Leaky Forms:
A Study of Email and Password Exfiltration Before Form Submission

Asuman Senol
imec-COSIC, KU Leuven

Gunes Acar
Radboud University

Mathias Humbert
University of Lausanne

Frederik Zuiderveen Borgesius
Radboud University

Abstract

Web users enter their email addresses into online forms for a variety of reasons, including signing in or signing up for a service or subscribing to a newsletter. While enabling such functionality, email addresses typed into forms can also be collected by third-party scripts even when users change their minds and leave the site without submitting the form. Email addresses—or identifiers derived from them—are known to be used by data brokers and advertisers for cross-site, cross-platform, and persistent identification of potentially unsuspecting individuals. In order to find out whether access to online forms is misused by online trackers, we present a measurement of email and password collection that occurs before the form submission on the top 100,000 websites. We evaluate the effect of user location, browser configuration, and interaction with consent dialogs by comparing results across two vantage points (EU/US), two browser configurations (desktop/mobile), and three consent modes. Our crawler finds and fills email and password fields, monitors the network traffic for leaks, and intercepts script access to filled input fields. Our analyses show that users' email addresses are exfiltrated to tracking, marketing and analytics domains before form submission and without giving consent on 1,844 websites in the EU crawl and 2,950 websites in the US crawl. While the majority of email addresses are sent to known tracking domains, we further identify 41 tracker domains that are not listed by any of the popular blocklists. Furthermore, we find incidental password collection on 52 websites by third-party session replay scripts.

1 Introduction

Websites commonly use third-party advertising and marketing services to monetize their content. Those services heavily depend on monitoring users' online activities, at times without their knowledge and consent. Stateful tracking mechanisms such as cookies are isolated by origins, and limited to the web platform. As users' online activities are spread over a number of connected devices, tracking users only on websites does not suffice to get a complete view of their profile. 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 or partitioning third-party cookies and trackers.

Email addresses are ideal identifiers to fill this gap, since they are unique, persistent, and can even be available in the offline realm—e.g., when a user signs up for a loyalty card. Compared to other personal information such as name or postal address, email addresses are more effective for tracking users across platforms, since they are long-term, unique, and available on many websites and applications to facilitate account login, registration, and newsletter subscriptions. Data brokers and advertisers already use email hashes to identify users, track them across devices, and match their online and offline activities [7,25,36].

The demand for a more global and persistent identifier, along with the ongoing phase-out of third-party cookies, makes email addresses typed into online forms an attractive target for collection by trackers. However, prior work on the collection of credentials typed into online forms is limited. Besides, the collection of information *before* form submission has been even less studied. Only a 2017 news article by Surya Mattu and Kashmir Hill reported that a third party called Navistone was collecting personal information from mortgage calculator forms before the user submitted the form [72]. This is despite the high dropout rates among web users (e.g., in signup forms [27,30]), which shows that many users indeed leave websites without submitting the form they started filling out. For instance, a survey by The Manifest found that 81% of the 502 respondents have abandoned forms at least once, and 59% abandoned a form in the last month [39].

In this paper, we investigate to what extent third-party trackers collect email addresses, and (incidentally) passwords, even if the user does not submit any form. Unlike prior work, we focus on leaks that occur before form submission, and we analyze the effect of location, of user consent to personal data processing, and of mobile vs. desktop browsing.

In addition, we evaluate the effect of users' location, of user consent to personal data processing, and of mobile vs. desktop browsing. In particular, we run crawls from two vantage points (EU & US), with desktop and mobile-emulated browsers. In addition, we use three different cookie consent settings to investigate the effect of user consent: accept all, reject all, and no action. Our contributions include the following:

- We develop an interactive, instrumented crawler based on DuckDuckGo's Tracker Radar Collector [35] to measure email and password exfiltration on Tranco top 100K sites. We fit the crawler with a pre-trained machine-learning (ML) classifier that can robustly detect email fields. Our crawler is further able to fill the email and password fields and to intercept script access to filled input fields (Section 3.1).

- Based on a crawl of 2.8 million pages from the top 100K sites, we find that trackers collect email addresses before form submission on thousands of websites in both EU (1,844 websites) and US (2,950 websites) crawls—60% more exfiltrations when the same sites are visited from the US. We uncover 41 previously unknown tracker domains that exfiltrate email addresses. We develop a proof-of-concept browser add-on that detects sniff and exfiltration attempts on online forms.

- We discuss whether email exfiltrations by trackers are compliant with the GDPR or not (Section 5). Further, we send GDPR requests to a sample of websites and third parties, asking the purpose of their email collection, retention period and further sharing policies (Section 6).

- Finally, we uncover incidental password collection by session replay providers on 52 websites (Section 4.2). Two third-party trackers with a combined presence of five million websites released fixes to address the issue, thanks to our disclosures.

2 Background and Related Work

2.1 Background

Web tracking is the process of collecting information about users' online activities 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 building 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 [54], evercookies [13] and cookie syncing [77] 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 [11, 89]. The countermeasures against traditional tracking mechanisms made alternatives such as tracking based on personal identifiers or "people-based marketing" [22] even more necessary.

2.2 Related Work

Online tracking Several studies investigated stateful [68, 82] and stateless [58, 61, 66] tracking techniques and their evolution over time. Taking an offensive approach, other studies proposed new tracking techniques that are difficult to detect such as canvas and GPU fingerprinting [65, 74]. Analyzing IAB Europe's Transparency and Consent Framework (TCF) cookie banners, Matte et al. found a widespread violation of the GDPR and the ePrivacy Directive; for instance by registering positive consent when the user has not made a choice [71]. Similar to our discussion on GDPR compliance of email exfiltration practices (Section 5), Mayer and Mitchell presented an overview of regulation that applies to online tracking–but their analysis predates modern privacy laws such as the GDPR [73].

Personal information leaks Lin et al. presented the first comprehensive study of privacy threats emanating from browsers' auto-fill functionality [69]. While relevant, auto-fill-related abuse is orthogonal to the types of exfiltration we investigate. Acar et al. studied personal data exfiltration by third parties, uncovering inadvertent password leaks by session replay scripts, and third parties that harvest (hashed) email addresses by injecting invisible login forms that trigger browsers' login managers [42].

Englehardt et al. built a corpus of emails by signing up to mailing lists, and they found that 30% of emails they received leaked the recipient's email address to one or more third-party servers when viewed in an email client program or web application [57]. Similar to our study, Englehardt et al. also searched and filled email fields, but their method aimed to identify leaks that occur when reading emails—not when typing email addresses on the page.

Starov et al. studied PII leakage on contact pages of the 100,000 most popular sites on the web [85]. They populated contact forms with a name, surname, email address and a sample contact message. Their results showed that, after removing accidental leakage, 6.1% $(1,035)$ of all contact forms leaked PIIs to third parties after form submission. They also

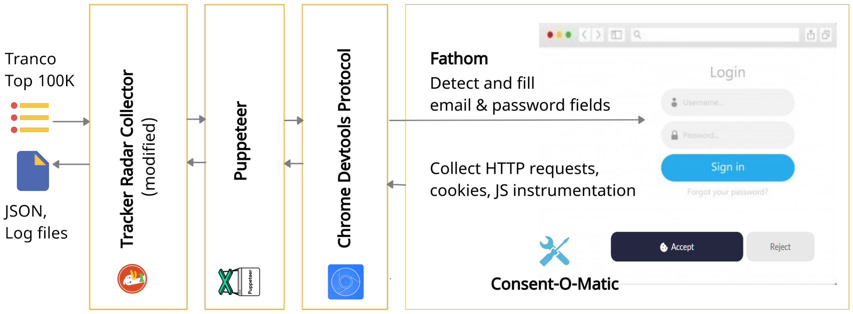

Figure 1: Components of our crawler. We integrate Firefox Relay's Fathom-based email field classifier [10] and Consent-O-Matic [46] to Tracker Radar Collector (TRC)—a web privacy measurement crawler developed by DuckDuckGo [35]. TRC is based on Puppeteer, which uses Chrome Devtools Protocol to interact with the underlying browser. We modify TRC to efficiently discover inner pages, and fill email and password fields.

found that PIIs were leaked to third parties before submitting the contact form on 13 websites. Unlike Starov et al.'s work, we ran comparative crawls (mobile/desktop, US/EU, consent modes); and our leak detection method did not require three visits. While not directly comparable, we identified substantially more personal information leaks.

Chatzimpyrros et al. [48] and Dao et al. [50] investigated PII leaks on top 200K websites, and on 307 popular shopping websites respectively. Chandramouli et al. measured the prevalence of email header injection vulnerabilities in web forms, which can be used for phishing, spoofing, and other attacks [47]. Other prior work investigated PII leaks on mobile devices [80, 81], or compared tracking on mobile and desktop devices [91].

Our study differs from these works by focusing on email and password exfiltration during the filling of the forms. We run crawls from multiple vantage points, with different consent modes to evaluate their effect on data exfiltration. We compare email and password collection on mobile and desktop crawls. In addition, we use GDPR requests to reach out to first and third parties to ask for the purposes of email address collection.

Web privacy measurement studies Many researchers developed their own tools to study web tracking techniques in the wild. In 2012, Mayer and Mitchell implemented FourthParty, a Firefox extension that instrumented browser APIs, HTTP traffic and cookies [73]. 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 the top million pages [43]. Englehardt and Narayanan developed OpenWPM, which consists of an instrumentation extension and automation code that drives a full-fledged Firefox browser [58]. Jueckstock and Kapravelos contributed VisibleV8, a modified V8 JavaScript engine that logs all native JS function calls and property accesses, without the need to add specific instrumentation [64]. Akhavani et al. inspected 33 Google Chrome, 31 Mozilla Firefox, and 33 Opera browser versions released from 2016 to 2020 by using VisibleV8, and showed that different browser versions have identifiable fingerprints [44]. Recently, DuckDuckGo developed Tracker Radar Collector [35], 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.

Login security Jonker et al. presented a framework called Shepherd, which detects login pages using a combination method of searching for login-based URLs, clickable elements and search engine APIs [63]. Shepherd also interacts with the login forms, and analyzes authentication cookies to determine whether the website is vulnerable to session hijacking. Analyzing the use of web authentication mechanisms on 100,000 domains, Van Acker et al. showed that login pages of certain open-source web frameworks and content management systems are vulnerable to several attacks under various adversary models [87]. They evaluated 51,307 login pages from 100K websites against man-in-the-middle attacks showing that 62.8% of login pages are vulnerable to adversaries with moderate resources. Van Acker et al.'s study also showed that password leaks to third parties are possible on many websites. Unlike these two studies measuring login page vulnerabilities, we measure the *actual misuse* by trackers on real-world websites.

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 capturing tracking-related behavior—that captures browser API accesses, cookies and requests. Unlike OpenWPM's inline instrumentation [64] 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 set by TRC, 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 input element. 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 [15]. 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 X and Y position of the links that led to pages with email or password fields: 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 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 and fill 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 links.

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. We search for iframes since a pilot crawl of top 1K Tranco sites showed that 3% of email fields are found in iframes. For detecting password fields, we search for input fields with type `password` (i.e. `input[type='password']`). However, email input fields do not need to have the `email` type (i.e. `input[type='email']`). In fact, through pilot crawls we found that many websites, including popular ones such as facebook.com, use text input elements to accommodate login with phone numbers or other username formats. To address this challenge, we integrated into our crawler a pre-trained email field classifier based on Mozilla Fathom [10]. Fathom is a supervised learning framework specialized to detect webpage parts such as popups [14]. We used the Fathom-based email field detector model used in Firefox Relay add-on [10]. Firefox Relay is a privacy-focused service from Mozilla that offers free email aliases [1]. 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 may indicate that earlier studies that relied on `email` input type could have missed a significant number of 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 on. 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. After typing into each field, we simulate pressing the 'Tab' key to switch to the next form field, while triggering the *blur* event on the previously filled element.

Englehardt et al. found that the "Show password" feature, which changes the `type` of the password field from `password` to `text`, caused certain session replay scripts to collect the passwords incidentally [55]. To measure such leaks at large, the crawler changes the password fields' type from `password` to `text` before filling the field. This allows us to simulate the effect of browser extensions such as ShowPassword [26], which displays passwords in cleartext. We then run a follow-up crawl without changing the password input type on websites where we identified password leaks. Overall, our password exfiltration measurements aim to identify the incidental collection, rather than malicious password theft.

[1] Coincidentally, Firefox Relay and similar email alias services can be used as countermeasures against email exfiltration we study in this paper.

3.5 Interaction with Consent Management Dialogs

After the introduction of the GDPR in 2018, more websites started to show 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 expects less tracking and data collection when refusing to give consent, prior research showed that in certain cases the opposite may be true: a recent study by Papadogiannakis et al. found that websites are more likely to use sophisticated tracking techniques such as ID syncing and fingerprinting when users reject cookies [79]. 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 [46] into 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 [76]. We configure Consent-O-Matic to log detected CMPs, and perform the following interactions with the CMPs:

accept-all: Allow processing for all purposes. **reject-all**: Disallow processing for all purposes. **no-action**: Continue without interacting with the CMP dialog, if any.

3.6 Measurement Configuration

We measure email and password exfiltration on the top $100,000$ Tranco websites [67][2]. Initially, we used the Tranco domains without any changes, but we encountered DNS errors even on most popular websites such as windowsupdate.com—the eighth most popular site in Tranco. To address this problem, we matched Tranco domains to URLs listed in the Chrome User Experience Report [1], which contains actual URLs visited by Chrome users. When matching domains to URLs, we pick the URL with the lower rank (more popular) if there are multiple alternatives. This minor change increased the successfully visited websites from $94,427$ (EU pilot crawl) to $99,380$ (EU final crawl). We used the March 2021 versions of both Tranco and Chrome UX Report lists.

To compare results based on user location, we run two simultaneous crawls from the EU (Frankfurt) and the US (New York City)—both using cloud-based servers hosted on Digital Ocean. For each crawl, we use one server with 16 cores and 32GB RAM.

We limit the maximum crawl duration on a site 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 the 99th percentile of the distribution (6 seconds).

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 [35]. For mobile crawls, we fill a different email address to distinguish emails we received due to mobile and desktop crawls. We omit experiments with different consent modes for mobile crawls due to limited time and space.

3.7 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 in web privacy measurement studies. Starov et al. compare data from three different crawls to identify PII in HTTP traffic [85]. Since Starov et al.'s method requires more crawls and manual analysis, we prefer Englehardt et al.'s method [57], which involves searching for different encodings and hashes of search terms, including Base64 encoding, and 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 we 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 Appendix 10.

We improve upon the original method by Englehardt et al. in several 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 incorporate it into the leak detector. We identified such missed leaks by using the received emails as 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

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

	EU				US			
Crawl Option	**no-action**	**accept-all**	**reject-all**	**mobile**	**no-action**	**accept-all**	**reject-all**	**mobile**
Crawled URLs	100K	7,720	7,720	100K	100K	7,720	7,720	100K
Successfully loaded websites	99,380	7,716	7,716	99,363	99,437	7,714	7,716	99,409
Crawled pages	625,143	44,752	40,385	597,791	690,394	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

Table 1: Desktop crawl statistics based on servers located in the EU and the US. no-action, accept-all, reject-all indicate consent modes. Crawled pages also include inner pages that we visited.

another third party that hashes email addresses with a fixed salt, which was hard-coded in their script. Note that using (salted) email hashes may prevent this third party to match identities with external entities such as data brokers—unless the data broker also uses the same salt for hashing emails.

3.8 Determining Tracker-related Leaks

There may be legitimate reasons why email addresses and—to some extent—passwords are collected before form submission: For instance, checking whether an email/username picked by a user is available before form submission. To avoid counting such cases, we exclude from our analysis all requests that are sent to first-party domains, or third-party domains that are not flagged as trackers. When determining *third partyness* we make use of Tracker Radar's entity list [12], which contains a list of domains owned by a company. Using entity-to-domains mapping allows us to better determine the third parties, and prevent overcounting the leaks. In addition, we exclude cases where we filled the email on a page or on an iframe that has a different domain than the crawled website. Note that throughout the study by *domain*, we mean registrable domain name or the effective top-level domain plus one (eTLD+1).

Lastly, we only consider requests that are sent to endpoints flagged as a tracker by one of Disconnect [52], Whotracks.me [33], DuckDuckGo [9] blocklists and uBlock Origin [16]. For the Disconnect list, we also consider domains in the “Content” category, which is only blocked if Firefox is in Private Browsing mode. For uBlock Origin, we use the blocklists enabled by default in the add-on. These include EasyList, EasyPrivacy and Peter Lowe's Ad and tracking server list, among others.

Manual tracker labeling Additionally, we label the leaky request domains that are not flagged as trackers by any of the Disconnect, Whotracks.me, DuckDuckGo and uBlock Origin. For each such domain, we follow a decision algorithm explained in Appendix 10 to determine the tracker status. Thanks to this manual analysis, we uncover 41 tracker domains that are not listed in any of the popular blocklists. Manually labeled domains accounted for an increase of 13.4% and 4.2% in the number of websites with email leaks, in the EU and US crawls, respectively (for no-action, desktop crawls). We plan to share these domains with blocklists providers.

3.9 Dataset

Our main dataset consists of eight crawls, all of which were run in May and June of 2021. A total of six desktop crawls were run from the EU and the 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 2.8M 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 can be used to debug the crawler. Each 100K website crawl took five days to run. The ethics considerations we took into account during the study can be found in Section 9.

4 Measurement Results

Results in this section are based on desktop crawls and no-action mode (no interaction with the cookie dialog) unless otherwise specified.

4.1 Email Leaks

Prevalence of leaks Table 3 shows that email addresses (or their hashes) are sent to a third-party tracker on $1,844$ (EU) vs. $2,950$ (US) distinct websites. This shows that, on more than a thousand websites, trackers only collect emails when the website is visited from the US.

Table 2 gives a more detailed overview of the most common trackers that emails are leaked to. Prom. stands for promi-

	EU						US					
Leak Type	Entity Name	Tracker Domain	Key by key	Num. sites	Prom.	Min. Rank	Entity Name	Tracker Domain	Key by key	Num. sites	Prom.	Min. Rank
email	Taboola	taboola.com	No	327	302.9	154	TowerData	rlcdn.com	No	524	553.8	217
	Adobe	bizible.com	Yes	160	173.0	242	Taboola	taboola.com	No	383	499.0	95
	FullStory	fullstory.com	Yes	182	75.6	1,311	Bounce Exchange	bouncex.net	No	189	224.7	191
	Awin Inc.	zenaps.com*	No	113	48.7	2,043	Adobe	bizible.com	Yes	191	212.0	242
		awin1.com*	No	112	48.5	2,043	Awin	zenaps.com*	No	119	111.2	196
	Yandex	yandex.com	Yes	121	41.9	1,688		awin1.com*	No	118	110.9	196
	AdRoll	adroll.com	No	117	39.6	3,753	FullStory	fullstory.com	Yes	230	105.6	1,311
	Glassbox	glassboxdigital.io*	Yes	6	31.9	328	Listrak	listrakbi.com	Yes	226	66.0	1,403
	Listrak	listrakbi.com	Yes	91	24.9	2,219	LiveRamp	pippio.com	No	138	65.1	567
	Oracle	bronto.com	Yes	90	24.6	2,332	SmarterHQ	smarterhq.io*	Yes	32	63.8	556
	TowerData	rlcdn.com	No	11	20.0	567	Verizon Media	yahoo.com*	Yes	255	62.3	4,281
	SaleCycle	salecycle.com	Yes	35	17.5	2,577	AdRoll	adroll.com	No	122	48.6	2,343
	Automattic	gravatar.com*	Yes	38	16.7	2,048	Yandex	yandex.ru	Yes	141	48.1	1,648
	Facebook	facebook.com	Yes	21	14.8	1,153	Criteo SA	criteo.com*	No	134	46.0	1,403
	Salesforce	pardot.com*	Yes	36	30.8	2,675	Neustar	agkn.com*	No	133	45.9	1,403
	Oktopost	okt.to*	Yes	31	11.4	6,589	Oracle	addthis.com	No	133	45.9	1,403
pswd	Yandex	yandex.com	Yes	37	12.12	4,699	Yandex	yandex.ru	Yes	45	17.23	1,688
		yandex.ru		7	2.41	12,989						
	Mixpanel	mixpanel.com	Yes	1	0.12	84,547	Mixpanel	mixpanel.com	Yes	1	0.12	84,547
	LogRocket	lr-ingest.io	Yes	1	0.12	82,766	LogRocket	lr-ingest.io	Yes	1	0.12	82,766

Table 2: Top tracker domains and associated entities that emails or passwords are exfiltrated to in desktop crawls using the no-action mode which was conducted in May'21. *: Third-party domain is not among the request initiators; that means the leak could have been triggered by another party. Prominence (Prom.) values have been multiplied by 1,000 for readability.

	EU			US		
	All	Third party	Tracking related	All	Third party	Tracking related
Email	4,395	2,633	**1,844**	5,518	3,790	**2,950**
Password	89	87	**48**	92	87	**49**

Table 3: The number of distinct websites where email and passwords are sent to first-party domains vs. third-party domains in desktop crawls using the no-action mode.

nence, a metric developed by Englehardt and Narayanan [58] which captures both the quantity and popularity of websites a third party is embedded on. We use prominence to sort third parties in Table 2 because it better represents the scale of a given third party's reach.

In the US crawl, rlcdn.com (TowerData, formerly Rapleaf) is the most prominent tracker domain that collects hashed email addresses. In their marketing material titled "Website Visitor Identification" and "Know your anonymous users", TowerData mentions matching anonymous visitors using email hashes [29]. On WebMD and Fox News websites, TowerData collected the MD5, SHA-1 and SHA-256 hashes of the email address typed into the login form. The EU list, on the other hand, is dominated by Taboola—an advertising company that was found to promote clickbait and other problematic content and ads [83, 92]. According to their help pages, Taboola accepts hashed emails to create target audiences [38] based on over 1.4 billion unique visitors they reach every month [31].

Cross-domain email sharing for identity matching On 565 of the 1,844 distinct websites (EU) where we identified email leaks to tracker domains, no script from the request domain was among the request's initiators. This means that these requests are initiated by other parties. Analyzing HTTP request initiators, and JavaScript stack traces of access to input fields we found that email leaks to yahoo.com, criteo.com and dotomi.com are always initiated by other parties. The email hashes to yahoo.com, for example, are sent by a script from adthrive.com (CafeMedia)—a digital publishing and ad monetization network that Yahoo has a partnership with [84]. The Yahoo endpoint (`ups.analytics.yahoo.com`) that email hashes are sent to, is described in Yahoo's ConnectID API documentation [32]. The documentation mentions that the API can be used for ID matching and is built on Verizon Media's *ID Graph*, "delivering a higher find rate of audiences on publishers' sites [sic] user targeting". Clickagy(.com), on the other hand, sends email hashes to up to seven other tracker domains including agkn.com (Neustar) and pippio.com (LiveRamp, formerly Acxiom), both of which accepts hashed emails for various services according to their public documentation and privacy policies [23, 37].

EU				US			
Rank	**Website**	**Third-party**	**Hash/encoding/compression**	**Rank**	**Website**	**Third-party**	**Hash/encoding/compression**
154	**usatoday.com***	taboola.com	Hash (SHA-256)	95	**issuu.com**	taboola.com	Hash (SHA-256)
242	**trello.com***	bizible.com	Encoded (URL)	128	**businessinsider.com**	taboola.com	Hash (SHA-256)
243	**independent.co.uk***	taboola.com	Hash (SHA-256)	154	**usatoday.com**	taboola.com	Hash (SHA-256)
300	**shopify.com**	bizible.com	Encoded (URL)	191	**time.com**	bouncex.net	Compression (LZW)
328	**marriott.com**	glassboxdigital.io	Encoded (BASE-64)	196	**udemy.com**	awin1.com	Hash (SHA-256 with salt)
567	**newsweek.com***	rlcdn.com	Hash (MD5, SHA-1, SHA-256)			zenaps.com	Hash (SHA-256 with salt)
705	**prezi.com***	taboola.com	Hash (SHA-256)	217	**healthline.com**	rlcdn.com	Hash (MD5, SHA-1, SHA-256)
754	**branch.io***	bizible.com	Encoded (URL)	234	**foxnews.com**	rlcdn.com	Hash (MD5, SHA-1, SHA-256)
1,153	**prothomalo.com**	facebook.com	Hash (SHA-256)	242	**trello.com***	bizible.com	Encoded (URL)
1,311	**codecademy.com**	fullstory.com	Unencoded	278	**theverge.com**	rlcdn.com	Hash (MD5, SHA-1, SHA-256)
1,543	**azcentral.com***	taboola.com	Hash (SHA-256)	288	**webmd.com**	rlcdn.com	Hash (MD5, SHA-1, SHA-256)

Table 4: Top ten websites where the filled email was collected by a tracker before form submission in desktop crawls using the no-action mode. *: Not reproducible anymore as of February 2022.

Our findings showed that email addresses or their hashes are sent to facebook.com on 21 distinct websites in the EU. On 17 of these, Facebook Pixel's Automatic Advanced Matching feature [21] was responsible for sending the SHA-256 of the email address in a `SubscribedButtonClick` event, despite not clicking any submit button. According to its documentation, Automatic Advanced Matching captures hashed customer data including email addresses, phone numbers, first and last names; from checkout, sign-in and registration forms. We believe the leaks are due to Facebook's script interpreting clicks on irrelevant buttons as "submit button clicked" events.

Website categories In order to compare email exfiltration across website categories, we query McAfee's categorization service [6]. Note that a website may have multiple categories. As shown in Table 5, Fashion/Beauty and Online Shopping are the two categories where we detect the most email exfiltrations—considering only the categories with more than 1,000 websites in our 100K sample. On the other hand, websites categorized as Public Information, Government/Military, and Games leaked less than 1% of the filled email address. A somehow surprising result was the following: despite filling email fields on hundreds of websites categorized as Pornography, we have not a single email leak. While surprising, this is in line with limited prior research on tracking on the adult websites: a limited 2016 study by Altaweel et al. found that adult websites have relatively fewer third-party trackers compared to non-adult websites with comparable popularity [45].

Effect of website popularity The number of websites with email leaks follows a close to a uniform distribution in the US crawl. On the other hand, in the EU crawl, there are substantially fewer sites with email leaks on the Tranco top 5K: only 1.28% sites on the top 5K has email leaks, compared to the average of 1.87% in websites with rank >5000 (cf. US top 5K: 2.96%, 5K-100K: 2.95%). Popular websites and trackers may be using questionable data collection methods sparingly in the EU to avoid GDPR fines or investigations.

Top websites with leaks Table 4 shows the top ten websites with email leaks for each vantage point. We list the third-party tracker found to collect emails on these sites, along with the hashing/encoding method used when exfiltrating the email. News websites such as usatoday.com, foxnews.com and independent.co.uk, appear high on the lists. This is in line with prior work which found that news websites contain the highest number of third parties compared to other website categories [58]. Medical news and information websites webmd.com and healthline.com are other notable entries for their sensitive content.

Emails sent key by key As shown in Table 2, certain third parties send email addresses character-by-character, as the user types in their address. This behavior appears to be due to session replay scripts that collect users' interactions with the page including key presses and mouse movements [42].

HTTP and WebSocket usage Finally, we observed that the leaked emails are almost always sent over encrypted (HTTPS) connections. We only found 15 and 14 websites where emails are leaked over HTTP in the EU and the US, respectively. In addition, on 67 websites in the EU and on 132 websites in the US, the leaks were sent over the WebSocket protocol—to hotjar.com, freshrelevance.com, noibu.com and decibelinsight.net.

4.2 Password Leaks

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.

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 follow up crawls of websites where we detected

	EU/US	EU		US	
Categories	**Sites**	**Filled sites**	**Leaky sites**	**Filled sites**	**Leaky sites**
Fashion/Beauty	1,669	1,176	131 (11.1%)	1,179	224 (19.0%)
Online Shopping	5,395	3,658	345 (9.4%)	3,744	567 (15.1%)
General News	7,390	3,579	235 (6.6%)	3,848	392 (10.2%)
Software/Hardware	4,933	2,834	138 (4.9%)	2,855	162 (5.7%)
Business	13,462	7,805	377 (4.8%)	7,924	484 (6.1%)
...	...	...	...	...	...
Games	2,173	925	9 (1.0%)	896	11 (1.2%)
Public Information	2,346	1,049	8 (0.8%)	1,084	27 (2.5%)
Gov't/Military	3,754	939	5 (0.5%)	974	7 (0.7%)
Uncategorized	1,616	636	3 (0.5%)	646	2 (0.3%)
Pornography	1,388	528	**0 (0.0%)**	645	**0 (0.0%)**

Table 5: Per-category number of websites we crawled, filled an email field, and observed an email leak to a tracker domain (based on desktop crawls using the no-action mode). The percentage under the Leaky sites column is based on total websites where we could fill an email field (i.e. 100 * Num. of leaky sites / Num. of filled sites).

a password leak; but this time we did not change the input type from *password* to *text*. We ran two such crawls, one from the EU, and one from the US; both desktop crawls. Unless otherwise specified, password leaks presented throughout this paper are based on these latter crawls, without input type swapping. We found that passwords are collected by trackers on 52 distinct websites even for users who do not use ShowPassword or similar extensions. An overwhelming majority (50/52) 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. Note that 7 of the 52 affected websites are in the Tranco top 20K, and some of them are major banks and other highly visible websites such as toyota.ru. We have already reported this problem to Yandex, and reached out to the affected first parties as explained in Section 6.

4.3 Vantage Points: EU vs. US

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 crawl, the number of websites with email leaks is 60% higher than that of the EU: 1,844 vs 2,950.

Comparing the websites where we detected an email leak, we find that 2,950 websites identified in the US crawl are roughly a superset of the (1,844) websites identified in the EU crawl: 94.4% of the 1,844 websites detected in the EU crawl also appears in the list of websites in the US crawl.

Tracker domains such as addthis.com, yahoo.com, doubleclick.net and criteo.com only seem to receive email addresses in the US crawls, perhaps due to stricter data protection regulations in the EU. In addition, the most prominent email collecting tracker across both crawls (rlcdn.com, TowerData), is not even among the top ten trackers in the EU in Table 2. [3] In fact, according to its webpage on GDPR compliance, TowerData does not seem to provide identity matching services in the EU [78]. In certain cases, the same tracking script is served with different content based on the vantage point. For instance, securedvisit.com, the tracker that uses a substitution cipher to encrypt its payload (Section 3.7), serves a slightly different script to EU visitors that disables email collection.

Consent modes	**EU**	**US**
accept-all	239	242
reject-all	201	199
no-action	202	228

Table 6: The number of distinct websites where emails were leaked and a CMP was detected in desktop crawls using the no-action mode.

Overall, our results appear to indicate that certain third parties avoid collecting EU visitors' email addresses. In Section 5, we provide a legal analysis of whether the practice of collecting emails before form submission complies with the GDPR.

4.4 The Effect of Consent

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). Crawling these websites with three consent modes, we obtain the results in Table 6, which shows the number of websites where we detect CMPs and email leaks to trackers. When we reject all data processing, the number of sites with leaks to trackers decreases by 13% in the US, 0.05% in the EU. The reduction in leaks in both cases is limited confirming Papadogiannakis et al.'s conclusion that cookie consent choices are not effective in preventing tracking [79]. Almost no reduction in the EU leaks, however, may be counter-intuitive. This is likely due to the limited number of websites where we could detect CMPs and observe leaks.

4.5 Mobile

We detected leaks on 1,745 and 2,744 distinct mobile websites in the EU and US crawls, respectively (Table 7). Although the number of sites with leaks is lower compared to desktop crawls, the ratio of the sites with leaks to the sites where we could fill email is nearly the same in both vantage points.

[3] In fact, TowerData sent a 451 HTTP error code (*Unavailable For Legal Reasons*) in responses to requests made in the EU crawl.

The mobile and desktop websites where emails are leaked to tracker domains overlap substantially but not completely. The Jaccard similarity of (leaky) desktop and mobile websites is equal to 66% in the EU and 64% in the US. The difference between the desktop and mobile results could be due to website dynamism and the time difference between the mobile and desktop crawls (more than a month).

We also found 18 tracker domains that only received email leaks on mobile crawls such as yieldify.com, td3x.com and getdrip.com. However, checking the websites associated with these domains did not suggest that they are only targeting mobile web visitors. Further, we found 24 domains that only appear in desktop crawls, further indicating that the difference could be due to factors such as time difference and website dynamism.

	Leaky/ Filled Sites EU	**Leaky/ Filled Sites US**
Desktop	1,844 / 60,008 (3.0%)	2,950/ 60,999 (4.8%)
Mobile	1,745 / 55,738 (3.1%)	2,744 / 57,715 (4.8%)

Table 7: The number of sites leaking emails or passwords to trackers, compared to the number of sites where we could fill an email address in desktop and mobile crawls using the no-action mode.

4.6 Emails Received on the Filled Addresses

Since our crawler fills a distinct email address for each website, we are able to attribute the received emails to distinct websites.[4] In the 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. 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 Does Email Exfiltration Comply With the GDPR?

In this section, we discuss how email exfiltration can breach at least three core rules of the General Data Protection Regulation (GDPR) [49]. Roughly speaking, the GDPR could be seen as a Europe-wide data privacy law. Because of length constraints, we focus on three main principles of the GDPR, omitting greater detail.

We discuss email exfiltration in general. We do not discuss to what extent specific companies comply with the GDPR. For such a company-specific analysis, each example of email exfiltration would have to be assessed separately, considering all the circumstances of that case.

Does the GDPR apply? The GDPR applies when 'personal data' are processed. Personal data are defined broadly in the GDPR. Essentially, any information that relates to an identifiable person is personal data (Article 4.1). For instance, an email address, an IP address, a tracking cookie, an identification number, and an 'online identifier' are almost always personal data. But even hashed or encrypted email addresses are generally personal data, as far as they contain a unique identifier that can be linked to a person [4]. Moreover, hashed email addresses can often be reversed [41]. 'Processing' is defined broadly too in the GDPR: virtually everything that can be done with personal data is a type of processing (Article 4(2)). Hence, if website owners or third parties exfiltrate an email address, they process personal data and the GDPR applies.

An organization that processes personal data is a 'controller' in GDPR parlance. The 'controller' is responsible for complying with the GDPR, and can be fined for non-compliance. In the case of email exfiltration, the website owner and the third party are typically both responsible (as 'joint controllers') [34, 70].

Is the GDPR relevant for companies outside Europe? The territorial scope of the GDPR is complicated, but can be summarized as follows (Article 3 GDPR). If the controller is based in the EU, the GDPR applies. But the GDPR can also apply to controllers based outside the EU. For instance, offering goods or services to Europeans can trigger the GDPR. If a website owner sells something and allows payment in Euros, and processes the personal data of website visitors, the owner must comply with the GDPR. The GDPR also applies to controllers based outside the EU, if they 'monitor' the behavior of people in the EU. Tracking people online is an example of such monitoring [60]. Hence, if a company uses email exfiltration for tracking web users in the EU, it must comply with the GDPR.

Transparency principle The GDPR has six overarching principles relating to the processing of personal data. The first principle says that personal data must be processed 'fairly and in a transparent manner' (Article 5). The controller must provide comprehensive information about what it does with personal data, in an 'intelligible and easily accessible form, using clear and plain language' (Article 11). Moreover, the GDPR requires detailed information about, for instance, the processing 'purposes', and the 'recipients of the personal data' (Article 13 and 14). Controllers can provide such information in a privacy notice.

Does email exfiltration comply with the transparency

[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.

principle? If the website does not clearly disclose that it or a third party exfiltrates email addresses, the exfiltration breaches the transparency principle. A phrase such as 'we share your personal data with selected marketing partners' does not provide sufficient transparency.

Purpose limitation principle Does email exfiltration comply with the GDPR's purpose limitation principle? Roughly summarized, the purpose limitation says that controllers can only collect personal data if they specify a clear purpose in advance. And the controller is not allowed to use the data for 'incompatible' new other purposes (Article 6(1)(b)). Suppose that the first purpose is enabling website visitors to manage their website account. The first purpose will be something like 'remembering the website visitors' login credentials so that they can open and maintain an account'. Say that the third party uses the exfiltrated email address for behavioral advertising, email marketing or tracking people around the web. Those purposes are incompatible with the original purpose, and thus prohibited.

The requirement for a legal basis such as consent Another important GDPR requirement is that the controller always needs a 'legal basis' to process personal data (Article 6). There are six possible legal bases, including consent. The requirements for valid consent are strict. For instance, a consent request that is hidden in the small print of a contract or privacy notice cannot lead to valid consent. Further, a controller cannot assume consent if people fail to opt-out (Article 4(11)). The GDPR does not always require the person's consent. However, for online tracking and behavioral advertising, the GDPR does require prior consent [3, 88].

To obtain valid consent to collect website visitors' email addresses before they click submit, the consent request would have to be specific; such as: 'Do you agree with us collecting your email address and sharing it with company, A, B, and C for email marketing before you click submit?'. Only if the website visitor clearly agrees to such a request, the visitor gives valid consent to email exfiltration. If the request was vague, or if the visitor did not clearly express their choice, the consent is invalid.

In certain situations, email exfiltration might be allowed under the GDPR without the website visitor's consent. Suppose that a security firm (third party) exfiltrates a website visitor's email address for an extra security check. Assuming that the security firm complies with all the other GDPR norms, the firm could be allowed to exfiltrate the email address without consent (based on Article 6(1)(f)).

Conclusion Email exfiltration by third parties can breach at least three GDPR requirements. First, if such exfiltration happens surreptitiously, it violates the transparency principle. Second, if such exfiltration is used for purposes such as behavioral advertising, marketing and online tracking, it also breaches the purpose limitation principle. Third, if the email exfiltration is used for behavioral advertising or online tracking, the GDPR typically requires the website visitor's prior consent. For breaching any of these three rules, controllers can be fined up to 20,000,000 Euro or up to 4% of their total worldwide annual turnover (Article 83(5)).

6 Security Disclosures, GDPR Requests, and Leak Notifications

Our methods allow us to detect email and password leaks from clients to trackers, but what happens after the leaks reach third party's servers is unknown to us. In order to better understand the server-side processing of collected emails, and to disclose cases of password collection, we have reached out to more than a hundred first and third parties. We used the real identity and university email account of one of the authors when reporting the issues or sending the GDPR requests. Moreover, we made it clear that our inquiries are sent within the context of an academic research.

Password collection disclosures Once again we note that we believe all password leaks to third parties mentioned below are incidental. We reached out to all third parties listed in Table 2. Yandex, the most prominent tracker that collects users' passwords, has quickly responded to our disclosure and rolled out a fix to prevent password collection. We have also notified more than 50 websites where passwords were collected. Since the majority of the websites embedding Yandex were in Russian, we have enclosed a Russian translation of our message in the notification email, along with our message in English. Mixpanel released an update only two days after we disclosed the issue. With this change, even the users with outdated SDKs were protected from collecting passwords involuntarily. LogRocket, who collected passwords on publicize.co's login page, have never replied to our repeated contact attempts[5]; and the password leak remained on Publicize's website for more than ten weeks, before it was fixed.

GDPR requests on email exfiltration We reached out to 58 first and 28 third parties with GDPR requests. We avoided sending blanket data access requests to minimize the overhead for the entities who were obliged to respond to our GDPR requests. Instead, we asked specific questions about how the collected emails are processed, retained and shared. In addition, we notified the top 33 websites[6] where we detected email exfiltration in the US crawl. We sent a friendly notification to these websites about the email exfiltration, rather than a formal GDPR request. We did not get any response from these 33 websites.

When selecting the first parties to send GDPR requests to, we included the most popular websites from the EU crawl, for which we could reproduce the email leaks. We asked the

[5] We have also enrolled the help of a contact at the Electronic Frontier Foundation, who tried calling LogRocket's phone number, emailed their privacy contact address, and their cofounder—all to no avail. Our attempts to disclose the issue via LogRocket's chatbot have also failed. We have also contacted Publicize, and have not heard back.

[6] 33 out of the top 50 websites for which we could reproduce the exfiltration.

first parties if they were aware of the email collection on their websites, how they used the collected email addresses, and how long they retained them.

Responses from first parties: Almost half of the first parties (30/58) responded to our requests.

- fivethirtyeight.com (via Walt Disney's DPO), trello.com (Atlassian), lever.co, branch.io and cision.com were among the websites that said they had not been aware of the email collection prior to form submission on their websites and removed the behavior.

- Marriott said that the information collected by Glassbox is used for purposes including customer care, technical support, and fraud prevention.

- Tapad, a cross-device tracking company on whose website we found an email leak, said that they are not offering their services to UK & EEA users since August, 2021; and they have deleted all data that they held from these regions.

- stellamccartney.com explained that the emails on their websites were collected before the submission due to a technical issue, which was fixed upon our disclosure. According to their response, the SaleCycle script that collected email addresses had not been visible to their cookie management tool from OneTrust.

Responses from third parties: Roughly half (15/28) of the third parties responded. Eight third parties, including Adobe, FullStory and Yandex said they are data processors, and asked us to send our GDPR request to the corresponding first parties.

- Taboola said in certain cases they collect users' email hashes before form submission for ad and content personalization; they keep email hashes for at most 13 months; and they do not share them with other third parties. Taboola also said they only collect email hashes after getting user consent; however, our findings and subsequent manual verification showed that was not always the case.

- Zoominfo said their "FormComplete" product appends contact details of users to forms, when the user exists in ZoomInfo's sales and marketing database. They said the ability to capture form data prior to submission can be enabled or disabled by their clients.

- ActiveProspect said their TrustedForm product is used to certify consumer's *consent to be contacted* for compliance with regulations such as the Telephone Consumer Protection Act in the US. They said data captured from abandoned forms are marked for deletion within 72 hours, is not shared with anyone including the site owner.

We picked the above responses to reflect the diversity of reasons for which email addresses are collected prior to form submission. While some collection reportedly occurs due to technical glitches, or (surprisingly) for compliance purposes; other responses point to collection for marketing, analytics and identity matching purposes. In certain cases, companies suggested that the email data are not shared with any third parties, while others have not made the same promise. The limited number of responses we received, along with potential response bias, prevent us from making generalizations. Regardless, we note the benefit of reaching out to the respective parties, despite the substantial logistics overhead. Due to limited space, we could only include a selection of the responses. We plan to publish an overview of the responses as part of our dataset.

7 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 [89]. Since March 2020, Safari blocks all third-party cookies [90]. Mozilla introduced tracking protection in 2018 by stripping cookies from requests to tracker domains, based on a tracker list compiled by Disconnect [52, 75].

In order to find out whether major browsers with anti-tracking features (namely, Safari and Firefox) block the exfiltrations we uncovered, we manually analyzed ten websites, each 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 devtools interface. We found that neither Safari nor Firefox blocked email exfiltrations to tracking endpoints in our small sample. This result may be expected since both browsers try to strike a balance between minimizing breakage and curtailing cross-site tracking. To this end, they allow requests to tracker domains, but they strip cookies, partition network state [56], or block access to storage that may facilitate cross-site tracking.

Browser vendors may take further steps to protect against scripts that harvest email addresses for tracking purposes. Browsers may block requests to these trackers, prevent their scripts from accessing 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 [2]. Similar solutions are already used by different vendors: Firefox already blocks requests to third parties that use browser fingerprinting for advertisement, analytics and social network tracking [5]. DuckDuckGo's browser extension uses JavaScript stack traces to block certain tracker cookies [53]. We believe the scale of unconsented data collection uncovered in our study justifies a similar countermeasure for scripts that harvest email addresses.

Browser extensions such as uBlock Origin [16], and browsers such as Brave [24] block requests to tracker domains, which better protects against email exfiltration than countermeasures built-in to Firefox and Safari. On mobile, users may opt for browsers that support extensions (e.g., Firefox, Safari), or use a privacy-focused mobile browser that blocks trackers such as Brave [24] and DuckDuckGo [40].

Recently, Mozilla [20], Apple [18], and DuckDuckGo [19] 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 FormLock, an extension that detects and highlights forms that may leak PII. Further, to prevent PII leakage, FormLock temporarily blocks third-party requests and prevents stashing of PII into various storage mechanisms such as cookies, localStorage and indexedDB [86].

LEAKINSPECTOR Since none of the available countermeasures allow inspection of sniff and exfiltration attempts, we developed LEAKINSPECTOR, a proof-of-concept browser add-on that warns users against sniff attempts and blocks requests containing personal information.

While LEAKINSPECTOR has similarities to FormLock, it also supports detecting form sniff attempts and more precisely detects and prevents leak attempts to trackers. Further, LEAKINSPECTOR does not require user intervention, and logs technical details of the detected sniff and leak attempts to console to enable technical audits. The logged information includes the value and XPath of the sniffed element, the origin of the sniffer script, and details of the leaky request such as URL and POST data. LEAKINSPECTOR has two main features that users may enable:

Sniffer Detector When this feature is enabled, LEAKINSPECTOR detects and optionally prevents sniffing of input fields where users may enter personal information such as name, email and credit card details. We use code extracted from Firefox's autofill field detection heuristics [17] to detect such input fields.

We overwrite the getter method of the HTMLInputElement prototype to intercept input field sniff attempts. We add an event listener for `input` event to all auto-fill fields to keep track of their current values. These input field values are then used to detect leaks in outgoing requests. When a script attempts to read a monitored field's value, LEAKINSPECTOR processes the JavaScript stack trace and extract the script addresses. It then highlights the sniffed input field if there is a third-party script in the stack trace categorized as a tracker by DuckDuckGo's blocklist [9]–which we also use in Section 3.8. When determining third party scripts, LEAKINSPECTOR takes into account domain-entity relationships [12].

Leak Detector LEAKINSPECTOR intercepts HTTP requests and runs the leak detector algorithm presented in Section 3.7. It detects encoded, hashed, compressed or cleartext leaks from the monitored fields. While LEAKINSPECTOR currently only uses DuckDuckGo's blocklist [9], it is possible to extend it to use other blocklists.

LEAKINSPECTOR also features a user interface where recent sniff and leak attempts are listed, along with the tracker domain, company and tracker category. The user interface module is based on DuckDuckGo's Privacy Essentials add-on [8]. We believe LEAKINSPECTOR may help publishers and end-users to inspect third parties that harvest personal information from online forms without their knowledge and consent.

8 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, based on data from pilot crawls. However, certain limitations apply to our data collection and analysis methods.

Leak detection 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 leak detection by applying methods such as multi-stage filtering [62], and JavaScript information flow tracking [59].

Shadow DOM and crawl depth During our pilot crawls we found that we cannot detect email and password fields if they are in the Shadow DOM [28] 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. These limitations make our results likely lower bounds.

Blocklists 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.

Domain aliases Although we only consider leaks to third-party tracker domains, we also analyzed a sample of exfiltration to first-party domains. The use cases we identified included email address verification and self-hosted analytics services. Future work could investigate exfiltrations to CNAME-based trackers that appear as first parties [51].

Bypassing cookie consent banners During the manual labeling process, we encountered modal GDPR consent dialogs that disallow proceeding without giving/rejecting to give consent. A real user would have to accept or reject data processing to interact with the page; but our web crawler could have bypassed the consent dialog, depending on how it is implemented. On a random sample of $1,000$ websites, we detected 168 modal consent dialogs.

Anti-bot measures Finally, our crawler might have been served CAPTCHA pages, or treated differently due to crawling from cloud IP addresses. During a 1K website pilot crawl, we identified only three CloudFlare CAPTCHA pages that blocked our crawler.

9 Ethics Considerations

Data collection: When crawling, we took adequate measures to avoid overloading the websites. For instance, we avoided making concurrent visits to the same website.

Disclosures: We reported password leaks to both trackers and to the websites where we detected a password leak. In our emails, we provided technical details and reproduction instructions so that it is easier for the parties to reproduce and address the issue we reported. To the third parties, we sent the list of websites where they caused a password leak. To avoid any misunderstanding, we made it clear to all parties that we did not collect any visitors' email or password during our study. We did not send GDPR requests to trackers that incidentally collected passwords.

10 Conclusion

We presented a large-scale study of email and password exfiltration by online trackers before form submission. In order to address the challenges of finding and filling input fields, we integrated into our crawler a pre-trained ML classifier that detects email fields. Our results—likely lower bounds—show that on thousands of sites email addresses are collected from login, registration and newsletter subscription forms; and sent to trackers before users submit any form or give their consent. Further, we found tens of sites where passwords are incidentally collected by third parties providing session replay services. Comparing results from the EU and the US vantage points, we found that 60% more websites leaked users' emails to trackers, when visited from the US. Measuring the effect of consent choices on the exfiltration, we found their effect to be minimal. Based on our findings, users should assume that the personal information they enter into web forms may be collected by trackers—even if the form is never submitted. Considering its scale, intrusiveness and unintended side-effects, the privacy problem we investigate deserves more attention from browser vendors, privacy tool developers, and data protection agencies.

Code and Data

The source code and the dataset from our study are publicly available at https://github.com/leaky-forms.

Acknowledgments

We thank Alexei Miagkov, Arvind Narayanan, Bart Jacobs, Claudia Diaz, David Roefs, Dorine Gebbink, Galina Bulbul, Gwendal Le Grand, Konrad Dzwinel, Pete Snyder, Sergey Galich, Steve Englehardt, Vincent Toubiana, our shepherd Alexandros Kapravelos, SecWeb and USENIX Security reviewers for their valuable comments and contributions. The idea for measuring email exfiltration before form submission is initially developed with Steve Englehardt and Arvind Narayanan during an earlier study [42]. Asuman Senol was funded by the Cyber-Defence (CYD) Campus of armasuisse Science and Technology. Gunes Acar was initially supported by a postdoctoral fellowship from the Research Foundation Flanders (FWO). The study was supported by CyberSecurity Research Flanders with reference number VR20192203.

References

[1] Adding Rank Magnitude to the CrUX Report in BigQuery. https://developers.google.com/web/updates/2021/03/crux-rank-magnitude.

[2] advertisingIdentifier | Apple Developer Documentation. https://developer.apple.com/documentation/adsupport/asidentifiermanager/1614151-advertisingidentifier.

[3] Article 29 Working Party, 'Opinion 03/2013 on purpose limitation' (WP 203), 2 April 2013. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2013/wp203_en.pdf.

[4] Article 29 Working Party, 'Opinion 05/2014 on Anonymisation Techniques' (WP 216) 10 April 2014. https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf.

[5] Cookie Status: Current Status Of Browser Tracking Prevention | cookiestatus.com. https://www.cookiestatus.com.

[6] Customer URL Ticketing System. https://www.trustedsource.org.

[7] Data Services API: Endpoints. https://developer.myacxiom.com/code/api/endpoints/hashed-entity.

[8] DuckDuckGo Browser Extensions. https://github.com/duckduckgo/duckduckgo-privacy-extension.

[9] DuckDuckGo Tracker Blocklist. https://staticcdn.duckduckgo.com/trackerblocking/v2.1/tds.json.

[10] email_detector.js - Private Relay. https://github.com/mozilla/fx-private-relay/blob/v1.2.2/extension/js/email_detector.js.

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

[12] entity_map.json - DuckDuckGo Tracker Radar. https://github.com/duckduckgo/tracker-radar/blob/main/build-data/generated/entity_map.json.

[13] Evercookie - Virtually irrevocable persistent cookies. https://samy.pl/evercookie.

[14] Fathom documentation. https://mozilla.github.io/fathom/.

[15] Firefox Password Manager Module. https://searchfox.org/mozilla-central/source/toolkit/components/passwordmgr/NewPasswordModel.jsm.

[16] gorhill/uBlock: uBlock Origin - An efficient blocker for Chromium and Firefox. Fast and lean. https://github.com/gorhill/uBlock.

[17] heuristicsRegexp.js - Mozilla Autofill. https://searchfox.org/mozilla-central/source/toolkit/components/formautofill/content/heuristicsRegexp.js.

[18] Hide My Email for Sign in with Apple. https://support.apple.com/en-us/HT210425.

[19] Introducing Email Protection: The easy way to block email trackers and hide your address. https://spreadprivacy.com/introducing-email-protection-beta/.

[20] Mozilla Relay | Protect your real email address to help control your inbox. https://relay.firefox.com/.

[21] Optimise: Automatic advanced matching. https://www.facebook.com/business/m/signalshealth/optimize/automatic-advanced-matching.

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

[23] Privacy | Neustar. https://www.home.neustar/privacy.

[24] Secure, Fast & Private Web Browser with Adblocker | Brave Browser. https://brave.com/.

[25] Sending oHashes to Oracle Data Cloud platform. https://docs.oracle.com/en/cloud/saas/data-cloud/data-cloud-help-center/IntegratingBlueKaiPlatform/IDManagement/sending_ohashes.html.

[26] ShowPassword - Chrome Web Store. https://chrome.google.com/webstore/detail/showpassword/bbiclfnbhommljbjcoelobnnnibemabl.

[27] Signup Abandonment Emails Case Study: How Drip Increased Trial Signups by 15%. https://www.saasemailmarketing.net/articles/signup-abandonment-emails-increase-trial-signups/.

[28] Using shadow DOM. https://developer.mozilla.org/en-US/docs/Web/Web_Components/Using_shadow_DOM.

[29] Website Visitor Identification Software Solutions | TowerData. https://www.towerdata.com/website-visitor-identification.

[30] What is a "good" conversion rate for your signup flow? https://heap.io/blog/good-conversion-rate-signup-flow.

[31] Why Taboola? https://pubhelp.taboola.com/hc/en-us/articles/360003157074-Why-Taboola-.

[32] yahoo-connectid/sync.spec.js. https://github.com/yahoo/yahoo-connectid/blob/d0b56d47a7/src/sync.spec.js#L33-L34.

[33] whotracks.me | Data from the largest and longest measurement of online tracking. https://github.com/ghostery/whotracks.me, 2017.

[34] Court of Justice of the European Union, Case C-40/17, Fashion ID GmbH & Co. KG v Verbraucherzentrale NRW e.V., judgment of 29 July 2019 (ECLI:EU:C:2019:629). https://curia.europa.eu/juris/liste.jsf?num=C-40/17, 2019.

[35] Tracker Radar Collector. https://github.com/duckduckgo/tracker-radar-collector, 2020.

[36] About the customer matching process - Google Ads Help. https://support.google.com/google-ads/answer/7474263?hl=en, 2021.

[37] Hashing Identifiers. https://docs.liveramp.com/connect/en/hashing-identifiers.html, 2021.

[38] Uploading and Targeting a Customer File. https://help.taboola.com/hc/en-us/articles/360021908874-Uploading-and-Targeting-a-Customer-File, 2021.

[39] 6 Steps for Avoiding Online Form Abandonment. https://themanifest.com/web-design/blog/6-steps-avoid-online-form-abandonment, 2022.

[40] DuckDuckGo Privacy Browser - Apps on Google Play. https://play.google.com/store/apps/details?id=com.duckduckgo.mobile.android, 2022.

[41] Gunes Acar. Four cents to deanonymize: Companies reverse hashed email addresses. https://freedom-to-tinker.com/2018/04/09/four-cents-to-deanonymize-companies-reverse-hashed-email-addresses/, 2018.

[42] Gunes Acar, Steven Englehardt, and Arvind Narayanan. No boundaries: data exfiltration by third parties embedded on web pages. *Proceedings on Privacy Enhancing Technologies*, (4):220–238, 2020.

[43] Gunes Acar, Marc Juarez, Nick Nikiforakis, Claudia Diaz, Seda Gürses, Frank Piessens, and Bart Preneel. FPDetective: Dusting the Web for Fingerprinters. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security*, pages 1129–1140, 2013.

[44] Seyed Ali Akhavani, Jordan Jueckstock, Junhua Su, Alexandros Kapravelos, Engin Kirda, and Long Lu. Browserprint: An analysis of the impact of browser features on fingerprintability and web privacy. In *International Conference on Information Security*, pages 161–176. Springer, 2021.

[45] Ibrahim Altaweel, Maximillian Hils, and Chris Jay Hoofnagle. Privacy on adult websites. In *Altaweel et al., Privacy on Adult Websites, Workshop on Technology and Consumer Protection (ConPro)*, 2017.

[46] Rolf Bagge, Célestin Matte, Éric Daspet, Kaspar Emanuel, Sam Macbeth, and Steven Roeland. Consent-O-Matic. https://github.com/cavi-au/Consent-O-Matic/, 2019.

[47] Sai Prashanth Chandramouli, Pierre-Marie Bajan, Christopher Kruegel, Giovanni Vigna, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. Measuring E-Mail Header Injections on the World Wide Web. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pages 1647–1656, 2018.

[48] Manolis Chatzimpyrros, Konstantinos Solomos, and Sotiris Ioannidis. You Shall Not Register! Detecting Privacy Leaks Across Registration Forms. In *Computer Security*, pages 91–104. Springer, 2019.

[49] Council of European Union. EU General Data Protection Regulation (GDPR). https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[50] Ha Dao and Kensuke Fukuda. Alternative to third-party cookies: investigating persistent PII leakage-based web tracking. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 223–229, 2021.

[51] Yana Dimova, Gunes Acar, Lukasz Olejnik, Wouter Joosen, and Tom Van Goethem. The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion. *Proceedings on Privacy Enhancing Technologies*, (3):394–412, 2021.

[52] Disconnect. Disconnect Tracking Protection. https://github.com/disconnectme/disconnect-tracking-protection.

[53] DuckDuckGo. DuckDuckGo Privacy Essentials browser extension. https://github.com/duckduckgo/duckduckgo-privacy-extension/blob/bfbd47a/shared/js/content-scope/tracking-cookies-1p-protection.js#L30, 2021.

[54] Peter Eckersley. How unique is your web browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies (PETS)*, page 1–18.

[55] Steve Englehardt, Gunes Acar, and Arvind Narayanan. No boundaries for credentials: New password leaks to Mixpanel and Session Replay Companies. https://freedom-to-tinker.com/2018/02/26/, 2018.

[56] Steven Englehardt and Arthur Edelstein. Firefox 85 Cracks Down on Supercookies – Mozilla Security Blog. https://blog.mozilla.org/security/2021/01/26/supercookie-protections, 2021.

[57] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. I never signed up for this! Privacy implications of email tracking. *Proceedings on Privacy Enhancing Technologies (PETS)*, 2018(1):109–126, 2018.

[58] Steven Englehardt and Arvind Narayanan. Online Tracking: A 1-million-site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1388–1401, 2016.

[59] Daniel Hedin, Arnar Birgisson, Luciano Bello, and Andrei Sabelfeld. JSFlow: Tracking information flow in JavaScript and its APIs. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pages 1663–1671, 2014.

[60] Chris Jay Hoofnagle, Bart van der Sloot, and Frederik Zuiderveen Borgesius. The European Union general data protection regulation: what it is and what it means. *Information & Communications Technology Law*, 28(1):65–98, 2019.

[61] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. Fingerprinting the Fingerprinters: Learning to Detect Browser Fingerprinting Behaviors. In *IEEE Symposium on Security and Privacy (SP)*, pages 1143–1161, 2021.

[62] Sakshi Jain, Mobin Javed, and Vern Paxson. Towards Mining Latent Client Identifiers from Network Traffic. *Proceedings on Privacy Enhancing Technologies (PETS)*, (2):100–114, 2016.

[63] Hugo Jonker, Stefan Karsch, Benjamin Krumnow, and Marc Sleegers. Shepherd: A generic approach to automating website login. In *Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb)*, 2020.

[64] Jordan Jueckstock and Alexandros Kapravelos. VisibleV8: In-browser Monitoring of JavaScript in the Wild. In *Proceedings of the Internet Measurement Conference*, pages 393–405, 2019.

[65] Tomer Laor, Naif Mehanna, Antonin Durey, Vitaly Dyadyuk, Pierre Laperdrix, Clémentine Maurice, Yossi Oren, Romain Rouvoy, Walter Rudametkin, and Yuval Yarom. DRAWNAPART: A Device Identification Technique based on Remote GPU Fingerprinting. In *Network and Distributed System Security Symposium (NDSS)*, 2022.

[66] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web (TWEB)*, 14(2):1–33, 2020.

[67] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter 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.

[68] Adam Lerner, Anna Kornfeld Simpson, Tadayoshi Kohno, and Franziska 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*, 2016.

[69] Xu Lin, Panagiotis Ilia, and Jason Polakis. Fill in the Blanks: Empirical Analysis of the Privacy Threats of Browser Form Autofill. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 507–519, 2020.

[70] René Mahieu and Joris Van Hoboken. Fashion-ID: Introducing a phase-oriented approach to data protection? *European Law Blog*, 2019.

[71] Célestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie Banners Respect My Choice?: Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *IEEE Symposium on Security and Privacy (SP)*, pages 791–809, 2020.

[72] Surya Mattu and Kashmir Hill. Before You Hit 'Submit,' This Company Has Already Logged Your Personal Data. *Gizmodo*, 2017. https://gizmodo.com/before-you-hit-submit-this-company-has-already-logge-1795906081.

[73] Jonathan R Mayer and John C Mitchell. Third-Party Web Tracking: Policy and Technology. In *2012 IEEE Symposium on Security and Privacy*, pages 413–427. IEEE, 2012.

[74] Keaton Mowery and Hovav Shacham. Pixel Perfect: Fingerprinting Canvas in HTML5. *Proceedings of W2SP*, 2012.

[75] Nick Nguyen. Changing Our Approach to Anti-tracking – Future Releases. https://blog.mozilla.org/futurereleases/2018/08/30/changing-our-approach-to-anti-tracking.

[76] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. Dark Patterns after the GDPR: Scraping Consent Pop-Ups and Demonstrating Their Influence. In *CHI Conference on Human Factors in Computing Systems*, pages 1–13, 2020.

[77] Lukasz Olejnik, Tran Minh-Dung, and Claude Castelluccia. Selling Off Privacy at Auction. *In Network and Distributed System Security Symposium (NDSS)*, 2014.

[78] Kirsten Onsgard. TowerData's GDPR Compliance. https://www.towerdata.com/blog/towerdata-gdpr-compliance.

[79] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users. In *Proceedings of the Web Conference 2021*, pages 2130–2141, 2021.

[80] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, Phillipa Gill, et al. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *The 25th Annual Network and Distributed System Security Symposium*, 2018.

[81] Jingjing Ren, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David 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*, pages 361–374, 2016.

[82] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. Detecting and Defending Against Third-Party Tracking on the Web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, pages 155–168, 2012.

[83] Md Main Uddin Rony, Naeemul Hassan, and Mohammad Yousuf. Diving Deep into Clickbaits: Who Use Them to What Extents in Which Topics with What Effects? In *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 232–239, 2017.

[84] Shobha Doshi. CafeMedia integrates with Verizon Media ConnectID. `https://cafemedia.com/integrating-with-verizon-media-connectid/`.

[85] Oleksii Starov, Phillipa Gill, and Nick Nikiforakis. Are You Sure You Want to Contact Us? Quantifying the Leakage of PII via Website Contact Forms. *Proceedings on Privacy Enhancing Technologies (PETS)*, (1):20–33, 2016.

[86] Oleksii Starov, Phillipa Gill, and Nick Nikiforakis. FormLock. `https://github.com/ostarov/Formlock`, 2021.

[87] Steven Van Acker, Daniel Hausknecht, and Andrei Sabelfeld. Measuring Login Webpage Security. In *Proceedings of the Symposium on Applied Computing*, pages 1753–1760, 2017.

[88] Michael Veale and Frederik Zuiderveen Borgesius. Adtech and Real-Time Bidding under European Data Protection Law. *German Law Journal*, 2021.

[89] John Wilander. Intelligent Tracking Prevention. `https://webkit.org/blog/7675/intelligent-tracking-prevention`, 2017.

[90] John Wilander. Full Third-Party Cookie Blocking and More. `https://webkit.org/blog/10218/full-third-party-cookie-blocking-and-more`, 2020.

[91] Zhiju Yang and Chuan Yue. A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments. *Proceedings on Privacy Enhancing Technologies*, (2):24–44, 2020.

[92] Eric Zeng, Tadayoshi Kohno, and Franziska Roesner. Bad News: Clickbait and Deceptive Ads on News and Misinformation Websites. In *Workshop on Technology and Consumer Protection (ConPro)*, 2020.

Appendix A Supported Hash and Encoding Methods for Leak Detection

Hashes and Checksums: MD2, MD4, MD5, SHA1, SHA256, SHA224, SHA384, SHA512, SHA3 (224, 256, 384, 512-bit), MurmurHash3 (32, 64, 128-bit), RIPEMD-160, Whirlpool, Salted SHA1 (salt=`QX4QkKEU`)
Encodings: Base16, Base32, Base58, Base64, Urlencode, Entity, Deflate, Zlib, Gzip, LZstring, Custom Map (`kibp8A4EWRMKHa7gvyz1dOPt6UI5xYD3nqhVwZBXfCcFe...` `0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghi...`)

Appendix B Labeling email-collecting 3rd-party domains that are not blocked by blocklists

For each domain:

1. Is the 3rd-party domain is owned by the same entity as the first party?
    a. Yes: not tracking-related (first-party exception)
2. Did we receive any email from websites where this domain collected email addresses?
    a. Yes: tracking-related
3. Identify the company website—use the initiator script (URL, source code, copyright preamble, comments) if necessary.
    a. Is the 3rd party used for email validation (check on an example first-party site taking into account UI messages (e.g. "Invalid email") and HTTP response content (e.g., "bogus email" when we enter test@gmail.com)?
        i. Yes: not tracking-related (validation exception)
    b. Identify the business category using BuiltWith and the company website (esp. check for solutions, products, and other marketing materials). Does the business category include one of marketing, advertising, analytics?
        i. Yes: tracking-related
        ii. No: not tracking-related