usenix
THE ADVANCED
COMPUTING SYSTEMS
ASSOCIATION

Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps

Trung Tin Nguyen, *CISPA Helmholtz Center for Information Security; Saarbrücken Graduate School of Computer Science, Saarland University;* Michael Backes, Ninja Marnau, and Ben Stock, *CISPA Helmholtz Center for Information Security*

https://www.usenix.org/conference/usenixsecurity21/presentation/nguyen

This paper is included in the Proceedings of the
30th USENIX Security Symposium.

August 11–13, 2021

978-1-939133-24-3

Share First, Ask Later (or Never?)
Studying Violations of GDPR's Explicit Consent in Android Apps

Trung Tin Nguyen[§*], Michael Backes[§], Ninja Marnau[§], and Ben Stock[§]
[§] CISPA Helmholtz Center for Information Security
[*] Saarbrücken Graduate School of Computer Science, Saarland University
{tin.nguyen,backes,marnau,stock}@cispa.de

Abstract

Since the General Data Protection Regulation (GDPR) went into effect in May 2018, online services are required to obtain users' explicit consent before sharing users' personal data with third parties that use the data for their own purposes. While violations of this legal basis on the Web have been studied in-depth, the community lacks insight into such violations in the mobile ecosystem.

We perform the first large-scale measurement on Android apps in the wild to understand the current state of the violation of GDPR's explicit consent. Specifically, we build a semi-automated pipeline to detect data sent out to the Internet without prior consent and apply it to a set of 86,163 Android apps. Based on the domains that receive data protected under the GDPR without prior consent, we collaborate with a legal scholar to assess if these contacted domains are third-party data controllers. Doing so, we find 24,838 apps send personal data towards data controllers without the user's explicit prior consent. To understand the reasons behind this, we run a notification campaign to inform affected developers and gather insights from their responses. We then conduct an in-depth analysis of violating apps as well as the corresponding third parties' documentation and privacy policies. Based on the responses and our analysis of available documentation, we derive concrete recommendations for all involved entities in the ecosystem to allow data subjects to exercise their fundamental rights and freedoms.

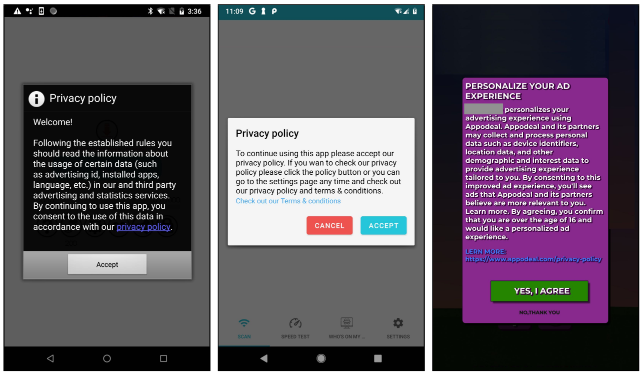
1 Introduction

Increasing data collection and tracking consumers by today's online advertising industry is becoming a major problem for individuals' rights regarding their personal data (e.g., users are secretly tracked and profiled [29, 43, 45]). To protect user privacy, regulatory efforts around the globe such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA) have been made in recent years [14, 54] — which mandate online services to disclose *transparently* how they handle personal data and grant users crucial data protection rights.

In mobile apps, researchers have analyzed the app privacy policies to identify legislation violations, i.e., determining whether an app's behavior is consistent with what is declared in the privacy policy [4, 56, 58, 72, 73]. However, irrespective of a privacy policy, under the GDPR [54], to be legally compliant, an app is required to obtain users' explicit consent before sharing personal data with third parties if such parties use the data for their own purposes (e.g., personalized advertising) [21]. The GDPR requires the consent to be *freely given, specific, informed, and unambiguous* (Section 2.2). That is, personal data transfer must only occur after the user has actively agreed (e.g., by clicking accept), i.e., "consent" packaged in terms and conditions or privacy policies is not compliant [29].

While many researchers have worked to detect and analyze consent notices (i.e., cookie banners) and their impact on the Web advertising and tracking industry after the GDPR went into effect [17, 37, 43, 57, 61, 62, 65], the community lacks insight into such violations in the mobile ecosystem. Recently, Weir et al. [69] surveyed app developers and observed that most developers' changes were cosmetic due to the GDPR legislation (e.g., adding dialogues) — which raises a serious question about whether these changes fulfill the legal conditions for collecting valid consents. Figure 1 shows examples of consent dialogues that mobile users in the European Union observe on many apps they use today. Notably, neither (a) nor (b) are valid consent dialogues required before data sharing with third parties, and even dialogue (c) is meaningless if data sharing occurs before the user has the ability to reject this.

To understand how prevalent violations of GDPR's *explicit* consent requirement are in the wild, we conduct a study with 86,613 Android apps available through the German Play Store, allowing us to provide a comprehensive overview of the current state of the violation of GDPR's explicit consent on mobile apps in the wild. Specifically, we first build a semi-automated and scalable pipeline to detect personal data sent to the Internet by analyzing the network traffic generated



Privacy policy

Welcome!

Following the established rules you should read the information about the usage of certain data (such as advertising id, installed apps, language, etc.) in our and third party advertising and statistics services. By continuing to use this app, you consent to the use of this data in accordance with our privacy policy.

(a)

Privacy policy

To continue using this app please accept our privacy policy. If you wan to check our privacy policy please click the policy button or you can go to the settings page any time and check out our privacy policy and terms & conditions.

Check out our Terms & conditions

(b)

PERSONALIZE YOUR AD EXPERIENCE

[redacted] personalizes your advertising experience using Appodeal. Appodeal and its partners may collect and process personal data such as device identifiers, location data, and other demographic and interest data to provide advertising experience tailored to you. By consenting to this improved ad experience, you'll see ads that Appodeal and its partners believe are more relevant to you. Learn more. By agreeing, you confirm that you are over the age of 16 and would like a personalized ad experience.

LERN MORE: https://www.appodeal.com/privacy-policy

(c)

Figure 1: Example of consent dialogues in Android apps.

by apps without user explicit prior consent and apply this to our dataset, which consists of both high-profile and long-tail apps. Based on the domains that receive data protected under the GDPR without prior consent, we collaborate with a legal scholar to assess the extent to which contacted domains are third-party data controllers — which require explicit consent.

Doing so, we find 24,838 apps sent personal data towards advertisement providers that act as data controllers without the user's explicit prior consent. To inform developers about these issues and understand the reasons behind them, we run a notification campaign to contact 11,914 affected developers and gather insights from 448 responses to our notifications. Inspired by the responses, we conduct an in-depth analysis of available documentation and default data collection settings of third-party SDKs. Based on the insights from both developers and our own analysis, we find that GDPR issues are widespread, often misunderstood, and require effort from advertisement providers, app stores, and developers alike to mitigate the problems. In summary, our paper makes the following contributions:

- We build a semi-automated and scalable solution (which is publicly available at [1]) to detect personal data sent to the Internet by analyzing the network traffic generated by apps without user explicit prior consent (Section 3).
- We perform a large-scale measurement on the mobile apps in the wild to understand the current state of the violation of GDPR's explicit consent (Section 4).
- We run a notification campaign to inform affected developers and gather insights from their responses. We then conduct an in-depth analysis of violating apps and the corresponding third parties' documentation (Section 5).
- We derive concrete recommendations to all concerned parties and make an urgent call to help developers comply with GDPR (Section 6).

2 Research Questions and Legal Background

Our work focuses on the violation of GDPR's explicit consent requirement in the realm of Android apps available through the European Play Store (i.e., from Germany). In the following, we briefly outline prior work in the area of GDPR and related privacy legislation, as well as more general privacy analyses for mobile apps. We subsequently present the legal background on GDPR and present our research questions.

2.1 Context of our Work

In recent years, many researchers have started to study the impact of GDPR on the online advertising and tracking industry and proposed different techniques to detect legislation violations. A related line of work aims to study the consent notices in the Web ecosystem, which are usually presented in cookie banners. Researchers have shown that many websites potentially violate the GDPR consent requirements, such not allowing users to refuse data collection or installing tracking and profiling cookies before the user gives explicit consent [17, 37, 43, 57, 61, 62, 65]. While many researchers have worked to detect and analyze consent notices and their impact on the Web advertising and tracking industry after the GDPR, no study has measured the GDPR violations of explicit consent on mobile apps. For mobile apps, researchers mostly focused on analyzing the app privacy policies to identify legislation violations, i.e., determining whether an app's behavior is consistent with what is declared in the app privacy policy [4, 56, 58, 72, 73].

Researchers have proposed different techniques to detect privacy violations by mobile apps and identify third-party advertising and tracking services. Many techniques have relied on the static program analysis of app binary code to detect malicious behaviors and privacy leaks [7, 8, 45, 48] as well as third-party library use [9, 40, 42]. While the static analysis techniques are well known for producing high false positives (e.g., do not produce actual measurements of privacy violations) [12, 39, 67], the dynamic analysis shows precisely how the app and system behave during the test (i.e., by running the app and auditing its runtime behavior) [10, 56, 70, 71]. However, an effective dynamic analysis requires building an instrumentation framework for possible behaviors of interest, which involves extensive engineering effort [53]. Another line of work aims to inspect network communications to identify third-party advertising and tracking services and privacy leaks [13, 25, 36, 52, 55] — which is closely related to our work. However, while prior works primarily focused on data protected by OS permissions (e.g., GPS data), we further detect potential unique identifiers which could be used to track an individual (Section 3.2.2). We believe our work is an important first step in understanding the magnitude of violations of GDPR consent requirements and potential causes, and can spark further research into addressing these problems.

Research Questions Orthogonal to prior work, we aim to understand how often GDPR's explicit consent mandate is violated in the mobile ecosystem, focusing on Android. To that end, we derive a semi-automated system that allows us to detect apps which sent out users' personal data without prior consent. By further analyzing the parties involved in receiving such data, this allows us to determine which parties act as *data controllers*, which require explicit consent, including specific explanations of what the data is used for. Specifically, our research aims at answering the following research questions:

- *RQ1: How many apps send out personal data without any prior consent?* By developing a semi-automated system to tackle this question, we analyze a dataset of 86,163 apps to detect to which hosts the apps send data without any prior explicit consent from the user.
- *RQ2: Of the apps which send out any data, how many send it towards parties that act as data controllers under the GDPR?* By analyzing the legal documents provided by third-party vendors, we determine which of them unequivocally must be considered data controllers, allowing us to reason about GDPR consent violations.
- *RQ3: Are developers aware of the requirements of GDPR and the issues that might arise from not following the outlined laws?* To answer this, we notify affected developers, provide details on which parties their apps contacted without prior consent, and survey the issues they face in integrating third-party SDKs in a GDPR-compliant way.

2.2 Legal Background

In this work, the GDPR is used as the base for our legal analysis. The GDPR governs all processing of personal data related to individuals situated in the EU and EEA. Additionally, the ePrivacy Directive applies to how third parties gather consent to accessing information stored on the consumers' device (also known as "cookie law"), but this is outside our scope.

2.2.1 Definition of Personal Data

Under GDPR's Article 4 [30], "**personal data**" (referred to as "PD") means any information relating to an identified or identifiable natural person ("**data subject**"). This definition includes unique identification numbers, which may include Advertising IDs, location data, and online identifiers (such as IP addresses) — when they can be used to identify users over a long period across different apps and services [4].

The definition of personal data under the GDPR is much broader than personal identifiable data (PII) under US laws. Instead of only including directly identifying data, GDPR also considers personal data such data that can be used alone or in combination to single out an individual in a data set. The EU Court of Justice has already declared that even dynamic IP addresses may be considered personal data in its Breyer v. Germany ruling [2].

Android's Advertising ID (AAID) is an interesting subject for the courts, which lacks a ruling as of yet. Google describes the ID as "*a unique, user-resettable ID for advertising, provided by Google Play services. [...]. It enables users to reset their identifier or opt-out of personalized ads*" [34]. While Google itself remained vague on characterisation of the AAID as personal data, the IAB Europe GDPR Implementation Working Group already established in their 2017 Working Paper on personal data that "*Cookies and other device and online identifiers (IP addresses, IDFA, AAID, etc.) are explicitly called out as examples of personal data under the GDPR*" [35]. In May 2020, NOYB – European Center for Digital Rights [46], a European not-for-profit privacy advocacy group, lodged a formal complaint over the AAID with Austria's data protection authority. The complaint states that although the AAID is personal data, Google does not adhere to the requirements of valid consent. Android users have no option to deactivate or delete the tracking ID, only to reset it to a new one. Furthermore, even Google's own brand Admob explicitly lists the AAID as personal data in their documentation about the User Messaging Platform used to deliver their ads [33]. Meanwhile, Apple has recently taken actions for mandatory prior consent for sharing of Advertising Identifiers for its iOS 14 update [6], clarifying that even dynamic advertising identifiers are considered personal data.

2.2.2 Legal Basis for Processing of Personal Data

Under the GDPR, all processing of European residents' personal data has to have a legal justification. App developers (first parties) process user data in order to provide the app's functionalities and services. By deciding on the means and purposes for processing the user's personal data, they act as **data controllers**, the legal role that is the responsible party for data processing. Parties external to this app developer (third parties) that also receive the user's data could act in two possible capacities. If they act purely on behalf of the first party with no data use for their own purposes and under the complete control of the first party (e.g., error logging), they act as **data processors**. If they use the user's data for their own purposes and gains, i.e., in order to do market research, create and monetize user profiles across customers or improve their services, and are not controlled by the first party, they act as data controllers.

GDPR Article 6 [31] contains the six general justifications for processing. Among others, the processing may be based on consent, the fulfillment of a contract, compliance with a legal obligation, or the data controller's legitimate interests when such interest outweighs the fundamental rights and freedoms of the data subjects. In practice, most advertising companies rely on consent or legitimate interests as the legal basis for processing personal data for profiling and targeted advertising (i.e., since the legal ground necessary for the performance of a contract does not apply in these circumstances [11, 29]).

However, a recent study from the Norwegian Consumer Council [29] shows that data subjects do not have a clear understanding of the amount of data sharing and the variety of purposes their personal data is used for in targeted ads. A large amount of personal data being sent to various third parties, who all have their own purposes and policies for data processing, are detrimental to the data subjects' privacy. Even if advertising is necessary to provide services free of charge, these privacy violations are not strictly necessary to provide digital ads. Consequently, it seems unlikely that these companies' legitimate interests may claim to outweigh the fundamental rights and freedoms of the data subject. This means that many of the ad tech companies would most likely have to rely on consent as the legal basis for their processing operations. In case the data transfer in question relies on user consent, the GDPR requires the consent to be *freely given, specific, informed, and unambiguous*. Further, the data subject must have given consent through a statement or by a clear affirmative action prior to the data processing in question (GDPR Art. 4(11) [30] and Art. 7 [32]).

Unambiguous consent under the GDPR must meet certain conditions. The GDPR Art. 7(2) states that: "*If the data subject's consent is given in the context of a written declaration which also concerns other matters, the request for consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language*". The user's consent has to be easily differentiated from other declarations or even consent to other processing activities. The user has to be specifically asked to consent to data sharing and processing for advertising purposes and this consent must not be grouped together with, e.g., consent to download the app or consent to access certain APIs on the phone.

In order to be legally valid, consent with regard to the processing of personal data has to be *explicit*. This means that the controller should obtain verbal or written confirmation about the specific processing [Recital 32]. According to the Article 29 Working Party, consent cannot be based on an opt-out mechanism, as the failure to opt-out is not a clear affirmative action [49]. The user has to actively give their consent, i.e., by clicking "I agree" on a consent form. Merely continuing to use an app or other passive behavior does not constitute explicit consent. Lastly, the consent has to be obtained prior to the data processing to be considered valid.

Our research focuses explicitly on these aspects of user consent. In particular, with respect to the aforementioned regulations, transmitting data to an advertisement company without *prior*, *explicit* consent by the user for the purpose of targeted advertisement is considered violating GDPR.

Data Type	Description
AAID	Android Advertising ID
BSSID	Router MAC addresses of nearby hotspots
Email	Email address of phone owner
GPS	User location
IMEI	Mobile phone equipment ID
IMSI	SIM card ID
MAC	MAC address of WiFi interface
PHONE	Mobile phone's number
SIM_SERIAL	SIM card ID
SERIAL	Phone hardware ID (serial number)
SSID	Router SSIDs of nearby hotspots
GSF ID	Google Services Framework ID

Table 1: Overview of personal data tied to a phone.

3 Methodology

Our main goal is to have a mostly automated and scalable solution to detect personal data that is being sent to the Internet without users' *explicit* consent, as is mandated by the GDPR. We set up an array of Android devices, on which we run each app (without any interaction) and capture the network traffic (Section 3.1). Based on personal data which is directly tied to the phone (see Table 1), we automatically detect this data in both plain and encoded form through string matching. Moreover, we derive a methodology that allows us to pinpoint data that may be other unique identifiers and manually validate whether this can be used to track the user/device (Section 3.2). In the following, we outline how we conduct each of the steps in more detail.

3.1 App Setup and Network Traffic Collection

We run each app and capture its network traffic. Here, we aim to detect apps' network traffic without users' explicit consent. To achieve this, we simply open the app but do not interact with it at all. The underlying assumption is that if network traffic occurs when this app is opened without any interactions, we have naturally not consented explicitly to any type of data collection by third parties. Hence, any data being sent out must not be PD, so as not to violate GDPR. Orthogonal to that, in practice, users may not grant all the apps' permission requests, or the app may use the runtime-permission mechanism (i.e., the permissions are not granted at installation time, and users will allow/deny permission requests at runtime when using the app). As such, it may be the case that PD (e.g., the IMEI) can only be accessed after the user consents to API usage. However, this API consent does not imply consent to have sensitive data shared with third parties. Therefore, to be legally compliant, the app must respect explicit consent even if it is authorized to access the data through a granted permission.

Recall that our goal is to analyze apps on a large scale. Hence, relying on static analysis techniques, which may produce a vast amount of false positives, is not an option [12, 39, 67]. Furthermore, we aim to have a lightweight solution to allow us to check thousands of apps in a reasonable time. Hence heavyweight instrumentation of the app itself is out of the question. Therefore, our approach is prone to miss certain instances (e.g., if we are unable to detect unique identifiers in the outgoing traffic or the app crashes in our lightweight instrumentation).

We rely on six rooted devices (Pixels, Pixel 3a, and Nexus 5) running Android 8 or 9 to analyze a given app. To intercept the TLS traffic, the devices are instrumented with our own root certificate (i.e., by using MitM proxy [15]). Further, we use *objection* to detect and disable SSL Pinning [47]. In the first step of our analysis pipeline, we aim to identify apps that send *some* data when started. To achieve that, we install the app in question and grant all requested permissions listed in the manifest, i.e., both install time and runtime permissions. Subsequently, we launch the app and record its network traffic. As our initial tests showed that apps sometimes did not load on first start, we close the app and reopen it, so as to increase the chances of observing any traffic. If an app shows no traffic in either of these starts, we discard it from further analysis.

3.2 Traffic Log Analyzer

Under the GDPR, personal data includes the Advertising IDs [46], location data, and online identifiers (e.g., IP addresses, any unique tracking identifiers) which can be used to identify users over a long period, potentially across different apps and services [4]. Next to data protected through OS permissions (e.g., IMEI, MAC), an app may also use other types of persisted, unique identifiers to track users. Hence, our analysis focuses on all possibly sensitive data as well as data that can be used to uniquely track a user or a specific instance of the app on their phone.

3.2.1 String-Matching Device-Bound Data

The first type of data we consider is such data that is tied to the phone, such as the location, the AAID, or the MAC address. Since such information is accessible by apps, we extract the relevant values from the phone through the Android debug bridge to ensure we know these values for each phone. The data selected for this (see Table 1) is inspired by the data used in prior work [52, 56]. Specifically, we first use simple string matching to identify PD that is static and known in advance. This information includes persistent identifiers bound to the phone (e.g., the IMEI, the MAC address of the WiFi interface, and the AAID) and those that are otherwise sensitive, such as the device owner's name, email address, or phone number. For the geolocation data, we search for the precise latitude and longitude written as a floating-point number, and those values are rounded to 3, 4, and 5 decimal places.

Beyond simple string-comparison, we also search for common transformations, such as upper/lower case, hashing (e.g., MD5, SHA-1), or encoding (e.g., base64) in our analysis. Naturally, this may miss cases in which, e.g., a custom hash function is used on the sensitive data by the app. To identify such cases as well as cases in which an app creates a persistent identifier itself, we conduct a second check for potential unique tracking identifiers.

3.2.2 Potentially Unique Tracking Identifiers Detector

This step aims to identify parameters that could be used to track and profile an individual, but do not obviously string-match with known values such as the IMEI. We aim to cover both cases of obfuscated usage of common identifiers as well as those cases where the app generates a persistent identifier and stores it locally. For example, from Android 8.0, the Android ID scopes to *{user, app signing key, device}* that does not obviously string-match with known identifiers.

More specifically, for a given app, we perform multiple runs (R_i) with a different set of devices (P_i) to monitor and record its network traffic. For each run R_i on a particular device P_i, we first install the app in question and grant all necessary requested permissions. While monitoring the app network traffic, we start the app, close the app and start it once more. By analyzing the captured traffic in run R_i, we extract all contacted hosts (domain names) as well as the GET and POST parameters (including parsing JSON if the content type of the request is set accordingly). This allows us to identify all contacted domains as well as the parameters and the values the app sent out. The output contains a set of triples t_i={<*domain,parameter,value*>}. Each triple <*domain,parameter,value*> is the identified contacted domain together with its parameter and the value that is being sent in the run R_i by the analyzed apps.

To this end, we further define two functions: (1) $diff(t_i,t_j)$ outputs all triplets of <*domain,parameter,value*> in t_i for triples that have the same domain and parameter but different value between t_i and t_j; (2) the function $stable(t_i,t_j)$ outputs all triplets of parameters which remained unchanged between two sets. Figure 2 shows an overview of our approach to detect potential unique identifiers (which we refer to as *UID* in the following). In general, four steps are involved:

1. On phone P_1, we first perform a run R_1 to discover all the app's network traffic. Then, by analyzing the R_1 traffic, we identify all of the contacted domains and their parameters (t_1). If there is no data sent to the Internet by the app ($t_1 = \{\}$), no further step is required.

2. On the same phone P_1, we now conduct run R_2 (installation and two open/close cycles). In between the two runs, we uninstall the app and set the time one day into the future. The intuition here is that if an app is sending some persistent identifier, this would be the same across time and remain on the device (e.g., in persistent storage or based on some information about the phone). Again,

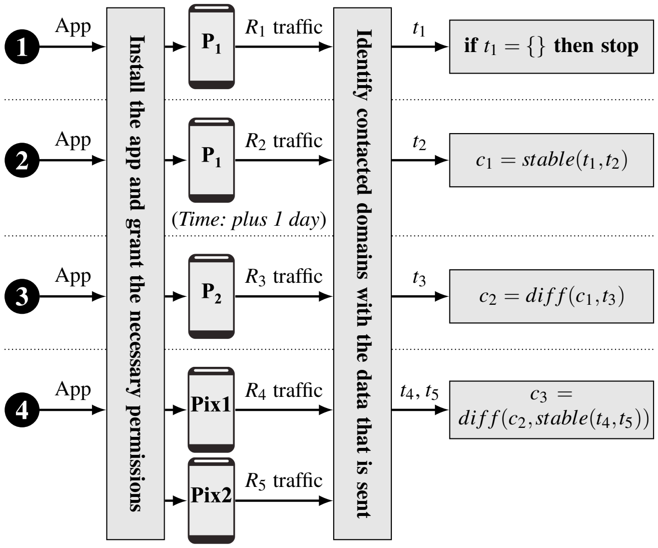


Figure 2: Overview of our methodology to identify potential UIDs. After each step, the analysis terminates if the resulting set of candidate parameters is empty.

we analyze the traffic to extract tuples (t_2). All parameters which are not stable between these runs cannot be persistent identifiers (e.g., the date) and are hence discarded. Suppose an app has any parameters with stable values across the two runs ($c_1 = stable(t_1, t_2)$). In that case, we consider a first candidate list for the next step — otherwise, we terminate the analysis (if $c_1 = \{\}$).

3. We now perform a run R_3 and extract the triplets from its traffic (t_3) on a different phone P_2. For each parameter value that remains stable across the two phones ($stable(c_1, t_3)$), we assume the stable value is tied to the *app* (such as some app-specific token) and hence discard these. If an app has at least one parameter, for which the value remained stable between R_1 and R_2 (both on P_1), but differs between R_2 (P_1) and R_3 (P_2), we consider this app further (naturally only considering those parameters that differed), i.e., $c_2 = diff(c_1, t_3) \neq \{\}$.

4. Given the diversity in our used phones, such a difference may simply be caused by the make and model or the OS version that is being sent out. To remove such cases, we now conduct further two runs R_4 and R_5, this time on two phones with the same make and model and OS version (Pixel 3a with Android 9). Suppose data is stable between these two runs ($stable(t_4, t_5) \neq \{\}$). In that case, we deem the corresponding parameter to be related to the device's make, model, or OS version, which is not a viable tracking identifier, and hence discard the entries. The outputs of the final step is then $c_3 = diff(c_2, stable(t_4, t_5))$.

Finally, this leaves us with a highly filtered list of candidates for persistent identifiers (c_3). In the final step, we manually check the parameters identified in this fashion, to ensure that they do not contain false positives. Particularly, we removed low-entropy entries such as carriers, time zones, or LAN IPs. Moreover, we also took into account the names of the parameters, disregarding parameter names that did not indicate any identifying capabilities (such as `rs` parameter on `helpshift.com`, which has sufficient entropy but lacks the clear indication of an identifier in its name). For our analysis, which we present in detail in Section 4, we identified 2,113 potential parameter/domain pairs that matched the criterion in the final stage. Of those, we discarded 412, e.g., because they were related to the (different) carriers or install times. Examples of different UIDs we detected this way are shown in Table 2. That is, given an app, our pipeline can automatically detect the sending of personal data (such as IMEI, IMSI, UIDs) without users' prior explicit consent. However, we have to *manually* vet the potential UIDs to avoid false-positive reports. Notably, we therefore may have missed true positives, which we nevertheless favor over a false positive.

Domains	Parameter
appsflyer.com	`deviceFingerPrintId=<UUID>`
branch.io	`hardware_id=6fd9a2e0f2721498`
tapjoy.com	`managed_device_id=tjid.36cec2b4196...`
unity3d.com	`common.deviceid=d3d55baf21d8f31839...`

Table 2: Examples of the UIDs identified by our approach.

3.3 Limitations

Our approach naturally suffers from certain limitations, some of which are desired. As an example, an app might show a welcome screen unrelated to data collection consent and only send out data once the user interacts with the app. Our framework would miss such cases of incorrect consent handling. We consciously decided to allow these false negatives, as understanding whether or not the welcome screen asks explicit consent and opt-out is infeasible to be done automatically.

Second, it might be possible that the consent notices are part of the runtime permissions request (e.g., apps have rationales that indicate data collection and use). By automatically granting all apps' permission requests, our approach might have false positives for such cases. However, in practice, Liu et al. [41] show that most developers do not provide rationales for permission requests (less than 25% of apps in their study). Moreover, before Android 6, only install-time permissions existed, meaning that any app compatible with Android 5 or lower *could not* ask for consent in the permission request. Out of the apps that we detected to send PD (see Section 4), about 96% support Android prior to 6.

Third, given that we rely on software that attempts to bypass security mechanisms (in particular SSL pinning), the apps may be able to detect such manipulation, e.g., by checking which CA is the root of the trust chain. Similarly, an app may also simply not start on a rooted device. Moreover, apps may not be supported on the Android 8 devices, which means they might not start and hence cannot be analyzed. Generally speaking, all these are potential causes for false negatives.

Finally, an app may also transmit a persistent identifier in some encrypted form with changing encryption keys or use a custom serialization format. Naturally, this is not something we can account for, and we would miss the parameter (as we could not decode the serialization protocol or, in case of the encryption case, its values already change between R_1 and R_2). However, we argue that if any app is detected to send out PD in our automated system, we have never granted explicit consent; hence we do not suffer from false positives.

4 Large-Scale Analysis

In this section, we present the results of our empirical study of Android apps on Google Play, with respect to the violation of GDPR's explicit consent mandate. We first outline which datasets we consider and subsequently present our analysis results. We note that all technical testing was done in Germany where the GDPR applies, i.e., our geolocation is Germany and the app store is set to the German variant. Based on our findings, we manually analyze the contacted domains with the help of a legal scholar to determine which third parties are *data controllers* for which GDPR mandates explicit consent.

4.1 App Dataset Construction

Our analysis aims to assess the state of GDPR violations in both high-profile and long-tail apps on the Play Store, and to understand if the violations are specific to either of them. To achieve this and compare these types of apps, we sample two datasets, totaling 86,163 apps:

High-profile app dataset: We crawled the top free high-profile apps in May 2020 from the Google Play store based on the AppBrain statistic [5]. For each country and 33 categories, AppBrain lists the top 500 apps. However, for some categories, AppBrain does not provide a full list of 500 apps (e.g., Events with only 271 apps). Therefore, as a result, our crawler obtained 16,163 high-profile apps from 33 app categories.

Long-tail app dataset: Between May and September 2020, we crawled all free Android apps available in the store from Germany and successfully obtained more than 1 million apps. Rather than running the analysis of the entire dataset, we decided to filter the list of apps through two steps to reach a more manageable dataset: we first rely on *Exodus-Privacy* [22] to identify apps that have integrated tracking or advertising libraries. As a result, we obtained more than 700,000 apps with embedded tracking or advertising libraries (304 of which are detected by Exodus-Privacy) in their code. Of these apps, we randomly sampled approx. 10% of apps with at least 10,000 downloads and excluded those in the high-profile set already, yielding 70,000 distinct apps for testing.

We note that this pre-selection strategy of filtering out apps which Exodus-Privacy did not flag as containing advertising or tracking libraries results in a sampling bias compared to the high-profile apps. To account for that, when comparing the statistics later, we only compare our data against those high-profile apps that Exodus-Privacy also flagged.

4.2 Network Traffic Analysis

As mentioned in Section 3.3, our approach suffers from certain limitations which keep us from analyzing all apps in the dataset. We were able to successfully analyze 72,274 (83.9%) apps, i.e., 14,975 high-profile apps and 57,299 long-tail apps. The remaining 13,889 either crashed or detected the analysis environment, making all of them potential false negatives.

Out of the 72,274 successfully analyzed apps, we identified 41,900 apps that contacted the Internet in either of the launches in R_1. Specifically, we identified 10,290 unique fully-qualified domain names being contacted. However, we found that a single registerable domain uses many subdomains (e.g., `rt.applovin.com`, `d.applovin.com`). To normalize these hosts to their registerable domain (`applovin.com` in the above cases), we rely on the public suffix list [51]. We refer to these resolved domains as *domain names* in the following. As a result, we identified 7,384 domain names that were contacted by 41,900 apps.

Among the 7,384 domain names, we found 1,744 (23,6%) domain names that received one or more of the types of PD listed in Table 1. Each time any of the relevant data is sent by an app to a domain, we count this as a case of PD being sent out. Specifically, we identified 28,665 apps (see the first column of Table 3) that sent PD to these 1,744 domain names. We now rely on the assumption that a third party would serve multiple apps and hence flag those domains as third-party domains that are contacted by at least ten different apps. This leads us to detect 337 distinct third-party domains. We found that 28,065 (97.9% of 28,665; second column in Table 3) apps sent PD to 209 third-party domains. Notably, third-party domains, representing only 12.0% of domains which received PD, are responsible for a disproportionate fraction (94,7%) of cases of receiving PD without prior consent.

This result suggests that only a negligible number of first parties collect PD. In contrast, the majority of PD was sent to third parties, which developers heavily rely on for a variety of purposes such as monetization (e.g., personalized ads), error logging, analytic services, user engagement, or social network integration. We note that GDPR mandates explicit consent in case such a third party acts as a data controller (rather than a data processor, that does not itself benefit from processing the data). Hence, in the following, we specifically focus on domains for which we can unequivocally determine that they control data for their own purposes, namely advertisement.

	Any Domains (N=28,665)		Third-Party Domains (N=28,065)		Advertisement Domains (N=24,838)	
	High-Profile Apps	Long-Tail Apps	High-Profile Apps	Long-Tail Apps	High-Profile Apps	Long-Tail Apps
AAID	5,177 (34.6 %)	22,152 (38.7 %)	5,072 (33.9 %)	21,957 (38.3 %)	4,366 (29.2 %)	19,904 (34.7 %)
BSSID	86 (0.6 %)	107 (0.2 %)	71 (0.5 %)	88 (0.2 %)	16 (0.1 %)	12 (0.0 %)
EMAIL	48 (0.3 %)	113 (0.2 %)	42 (0.3 %)	108 (0.2 %)	—	—
GPS	459 (3.1 %)	1,151 (2.0 %)	363 (2.4 %)	946 (1.7 %)	136 (0.9 %)	244 (0.4 %)
GSF	4 (0.0 %)	3 (0.0 %)	3 (0.0 %)	1 (0.0 %)	—	—
IMEI	107 (0.7 %)	611 (1.1 %)	51 (0.3 %)	444 (0.8 %)	36 (0.2 %)	356 (0.6 %)
IMSI	22 (0.1 %)	26 (0.0 %)	8 (0.1 %)	6 (0.0 %)	—	—
MAC	68 (0.5 %)	126 (0.2 %)	30 (0.2 %)	41 (0.1 %)	27 (0.2 %)	17 (0.0 %)
PHONE	1 (0.0 %)	4 (0.0 %)	1 (0.0 %)	—	—	—
SERIAL	49 (0.3 %)	158 (0.3 %)	17 (0.1 %)	91 (0.2 %)	3 (0.0 %)	3 (0.0 %)
SIM_SERIAL	9 (0.1 %)	29 (0.1 %)	5 (0.0 %)	19 (0.0 %)	—	—
SSID	73 (0.5 %)	108 (0.2 %)	67 (0.4 %)	78 (0.1 %)	17 (0.1 %)	15 (0.0 %)
UID	1,044 (7.0 %)	4,471 (7.8 %)	938 (6.3 %)	4,236 (7.4 %)	679 (4.5 %)	3,533 (6.2 %)
Any	5,455 (36.4%)	23,210 (40.5%)	5,276 (35.2%)	22,789 (39.8%)	4,415 (29.5%)	20,423 (35.6%)

Table 3: Types of data and number of apps sending this to any, third-party, and ad domains (percentages relative to dataset sizes).

4.3 Identifying Advertisement Domains

Under the GDPR, all personal data processing has to have a legal justification. The first party acting as a data controller may rely on several potential legal justifications for their data processing: fulfillment of a contract, legitimate interest, or consent. This legal justification extends to any third party acting as a data processor for the app developer. Since the third party acts completely under the app developer's control they are viewed as in the same legal domain as the first party. Meanwhile, a third party acting as a data controller would need its own legal justification to receive and process the user's PD. As such, they cannot rely on the original controller (app developer) to be the only responsible party to obtain a valid legal basis for their processing operations, or to ensure compliance with other obligations under the GDPR, particularly regarding the exercise of data subjects' rights. [29].

As the most prominent business case of third parties receiving and processing user data for their own business purposes, we chose (targeted) advertising to have a conservative lower bound for the cases of GDPR violations in the wild. An app which relies on external data controllers for targeted advertising needs to explicitly ask for the user's consent to share her PD with the third party. We found that third-party domains received 94,7% of all PD being sent out to the Internet. In order to analyze whether this data transfer would most likely require the user's prior consent, we first need to identify whether a third party is an advertising company, and second need to differentiate between those third parties that act as data processors and those that act as data controllers.

To determine whether a party is a potential advertisement company, we first rely on Webshrinker's categorization to identify the main topic of a domain [68] for all 209 third-party domains that received PD in our analysis. For all domains *not* flagged as ad-related, we manually review the Web pages of the domains to assess if the domain is related to a company offering in-app advertising services. For example, while Facebook is categorized by Webshrinker as a social network, they are also an advertising company, which relies on `graph.facebook.com` for advertising and tracking [52]. In this fashion, we identified 69 domains which are operated by ad-related companies. However, not all these domains actually act as data controllers under the GDPR. To distinguish between data controllers and processors, we analyzed the legal documents provided by the third parties.

Particularly, we manually analyzed the terms of service, privacy policies, developer guidelines and contracts, if available. The GDPR requires companies processing personal data to transparently provide their processing purposes and justification. We relied on the third party's legal self-assessment whether they describe themselves and their data use as a data controller or data processor. If they described their data use as mainly for their own company's gain, e.g., assembling and selling user profiles across several different apps, we would classify them as data controllers. If they limit their described data use as purely on behalf of and instructed by the app developer and if they would provide the additional necessary data processor agreement documents, we classify them as data processors. If a company's legal statements were too vague or they offered services as both data controller *and* processor, we classified them as data processors in order to conservative estimate the number of potential GDPR violations and to not unjustly notify app developers that commissioned these companies as data processors.

Out of 69 third-party domains which are operated by ad-related companies, we identified 45 domains of data *controllers* (full list in Appendix, Table 5), which would require explicit consent to receive data. In the next section, based on these 45 ad-related domains, we present our analysis on the GDPR compliance of apps regarding consent requirements.

4.4 In-Depth Analysis of Violations

We now focus on the set of apps which contacted any of the aforementioned 45 domains which we determined to be ad-related data controllers. Based on these domains, we find that the vast majority of apps that contact third parties with PD in fact send this to ad-related domains (24,838/28,065 or 88.5%, as shown in the third column of Table 3). Moreover, 86.6% (24,838/28,665) of apps which sent out *any* data do so towards advertisement domains. Relative to the number of apps we could successfully analyze, this means that 34.4% of them sent out PD to third-party data controllers, thereby violating GDPR's mandated consent to such collection. We note that this is in light of a mere 45/1,774 (2.5%) contacted domains being flagged as advertisement domains, which shows the significant skew towards apps sending out PD to advertisement companies without user's explicit prior consent. Notably, there is a significant skew towards the AAID as being the most frequently transferred piece of PD. However, according to both the advertising industry [35] and Google's own brand [33], the AAID is considered PD and regularly requires consent before being collected by third-party data controllers.

Identifying Major Players We now turn to analyze which are the most frequent parties that receive PD. Figure 3 shows the top 10 ad-related domains that received PD in our dataset, counting the number of apps that sent data towards them. We find that more than half of the apps which sent data without consent sent data to (at least) Facebook. It is noteworthy that Facebook makes GDPR compliance particularly tough for developers to implement. According to their own documentation [23], their SDK defaults to assuming user consent. That is, a developer must actively disable the automated transmission of PD and implement their own consent dialogue. In the case of Facebook, they operate in multiple capacities (e.g., social media integration *and* advertisement), yet their terms allow data sharing between the different services in their privacy policies. Specifically, the Facebook Graph API can share its data with Facebook Ads [4], which in turn can again be used to optimize advertisement.

The current version of the SDK of the second most-prevalent recipient of data, namely Unity, supports two variants of consent [63]: either, the developer provides consent through an API call (naturally after having acquired explicit consent) or the app developer can rely on Unity's solution which asks the user when the first ad is shown. However, as per their legal documentation, this is an *opt-out* mechanism rather than opt-in [64]. We believe this to be the major driving force behind the large number of requests towards Unity, as their ads first transmit data and then ask users to opt-out.

As for the third-largest recipient, we note that Flurry also supports a consent API, but the documentation is unclear about the default behavior and lacks important details about the proper implementation [26]. More notably, Flurry delegates the responsibility to acquire consent to the app developer. Moreover, they explicitly state that they assume any data is only sent after the developer has received consent. Overall, this implies that library providers make it very cumbersome for developers to be compliant with GDPR.

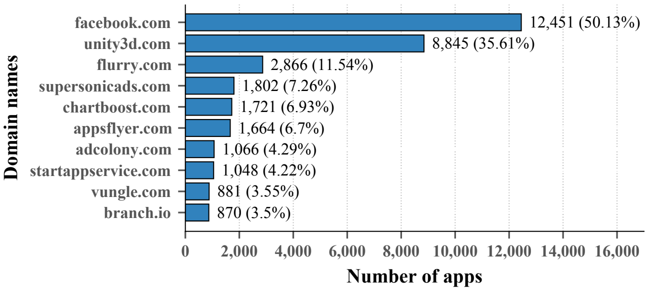


Figure 3: Top 10 ad domains that frequently received PD from 24,838 apps that sent PD to all ad-related domains.

Combining Multiple Identifiers As our data indicates, the vast majority of apps send out the AAID. While this in itself is already problematic with respect to the GDPR, apps sending out any other information alongside the AAID also violate the Google policy for AAID usage. In particular, according to said policy [3, 50], the AAID must only be used for advertising and user analytics. For both purposes, the AAID may not be connected to persistent device identifiers (e.g., SSID, IMEI). The AAID may only be connected to other personally-identifiable information with the user's explicit consent. In our dataset, we found that a total of 3,840 apps combined the AAID with some other type of personal information. Hence, all these apps not only infringe on the explicit consent required by GDPR, but also violate Google's policy, which means they could be removed from the app store.

For each app, we investigated to which ad-related domain they sent out the combination of the AAID and other PD. The results of this analysis are shown in Table 4. Note that, on purpose, we do not include the UID here, as we cannot identify whether a particular unique ID is just the AAID (e.g., hashed with an unknown algorithm). The results indicate that there are numerous domains that receive the combination of AAID and some other identifiers. Specifically, for cases such as the 190 apps that sent out the AAID with the IMEI to Flurry, Google can remove the apps without prior notice from the app store. To further understand this violation of Google's policy (combined with the fact that only relatively few apps conduct this practice), we analyzed the versions of SDKs used in apps which sent out the data to these top 5 ad-related domains. To that end, we rely on a two-step approach. First, based on the collected traffic, we identify SDK version numbers from the requests, such as GET or POST parameters (see Appendix B for details). For apps which lack such version information in the HTTP traffic, we instead rely on a more involved analysis

Domains	Data Types	No. Apps
flurry.com	IMEI	190
	GPS	156
	SERIAL	4
	SSID	2
	MAC	1
my.com	BSSID;GPS;MAC;SSID	22
	MAC	17
	GPS;MAC	2
	BSSID;GPS;IMEI;MAC;SSID	1
amazon-adsystem.com	GPS	30
unity3d.com	IMEI	29
vungle.com	GPS	26

Table 4: Top 5 of ad domains receiving AAID along with other PD in our two app datasets.

through LibScout [9]. We chose not to apply LibScout to all apps given the significant runtime overhead this would have caused (99.47 seconds per app for 100 randomly tested apps on macOS/Core-i7/16GB RAM).

Out of the 353 apps which contacted Flurry, we were unable to extract SDK version information for 202 apps from their traffic. For these 202 apps, LibScout successfully detected SDK versions for 53 of the apps. In particular, it detected SDK versions for 45 of the 190 apps which sent the IMEI, all of which were pre-GDPR versions. We note that based on the release notes of Flurry [28], the feature for the IMEI collection was removed already in 2016. Since we are unable to download Flurry SDKs before version 6.2.0 (released in November 2015), it is highly likely that LibScout's failure to detect the version stems from pre-6.2.0 versions being used by the apps in question. Hence, we believe that the IMEI collection can be attributed to extremely old versions of the SDK still in use in the apps we tested. For the versions which sent the serial, LibScout detected two out of the four SDK versions, both of which ran pre-GDPR versions. For the single detected case of sending out the MAC address, LibScout detected a version with GDPR support. Finally, for the class of apps that sent out GPS, we could detect SDK versions for 151/156 based on the traffic, and LibScout successfully detected the SDK version for the remaining five. Notably, all these apps used current versions of the Flurry SDK. However, based on the Flurry manual, it appears that if an app has GPS permissions, Flurry defaults to sending this unless the developer explicitly opts-out [27].

For the 42 cases in which `my.com` received (at least) the MAC with the AAID, we found that 23 ran SDK versions which support GDPR. However, the documentation is sparse [44] and it remains unclear if the default behavior is to collect such data (or the developer has to set `setUserConsent` to true first). For the remaining 19 cases, they all used outdated SDK versions without GDPR support. Considering the data sent out to Amazon, we find that 20/30 apps are running a current version of the Mobile Ads SDK. For the 29 cases of apps which sent the AAID along with the IMEI to Unity, these all used outdated SDKs (released before 2018 when GDPR came into effect). For Vungle, 16/26 apps which sent out GPS with the AAID ran pre-GDPR versions of the library (added in version 6.2.5 [66]). Yet, for the remaining ten, the version numbers indicated GDPR support; i.e., in these cases developers opted into sending said data.

Further, out of 24,838 apps that sent PD to ad-related domains, we found that 2,082 (8.4%) of these apps have a pre-GDPR update date (before May 2018). We note from this analysis that the most egregious violations can be attributed both to extremely old versions of libraries (e.g., developers often neglect SDK updates when adding functionality [18]), but also to the complex configuration required to make apps GDPR (and Play Store)-compliant. This is particularly obvious for the collected GPS coordinates in Flurry's SDK, which seems to be enabled by default unless developers opt-out. In the following, we aim to understand if the violations discussed thus far are specific to either high-profile or long-tail apps.

Comparing the Datasets A natural question that arises is about potential differences between the datasets. As mentioned earlier, we filtered the long-tail apps through Exodus-Privacy, which introduces a selection bias. To account for that, before comparing the datasets, we apply the same filtering to the high-profile apps. After that, we find that only 10,799 of the 14,975 high-profile apps we could successfully analyze would have passed this filtering step. Notably, we would have missed 888 high-profile apps which sent out data if we had prefiltered the high-profile apps. Assuming similar distributions of undetected ad-related libraries in the long-tail set, this is an obvious limitation of our sampling approach and should be kept in mind for future work. After applying the filtering to the high-profile set, we consider 3,527/10,799 (32.6%) high-profile apps which sent out data as well as 20,423/57,299 (35.6%) apps from the long-tail dataset in the following.

We first compare the number of apps in each dataset which send out PD to ad-related domains. Our null hypothesis H_0 is that there is no difference between high-profile and long-tail apps in terms of sending out PD. By applying χ^2, we find that with $p < 0.01$, H_0 is rejected. However, computing Cramer's V ($v = 0.0228$) we find that the effect is negligible [20]. Next, we investigate to what extent the apps in the datasets differ in terms of domains to which they send PD. For this, we apply the Kruskal-Wallis test, which rejects H_0 of no difference between the sets with $p < 0.01$; however, computing the effect size $\varepsilon^2 = 0.0178$, there is again only a small to negligible effect [20]. Similarly, for the number of different types of PD sent out, Kruskal-Wallis shows $p = 0.022$, but $\varepsilon^2 = 0.0002$, i.e., again significant difference, yet negligible effect.

In addition to the overall trend, we also analyzed whether we can observe differences in the parties which are contacted by apps in each category. Figure 4 shows the most frequently

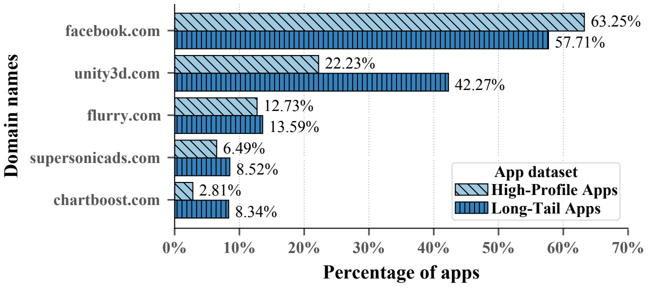


Figure 4: Top 5 ad domains receiving PD in each app set after applying the Exodus-Privacy filtering to the high-profile set (percentages relative to the PD-sending apps per dataset).

contacted domains across both datasets together with the percentage of apps that sent PD to them. In particular, we consider the percentages relative to the apps after the Exodus filtering step. We note that `facebook.com` is the most prevalent in both sets, yet occurs more often in the high-profile than the long-tail apps. Contrary to that, we find that `unity3d.com` is more pronounced in the long-tail apps.

By analyzing the categories of the apps, we found that Unity is frequently used in games, which are at the core of Unity's business model. Notably, AppBrain combines all subcategories of games into a single category, meaning our high-profile apps set contains only 500 such games. In contrast, in the long-tail apps set, almost 20% of the apps are related to games, explaining the significant skew towards Unity in that dataset. Generally, out of 72,274 successfully analyzed apps, the top 5 categories that have more violating apps than others in both app sets are *game* (73.29%), *comics* (64.97%), *social* (41.39%), *shopping* (37.04%), and *weather* (36.59%).

Overall, the results of our comparative analysis lead us to conclude that the phenomenon of sending out personal data without prior explicit consent occurs as frequently and with as many parties in both high-profile and long-tail apps. While we did observe statistically significant differences, the associated effect size was negligible. And while there certainly exists a difference between the two datasets in terms of *who* receives data, we cannot observe a difference that would warrant the hypothesis that high-profile apps violate GDPR less than long-tail ones.

Manually Analyzing Consent Dialogues To investigate whether developers may have merely misunderstood the concept of consent (or the GDPR requirements thereof), we randomly sampled 100 apps which sent out data in our experiment and checked the screenshots (which we had taken as part of our analysis to show on the Web interface). Specifically, we checked for both implicit consent dialogues (such as those indicating that by using the app, we consent to data being sent out) or explicit opt-out dialogues. We note here that even having an opt-out dialogue which – after negative confirmation – stops collecting of data still meant the app sent out data *before* asking for the user's consent. Among these 100 apps, we found only 25 apps present any type of consent notices to users. Of these, only 11 apps provide an option to reject the data collection, while the remaining 14 apps ask users to accept the data collection to use without options to reject the data collection and sharing. Overall, this indicates that the vast majority of apps do not even attempt to achieve GDPR compliance, nor do they add meaningful ways of rejecting consent after the fact.

5 Developer Notification

In addition to the technical analyses described thus far, we also notified affected developers. This had two main goals: first, to inform them about the potential breach of GDPR regulations, which may lead to severe fines [16]. Second, we wanted to gain insights into the underlying reasons that caused the observed phenomena in the first place. Since disclosing the findings to authorities (e.g., regulators, Google) might cause financial harm to developers, we consciously decided not to involve authorities but rather notify developers directly to remedy compliance issues. We note that our institution's ethics guidelines do not mandate approval for such a study.

To notify the developers, we extracted the email addresses used to upload the apps into the Play Store. To assess how many developers actually received our reports, rather than including the technical details about the issues in the email, we sent developers a link to our web interface. On this, we briefly explained our methodology of testing and showed the developers information about which hosts received which type of data (see the previous section). In addition, in our email, we asked recipients if they had been aware of the potential violation of their apps, their general understanding of what is personal data under GDPR, and their plans to address the issues as well as proposals for tool support (see Appendix A for the full email). We decided to have this rather than a full-fledged survey, as we wanted to keep the overhead for respondents as low as possible to prompt more responses. We note that the notification was carefully worded not to make legally conclusive statements, since this could amount to legal consulting which is strictly regulated by German law.

5.1 Notification and Accessed Reports

Out of the 24,838 apps for which we had discovered some potential GDPR issue, 7,043 had been removed from the Play Store by the time we conducted our notification. For the remaining 17,795 apps, we sent out notifications in two batches, each with a reminder. The first batch of apps were notified on December 15, 2020, with a reminder on January 5, 2021. In this batch, we only included such apps that had not been updated since our download from the Play Store until the day of the notification, totalling 8,006 apps. We took this step to ensure that we would not notify developers who had removed the problematic code between our dataset download and notification date. For these 9,789 apps with recent changes, we re-downloaded the latest version, conducted our analysis again to confirm our findings, and added those apps to the second batch which still had some issues. The second batch of notifications was sent on January 6, 2021, with reminders on January 20, 2021. Note that we decided to give an additional week between notification and reminder for round 1 of the notifications, given the overlap with the Christmas vacation. In both cases, we grouped emails to developers (i.e., if a single developer had more than one app in the store, they only received one email with multiple links). We followed best practices established by prior work [19, 38, 59, 60] allowing developers to opt-out and not send reminders for those apps for which we had previously seen an access to the report.

In total, we notified 11,914 developers responsible for 17,795 apps. Of those developers, eight asked to be removed from our experiment. Until February 1, 2021, we saw 2,199 accessed reports. Notably, some accesses were related to spam checking (e.g., from Barracuda's IP range or clients not downloading subresources like CSS files), which we ignore in our analysis. This leaves us with accessed reports for 2,083 apps. Notably, considering that a single owner may have multiple apps affected by the same issue, we count the overall number of apps for which their developer accessed *some* report, totalling 2,791 (15.7%) apps for which we reached their owner.

5.2 Developer Responses

In addition to the accessed reports and the updated apps, we also analyzed the responses we received from developers. In total, this amounted to 448 distinct senders that we classified emails for. Based on an initial set of responses, three coders developed an initial code book and then separately analyzed the entire set of responses. For all cases in which their assessment of an email/thread differed, they discussed the cases in a group until they agreed on a classification. Note that not all respondents answered the stated questions from our email [1].

Of the 448 respondents, 114 acknowledged receipt of our email and wanted to take it under advisement. 54 stated that they required further investigation, either within their respective companies or their third-party SDK vendor. 48 further inquired with us about potential solutions to the problem, such as adding privacy policies to explain the data collection. We faithfully answered these emails while stating that we cannot provide conclusive individual legal assessments. Notably, 20 respondents argued that the EU was not their main market, and that hence either GDPR would not apply to them or they did not feel the need to implement consent, either being unaware of their app being downloadable from an EEA country Play Store or that having users resident in the EU leads to applicability of the GDPR.

On the other side of the spectrum, 116 respondents disagreed with our assessment. These ranged from comments like *"i am not aware that my app might not be GDPR compliant"*, *"I show my privacy policy at the start of the app"*, *"where seems to be the problem"* to simple claims that their apps do not transmit any user information, but also argued that their advertisement libraries first ask for user consent before transmitting any data. This highlights a misconception that having a privacy policy supersedes the need to have explicit consent under the GDPR. Notably, as also highlighted by our manual analysis, many ad-related libraries sent out data before showing the consent dialogue. In the most extreme cases, developers also argued that it was infeasible for them to fully support GDPR, with one developer stating: *"I haven't done anything wrong in my eyes. I show adverts in my Apps from BIG F****** COMPANIES like Google and Facebook and it's up to them to tell developers what they collect so we can then pass that on to our users. Like how the hell am I supposed to know what a black box SDK from Google does with data in the App I publish to people???"*.

When asked about the data collection, 70/151 respondents (46%) said they were aware of the types of data being collected and 53/122 (43%) said they knew this data was protected by GDPR. Of the 139 respondents who answered our question regarding reasons for lacking explicit consent, 66 (47%) argued they rely on a third-party app builder or SDK to make their apps compliant and 40 (29%) believed their app to be compliant already. Ten explained their app was outdated, seven noted that they lacked resources for proper implementation, and seven said this was a bug. Finally, nine respondents stated there was no particular reason.

When asked about their plans to change their apps (218 answers), 136 (62%) stated to update their app, with another 29 (13%) claiming to plan to remove the app from the Play Store altogether or make it inaccessible from EEA countries. Eleven said to conduct additional research into GDPR and their responsibilities as the developers, and 17 said they did not feel the necessity to take any steps. Our data is heavily skewed towards those developers that plan to take action; we attribute this to the fact that developers that disagreed with our assessment rarely answered our follow-up questions.

Regarding our final question about developer support, we received 72 answers. Of those, 44 wanted to have an automated tool like ours to analyze their apps for compliance, while 19 asked for better documentation around how to implement GDPR compliance. Finally, nine respondents argued that third-party tools should be compliant by default (e.g., *"requiring ad providers to take responsibility for all the compliance to this unnecessarily complicated law"*). Naturally, the skew towards automated tooling is not surprising, given that we notified developers after applying our automated toolchain. It is noteworthy, though, that fewer developers answered this final question, implying that it is not even clear to them how they could be better supported in this particular issue space.

[1] All developers we quote in this paper have given their explicit consent.

5.3 Updates to Notified Apps

To assess our notification's impact on the affected apps, we downloaded new versions of all apps that had looked at our reports at least one by April 06, 2021. We re-ran our pipeline for each app with an updated version to assess if the changes were related to the reported GDPR infringement. For the 2,791 apps for which we reached a developer (i.e., they looked at at least one report for their apps), 91 apps were removed from Google Play, and 8 apps were no longer available to download from Germany. We found 1,075 apps with updates since our notification for the remaining apps. By rerunning our pipeline on these 1,075 apps, we observed that 250 (23%) apps no longer sent PD to ad-related domains without prior consent. Considering those 136 respondents that claimed to plan to update their apps to incorporate proper GDPR consent, we found that 92 apps (for which the respondents were responsible) had been updated until the end of our experiment. Notably, though, only 43 were updated in such a fashion that they did not send out any data without interaction.

We note here that the overall number of apps which addressed the issue is low. Based on the responses we received, we believe this to have two core reasons. First, many apps are developed by small teams (if not individuals) who would rather focus on functionality updates. Second, as shown in our analysis of popular SDKs such as the one from Facebook, they do not provide a consent dialogue, but rather put the burden on the developer to integrate a new UI to ask for consent, which is then passed to the SDK. Hence, we believe the number of apps which address this issue will rise over time and the seemingly small change in overall numbers can be attributed to a lack of time to properly address the issue.

6 Calls to Action

Our results thus far have shown that the sharing of PD with third-party data controllers is very pronounced in the datasets we tested. More than one-third of all apps we tested sent out PD before any users' interaction. More notably, we could not find a significant difference between high-profile and long-tail apps, i.e., the problem affects both high-profile and long-tail apps. Given these insights, we now discuss which involved parties can take which steps to remedy the situation.

6.1 Third Parties Should Take Responsibility

Today, digital content is largely funded by advertising, which means that companies monetize our behavior, attention, and PD rather than us paying for services with money [29]. To maximize revenue, advertising services heavily rely on continuous data collection and tracking PD from users [43]. Our results show a significant skew towards apps sending out PD to advertisement companies without user's explicit prior consent (i.e., 86.6% of all apps that sent PD to the Internet) — which is the most prominent business case of third parties receiving and processing user data for their own business purposes. However, we found that these third parties make it cumbersome for developers to comply with GDPR or shift the responsibility to app developers.

For example, Facebook required developers to obtain users' consent before sending data via the SDK [23], whereas the default behavior is automatically collecting user PD such as AAID [24]. Our insights further show that such popular companies play key roles in the widespread receiving of PD without users' explicit consent, i.e., more than half of apps that sent data without consent sent it to (at least) Facebook. However, many developers believed that their apps are compliant by default when using these popular companies' services, as noted by one respondent as *"These third party SDKs are from industry leading ad networks that only accept those apps that are GDPR compliant. So a GDPR compliance is must before the app is being approved by these advertising networks (i.e. Facebook & Admob). So our app is a GDPR complaint"* (sic). In addition, some respondents claimed to be aware of GDPR-relevant data, but were surprised by our reports which showed that the SDKs collected information; *"We were already aware of this topic and we were already working on it. We did not send any events to Facebook (we eliminated this feature long ago) – the SDK itself sent out data to Facebook without any trigger from our side"*. While this lack of knowledge does not absolve the first party of their responsibility, the lack of clear guidelines and safe defaults for GDPR-compliant data collection by the advertisement industry inevitably puts their customers, i.e., the app developers, at risk of the draconian fines which can be imposed for GDPR violations [29].

Our findings show the urgent need for advertisement companies and third parties (data controllers) to make comprehensive changes to help app developers comply with European regulations and exercise the data subjects' fundamental rights and freedoms. Particularly, third parties first should limit the data collection to respect principles of data protection by design and by default. For obtaining user consent, third parties should provide the consent mechanism that automatically shows the consent dialogue to users and explicitly ask for opt-in to data sharing and collection, without forcing the developers to implement this mechanism in a legally compliant way. Further, to support developers, third parties should make their documentation transparent and easy to access, including explicit discussions of implications of violating GDPR.

6.2 App Stores Should Take Actions

App stores such as Google Play are a channel for the distribution of developers' apps to the users, which play an important role in supporting developers to be informed about each territory's related regulations and protect user privacy. However, we found that developers lack such support, e.g., *"This game was designed to Brazil and we published the game to Europe* *in March 2019 to expand our potencial [sic] customers, targeting Portugal. GooglePlay allows this, checking a button, without any restrictions. So, I feel protected by Google somehow"*. Therefore, we strongly suggest that app stores should take more decisive actions in this area. For example, when developers upload their apps, the store should tell them about the selected countries' associate regulations.

Besides a large number of apps that sent PD to ad-related domains without users' explicit consent (24,838 apps), we further detected a total of 3,840 apps that combined the AAID with some other type of PD. Hence, all these apps not only infringe on the explicit consent required by GDPR, but also violate Google's policy [3, 50]. Such behaviors happened due to developers' opt-in or the usage of outdated libraries that do not support GDPR. Given that, app stores could also employ such techniques as our to identify the potential violations of GDPR explicit consent, or the usage of outdated SDK by or LibScout [9], and then inform developers before delivering the apps to end-users. To support this effort, we make our analysis pipeline available as open-source [1].

6.3 Support for Developers

Obviously developers play a major role in making their apps compliant with the GDPR. Our findings show that they are currently put in a disadvantaged position. Out of the responses we received, more than half noted that they were unaware of what counts as *personal* data under GDPR. From the received responses, there is a clear need for better information and documentation as well as tools which help developers avoid such pitfalls. Further, based on our in-depth analysis of third party's developer and legal documentation, we observed that third parties make it cumbersome for developers to comply with GDPR. We therefore strongly call on third-party vendors for better documentation and transparency in legal documents, which should in turn be thoroughly checked by developers when building their apps.

7 Conclusion

In this paper, we performed an empirical study of 86,163 Android apps to understand the current state of the violation of GDPR's explicit consent. Doing so, we found 24,838 (34.3% of the successfully analyzed) apps sent personal data towards advertisement providers that act as data controllers without the user's explicit prior consent. We believe that our results shed new light on the current state of the violation of GDPR's explicit consent in the wild. Based on our insights from our notifications and in-depth analysis, we find that this problem is widespread, misunderstood among developers, and requires effort from ad providers, app stores, and developers alike to mitigate. Finally, we derived concrete recommendations to all concerned parties and make an urgent call to help developers comply with the GDPR and honor users' rights and freedoms.

Acknowledgments

We thank our shepherd Roya Ensafi and the anonymous reviewers for their valuable feedback. Many thanks to Michael Schilling for his help and support, and Richard Bergs, Johanna Wüst, Xin Yun who categorized the developer responses.

References

[1] GDPR-Consent, 2021. URL https://github.com/cispa/gdpr-consent.

[2] Breyer v. Germany, 2021. URL http://hudoc.echr.coe.int/eng?i=001-200442.

[3] Google Developer Distribution Agreement. Developer distribution agreement. https://play.google.com/about/developer-distribution-agreement.html#use, 2021. 2021/01/17.

[4] B Andow, S Y Mahmud, J Whitaker, W Enck, B Reaves, K Singh, and S Egelman. Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with policheck. In *USENIX Security*, 2020.

[5] appbrain. Google play ranking. https://www.appbrain.com/stats/google-play-rankings/, 2021. 2021/05.

[6] Apple. User privacy and data use. https://developer.apple.com/app-store/user-privacy-and-data-use/, 2021. 2021/02/01.

[7] S Arzt, S Rasthofer, C Fritz, E Bodden, A Bartel, J Klein, Y Le Traon, D Octeau, and P McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. *Acm Sigplan Notices*, 2014.

[8] K W Y Au, Y F Zhou, Z Huang, and D Lie. Pscout: analyzing the android permission specification. In *CCS*, 2012.

[9] M Backes, S Bugiel, and E Derr. Reliable third-party library detection in android and its security applications. In *CCS*, 2016.

[10] R Bhoraskar, S Han, J Jeon, T Azim, S Chen, J Jung, S Nath, R Wang, and D Wetherall. Brahmastra: Driving apps to test the security of third-party components. In *USENIX Security*, 2014.

[11] European Data Protection Board. Guidelines 2/2019 on the processing of personal data under article 6(1)(b) gdpr in the context of the provision of online services to data subjects". https://edpb.europa.eu/sites/edpb/

files/files/file1/edpb_guidelines-art_6-1-b-adopted_after_public_consultation_en.pdf, 2019. 2019/02.

[12] R Bonett, K Kafle, K Moran, A Nadkarni, and D Poshyvanyk. Discovering flaws in security-focused static analysis tools for android using systematic mutation. In *USENIX Security*, 2018.

[13] P Calciati, K Kuznetsov, X Bai, and A Gorla. What did really change with the new release of the app? In *MSR*, 2018.

[14] CCPA. California consumer privacy act (ccpa). https://oag.ca.gov/privacy/ccpa, 2021. 2021/02/01.

[15] A Cortesi, M Hils, T Kriechbaumer, and contributors. mitmproxy: A free and open source interactive HTTPS proxy, 2010–. URL https://mitmproxy.org/.

[16] Datatilsynet. Intention to issue eur 10 million fine to grindr llc, 2021. URL https://www.datatilsynet.no/en/news/2021/intention-to-issue--10-million-fine-to-grindr-llc2/. 2021/02/04.

[17] M Degeling, C Utz, C Lentzsch, H Hosseini, F Schaub, and T Holz. We value your privacy... now take some cookies: Measuring the gdpr's impact on web privacy. In *NDSS*, 2019.

[18] E Derr, S Bugiel, S Fahl, Y Acar, and M Backes. Keep me updated: An empirical study of third-party library updatability on android. In *CCS*, 2017.

[19] Z Durumeric, E Wustrow, and J A Halderman. Zmap: Fast internet-wide scanning and its security applications. In *USENIX Security*, 2013.

[20] P D Ellis. *The essential guide to effect sizes: Statistical power, meta-analysis, and the interpretation of research results*. Cambridge university press, 2010.

[21] europa.eu. "opinion 06/2014 on the notion of legitimate interests of the data controller under article 7 of directive 95/46/ec (article 29 working party). https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp217_en.pdf, 2020/09/02.

[22] Exodus-Privacy. Exodus standalone. https://github.com/Exodus-Privacy/exodus-standalone, 2020. 2020/09/14.

[23] Facebook. Fb sdk best practices for gdpr compliance. https://developers.facebook.com/docs/app-events/gdpr-compliance/, 2021. 2021/02/01.

[24] Facebook. Get started – android. https://developers.facebook.com/docs/app-events/getting-started-app-events-android#auto-events, 2021. 2021/02/01.

[25] Á Feal, P Calciati, N Vallina-Rodriguez, C Troncoso, and A Gorla. Angel or devil? a privacy study of mobile parental control apps. *PoPETS*, 2020.

[26] Flurry. Flurry monetization and gdpr. https://developer.yahoo.com/flurry/docs/publisher/gdpr/, 2021. 2021/02/01.

[27] Flurry. Manual flurry android sdk integration. https://developer.yahoo.com/flurry/docs/integrateflurry/android-manual/, 2021. 2021/02/01.

[28] Flurry. Android sdk release notes. https://developer.yahoo.com/flurry/docs/releasenotes/android/#version-6-3-0-03-22-2016, 2021. 2021/02/01.

[29] forbrukerradet.no. Out of control. https://fil.forbrukerradet.no/wp-content/uploads/2020/01/2020-01-14-out-of-control-final-version.pdf, 2020/09/02.

[30] GDPR. Art. 4 Definitions. URL https://gdpr.eu/article-4-definitions/.

[31] GDPR. Art. 6 Lawfulness of processing, 2021. URL https://gdpr.eu/article-6-how-to-process-personal-data-legally/. 2021/02/01.

[32] GDPR. Art. 7 Conditions for consent, 2021. URL https://gdpr.eu/article-7-how-to-get-consent-to-collect-personal-data/. 2021/02/01.

[33] Google. Obtaining consent with the user messaging platform. URL https://developers.google.com/admob/ump/android/quick-start.

[34] Google. Play console help for android developers - advertising id. https://support.google.com/googleplay/android-developer/answer/6048248?hl=en, 2021/02/02.

[35] IAB Europe GDPR Implementation Group. The definition of personal data - working paper 02/2017. https://iabeurope.eu/wp-content/uploads/2019/08/20170719-IABEU-GIG-Working-Paper02_Personal-Data.pdf, 2017.

[36] P Hornyack, S Han, J Jung, S Schechter, and D Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *CCS*, 2011.

[37] R Leenes and E Kosta. Taming the cookie monster with dutch law–a tale of regulatory failure. *Computer Law & Security Review*, 2015.

[38] F Li, Z Durumeric, J Czyz, M Karami, M Bailey, D McCoy, S Savage, and V Paxson. You've got vulnerability: Exploring effective vulnerability notifications. In *USENIX Security*, 2016.

[39] L Li, T F Bissyandé, M Papadakis, S Rasthofer, A Bartel, D Octeau, J Klein, and L Traon. Static analysis of android apps: A systematic literature review. *Information and Software Technology*, 2017.

[40] M Li, W Wang, P Wang, S Wang, D Wu, J Liu, R Xue, and W Huo. Libd: Scalable and precise third-party library detection in android markets. In *ICSE*, 2017.

[41] X Liu, Y Leng, W Yang, W Wang, C Zhai, and T Xie. A large-scale empirical study on android runtime-permission rationale messages. In *VL/HCC*, 2018.

[42] Z Ma, H Wang, Y Guo, and X Chen. Libradar: Fast and accurate detection of third-party libraries in android apps. In *ICSE*, 2016.

[43] C Matte, N Bielova, and C Santos. Do cookie banners respect my choice?: Measuring legal compliance of banners from iab europe's transparency and consent framework. In *SP*, 2020.

[44] myTarget. Privacy and gdpr. https://target.my.com/help/partners/privacygdpr/en, 2021. 2021/02/01.

[45] T T Nguyen, D C Nguyen, M Schilling, G Wang, and M Backes. Measuring user perception for detecting unexpected access to sensitive resource in mobile apps. In *ASIA CCS*, 2020.

[46] NOYB – European Center for Digital Rights. Google: If you don't want us to track your phone – just get another tracking id! https://noyb.eu/en/complaint-filed-against-google-tracking-id, 2021. 2021/01/17.

[47] objection. Runtime mobile exploration. https://github.com/sensepost/objection, 2021. 2021/01/17.

[48] X Pan, Y Cao, X Du, B He, G Fang, R Shao, and Y Chen. Flowcog: context-aware semantics extraction and analysis of information flow leaks in android apps. In *USENIX Security*, 2018.

[49] Data Protection Working Party. Guidelines on consent under regulation 2016/679 (wp259rev.01). https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=623051, 2016. 2020/09/04.

[50] Google Policy. Monetization and ads. https://support.google.com/googleplay/android-developer/answer/9857753/#zippy=, 2021. 2021/01/17.

[51] publicsuffixlist. publicsuffixlist. https://github.com/ko-zu/psl, 2021. 2021/05.

[52] A Razaghpanah, R Nithyanand, N Vallina-Rodriguez, S Sundaresan, M Allman, C Kreibich, and P Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *NDSS*, 2018.

[53] J Reardon, Á Feal, P Wijesekera, A E B On, N Vallina-Rodriguez, and S Egelman. 50 ways to leak your data: An exploration of apps' circumvention of the android permissions system. In *USENIX Security*, 2019.

[54] General Data Protection Regulation. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46. *OJEU*, 2016.

[55] J Ren, A Rao, M Lindorfer, A Legout, and D Choffnes. Recon: Revealing and controlling pii leaks in mobile network traffic. In *MobiSys*, 2016.

[56] I Reyes, P Wijesekera, J Reardon, A E B On, A Razaghpanah, N Vallina-Rodriguez, and S Egelman. "won't somebody think of the children?" examining coppa compliance at scale. *PETS*, 2018.

[57] I Sanchez-Rola, M Dell'Amico, P Kotzias, D Balzarotti, L Bilge, P-A Vervier, and I Santos. Can i opt out yet? gdpr and the global illusion of cookie control. In *Asia CCS*, 2019.

[58] R Slavin, X Wang, M B Hosseini, J Hester, R Krishnan, J Bhatia, T D Breaux, and J Niu. Toward a framework for detecting privacy policy violations in android application code. In *ICSE*, 2016.

[59] B Stock, G Pellegrino, C Rossow, M Johns, and M Backes. Hey, you have a problem: On the feasibility of large-scale web vulnerability notification. In *USENIX Security*, 2016.

[60] B Stock, G Pellegrino, F Li, M Backes, and C Rossow. Didn't you hear me? - towards more successful web vulnerability notifications. In *NDSS*, 2018.

[61] S Traverso, M Trevisan, L Giannantoni, M Mellia, and H Metwalley. Benchmark and comparison of tracker-blockers: Should you trust them? In *TMA*, 2017.

[62] M Trevisan, S Traverso, E Bassi, and M Mellia. 4 years of eu cookie law: Results and lessons learned. *PETS*, 2019.

[63] Unity3d. Gdpr compliance. https://docs.unity3d.com/Packages/com.unity.ads@3.3/manual/LegalGdpr.html, 2021. 2021/02/01.

[64] Unity3d. Privacy policy. https://unity3d.com/legal/privacy-policy, 2021. 2021/02/01.

[65] P Vallina, Á Feal, J Gamba, N Vallina-Rodriguez, and A F Anta. Tales from the porn: A comprehensive privacy analysis of the web porn ecosystem. In *IMC*, 2019.

[66] Vungle. Vungle sdk for android. https://github.com/Vungle/Android-SDK/blob/master/CHANGELOG.md, 2021. 2021/02/01.

[67] Y Wang, H Zhang, and A Rountev. On the unsoundness of static analysis for android guis. In *PLDI*, 2016.

[68] webshrinker. webshrinker. https://www.webshrinker.com/, 2021. 2021/05.

[69] C Weir, B Hermann, and S Fahl. From needs to actions to secure apps? the effect of requirements and developer practices on app security. In *USENIX Security*, 2020.

[70] P Wijesekera, A Baokar, A Hosseini, S Egelman, D Wagner, and K Beznosov. Android permissions remystified: A field study on contextual integrity. In *USENIX Security*, 2015.

[71] Z Yang, M Yang, Y Zhang, G Gu, P Ning, and X S Wang. Appintent: Analyzing sensitive data transmission in android for privacy leakage detection. In *CCS*, 2013.

[72] L Yu, X Luo, X Liu, and T Zhang. Can we trust the privacy policies of android apps? In *DSN*, 2016.

[73] S Zimmeck, Z Wang, L Zou, R Iyengar, B Liu, F Schaub, S Wilson, N M Sadeh, S M Bellovin, and J R Reidenberg. Automated analysis of privacy requirements for mobile apps. In *NDSS*, 2017.

A Email Notification Template

Dear $developer team,

We are a team of academic researchers from the $affiliation, conducting a research project on user consent and GDPR (EU General Data Protection Regulation) compliance of mobile apps. Please note that this email is part of an academic research project and is not meant to sell any products or services.

As part of our analysis, we investigate the sharing of users' personal information (e.g., user IP address, persistent identifiers, tracking identifiers) with third-party services to show personalized or behavioral advertising. Based on our analysis, your app shares some personal user information to such services without obtaining prior explicit consent from users. We have prepared a detailed report on the analysis methodology, the data being sent out, and the parties involved. You can access this through our (password-protected) Web interface at $report_url (please do not publish this URL as it is personalized for your app). By analyzing the legal documents (e.g., the terms of service, privacy policies, developer guidelines, and contracts) provided by the third-party services in question, we concluded that your app might be non-compliant with the consent requirements by the GDPR [1]. In most cases, in order to be legally compliant, an app is required to obtain explicit consent from users situated in the European Union before sharing users' personal data with third parties for personalized ads, if those third parties act as a data controller. Please note that we do not offer a conclusive legal assessment or consultancy on an individual app's compliance as there might be an alternative lawful basis present for data sharing with a third party other than consent. As this email is part of a research project in which we are trying to understand the reasons for GDPR compliance issues of mobile apps in the wild, it would be immensely helpful to provide us with feedback regarding your apps.

(1) Were you aware of the types of data that are being collected and transmitted when you include third-party SDK(s) into your apps? Were you aware that these types of data could be considered personal data under the GDPR?

(2) Are there specific reasons why your app does not implement explicit consent?

(3) Are there any changes you plan to apply to remedy the outlined issues? What type of support (e.g., documentation or automated tools) would be beneficial for you?

Should you have further questions or wish not to receive any further communication, please contact us, and we will diligently follow the request.

Best regards
$researchers
[1] (The full-text reference of [54] was added in the email.)

B Manual Version Analysis

We first analyze the app network traffic to find parameters that indicate the SDK version. These are: `unity3d.com` (*x-unity-version*, *sdkversion*, *sdk_version_name*, *sdk_ver*, *sdkversion*); `flurry.com` (*fl.sdk.version.code*); `vungle.com` (*user-agent*, *sdk*); `my.com`: (*mytracker_ver*); `amazon-adsystem.com` (*adsdk*). Further, we manually verify the results with the identified SDK(s) release notes. We then used this knowledge to detect versions all apps that sent PD.

No.	Name	Domain Names	GDPR Solution	Earliest consent support SDK version	Note
1	Facebook	facebook.com	Do not require consent	—	Under GDPR, developer are required to obtain end User consent before sending data via our SDK
2	Unity	unity3d.com	Consent API	3.3.0	
3	Flurry	flurry.com	Consent API	10.0.0	
4	AppsFlyer	appsflyer.com	Do not require consent	—	Providing APIs for opt-in and opt-out
5	Chartboost	chartboost.com	Consent API	7.3.0	
6	SuperSonic	supersonicads.com	Consent API	6.7.9	
7	StartApp	startappservice.com	Consent API	1.2.0	
8	AdColony	adcolony.com	Consent API	3.3.4	
9	Branch	branch.io	Do not require consent	—	Providing APIs for opt-in and opt-out
10	Vungle	vungle.com	Consent API	6.2.5	
11	Applovin	applovin.com	Consent API	8.0.1	
12	Tapjoy	tapjoy.com	Consent API	11.12.2	GDPR-compliant based on "legitimate interest"
13	ConsoliAds	consoliads.com	Consent API	—	
13	BidMachine	bidmachine.io	Consent API	1.3.0	
14	MoPub	mopub.com	Consent API	5.0.0	
15	Presage	presage.io	—	—	
16	AdinCube	adincube.com	—	—	
17	Ogury	ogury.io	Consent API	4.1.4	
18	Amazon	amazon-adsystem.com	—		
19	InMobi	inmobi.com	Consent API	7.1.0	
20	Adbrix	ad-brix.com	Do not require consent	—	Providing APIs for opt-in and opt-out
21	Adbrix	adbrix.io	Do not require consent	—	Providing APIs for opt-in and opt-out
22-23	Tenjin	tenjin.com, tenjin.io	Do not require consent	-	Providing APIs for opt-in and opt-out
24	Mobvista	rayjump.com	—	—	
25	Tenjin	tenjin.io	—	—	
26	Appnext	appnext.com	Consent API	2.3.0	
27	Pollfish	pollfish.com	Do not require consent	—	Have to provide disclosure for using this SDK
28	My.com	my.com	—	—	
29	Soomla	soom.la	Consent API	—	Should be a bad practice since default behavior is TRUE
30	Localytics	localytics.com	Do not require consent	2.1.0	Providing APIs for opt-in and opt-out
31	Tapdaq	tapdaq.com	Consent API	6.2.2	
32	Leanplum	leanplum.com	Do not require consent	-	Providing APIs for opt-in and opt-out
33	Criteo	criteo.com	Consent Management Provider	3.7.0	
34	WebEngage	webengage.com	—		
35	Smart AdServer	smartadserver.com	Consent Management Provider	1.2.0	
36	Umeng	umeng.com	—	—	
37	omtrdc.net	omtrdc.net	—	—	
38	MobiRoller	mobiroller.com	—	—	
39	Kiip	kiip.me	not clear	—	Due to GDPR regulations, NinthDecimal is now blocking all ad requests from the affected EEA regions.
40	Adtrace	adtrace.io	Do not require consent	-	Have to provide disclosure for using this SDK
41	Airpush	airpush.com	—	—	
42	Inloco	inlocomedia.com	Consent API	4.0.0	
43	PubMatic	pubmatic.com	—	—	
44	Tapstream	tapstream.com	—	—	
45	YovoAds	yovoads.com	—	—	

Table 5: Companies detected as ad-related, for which our analysis of legal documents indicate they act as data controllers.