User Tracking in the Post-cookie Era: How Websites Bypass GDPR Consent to Track Users

Emmanouil Papadogiannakis
FORTH/University of Crete
Greece

Nicolas Kourtellis
Telefonica Research
Spain

Panagiotis Papadopoulos
Telefonica Research
Spain

Evangelos P. Markatos
FORTH/University of Crete
Greece

ABSTRACT

During the past few years, mostly as a result of the GDPR and the CCPA, websites have started to present users with cookie consent banners. These banners are web forms where the users can state their preference and declare which cookies they would like to accept, if such option exists. Although requesting consent before storing any identifiable information is a good start towards respecting the user privacy, yet previous research has shown that websites do not always respect user choices. Furthermore, considering the ever decreasing reliance of trackers on cookies and actions browser vendors take by blocking or restricting third-party cookies, we anticipate a world where stateless tracking emerges, either because trackers or websites do not use cookies, or because users simply refuse to accept any.

In this paper, we explore whether websites use more persistent and sophisticated forms of tracking in order to track users who said they do not want cookies. Such forms of tracking include first-party ID leaking, ID synchronization, and browser fingerprinting. Our results suggest that websites do use such modern forms of tracking even before users had the opportunity to register their choice with respect to cookies. To add insult to injury, when users choose to raise their voice and reject all cookies, user tracking only intensifies. As a result, users' choices play very little role with respect to tracking: we measured that more than 75% of tracking activities happened before users had the opportunity to make a selection in the cookie consent banner, or when users chose to reject all cookies.

CCS CONCEPTS

• **Information systems** *Web applications*; • **Social and professional topics** **Governmental regulations**; **Governmental surveillance**.

KEYWORDS

User Tracking, GDPR, User Consent, Web fingerprinting

1 INTRODUCTION

Over the past few years, we have seen an increasing concern about user data protection with respect to the data of European users. This was probably the result of the General Data Protection Regulation (GDPR) which was adopted in April 2016 and came into force in May 2018. The main difference of this regulation compared to previous legislation is that it includes significant fines for companies which collect users data without the users' consent or some other legal basis. Such fines can reach up to 20 million euros, or up to 4% of the annual worldwide turnover of the preceding financial year, whichever is greater. As a result, several companies, and their associated websites, have started asking their visitors and users for their consent, before collecting (and processing) their data.

Such a consent has been usually collected via cookie banners, which ask users for consent and may give some choices as well. Indeed, users may be given the choice to accept all cookies, to accept some cookies, or even to reject all cookies. The choice is entirely up to the user, and the correct implementation of this choice is the responsibility of the website. Although this sounds completely legal and fully straightforward, deviations have been reported in literature [12, 17, 26, 40, 46]. For example, some websites claim that some cookies are absolutely necessary for their operation (e.g., for the page to be delivered) or due to legitimate interest (e.g., to improve the product), and can not be rejected by the users. Thus, users cannot really choose to reject *all* cookies: these necessary cookies cannot be rejected. Past research studies have noticed some discrepancies between what the users type and what is registered in the website. For example, the users may provide a negative response (i.e., reject all cookies), but the cookie banners may register a positive one (i.e., accept all cookies), or the cookie banners may register a positive response even before the users had the opportunity to provide any choice [26].

All these previous studies focus on cookies and compliance of cookie processing with the GDPR. In this paper, we set out to explore a slightly different question:

> *If a user does not provide consent, or chooses to **reject** all cookies, do websites use other forms of tracking to*

track this user? If so, what are these forms of tracking, and what is the extent of this tracking?

Considering the (i) ever less reliance of third-party trackers on non-permanent, erasable state-like cookies [21] and (ii) recent advances of browser vendors against third-party cookies [6, 20], it is apparent that the need for identifying how websites treat user consent in case of stateless (cookie-less) tracking is more than timely and urgent. We address this need and try to fill this exact gap in our understanding, by being the first to investigate what is the GDPR compliance across the Web in the case where websites and trackers do not use cookies, or users do not accept cookies.

Sadly, our results suggest that even when users reject all cookies, websites *do use other forms of tracking* to track users, and process personal data, in violation of GDPR. Such forms of tracking include *first-party ID leaking*, *ID synchronization* and *browser fingerprinting*.[1] First-party ID leaking and ID synchronization are used to pass an identifier (such as a cookie) as an "argument" in an HTTP request to a website - different from the website that planted this ID in the first place. In fact, according to past studies [16, 35, 36], Web entities may share IDs they have assigned to users and help third-parties re-identify users or create universal IDs. Browser fingerprinting [2, 14] uses elaborate approaches to uniquely identify a user through characteristics of her device - characteristics which can be easily found by a website. Such characteristics may include screen resolution and rendering characteristics, browser fonts and installed plugins etc. [11, 27, 30, 34]. Combining several of these characteristics can provide a large enough number of entropy bits to uniquely identify a user.

Although these cases of user identification are considered "personal data processing" according to GDPR and ePrivacy [15] regulations, and must be visible to users, they often do not appear in request forms of consent managers deployed by modern websites. In this study, we highlight exactly that: the lack of transparency and user consent when it comes to websites that deploy user identification techniques like ID synchronization and browser fingerprinting. The contributions of this work are as follows:

- We propose a fully automated method for detecting browser fingerprinting on websites using the Chromium Profiler.
- We crawl close to one million websites and record how they track users using sophisticated forms of tracking (such as first-party ID leaking, ID synchronization and browser fingerprinting) as a function of users' choices.
- We find that: (1) More than 75% of leaks happen despite the fact that users have chosen to reject all cookies; (2) Websites embedded with ID synchronizing third-parties force browsers to engage in several ID synchronizations (3.51 per ID, on average) even before users had a chance to accept or deny consent; (3) Popular websites are more likely to disregard users' consent choices and engage in first-party ID leaking and ID synchronization; (4) Browsers leak more information when users choose to reject all cookies than when they choose to take no action at all; (5) Our analysis of

---

[1]One might think that ID synchronization is a form of tracking using cookies. This is not really true: although ID synchronization does use (values stored in) cookies, passing such values around is done in an unorthodox manner, completely different from the way cookies are used.
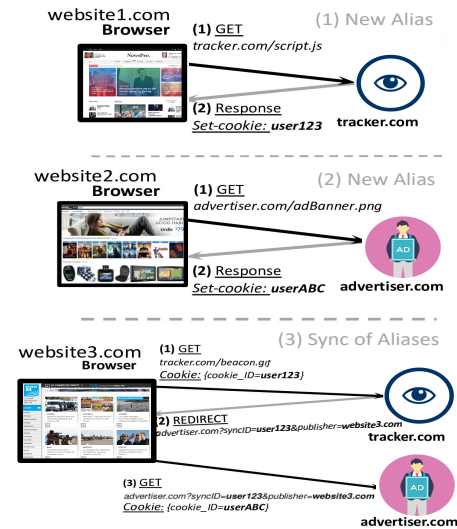

Figure 1: Example of an ID synchronization operation. Two entities match the IDs they have assigned to the same user.

tracking per country code reveals significant discrepancies across EU countries.
- Our methodology can be transformed into an auditing tool for regulators, stakeholders and privacy-policy makers, for verifying compliance with GDPR and users' privacy rights.

2 BACKGROUND

In the world of Web, cookies are used to store identifying information for a given user. However, recent policies and regulations from browser vendors and government bodies [7, 43, 48] try to control the exposure of this identifying information to third-parties and for how long. These policies restrain the ad and tracking industry that relies on re-identifying a user for long periods to serve more targeted ads. Some of the most popular techniques used by the third-parties include ID synchronization (e.g., cookie synchronization [2, 31, 35, 37]) and canvas fingerprinting [28], but also the font-based fingerprinting [11], WebRTC-based fingerprinting, AudioContext fingerprinting, and Battery API fingerprinting [14].

2.1 ID Sharing

Whenever a user visits a new website, a plethora of cookies and IDs are assigned to her, allowing first or third-parties to re-identify her across the Web and build a profile based on her browsing behavior. These profiles can be later centralized in Data Management Platforms [13], sold by data brokers [4], or used by advertisers to bid in ad auctions [33], ad-retargeting [18] and cross-device tracking [41]. For the different Web entities (e.g., publishers, analytics, data brokers, advertisers, etc.) to perform such transactions, all of the different assigned aliases (i.e., IDs) that each entity has assigned to the same user, need to be linked (i.e., synced) together. This would reveal that the user that the entity A knows as userABC is the same user that entity B knows as user123.

Figure 1 illustrates an example of how this ID synchronization takes place. Assume a user browsing website1.com and website2.com, in which there are third-parties like tracker.com
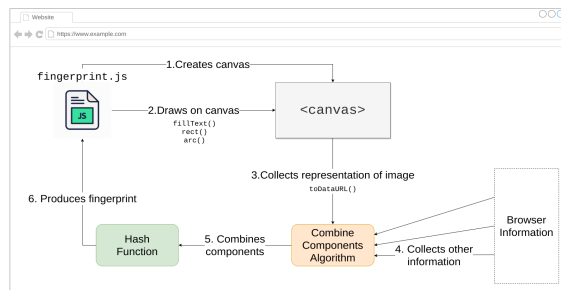
Figure 2: Canvas Fingerprinting process as part of the browser fingerprinting methodology used by popular libraries. The website can extract a fingerprint of the user's browser.

and `advertiser.com`, respectively. Consequently, these two third-parties have the chance to assign an alias to the user and re-identify them in the future. From now on, `tracker.com` knows the user with the ID `user123`, and `advertiser.com` knows the same user with the ID `userABC`. Next, assume that the user lands on `website3.com`, which includes some JavaScript code from `tracker.com` making the browser issue a GET request to `tracker.com` (step 1), who responds back with a REDIRECT request (step 2), instructing the user's browser to issue another request to its collaborator `advertiser.com` this time, using a specifically crafted URL (step 3) where the alias it uses (i.e., `user123`) is piggybacked. When `advertiser.com` receives the above request from the user it knows as `userABC`, it learns that the user whom `tracker.com` knows as `user123`, and the user `userABC` are basically the same user. This allows the two entities to join the different aliases (e.g., cookies, device IDs, user IDs, etc.) a user has on the Web.

In this paper, we study two types of ID sharing: (i) *first-party ID leaking*, where a first-party alias (e.g., a cookie or device ID) is leaked from the visited website to different third-parties, and (ii) *third-party ID synchronization*, where third-parties link together the different third-party aliases they use for the same users.

2.2 Browser Fingerprinting

Browser Fingerprinting is a sophisticated set of techniques, which can be used to uniquely identify browser instances without storing any information on the user side (stateless). It can be used to detect malicious users that create multiple accounts in social networking services, or even stop deceitful orders in e-commerce platforms. However, this technique can be abused by privacy-violating websites and, therefore, track users across sites, or even de-anonymize private sessions. In fact, previous work [23, 28] has shown that this technique provides sufficient bits of entropy to effectively track users, even through the usage of the Tor Browser.

One of the most prevalent and stealthy such fingerprinting techniques is Canvas Fingerprinting: named after the HTML canvas element, which was introduced in the latest version (i.e., HTML5). A canvas element provides the required functionality for drawing graphics using client-side code. Moreover, canvas fingerprinting relies on WebGL, a cross-platform JavaScript API that enables developers to render advanced graphics using shaders. As a result,

developers have access to rendering functionality, which is performed in a GPU, however, in an HTML context via the canvas element.

Figure 2 demonstrates the process of canvas fingerprinting as part of browser fingerprinting. Assume (i) a website that contains the fingerprinting code and (ii) a browser instance that can execute JavaScript code. As a first step, the fingerprinting script creates a canvas element using the built-in interface provided by almost all modern browsers. Next, the script renders some 2D graphics and text on the canvas. Usually, the text that is drawn is a pangram. This means, that it contains all the letters of the English alphabet in order to increase the number of entropy bits. Different font sizes and font families result in a slightly different text that can affect the final fingerprint. As a next step, the fingerprinting script needs to extract the content of the canvas and inspect its pixel values (step 3). This is achieved using the method `toDataURL()`, provided by the canvas object. This method returns the Base64 encoding of the canvas' content. Based on various factors, including fonts that are installed on the user's machine, version of OpenGL and browser's rendering engine, this string can be sufficiently different per user.

Then, the script combines this canvas fingerprint with other information, which can be used as an additional source of entropy (step 4). This information includes, among others, the host operating system and timezone, its screen resolution, installed plugins, preferred language set in the browser and number of logical processors available on the host. The output of the combination algorithm is a long string that uniquely identifies the specific browser instance (step 5). Finally, the identifier is hashed, to produce a fingerprint for this specific browser (step 6) and is usually sent across the network, or even stored as a cookie.

Tracking techniques need to be transparent to users to avoid raising suspicion or harm the user experience. As such, browser fingerprinting can be performed in minimum time on any browser that supports JavaScript by using invisible HTML elements and without requiring any privileges or permission from the user. Consequently, even privacy-aware advanced users that block cookies can be tracked. Furthermore, browser fingerprinting is difficult to prevent because it relies on native functionality, built in modern browsers. Users need to either disable JavaScript, or use external browser extensions. These techniques usually add random noise to some built-in functions, making the fingerprint different, each time the same website attempts to (re)identify a user [5, 22, 29].

3 METHODOLOGY

To investigate the effect of the different options a user is provided with while visiting websites with a consent form, we leverage the Consent-O-matic tool [1]. Consent-O-matic is the state-of-the-art browser extension to automatically detect and handle GDPR consent forms. Whenever the extension detects a Consent Management Platform (CMP), it logs its info (e.g., vendor, encoding, IDs). Additionally, it can be configured to either accept or reject the different categories of data processing purposes. In addition to this, we develop a puppeteer-based crawler that instruments a Chrome browser. By using Consent-O-matic, the browser can automatically perform one of the following three actions when a consent form is detected:
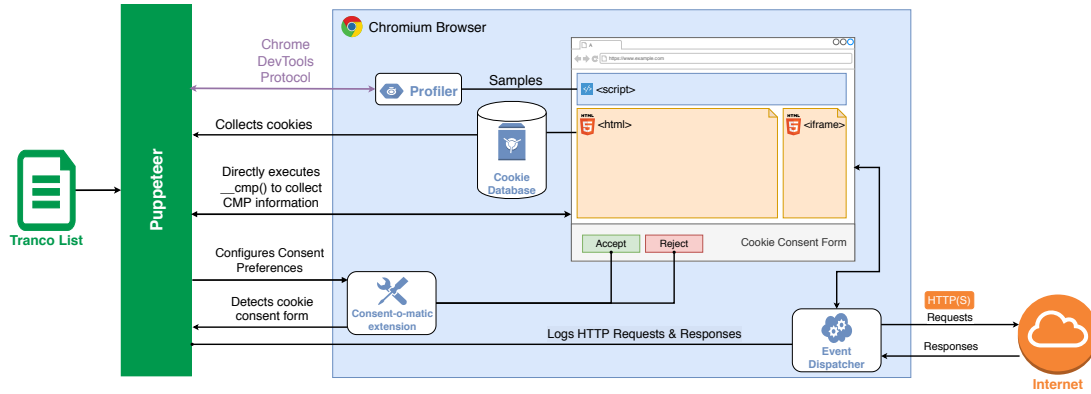
Figure 3: High level overview of our crawling methodology. We use Puppeteer to instrument a web browser and automatically visit websites. The Chrome Profiler is a built-in tool used to record and analyze run-time performance by collecting callsite information and execution statistics. The Cookie Database stores all cookies set by various domains. The Consent-O-matic tool is loaded on browser startup as an extension to handle cookie consent forms. Whenever a request is issued or a response is received, the event dispatcher emits the appropriate event, which is handled by our puppeteer-based crawler.

Table 1: Summary of our crawled dataset.

Description	Volume	% of total
Initial set of websites	850K	
Websites that errored	219,098	25.78%
Websites that were filtered out		
(pornographic or no-bots allowed)	2,689	0.32%
Total websites correctly parsed	628,213	73.90%
Websites with a CMP	27,953	3.29%
Websites with a CMP and no error		
in all three consent actions	27,180	3.20%

(1) Accept All: grant consent for all data processing purposes to all third-parties residing in the visited website.
(2) Reject All: deny consent for all data processing purposes to all third-parties residing in the visited website.
(3) No Action: avoid interacting with the form in any way.

By using our instrumented browser, we crawl (with clean state) the landing page of the top 850K sites of Tranco list [25]. This list aggregates the ranks from the lists provided by Alexa, Umbrella, and Majestic from 29.07.2020 to 27.08.2020 (pay-level domains retained)[2]. Whenever a CMP is detected, we crawl the given website 3 times (one for each of the different consent actions), and we store: the HTML, cookiejar, HTTP requests, HTTP responses, JS function calls and CMP info for each case. It is important to note that, we capture HTTP(S) requests and responses passively, via the emitted Chrome events without mutating or intercepting them. This ensures that the behavior of the website is not affected by our crawler. An overview of our crawling methodology is illustrated in Figure 3.

3.1 Data Description and Analysis

Overall, the crawler (located in EU) visited 850K sites from August 28th 2020 to September 17th 2020, the Consent-O-matic extension detected 27,953 sites with a CMP (or 4.44% of the successfully visited sites)[3], and we collected a total of 108 GB of data for these sites. Crawls failed at 25.78% of the initial set of websites (due to error,

puppeteer time-out, site inaccessibility, site did not serve EU-based users). Table 1 summarizes our dataset.

Detecting Third-party ID Synchronization: We perform an offline analysis on the collected data to detect ID synchronization operations. Specifically, we examine all application-level network traffic and search for requests that contain unique IDs. For HTTP GET requests, we inspect the URL of the requests and examine their path and parameters. For HTTP POST requests, we inspect the data stored in the request body. We report a case of ID synchronization only if a unique ID is delivered to a domain different from the one that assigned it to the user. This analysis is performed for both first-party and third-party set IDs and in a per-website base. The majority of these IDs are stored in cookies. Thus, we parse the value of each and look for strings that can be used as unique IDs. If this value is a text string representing a JSON object, we get the values stored in key-value pairs in the object[4]. If the object contains inner JSON objects, we recursively obtain all values in all nested levels.

To reduce false positives, we deliberately filter values that include consent information (e.g., values of the keys euconsent, eupubconsent, __cmpconsent and __cmpiab). As described in [26], such values can be used to share user's consent across different CMPs or third-parties present on the page. Additionally, we filter out values that are considered common and cannot be used as identifiers: strings that represent dates, timestamps, regions, locale, strings that end with a common file extension (e.g., jpg), strings that are URLs (e.g., start with www. or http://) and, finally, strings that are prevalent keywords. To construct a list of such keywords, we use a simplified puppeteer-based crawler to visit over 2.5K websites, and store all cookies. We manually inspect their values and we identify over 80 keywords that are frequently found in cookies but cannot be used for user identification. This list includes keywords such as "homepage", "undefined", "desktop", "not set" and "active", among others. We also exclude strings that have a length of 5 or

---

[2]https://tranco-list.eu/list/Q274/full
[3]Inline with related works which report detection rates of 3% [40] and 6.2% [26]. Designing a detection tool with better accuracy is very challenging due to the heterogeneity of the various existing consent management libraries and custom solutions.

[4]We purposely ignore the keys found in key-value pairs of JSON objects, since these keys rely on the API which the website uses, and do not contain any useful information that can uniquely identify users. Treating these keys as possible identifiers would result in multiple false positives.

less characters as they do not contain enough bits of entropy to uniquely identify a user. In addition, we see cookie values combining (with a delimiter) identifiers with non-identifying info (e.g., timestamp, locale, etc.), for example: *foo={userID};15693242;en-US*. We find less than 0.6% of such IDs being synced with third-parties.

The last step is to detect the possible IDs in the HTTP traffic. For each string of the previous step, we examine all HTTP requests targeting domains different than the one that set the cookie, and seek for an exact string match. We search for these possible IDs in (i) URL parameters, (ii) the body of requests and (iii) the referrer header. We tokenize the URL parameters using both default (i.e., &) and custom (i.e., ";") delimiters.

Detecting Browser Fingerprinting: As described in [28] and illustrated in Figure 2, browser fingerprinting techniques, such as canvas fingerprinting, can be performed using various native methods provided by the browser's run-time environment (e.g., `fillText`). Past work [2, 14, 24, 39] has focused on monitoring these native methods along with their returned values. By observing the sequence of function calls along with the arguments given to these functions, one can have indications of browser fingerprinting. Additionally, searching for common arguments found in popular fingerprinting libraries can help increase the level of certainty. We argue that this method produces multiple false positives, since websites which use the native methods or HTML elements, like the canvas element, legitimately, might be marked as fingerprinting websites. Indeed, in [39] manual revision of results was required in order to exclude false positives. To mitigate this, our approach focuses on a higher level of abstraction and does not examine the native (i.e., browser's built-in) methods. This way, we successfully disregard websites that use these methods legitimately (e.g., the canvas element for web graphics). Specifically, to detect browser fingerprinting, we perform JavaScript code profiling and search for specific function calls that indicate the presence of a fingerprinting library. Our method reduces the number of true positives, but ensures that the results are trustworthy. Moreover, this method can be utilized by a fully-automated crawler, without the need of manual intervention.

In particular, we analyse the open-source version of one of the most widely-used fingerprinting JavaScript libraries: FingerprintJS [47]. We extract the full list of functions used during the process of browser fingerprinting. We then focus on functions that consist of multiple operations and require a significant number of execution cycles. This ensures that they will always be sampled by the profiler. Moreover, we ignore functions that have common names (e.g., `map` or `isIE`) and functions that can be utilized by general purpose code to perform actions not necessarily related to fingerprinting (e.g., `getRegularPlugins`). As a result, we conclude that the execution of the functions `getCanvasFp`, `getWebglFp`, `Fingerprint2` and `Fingerprint2.get` indicate browser fingerprinting. These functions indicate clear intent to fingerprint the user's browser and uniquely identify them.

Next, to fully automate the detection of browser fingerprinting, we modify our puppeteer-based crawler to start with the built-in profiler tool of the Chromium browser, enabled. This was achieved using Puppeteer's ability to create a session for the Chrome DevTools protocol [44]. Additionally, we set the sampling interval
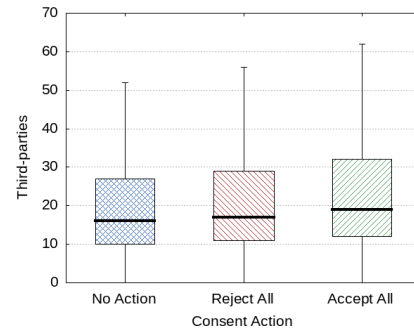

Figure 4: Number of third-parties running on the website during the three different types of visits (min, 25th percentile, median, 75th percentile, max). Surprisingly, for the median website, in the `Reject All` case there were more (i.e., 17) third-parties running than in the `No Action` case (i.e., 16).

of the profiler to 500 μs, which results to 2K samples per second. The output of the Chromium profiler is a list of profile nodes. Each node contains information about samples, in addition to a unique ID and a call frame. Using this call frame, we extract the function name along with the URL of the JavaScript script that contains the specific function. This enables us to search for fingerprinting functions, as well as identify the exact script that performs browser fingerprinting.

Limitation: Although our mechanism is fully automated, we must acknowledge that our fingerprinting detection process may miss minified or obfuscated fingerprinting scripts.

4 ANALYSIS OF CONSENT

In this section, we present our measurements and analyze the behavior of websites across three types of visits: when consent is (i) rejected (`Reject All`), (ii) granted (`Accept All`), and (iii) not responded to (`No Action`).

4.1 Cookie Consent and Third-Parties

Fist, we study how websites change their user tracking behavior depending on the consent provided (or not), via number of third-parties they interact with. Therefore, we measure the number of unique third-parties running a script on the websites of our dataset before and after a user consent action. In Figure 4, we plot the results (min, 25th percentile, median, 75th percentile, max) for the three user actions. Two-sided non-parametric Kolmogorov–Smirnov tests for the three cases demonstrated that the three distributions are statistically different, with p-value$<10^{-10}$ ($D_{(noaction, rejectall)}$=0.038, $D_{(rejectall, acceptall)}$=0.061, $D_{(acceptall, noaction)}$=0.097). As we can see in Fig. 4, in the `No Action` and in the `Accept All` case, there are 16 and 19 third-parties running in the median website, respectively. Surprisingly, in the `Reject All` case, there were more (i.e., 17) third-parties running in the median website and may reach up to 29 for the 75th percentile. This suggests that interacting with the consent manager has an impact on the number of third-parties in the visited website. Specifically, there are more third-parties running in the median website when consent was denied.

Table 2: Number of websites detected (i) leaking their first-party user IDs and (ii) having third-parties that perform synchronizations of user IDs.

Consent Action	Websites engaging in first-party ID leaking	Websites with third-party ID synchronization
No Action	14,238 (52.38%)	6,533 (24.03%)
Reject All	15,334 (56.41%)	7,123 (26.20%)
Accept All	17,764 (65.35%)	8,048 (29.61%)

Table 3: Average number of unique third-parties learning a user ID. A user's browser leaks first-party IDs to 2.14 third-parties and engages on 3.51 synchronizations per third-party ID, on average, even before the user accepted or rejected consent.

ID	No Action	Reject All	Accept All
First-party ID	2.14	2.49	3.04
Third-party ID	3.51	3.91	4.86

Table 4: Top-5 third-parties that learn the highest number of first-party IDs per consent action in our dataset.

#	No Action	Reject All	Accept All
1.	facebook.com 18.87%	facebook.com 18.29%	facebook.com 19.48%
2.	google-analytics.com 18.85%	google-analytics.com 17.28%	google-analytics.com 15.99%
3.	bing.com 9.64%	bing.com 8.84%	bing.com 10.27%
4.	hubspot.com 6.66%	doubleclick.net 6.60%	doubleclick.net 6.82%
5.	doubleclick.net 4.68%	hubspot.com 5.86%	hubspot.com 5.99%

4.2 Sharing User IDs with Third-Parties

First-party ID leaking: In our next experiment, we set out to explore the cases where a first-party ID (e.g., cookie, device ID [30]), previously set by the visited website, is getting leaked to third-parties. Therefore, we measure how many first-party ID leaking operations are being performed in a website as a function of the three aforementioned user choices. One would expect that there are *no* such operations before the user makes a choice (i.e., No Action), as well as when the users rejects all cookies (i.e., Reject All). However, as shown in Table 2, among the websites that present their users with a cookie consent banner, we found 14,238 of them to perform first-party ID leaking even before their users had the opportunity to register their preferences (No Action case). To our surprise, when users Reject All cookies, the first-party ID leaking only gets worse, with more than 15,334 of them engaging in it.

Next, we explore what is the extent of these leaks. Table 3 shows the average number of first-party ID leaking, as a function of the three user choices. There are 2.14 first-party ID leaks even before the user has the opportunity to accept cookies or not (blue bar-No Action). To make matters worse, if the user chooses to reject all cookies (red bar-Reject All), a first-party ID may be leaked to even more third-parties, on average (2.49). Furthermore, in Table 3, we measure the average number of third-parties that learn a first-party in the websites we detected this phenomenon. The difference between Reject All and Accept All is rather small: in the average website, choosing Accept All leaks first-party IDs to 3.04 third-parties, when Reject All leaks IDs to 2.49 third-parties, i.e.,

Table 5: Top-5 third-parties with highest number of third-party synchronisations per consent action in our dataset.

#	No Action	Reject All	Accept All
1.	doubleclick.net 21.15%	doubleclick.net 21.47%	doubleclick.net 20.22%
2.	everesttech.net 13.21%	everesttech.net 12.10%	everesttech.net 10.89%
3.	scorecardresearch.com 10.59%	facebook.com 9.95%	facebook.com 9.61%
4.	facebook.com 10.15%	scorecardresearch.com 9.61%	ad.gt 9.54%
5.	taboola.com 9.68%	google-analytics.com 8.30%	taboola.com 8.49%

about 25% less. The difference between the two is hardly significant, implying that more than 75% of the third-parties that will learn a first-party ID, do so without user's consent!

In Table 4, we show the top-5 third-parties in our dataset that learn the most first-party IDs across all websites for each of the three consent options. Facebook with its social plugin, Google with its analytics tracker and ad-exchange (Doubleclick) modules, and Microsoft (Bing) occupy the top positions in all three consent options.

> **Finding**: Browsers leak more information when users choose to reject all cookies than to take no action at all. In fact, more than 75% of the leaks happen despite the fact that users have chosen to reject all cookies.

Third-party ID synchronization: Apart from sharing the first-party IDs they assign to the visiting users, websites also host third-parties that (as described in Section 2.1) need to synchronize the different user IDs they use for the same users, in order to merge their databases on the back-end. This way, they can later target specific groups of users [3], sell their data [8], or use these data in ad-auctions [32, 38]. This type of leakage is worse than the first-party ID leaking, since (1) it is not in the control of the websites themselves, (2) via this mechanism, third-parties *that are not present on the website* can be alerted of a user's presence.

As shown in Table 2, from the websites that present their users with a consent manager, we found 6,533 websites hosting third-parties that conduct synchronization of IDs before users had the opportunity to register their choices (No Action). If users Reject All cookies, then even more websites (7,123) engage in ID synchronization. Although consistent with the finding of the previous subsection (first-party ID leaking), this fact sadly shows that websites employ sophisticated forms of tracking totally disregarding user consent preferences.

To quantify the extent of the phenomenon that happens as a function of the three consent choices examined, in Table 3 we measure the average number of unique third-parties synchronizing a user ID. When the user takes No Action, their browser engages in 3.51 synchronizations, on average. This means that when the user is asked for GDPR compliance, and before even responding, their browser already leaked at least one third-party ID to more than three other third-parties. To make matters worse, if the user responds negatively and chooses to Reject All cookies, their cookies may get synced with even more third-parties: 3.91, on average.

(a) Unique third-parties in ID leaking per website rank.

(b) Unique third-parties in ID leaking per website rank (normalized).

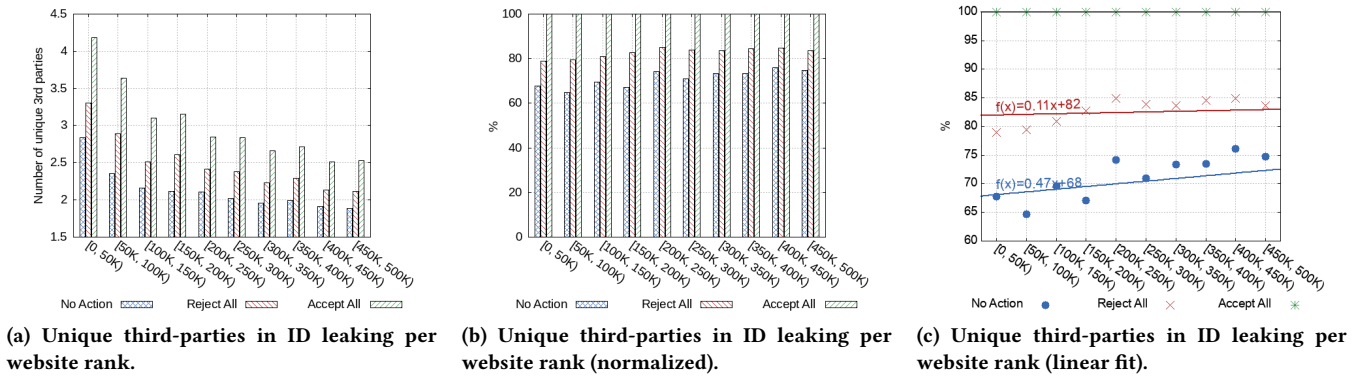**(c) Unique third-parties in ID leaking per website rank (linear fit).**

Figure 5: First-party ID leaking as a function of the website's popularity. (a) Average number of unique third-parties involved in ID leaking per rank range of the website (b) This figure plots the same information as Fig. 5(a), with the exception that all `Accept All` values are normalized to 100%. (c) This figure plots the same information as Fig. 5(b), with the exception that all `Accept All` values are normalized to 100%. `Reject All` and `No Action` points have been fitted with a straight line. The line suggests an increasing trend implying less popular sites are more aggressive at disregarding user choices.

Table 6: Websites performing browser fingerprinting.

Description	Volume	% of total
No Action	279	1.03%
Reject All	285	1.05%
Accept All	330	1.21%
In at least one consent action	336	1.24%
In all 3 cases	247	73.5%
Only in Accept All case	47	13.9%
Only in Reject All case	3	0.9%
Wait for action	7	2%

In Table 5, we show the top-5 third-parties conducting the highest number of synchronizations across websites, for each of the consent options. This time, Google's ad-exchange platform `doubleclick.net` and Amazon tracker `everestTech.net` are the top-2 in all three consent options.

> **Finding**: Websites with embedded third-parties that synchronize the IDs they have assigned for the same user, force browsers to engage in 3.51 synchronizations, on average, even before the users had any chance to accept or reject consent.

4.3 Browser Fingerprinting

In our next experiment, we set out to explore whether websites track users differently via browser fingerprinting, given the different user responses to the requested cookie consent. By using the methodology presented in Section 3.1, we detect the number of websites performing browser fingerprinting across the different types of visit. Table 6 presents our findings and, as we can see, the action of the user has no significant impact on the websites' fingerprinting operations. Specifically, 279 websites perform browser fingerprinting even before the user had the opportunity to respond to the consent request (i.e., `No Action`). Even worse, if the user chooses to `Reject All` cookies, even more websites engaged in browser fingerprinting: 285 websites. Interestingly, we see 73.5% of the fingerprinting websites perform browser fingerprinting no matter what the user consent action is. In addition,

we see that only 2% of these websites wait for user's action before starting their fingerprinting operation. Only 13.9% of them perform browser fingerprinting only when the user gives consent, and 0.9% of the websites perform browser fingerprinting *only if the user rejects giving consent*. It is apparent that these websites are using browser fingerprinting as a fallback mechanism in case they are not allowed (by the GDPR) to set a cookie on the user side. It is important to stress at this point that based on Article 4/Recital 30 [10], GDPR regards the process of user identifying information and not cookies per se.

> **Finding**: Although websites ask users for cookie consent, they do not take into account this consent when they perform browser fingerprinting.

4.4 Does website popularity matter?

In our next experiment, we explore whether a website's popularity impacts how the website respects the user's choices. For this reason, we grouped the websites into buckets based on their popularity: the first bucket contains the top 50K websites in the Tranco list, the next bucket contains the next 50K most popular sites (i.e., ranking 50K-100K), etc. In Figure 5(a), we show the extent of first-party ID leaking for the different buckets for the three cases we study: `No Action` (blue bar), `Reject All` (red bar), and `Accept All` (green bar). We see that as the popularity of the website decreases (right part of the plot), all bars tend to decrease, implying that the magnitude of tracking through first-party ID leaking decreases as well. In Figure 5(b), we normalize the values (so that the `Accept All` corresponds to the same 100% value). We see that in this case, blue bars and red bars tend to have a slightly increasing trend to the right. That is, less popular websites tend to be slightly more aggressive in disregarding user choices. For example, popular websites (0, 50K) do 67% of their first-party ID leaking before the user makes any choice, while less popular sites (400K, 450K) make 77% of their first-party ID leaking before the user makes any choice.

In Figure 5(c), we show an interpolation of the results using a straight line. In both consent actions (`No Action` and `Reject`

(a) Unique third-parties involved in ID synchronizations per website rank.

(b) Unique third-parties involved in ID synchronizations per website rank (normalized).

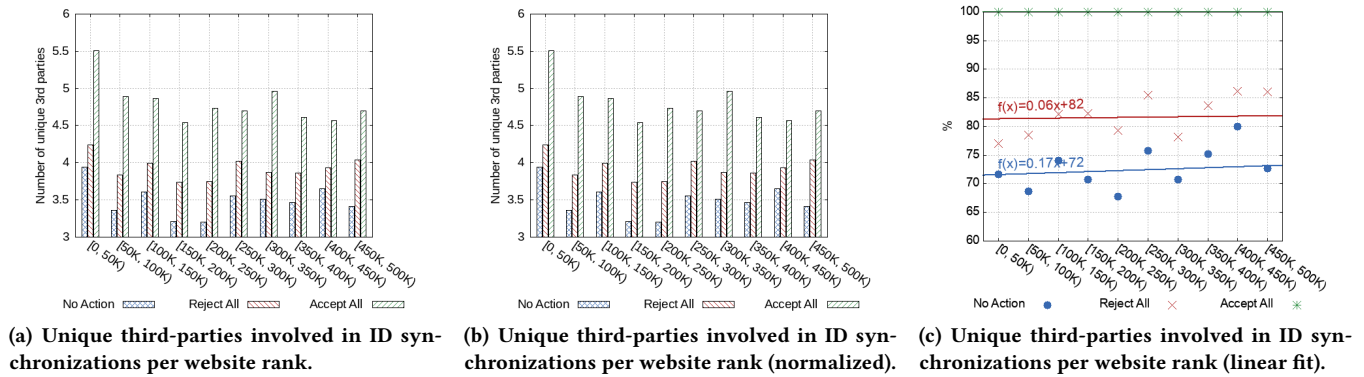(c) Unique third-parties involved in ID synchronizations per website rank (linear fit).

Figure 6: ID Synchronization as a function of the website's popularity. (a) Average number of unique third-parties involved in synchronizations per rank range of the website. (b) This figure plots the same information as Fig. 6(a), with the exception that all `Accept All` values are normalized to 100%. (c) This figure plots the same information as Fig. 6(b), with the exception that all `Accept All` values are normalized to 100%. `Reject All` and `No Action` points have been fitted with a straight line. We see that the line suggests an increasing trend implying that less popular sites are more aggressive at disregarding user choices.

All), we see a positive slope (R^2=0.42 and R^2=0.04, respectively). Similarly, in Figure 6, we see the same trend across the popularity buckets of websites hosting synchronizing third-parties: less popular websites (towards the right-end of the figures) tend to be more aggressive at disregarding user choices.

> **Finding**: Less popular websites are more aggressive at disregarding users' consent choices and engage in first-party ID leaking and third-party ID synchronizations.

4.5 Does the hosting country matter?

Next, we study how the websites hosted in different countries (represented by their country code top-level domain (ccTLD)) treat the user consent. In Figure 7(a), we present the results for the case of first-party ID leaks. As we see, a higher percent of Europe-based ccTLDs respect the choices of the users (i.e., less first-party ID leaking): websites with ccTLD=*fr* (France), *dk* (Denmark), *nl* (Netherlands), *at* (Austria) and *de* (Germany), leak first-parties to less number of third-parties than websites with non Europe-based ccTLDs (e.g., *uk* (UK), *ca* (Canada), *au* (Australia)), where the choices of the user do not have any impact.

In Figure 7(b), we normalize the results based on the `Accept All`. Websites on the right part of the figure tend to disrespect users choices: the difference between `Accept All` and `Reject All` in sites ending in *.cz* and *.ch* seems to be negligible. Thus, whether the user chooses `Accept All` or `Reject All` makes little difference. On the contrary, websites on the left part of the figure seem to respect user choices more. For example, the difference between `Accept All` and `Reject All` for *.fr* websites seem to be close to 50%. Similarly, the difference between `No Action` and `Accept All` for *.fr* websites seems to be more than 70%. Surprisingly, we see the ccTLD of *.eu* being on the right side of the figure, which means that there is an increased number of websites in this ccTLD, not yet compliant with GDPR. Thus, although not perfect, user choices for the websites on the left part of the figure have a meaningful effect, in contrast to websites on the right part.

Similarly, in Figure 8(b), we plot the same results for third-party ID synchronization. We see that, again, European ccTLDs: *.fr*, *.dk*, *.nl*, *.at* and *.de*, tend to perform less third-party ID synchronization when there is no consent given by the user, than websites with non Europe-based ccTLDs: (e.g., *uk* (UK), *ca* (Canada), *au* (Australia)), where the user choices have a much smaller impact. Surprisingly, we see two European ccTLDs, *.gr* (Greece) and *.cz* (Czech Republic), not performing like other European ccTLDs. To make matters worse, websites of *.cz* perform more synchronizations when users deny giving consent.

5 INEFFECTIVE CONSENT: EDGE CASES

In our dataset, we observed 73 websites that interact with over 100 unique third-parties each, in at least one of the three types of visit. One such example with extreme behavior is `laprovence.com`. When a user visits the website and gives `Accept All` consent, the website interacts with 159 different third-parties and performs synchronization for multiple IDs with 59 of these parties. We observed the values of 37 unique third-party cookies being leaked to third-parties different from the cookie's owner. In the `Reject All` case, the website interacts with 80 third-parties, and performs synchronization for at least one ID with 16 for them. Interestingly, when the user lands on the website with a clean session and performs `No Action`, but simply waits, the website interacts with 97 third-parties and performs synchronization with 29 of them.

Regarding first-party ID leaking, we observed that multiple websites store a cookie labeled as "necessary", but then proceed to leak its value to various third-parties. For example, `harryanddavid.com` leaks the values of 28 different first-party cookies in the `Reject All` and `No Action` cases. Also, `diariodepontevedra.es` and `asivaespana.com`, in the `Reject All` and `Accept All` cases, respectively, perform ID leaking with 38 different third-parties for more than one ID.

In addition, `camer.be` interacts with 91 unique third-parties in the `No Action` case, 94 in the `Accept All` case, and surprisingly with 131 in the `Reject All` case. For the `Reject All` case, this

(a) Number of unique third-parties learning a first-party ID.

(b) Normalized number of unique third-parties learning a first-party ID.
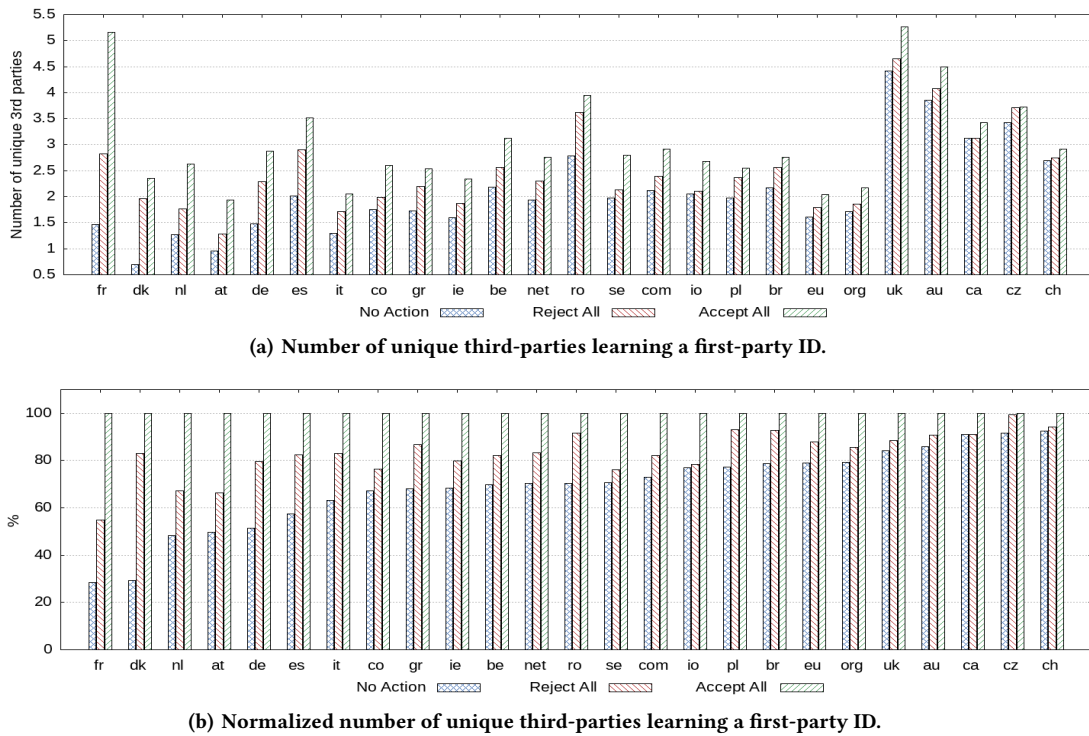
Figure 7: Number of unique third-parties learning a first-party ID as a function of the top-level domain per country code. (a) This figure plots the absolute values. (b) This figure plots the same information as in (a), with the difference that the max value (`Accept All`) is normalized to 100%. This enables us to compare websites that have different magnitudes of leakage. We see that websites in different domain names have very different behavior. For example websites in .*fr* make 1.5 leaks before the user gives consent, close to 2.8 leaks when the user rejects all cookies, and more than 5 leaks when the user accepts all cookies. On the other end of the spectrum, user choices in websites in .*cz* seem to have little impact: they leak to 3.7 third-parties both in cases when users choose to `Reject All` cookies, and in cases where users choose to `Accept All` cookies.

website is also involved in a major third-party ID synchronization operation. At the time of crawling, the website interacted with the third-party `taboola.com`, which stored a cookie with name `t_gid` and value `884d05cc-335c-4226-ab94-7ab6114fef6a-tuct65bfbc8`. This value was sent to 20 other third-parties. One interesting finding is that this cookie is stored only when the user declines consent (i.e., `Reject All`).

Similarly, `cnnturk.com` is also involved in a major third-party ID synchronization operation. Specifically, when the user lands on the website, a third-party called `lijit.com` stores the cookie `_ljtrtb_42`. The value of this cookie is then sent to 21 other third-parties. Interestingly, this behavior is observed only after the user has interacted with the cookie consent form (i.e., `Reject All` and `Accept All` cases). One example value of this cookie that we observed during the `Accept All` case is `c98d9202-8774-4e11-8c90-99d9cb879930-tuct65c0de5`, which can be used to uniquely identify a user. Note that `lijit.com` is an ad-serving domain, which can be found in multiple blacklists for tracking domains.

Finally, `glamour.com` leaks a unique identifier which is set as the value of a first-party cookie. Specifically, when a user lands on the website, a cookie called `CN_xid` is stored, with one example value being `73a4ff1f-ff45-4943-bdaa-73658b00bd42`.

Then, this value is sent to exactly 21 unique third-parties. The third-parties that receive the value of the cookie are exactly the same for all 3 types of visits. An interesting finding is that the third-parties that receive this value are not only domains known for advertising and analytics (e.g., `google-analytics.com` and `securepubads.g.doubleclick.net`), but also legitimate and mainstream websites like `vogue.com` and `wired.com`.

6 RELATED WORK

The recent increased interest of regulators and governments around the privacy rights of Internet users did not result only in rules like GDPR and California Consumer Privacy Act (CCPA), but also in an important body of research. In [17], authors investigate the legal compliance of purposes for 20K third-party cookies collected. Their findings show that purposes declared in cookie policies do not comply with the purpose specification principle in 95% of cases. In [9], authors collect cookies from the Alexa Top 100K websites and compare their cookie behavior from different vantage points, to investigate whether there are differences in cookie setting when accessing Internet services from different jurisdictions. Additionally, they study whether cookie setting behavior has changed over time by comparing today's results with a dataset from 2016.

(a) **Number of unique third-parties engaged in third-party ID synchronization.**

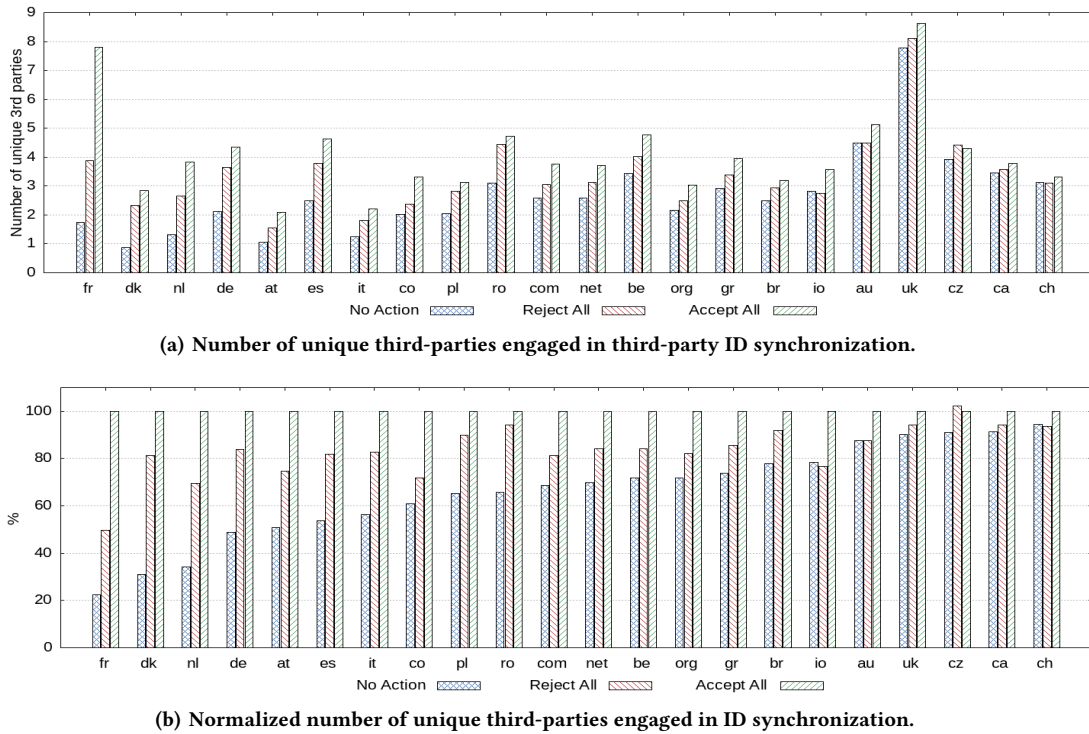(b) **Normalized number of unique third-parties engaged in ID synchronization.**

Figure 8: Number of unique third-parties engaged in third-party ID synchronization as a function of the top-level domain per country code. (a) This figure plots the absolute values. (b) This figure plots the same information as in (a), with the difference that the max value (Accept All) is normalized to 100%. This enables us to compare sites that have different magnitudes of leakage. As in first-party ID leaking, websites in *.fr* engage in less third-party ID synchronization without the user's consent. On the other end of the spectrum, user choices in sites in *.cz* seem to have little impact: they engage in approximately 4.4 third-party ID synchronizations both when users choose to Reject All cookies, and in cases where users choose to Accept All cookies.

In [40] authors perform an evaluation of the tracking performed in 2K high-traffic websites, hosted both inside and outside the EU. Specifically, they evaluate the information presented to users and the actual tracking implemented through cookies. Their results show that the GDPR has impacted website behavior in a truly global way. US-based websites behave similarly to EU-based ones, while third-party opt-out services reduce the amount of tracking, even for websites which do not put any effort in respecting GDPR. On the other hand, they show that cookies can identify users when visiting more than 90% of the websites they crawled, and they encountered a large number of websites that present deceiving information, making it it very difficult, if at all possible, for users to avoid being tracked. Similar to this work, in [12], authors crawl 1.5K EU, US, and Canadian websites from 18 countries and analyze the cookie notices they find. Using a series of regression models, they find that a website's Top Level Domain explains a substantial portion of the variance in cookie notice metrics, but the users vantage point does not, which means that websites follow one set of privacy rules for all their users.

In [46], authors study the common properties of the graphical user interface of consent notices and conduct three experiments with more than 80K unique users on a German website, to investigate the influence of notice position, type of choice, and content

framing on consent. Their results show that (i) users are more likely to interact with a notice shown in the lower left part of the screen, (ii) users are willing to accept tracking compared to mechanisms that require them to allow cookie use for each category or company individually, (iii) the wide-spread practice of nudging has a large effect on the choices users make. In [45] authors study the impact of the legislation on cookie syncing between third-parties. They show that the general structure of how the entities are arranged is not affected by the GDPR, but the new regulation has a statistically significant impact on the number of connections that shrunk by 40% in the GDPR era.

In an effort closest to ours, Matte et al. analyzed the GDPR and ePrivacy Directive across 23K European websites to identify legal violations in implementations of cookie banners based on the storage of consent [26]. That is, they (i) capture the user's choice (consent or not), (ii) measure whether the websites register the same response as the user's choice, (iii) measure whether websites register any response *before* the users click their preference. They found that: 141 websites register positive consent even if the user has not made their choice; 236 websites nudge the users towards accepting consent by pre-selecting options; and 27 websites store a positive consent even if the user has explicitly opted out. Performing extensive tests on 560 websites, they found at least one violation

in 54% of them. Although our work and [26] share similar goals, they clearly have significant differences. First, although [26] focuses on cookies as the main tracking mechanism, in this work, we focus on post-cookie tracking mechanisms including browser fingerprinting, ID leaking, and ID synchronization. In this aspect, we explore whether sites use such post-cookie tracking mechanisms to bypass any consent the user has provided for cookies. Second, [26] focuses on whether the Cookie Management Provider registers the same response as the user's input. We follow a different methodology and measure *not* the response registered, but the actual tracking mechanisms that are activated when the users access a website.

7 DISCUSSION

GDPR compliance: One question that comes to mind is whether these websites are in violation of the GDPR and the ePrivacy Directive. Obviously, one cannot make such an umbrella statement for all the websites studied in this paper. Such violations should be studied on a case-by-case basis. Even further, each website is different, and may have a legal basis to collect user data that goes beyond the user consent. What we identify in this paper is a *disparity* between (i) what the users perceive about the collection of their data, and (ii) what some websites implement with respect to data processing. Indeed, by being shown a cookie consent banner, users perceive that they are being asked to give their permission to the website to collect and process their data. Even further, when they are given several choices, users feel that they are empowered to give a fine-grain permission, which will obviously be taken into account.

Unfortunately, this perception of the users is completely different from what various websites implement. In this paper, we saw that several websites collect (and share with third-parties) information about their users, even before the users had the opportunity to register their preference. Even worse, when the users said that they would like to reject all cookies, collection of their data even intensified. Indeed, each website is ultimately responsible for the consent asked from their visitor. However, it is not obvious if the legal responsibility is shifted to the Consent Management Platform (CMP). Nonetheless, and considering our results, it is hard to believe that all these publishers do not respect the users' consent choices without intention (e.g., due to software bug, bad developer practices or wrong integration with their CMP).

Interestingly, existing literature, websites and blog-posts around the GDPR and changes it brought on the Internet and user tracking [42], focus solely on how identifiable information stored in cookies is maintained. However, as highlighted here, the GDPR is not only about cookies. Instead, we aim to increase user awareness regarding the GDPR (non)compliance of deployed stateless (i.e., cookie-less) tracking, and influence a change in language used in consent request statements, to be GDPR-compliant and reflect closely what the websites do in reality, in comparison to what is explained to the user.

Furthermore, our analysis of tracking per country code reveals significant discrepancies across EU (or not) countries. These results highlight the lack of effort from specific local governments regarding the digital privacy rights of their citizens. Our results can motivate them to take action and increase the GDPR enforcement

to make websites hosted in their countries aligned with the rest of EU countries with respect to the GDPR compliance.

Inbound vs. Outbound Information: Although user tracking without user consent is generally undesirable, in this paper, we studied some sophisticated approaches to user tracking (such as first-party ID leaking and ID synchronization) which involved not only data collection, but also data sharing with third-parties. Indeed, both approaches, provide to third-parties identifiers associated with the current user. In this way, third-parties will be able to know that this user has visited the specific website (even if they are not embedded in that website). And this happens even before the user has given any permission for data collection on the cookie consent banner. To put it simply: the website has already told third-parties that this user has just visited, while the user still makes up their mind whether to give consent for data collection or not. Thus, the user is asked for consenting to something that has already happened and it will keep on happening even if the user denies consent.

Edge Cases: Someone could argue that the edge-cases studied in this paper are momentary, and cannot be held against websites as proof of non-GDPR compliance. However, even though we acknowledge the dynamicity of websites, we made a best effort to provide results that were repeatable across multiple crawls. In fact, changes in third-parties embedded in a website could change their intensity of tracking. We anticipate such changes are transient and infrequent in websites, and that high intensity of tracking is repeatable.

Methodology: The methodology we presented in this paper can be transformed into an auditing tool for regulators, stakeholders and privacy-policy makers, for verifying compliance with the GDPR, ePrivacy Directive, and users' privacy rights. Our approach links together the (i) requested user consent of webmaster with (ii) actions taken by the website based on the particular consent given. Apart from these entities, browser vendors have already shown interest in blocking bad policies on websites [6, 19, 20] and our methodology can help towards exactly these goals. Specifically, by following our methodology, browser vendors can detect at run-time stateless device fingerprinting attempts [5] and compare these actions with given user consent.

8 CONCLUSION

Over the past couple of years, an increasing number of websites have started to present users with cookie consent banners: pop-up windows that ask for user's permission to send/receive cookies. Such banners provide a variety of choices including (i) accept all, (ii) reject all, and (iii) accept some cookies. In this paper, we study whether these websites that present users with cookie consent banners, track their users using "non cookie" approaches including first-party ID leaking, third-party ID synchronization and browser fingerprinting.

In our experiments, we found 15,334 websites that track their users using first-party ID leaking. Even further, this tracking happened despite the fact that users of these websites had rejected all cookies in the cookie consent banner! In fact, most of these websites (14,238) had started the first-party ID leaking tracking even before the users had any opportunity to register their consent choice.

Therefore, we highlight a gap between what users expect to happen when they see a cookie consent banner and what several websites do as a result of users' choices. We feel that research like this helps increase transparency on the Web and expose websites which do not correspond to users' expectations, and are non-GDPR compliant.

Future work could focus on even harder questions such as: How should third-parties connect into CMP prompts? Is it intentional that some third-parties only take action on "reject all" option? If yes, why? Are some CMPs better than others with respect to GDPR compliance? Are all these privacy violations the website's, the CMP's, or the third-party's fault?

ACKNOWLEDGMENTS

REFERENCES

[1] Aarhus Universitet. 2019. Consent-O-Matic. https://github.com/cavi-au/Consent-O-Matic.
[2] Gunes Acar, Christian Eubank, Steven Englehardt, Marc Juarez, Arvind Narayanan, and Claudia Diaz. 2014. The web never forgets: Persistent tracking mechanisms in the wild. In *ACM CCS*.
[3] Pushkal Agarwal, Sagar Joglekar, Panagiotis Papadopoulos, Nishanth Sastry, and Nicolas Kourtellis. 2020. Stop tracking me bro! differential tracking of user demographics on hyper-partisan websites. In *WWW*.
[4] Madhumita Murgia Aliya Ram. 2019. Data brokers: regulators try to rein in the 'privacy deathstars'. https://www.ft.com/content/f1590694-fe68-11e8-aebf-99e208d3e521.
[5] Brave Browser. 2020. What's Brave Done For My Privacy Lately? Episode #3: Fingerprint Randomization. https://brave.com/privacy-updates-3/.
[6] Cookiebot. 2021. Google ending third-party cookies in Chrome. https://www.cookiebot.com/en/google-third-party-cookies/.
[7] CookiePro. 2011. The Cookie Law Explained. https://www.cookielaw.org/the-cookie-law/.
[8] Joseph Cox. 2020. Leaked Documents Expose the Secretive Market for Your Web Browsing Data. https://www.vice.com/en/article/qjdkq7/avast-antivirus-sells-user-browsing-data-investigation.
[9] A. Dabrowski, G. Merzdovnik, J. Ullrich, G. Sendera, and E. Weippl. 2019. Measuring cookies and web privacy in a post-gdpr world. In *PAM*.
[10] DPO LLC. 2020. Article 4 - Recital 30. http://gdpr-text.com/read/article-4/.
[11] Peter Eckersley. 2010. How unique is your web browser?. In *PETS*.
[12] Rob van Eijk, Hadi Asghari, Philipp Winter, and Arvind Narayanan. 2019. The impact of user location on cookie notices (inside and outside of the European union). In *Workshop on Technology and Consumer Protection (ConPro'19)*.
[13] Hazem Elmeleegy, Yinan Li, Yan Qi, Peter Wilmot, Mingxi Wu, Santanu Kolay, Ali Dasdan, and Songting Chen. 2013. Overview of Turn Data Management Platform for Digital Advertising. (2013).
[14] Steven Englehardt and Arvind Narayanan. 2016. Online tracking: A 1-million-site measurement and analysis. In *ACM CCS*.
[15] European Commission. 2017. Proposal for a Regulation on Privacy and Electronic Communications. https://ec.europa.eu/digital-single-market/en/news/proposal-regulation-privacy-and-electronic-communications.
[16] Marjan Falahrastegar, Hamed Haddadi, Steve Uhlig, and Richard Mortier. 2016. Tracking personal identifiers across the web. In *PAM*.
[17] Imane Fouad, Cristiana Santos, Feras Al Kassar, Nataliia Bielova, and Stefano Calzavara. 2020. On Compliance of Cookie Purposes with the Purpose Specification Principle. In *IWPE 2020*.
[18] Costas Iordanou, Nicolas Kourtellis, Juan Miguel Carrascosa, Claudio Soriente, Ruben Cuevas, and Nikolaos Laoutaris. 2019. Beyond Content Analysis: Detecting Targeted Ads via Distributed Counting. In *ACM CoNEXT* (Orlando, Florida).
[19] John E Dunn. 2020. Google Chrome to start blocking downloads served via HTTP. https://nakedsecurity.sophos.com/2020/02/10/google-chrome-to-start-blocking-downloads-served-via-http/.
[20] John Wilander. 2019. Intelligent Tracking Prevention 2.3. https://webkit.org/blog/9521/intelligent-tracking-prevention-2-3/.
[21] Jonathan Mervis. 2020. Is Cookieless Tracking the Future of Web Analytics? www.amazeemetrics.com/en/blog/is-cookieless-tracking-the-future-of-web-analytics/.
[22] Pierre Laperdrix, Benoit Baudry, and Vikas Mishra. 2017. FPRandom: Randomizing core browser objects to break advanced device fingerprinting techniques. In *International Symposium on Engineering Secure Software and Systems*.
[23] Pierre Laperdrix, Walter Rudametkin, and Benoit Baudry. 2016. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *IEEE S&P*.
[24] Hoan Le, Federico Fallace, and Pere Barlet-Ros. 2017. Towards accurate detection of obfuscated web tracking. In *IEEE M&N*.
[25] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *NDSS*.
[26] C. Matte, N. Bielova, and C. Santos. 2020. Do Cookie Banners Respect my Choice? : Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *IEEE S&P*.
[27] Jonathan R Mayer and John C Mitchell. 2012. Third-party web tracking: Policy and technology. In *IEEE S&P*.
[28] Keaton Mowery and Hovav Shacham. 2012. Pixel perfect: Fingerprinting canvas in HTML5. *Proceedings of W2SP* (2012).
[29] Nick Nikiforakis, Wouter Joosen, and Benjamin Livshits. 2015. Privaricator: Deceiving fingerprinters with little white lies. In *WWW*.
[30] Nick Nikiforakis, Alexandros Kapravelos, Wouter Joosen, Christopher Kruegel, Frank Piessens, and Giovanni Vigna. 2013. Cookieless monster: Exploring the ecosystem of web-based device fingerprinting. In *IEEE S&P*.
[31] Lukasz Olejnik, Tran Minh-Dung, and Claude Castelluccia. 2013. Selling off privacy at auction. (2013).
[32] Michalis Pachilakis, Panagiotis Papadopoulos, Nikolaos Laoutaris, Evangelos P Markatos, and Nicolas Kourtellis. 2019. Measuring ad value without bankrupting user privacy. *arXiv preprint arXiv:1907.10331* (2019).
[33] Michalis Pachilakis, Panagiotis Papadopoulos, Evangelos P Markatos, and Nicolas Kourtellis. 2019. No more chasing waterfalls: a measurement study of the header bidding ad-ecosystem. In *IMC*.
[34] Elias P Papadopoulos, Michalis Diamantaris, Panagiotis Papadopoulos, Thanasis Petsas, Sotiris Ioannidis, and Evangelos P Markatos. 2017. The long-standing privacy debate: Mobile websites vs mobile apps. In *ACM WWW*.
[35] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos Markatos. 2019. Cookie synchronization: Everything you always wanted to know but were afraid to ask. In *WWW*.
[36] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2018. The Cost of Digital Advertisement: Comparing User and Advertiser Views. In *WWW*.
[37] Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2018. Exclusive: How the (Synced) Cookie Monster Breached My Encrypted VPN Session. In *EuroSec*.
[38] Panagiotis Papadopoulos, Nicolas Kourtellis, Pablo Rodriguez Rodriguez, and Nikolaos Laoutaris. 2017. If You Are Not Paying for It, You Are the Product: How Much Do Advertisers Pay to Reach You?. In *IMC*.
[39] Philip Raschke and Axel Küpper. 2018. Uncovering Canvas Fingerprinting in Real-Time and Analyzing ist Usage for Web-Tracking. In *Workshops der INFORMATIK 2018-Architekturen, Prozesse, Sicherheit und Nachhaltigkeit*.
[40] Iskander Sanchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. 2019. Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control. In *ACM Asia CCS*.
[41] Konstantinos Solomos, Panagiotis Ilia, Sotiris Ioannidis, and Nicolas Kourtellis. 2019. TALON: An Automated Framework for Cross-Device Tracking Detection. In *USENIX RAID*.
[42] Konstantinos Solomos, Panagiotis Ilia, Sotiris Ioannidis, and Nicolas Kourtellis. 2020. Clash of the Trackers: Measuring the Evolution of the Online Tracking Ecosystem. ACM/IFIP TMA.
[43] Nick Statt. 2020. Apple updates Safari's anti-tracking tech with full third-party cookie blocking. https://www.theverge.com/2020/3/24/21192830/apple-safari-intelligent-tracking-privacy-full-third-party-cookie-blocking.
[44] The Chromium Authors. 2014. Chrome DevTools Protocol. https://chromedevtools.github.io/devtools-protocol/.
[45] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2018. The unwanted sharing economy: An analysis of cookie syncing and user transparency under GDPR. *arXiv preprint arXiv:1811.08660* (2018).
[46] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (Un) informed Consent: Studying GDPR Consent Notices in the Field. In *CCS*.
[47] Alexey Verkhovsky, Artem Kuznetsov, Surgie Finesse, Dmitry Ivanov, Konstantin Darutkin, and Valentin V. 2021. FingerprintJS. https://github.com/fingerprintjs/fingerprintjs.
[48] World Wide Web Consortium (W3C). 2010. Same-origin policy. https://www.w3.org/Security/wiki/Same_origin_policy.