019, pp. 393–405.

[31] B. Krishnamurthy and C. E. Wills. “On the leakage of personally identifiable information via online social networks”. In: *Comput. Commun. Rev.* 40.1 (2010), pp. 112–117.

[32] B. Krishnamurthy and C. E. Wills. “Privacy diffusion on the web: a longitudinal perspective”. In: *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*. ACM, 2009, pp. 541–550.

[33] P. Laperdrix, O. Starov, Q. Chen, A. Kapravelos, and N. Nikiforakis. “Fingerprinting in Style: Detecting Browser Extensions via Injected Style Sheets”. In: *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, Aug. 2021.

[34] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. “Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation”. In: *Proceedings of the 26th Annual Network and Distributed System*

Security Symposium. NDSS 2019. Feb. 2019. DOI: 10.14722/ndss.2019.23386.

[35] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. "Internet Jones and the Raiders of the Lost Trackers: An Archaeological Study of Web Tracking from 1996 to 2016". In: *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*. USENIX Association, 2016.

[36] X. Lin, P. Ilia, and J. Polakis. "Fill in the Blanks: Empirical Analysis of the Privacy Threats of Browser Form Autofill". In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. Ed. by J. Ligatti, X. Ou, J. Katz, and G. Vigna. ACM, 2020, pp. 507–519.

[37] S. Mattu and K. Hill. "Before You Hit 'Submit,' This Company Has Already Logged Your Personal Data". In: *Gizmodo* (Oct. 2020). URL: https://gizmodo.com/before-you-hit-submit-this-company-has-already-logge-1795906081.

[38] J. R. Mayer and J. C. Mitchell. "Third-Party Web Tracking: Policy and Technology". In: *IEEE Symposium on Security and Privacy, SP 2012, 21-23 May 2012, San Francisco, California, USA*. IEEE Computer Society, 2012, pp. 413–427.

[39] by MDN contributors. *Using shadow DOM*. https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM. [Online; accessed 2-Jun-2021]. 2021.

[40] K. Mowery and H. Shacham. "Pixel Perfect: Fingerprinting Canvas in HTML5". In: *Proceedings of W2SP 2012*. IEEE Computer Society. May 2012.

[41] Mozilla. *Enhanced Tracking Protection in Firefox for desktop*. https://support.mozilla.org/en-US/kb/enhanced-tracking-protection-firefox-desktop. 2021.

[42] Mozilla. *fx-private-relay*. [Online; accessed 10. Jun. 2021]. June 2021. URL: https://github.com/mozilla/fx-private-relay/blob/v1.2.2/extension/js/email_detector.js.

[43] Mozilla. *Mozilla Relay*. https://relay.firefox.com/. [Online; accessed 1-Jun-2021]. 2020.

[44] *Mozilla Fathom*. https://mozilla.github.io/fathom/. [Online; accessed 1-Jun-2021]. 2019.

[45] N. Nguyen. *Changing Our Approach to Anti-tracking – Future Releases*. [Online; accessed 16. Jul. 2021]. July 2021. URL: https://blog.mozilla.org/futurereleases/2018/08/30/changing-our-approach-to-anti-tracking.

[46] N. Nikiforakis, A. Kapravelos, W. Joosen, C. Kruegel, F. Piessens, and G. Vigna. "Cookieless Monster: Exploring the Ecosystem of Web-Based Device Fingerprinting". In: (2013), pp. 541–555. DOI: 10.1109/SP.2013.43. URL: https://doi.org/10.1109/SP.2013.43.

[47] M. Nouwens, I. Liccardi, M. Veale, D. Karger, and L. Kagal. "Dark patterns after the GDPR: Scraping consent pop-ups and demonstrating their influence". In: *Proceedings of the 2020 CHI conference on human factors in computing systems*. 2020, pp. 1–13.

[48] Oberlo. *What Percentage of Internet Traffic is Mobile?* https://www.oberlo.com/statistics/mobile-internet-traffic. [Online; accessed 30-May-2021].

[49] L. Olejnik, M.-D. Tran, and C. Castelluccia. "Selling off User Privacy at Auction". In: *21st Annual Network and Distributed System Security Symposium, NDSS 2014, San Diego, California, USA, February 23-26, 2014*. The Internet Society, 2014.

[50] E. Papadogiannakis, P. Papadopoulos, N. Kourtellis, and E. P. Markatos. "User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users". In: *Proceedings of the Web Conference 2021*. 2021, pp. 2130–2141.

[51] P. Papadopoulos, N. Kourtellis, and E. Markatos. "Cookie synchronization: Everything you always wanted to know but were afraid to ask". In: *The World Wide Web Conference*. 2019, pp. 1432–1442.

[52] *Privacy, simplified. — DuckDuckGo Browser Extension & Mobile App*. [Online; accessed 17. Jul. 2021]. July 2021. URL: https://duckduckgo.com/app.

[53] A. Razaghpanah, R. Nithyanand, N. Vallina-Rodriguez, S. Sundaresan, M. Allman, C. Kreibich, and P. Gill. "Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem". In: *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*. The Internet Society, 2018.

[54] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. R. Choffnes. "ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic". In: *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys 2016, Singapore, June 26-30, 2016*. ACM, 2016, pp. 361–374.

[55] F. Roesner, T. Kohno, and D. Wetherall. "Detecting and Defending Against Third-Party Tracking on the Web". In: *Proceedings of the 9th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2012, San Jose, CA, USA, April 25-27, 2012*. USENIX Association, 2012, pp. 155–168. URL: https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner.

[56] *Samy Kamkar - evercookie - virtually irrevocable persistent cookies*. [Online; accessed 10. Jun. 2021]. June 2021. URL: https://samy.pl/evercookie.

[57] *Segment | #1 CDP to Manage Customer Data*. [Online; accessed 17. Jul. 2021]. July 2021. URL: https://segment.com.

[58] *Sending oHashes to Oracle Data Cloud platform*. [Online; accessed 11. Jun. 2021]. June 2021. URL: https://docs.oracle.com/en/cloud/saas/data-cloud/data-cloud-help-center/IntegratingBlueKaiPlatform/IDManagement/sending_ohashes.html.

[59] *ShowPassword*. [Online; accessed 11. Jun. 2021]. June 2021. URL: https://chrome.google.com/webstore/detail/showpassword/bbiclfnbhommljbjcoelobnnnibemabl.

[60] K. Solomos, P. Ilia, S. Ioannidis, and N. Kourtellis. "Clash of the Trackers: Measuring the Evolution of the Online Tracking Ecosystem". In: *CoRR* abs/1907.12860 (2019). arXiv: 1907.12860. URL: http://arxiv.org/abs/1907.12860.

[61] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle. “Flash Cookies and Privacy”. In: *Intelligent Information Privacy Management, Papers from the 2010 AAAI Spring Symposium, Technical Report SS-10-05, Stanford, California, USA, March 22-24, 2010*. AAAI, 2010.

[62] O. Starov. *FormLock*. https://github.com/ostarov/Formlock. [Online; accessed 14-July-2021]. 2021.

[63] O. Starov, P. Gill, and N. Nikiforakis. “Are you sure you want to contact us? Quantifying the leakage of PII via website contact forms”. In: *Proceedings on Privacy Enhancing Technologies* 2016.1 (2016), pp. 20–33.

[64] O. Starov and N. Nikiforakis. “Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions”. In: *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 2017, pp. 1481–1490.

[65] *Taboola*. https://www.taboola.com/. [Online; accessed 13-July-2021].

[66] *Taboola Help Center*. https://pubhelp.taboola.com/hc/en-us/articles/360003157074-Why-Taboola-. [Online; accessed 16-July-2021].

[67] TowerData. *Website Visitor Identification Software Solutions | TowerData*. [Online; accessed 17. Jul. 2021]. July 2021. URL: https://www.towerdata.com/website-visitor-identification.

[68] Z. Whittaker. *Many popular iPhone apps secretly record your screen without asking*. https://techcrunch.com/2019/02/06/iphone-session-replay-screenshots/. [Online; accessed 16-July-2021].

[69] *whotracks.me*. https://github.com/ghostery/whotracks.me. [Online; accessed 14-July-2021]. 2021.

[70] J. Wilander. *Full Third-Party Cookie Blocking and More*. [Online; accessed 10. Jun. 2021]. Apr. 2020. URL: https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more.

[71] J. Wilander. *Intelligent Tracking Prevention*. [Online; accessed 16. Jul. 2021]. Feb. 2017. URL: https://webkit.org/blog/7675/intelligent-tracking-prevention.

[72] Z. Yang and C. Yue. “A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments”. In: *Proc. Priv. Enhancing Technol.* 2020.2 (2020), pp. 24–44.

[73] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol. “Tracking the Trackers”. In: *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*. ACM, 2016, pp. 121–132.

Appendix A. Supported Hash and Encoding Methods for Leak Detection

Hashes and Checksums: md2, md4, md5, sha1, sha256, sha224, sha384, sha512, sha3224, sha3256, sha3384, sha3512, murmurhash3 32-bit, murmurhash3 64-bit, murmurhash3 128-bit, ripemd160, whirlpool, salted sha1

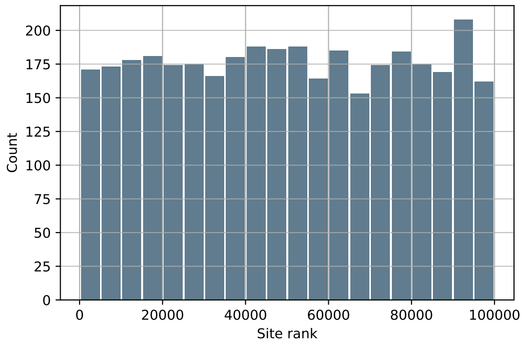

Figure 4. Distribution of tracking related leaks across various Tranco ranks in the US

Encodings: base16, base32, base58, base64, urlencode, entity, deflate, zlib, gzip, lzstring, custom map