The OK Is Not Enough:
A Large Scale Study of Consent Dialogs in Smartphone Applications

Simon Koch
TU Braunschweig
simon.koch@tu-braunschweig.de

Benjamin Altpeter
Datenanfragen.de e.V.
benni@datenanfragen.de

Martin Johns
TU Braunschweig
m.johns@tu-braunschweig.de

Abstract

Mobile applications leaking personal information is a well established observation pre and post GDPR. The legal requirements for personal data collection in the context of tracking are specified by GDPR and the common understanding is, that tracking must be based on proper consent. Studies of the consent dialogs on websites revealed severe issues including dark patterns. However, the mobile space is currently under-explored with initial observations pointing towards a similar state of affairs. To address this research gap we analyze a subset of possible consent dialogs, namely privacy consent dialogs, in 3006 Android and 1773 iOS applications. We show that 22.3% of all apps have any form of dialog with only 11.9% giving the user some form of actionable choice, e.g., at least an accept button. However, this choice is limited as a large proportion of all such dialogs employ some form of dark pattern coercing the user to consent.

1 Introduction

Mobile applications do collect a large amount of personal data of the user and transmit those to third parties. This behavior has been well documented both on Android [40, 42, 63, 73] as well as iOS [33, 39, 51]. The introduction of the GDPR as the European privacy legislation was supposed to better protect personal data of consumers and outlaw underhanded data collection. However, recent work studying the collection behavior of current mobile applications casts doubt on the overall impact and shows data collection is still happening [48, 49, 58].

However, even though collection of personal data can be questionable from an ethical standpoint the legislature does allow for collection. It is legal to collect data, e.g., if it is strictly required for the functionality of the applications or legally required. However, for the popular purpose of tracking, prior consent for data collection has to be given. The GDPR sets out clear rules on how a request for consent has to be structured and collected, explicitly stating that consent has to be voluntary and must not be coerced. Recent fines for Meta by the Irish Data Protection Commission do underline this principle [6].

Studies analyzing cookie consent dialogs on the web revealed that a large portion do not conform to the stated rules. Even worse, the analyzed dialogs widely employed stylistic choices to coerce or nudge a user towards giving consent [54, 55, 60, 71]. Such design choices have been termed 'Dark Patterns'. The overwhelming presence of Dark Patterns in consent dialogs led to the European privacy advocacy NGO *noyb*[1] launching two campaigns against deceptive cookie banners [16, 18] resulting in websites changing their dialog designs. This shows that effort towards the study of consent dialogs and action based on this information can improve the self-determination of users concerning their privacy choices.

Assuming tracking, mobile applications also have to collect the consent by the user prior to data collection, according to the common understanding of the GDPR. Mobile applications, thus, tend to push a *privacy consent dialog* on first start to get the user to consent to their data collection. Furthermore, Apple [3] and recently Google [10] require developers to provide privacy labels, informing the user of the intended data collection, giving an initial impression of mobile applications likely being honest concerning private data collection. However, recent work by Koch et al. [48] has shown that privacy labels on iOS are not being enforced and flouted by developers. Additionally, they performed a visual inspection of started mobile applications and report that there was a lack of privacy consent dialogs for applications that self-declare themselves to collect personal information in their privacy label as well as applications displaying privacy consent dialogs that employed Dark Patterns. Similar initial observations have been made for Android by Nguyen et al. [58]. However, no conclusive research into the design and effect of privacy consent dialogs in mobile appliations has been done yet.

We approach this research gap and develop a tool chain that covers both iOS and Android applications to extract and

---

[1] https://noyb.eu/en

analyze a subset of consent dialogs, namely privacy consent dialogs. We then apply this tool chain to 3006 Android and 1773 iOS applications to study the amount of privacy consent dialogs in mobile applications, their design choices, and the ratio of possibly GDPR-non-conforming ones. Furthermore, we leverage our tool chain to test the effect consenting or declining a dialog has on the transmitted personal information to known trackers.

Overall, we have two key contributions:

- A novel and mobile OS-agnostic privacy consent dialog analysis tool chain to download, run, and analyze mobile applications
- A large scale study of 3006 Android and 1773 iOS applications concerning:
  (a) the structure of a privacy consent dialog;
  (b) the effect of consenting or declining to a dialog

In the remainder of the paper, we first detail prior work in the area of mobile privacy and consent dialogs (Section 2). We follow up by detailing the legal background governing requirements for consent dialogs (Section 3). Based on those requirements, we first collect a large dataset of both Android and iOS Apps (Section 4), analyze the usage of Consent Management Platforms (Section 5), and develop an analysis tool chain to interact with mobile applications, detect consent dialogs, and extract the required features for our analysis (Section 6). We then apply our tool chain and detail the observed Dark Patterns as well as data transmission behavior (Section 7), followed by a discussion (Section 8). Finally, we summarize our key contributions and results (Section 9).

2   Related Work

Current work related to our research up until now can be split into work concerning mobile privacy on iOS [33,39,48,50,51] and Android [42, 57, 58, 62, 63, 69, 73], or comparing the two [49], as well as work concerned with analyzing privacy policies and dialogs or frameworks on the web [37,45–47,54–56,60,61,70,71] and in mobile applications [63,67,74].

Dynamic traffic analysis has been leveraged both on iOS as well as on Android and shown that apps send and receive data within the first seconds of launch, and share data with third-party libraries [48,51,58,68], detecting similar data sharing behavior regardless of operating system [49]. Ren at al. detected an increase in data collection across multiple app versions over time [62]. This shows that personal data collection in mobile applications is prevalent before and after introduction of the GDPR. Kollnig et al. analyzed Apps before and after the introduction of iOS 14 and its corresponding rules on privacy compliance, they found that tracking and data collection is still prevalent regardless of this change introduced by Apple [50]. Static analysis and symbolic execution have been used to detect leaks of sensitive information in mobile

apps [39,42,73], demonstrating that inferences concerning privacy-respecting behavior can be gained this way.

Work analyzing privacy policies in mobile apps showed mismatches between self-declared privacy policy and actual code behavior [67,74] and that privacy policies do not lead to improved privacy of the user [63]. Orthogonal work analyzing web consent dialogs and privacy policies mirrors those insights and indicates widespread violation of proper consent requirements for dialogs even when belonging to a consent management platform [54,55,60,71] and that tracking widely happens before any user interaction or even despite rejection of data collection [61,70]. Additional work focusing on the impact of the GDPR on privacy dialogs and policies showed changes due to the GDPR [37,46] but given the previously detailed plethora of work after 2018 on GDPR violations those changes clearly did not have a positive effect on actual privacy protection. Concurrent independent work by Nguyen et al. looked at mobile consent dialogs on Android but did not cover iOS [59]. To analyze privacy policies, several tool chains have been developed leveraging ML to extract features and enable querying the extracted model [45, 47]. Finally, Maryam et al. analyzed 100 popular websites online and in their corresponding mobile apps concerning tracking and privacy notices and found major inconsistencies for essentially the same service [56].

Overall, those works paint a picture of questionable design choices in consent dialogs, with clear indications that even if the user is presented with a choice, that choice is often ignored, a bleak picture for users concerned about their privacy on the web. However, up until now those works focus on the web and it is unexplored whether the situation on the web also corresponds to the situation in mobile applications. Furthermore, prior work demonstrates that data collection via apps on mobile devices is omnipresent and provides the technological basis for our work contrasting data collection on mobile devices with privacy consent dialog interaction.

3   The Makings of Legal Consent

Due to localization and consequent expertise, we perform our analysis and research in the context of European Union (EU) laws and, thus, the EU GDPR [24], which does not necessarily reflect the UK GDPR post BREXIT.

The GDPR is the EU privacy law protecting consumers from unwanted data collection and processing. It went into effect in 2016 with a grace period until 2018 [31]. While the GDPR as a law gives general direction on what constitutes personal data, processing, and consent, its specific interpretation and enforcement is done by the Data Protection Agencies (DPAs). The GDPR is not only binding for companies within the EU but also includes companies outside the EU if they process data of people within the EU related to the offering of goods and services, or the monitoring of behavior [66]. In this section, we first introduce required vocabulary, then state the

criteria that are required for consent, and finish with lessons learned summarizing the key points required to understand the legal context of our research.

We want to emphasize that we present the legal matter in a clear cut fashion to ease understanding and to focus on the key takeaways. However, this must not be understood as if the legal landscape is indeed unified in their opinion. Unless there is a verdict by the highest court (and even sometimes after) there are always possibly valid opposing opinions. Thus, each statement below should be read with an asterisk as a reminder that there is an ongoing discussion. Even though we did our due diligence, our summary should not be construed as legal advice as we are not legal scholars or practitioners.

3.1 What Constitutes Personal Data and Processing?

Article 2(1) GDPR specifies what constitutes personal data and processing. *Personal data* is any information relating to an identified or identifiable natural person and *processing* is defined as any operation performed on personal data, including collection, recording, and storage. This means that in essence, any data that can be connected to a natural person (*data subject*) and is accessed by an organization (*data controller*) falls under the GDPR. This even applies to pseudonymous data (Recital 26(2) GDPR), i.e., data that can be linked to a person in combination with other data (Article 4(5) GDRP). Rulings have even applied this to IP addresses as those could be attributed to a person by obtaining ISP records [13].

The GDPR also defines the role of *data processor*. While the data controller is the party that determines the purposes and means for which data is collected (Article 4(7) GDPR), a data processor is any party that processes the data on behalf of the controller (Article 4(8) GDPR). The GDPR requires data controllers to enter into data processing agreements with any employed data processor (Article 28(3) GDPR), to document this relationship, containing specifications on the form of processing. However, the liable party for the processing is the data collector.

Processing of properly anonymized data is not restricted by the GDPR (Recital 26(5) GDPR). Ways of such an anonymization can include statistical aggregation, or collection of information not related to a person such as application settings. However, research has shown that even a few benign data points presumed to be anonymous can be leveraged to uniquely identify a person [35, 36, 43, 44, 64] and that fingerprinting via settings can be used to identify a unique device [38]. The GDPR requires considering such indirect deanonymization before collection [52].

3.2 What Constitutes Proper Consent?

Processing of any personal data is generally prohibited by the GDPR unless there is a legal basis, with the GDPR providing a conclusive list (Article 6(1)(a-f) GDPR): (a) **Consent** by the data subject, (b) **contractual necessity**, (c) **legal necessity** such as know your customer laws, (d) necessity to protect a person's **vital interests**, (e) tasks in the **public interest**, usually by public authorities, (f) **legitimate interests** by the controller outweighing the data subject's interests and fundamental rights.

In the context of mobile applications and our analysis focus of traffic directed at known trackers (Section 6) lit. c, d, and e do not apply. Article 6(1)(c) GDPR does not apply as we are not opening up a bank account or otherwise moving into a business relationship with the company developing the app, we simply start the app and at most interact with the privacy dialog. Article 6(1)(d) GDPR does not apply as by simply starting the app one cannot construe any vital necessity or interest of ours to protect. Article 6(1)(e) GDPR does not apply as we are analyzing the traffic directed at known trackers which does not include any official authority. Furthermore, according to data protection authorities lit. b, and f do not apply as a legal basis for tracking either [11, 41]. Consequently, only consent remains.

The GDPR lists five basic conditions that need to be met for consent to be considered valid (Article 4(11) GDPR). Consent has to be: **(1) freely given** as data subjects need to have a genuine and free choice to refuse or withdraw (Recital 42 GDPR); **(2) specific** for each purpose of processing (Recital 32 GDPR); **(3) informed** such as the data subject knows at least the identity of the controller and the purposes of the processing (Recital 42 GDPR); **(4) unambiguous** such that it is clear, concise, and not unnecessarily disruptive to the use of the service (Recital 32 GDPR); and an **(5) affirmative action**, with silence, pre-ticked boxes, or inactivity not constituting consent (Recital 32 GDPR).

Article 7 GDPR defines further criteria for the legality of consent, specifically that consent can only be obtained through a declaration also concerning other matters if it is clearly distinguishable from those other matters (Article 7(2) GDPR). Finally, Article 7(3) GDPR requires that consent has to be withdrawable at any time, that the data subject has to be informed that they can withdraw their consent at any time, and finally that withdrawing consent needs to be as easy as giving it.

Any collection based on consent in violation of those criteria is risking a GDPR violation.

3.3 Resulting Criteria Catalog for Valid Consent Dialogs

We have discussed the wording of the GDPR, what constitutes personal data and processing, as well as why data collection

for tracking in the context of mobile applications requires prior consent. Furthermore, we established criteria on what constitutes consent under the GDPR and that data collection for tracking, prior to or based on consent lacking the laid out criteria risks running afoul of the GDPR.

Part of the requirements for consent are subjective or at least require some form of context understanding such as the requirement for the consent to be specific, informed, or unambiguous. A machine cannot decide whether a given text contains sufficient information for the data subject to fulfill either of those requirements. However, a machine can analyze design choices that affect affirmative action and indicate whether a human user is coerced toward consenting.

We, thus, define five criteria based on our understanding of the GDPR that are required (i.e., a lower bound) for consent dialogs to be sufficient and that can be assessed by a machine:

1. refusing consent has to be possible with the same number of clicks as consenting [14, 21, 41]

2. buttons have to have clear and non ambiguous labels [14, 41]

3. the dialog may not only provide a link or refer to information deeper in the consent flow [14, 21]

4 it must be possible to directly refuse [41]

We consider our criteria 1, 2, and 3 to be backed by [41], which is a verdict of the LG Rostock, i.e., a German court of law, and thus a legal requirement. Criteria 4 is only backed by [41] on a FAQ provided by the bureau of data protection of Baden-Württemberg, and should thus be considered a best practice until a corresponding verdict has been delivered. The lack of either of those design criteria increases the work required by a user to refuse consent and thus nudges them towards consent. Previous literature has termed such design decisions in consent dialogs as 'Dark Patterns' [54, 55, 60, 71]. We use this phrase accordingly to encompass violations of our set design criteria as each violation represents an additional challenge for the user to not give consent.

4 App Acquisition

Automatically downloading app files that can be run on a mobile phone is not trivial. The intended process by both iOS and Android is to use the provided app store on the phone as a logged-in user. In the app store, you can browse a selection of apps and the installation process is triggered by choosing to install the app within the app store. This process even registers the app to the user on iOS and an app that is not registered to a user account cannot be run by that account. Android does not have such a process and any application can be installed and run if one has access to the corresponding app binary (APK).

There exist different solutions to access large volumes of Android APKs [34] and it is possible to individually download them [22]. The main challenge here is gaining information on the ranking of apps to focus any analysis on popular and thus widely used apps. For iOS, only manual solutions are known. Koch et al. [48] leveraged the 3u tool[2] to manually download a large set of iOS apps but this approach is fairly cumbersome and labor-intensive. Consequently, we have two challenges: accessing the ranking of apps and subsequent download of identified apps of interest.

We solved both challenges by leveraging hidden API endpoints to register apps to the corresponding accounts used for analysis as well as to download the corresponding APK and IPA files for Android and iOS, respectively. Neither Google nor Apple provide publicly advertised endpoints to automatically download a list of the current most popular apps or the apps themselves. However, using open source tools and adapting existing web APIs, we are able to automate app acquisition of the top 100 apps across different categories for both Android and iOS. Those categories encompass all main categories of both Android and iOS.

Android To gain access to the top 100 apps across different categories on Android, we leverage the published charts by Google themselves. This ranking is not linked anywhere we know and was found by accident through search engines. We use the access point to access lists across the different categories whose IDs can be extracted via the Google Play Store interface[3] and used the PlaystoreDownloader [22] to download and store the corresponding APKs.

iOS By observing iTunes traffic, we discovered the required endpoint. We leverage the returned list for each category to download apps using IPATool [12]. However, IPATool used to only be able to download already purchased apps. We overcame this limitation by further traffic analysis and reverse engineering of tools capable of buying but not downloading. Those insights have already been fed back to the community and are now part of the current IPATool.

For our consent dialog detection and classification, we used our automated app acquisition to download the top 100 of each primary category for both Android and iOS. This resulted in the successful download of 3006 Android and 1773 iOS applications. We performed the downloads from 2022-11-17 to 2022-11-18 for Android and from 2022-12-04 to 2022-12-05 for iOS.

5 Consent Management Platforms

Consent Management Platforms (CMP) are off-the-shelf solutions to ensure legal compliance for data collection. They allow developers to forgo consent implementations and leverage standardization. CMPs are potentially useful for researchers as there is only a limited amount and each has a standardized method of recording consent. They also provide data collec-

---

[2]http://www.3u.com/
[3]https://play.google.com/store/apps

tors with a way to check if they are cleared for collection and processing purposes.

Given that a high prevalence of CMP usage could allow for a focused and precise analysis of privacy consent dialogs and their effect we conduct a static analysis of application binaries to ascertain their prevalence. We show that CMPs are not popular enough to pivot our privacy consent dialog analysis around them.

5.1 Prevalence of CMPs

Theoretically, CMPs allow developers to ensure that they stay within the boundaries of the law. However, due to their high configurability, this might not always be the case [30] and recent rulings have declared one large framework for CMPs to actually be in breach of the GDPR [7].

As CMPs provide a high degree of standardization across apps that use them, they provide an easy to leverage approach towards an analysis framework. However, a sufficiently high usage across mobile applications has to be present.

Analysis of the web has shown that between 6% and 13% of European websites deploy CMPs [46, 54] and roughly 40% of the top 10k US websites [5]. If this is also the case across sufficient mobile applications, this could ease our detection and analysis of privacy consent dialogs.

The Interactive Advertising Bureau (IAB) has designed the Transparency and Consent Framework (TCF) that intends to work across multiple CMPs. According to their website, they have 79 web CMPs compliant with their framework and 32 with their mobile framework. This makes the TCF and any TCF-compliant CMP a great target for further consent and consent dialog analysis. However, even usage of the TCF might not ensure compliance with the GDPR, as a recent ruling declared it not GDPR-compliant [7].

5.2 Static Analysis of CMP Usage

To estimate the popularity of CMPs and TCF-compliant CMPs, we use static analysis to detect whether an app potentially uses a CMP. A large prevalence of CMPs would allow us to leverage the corresponding standardization. To detect potential usage of a CMP, we search the included libraries of the APKs and IPAs with a similar approach to what the Exodus Privacy project uses to check for the presence of tracking libraries [8]: We extract a list of the contained libraries in the IPA and APK application packages and search for known CMP names. This provides a rough estimate on the minimal amount of CMPs being in use. We do not perform any further static analysis, e.g., analyzing liveness of the library code or configuration parameter.

Android: To gain access to the included libraries, we use `dexdump` for Android APK files that can statically extract the names of the included classes from an APK. Those names are then searched for known CMPs.

iOS: An IPA is a essentially a ZIP file that that stores all included libraries[4]. The included subdirectory names are then searched for known CMP identifiers.

5.3 Detected CMP Usage

We first curated a list of 26 different CMPs based on the official IAB TCF vendor list [26] as well as external resources [27, 28]. This list is a best effort approach towards detecting CMPs but not necessarily exhaustive. We were able to determine the CMP vendor names for 20 TCF-compliant CMPs and 6 additional CMPs. We searched 4779 apps and detected 252 (5.1%) applications containing libraries or classes matching our curated list.

We detected IAB TCF-compliant CMPs in 252 apps (5.1%). 199 (6.6%) Android apps and 53 (2.7%) iOS apps contained at least one CMP.

Note that simply including a CMP-related class or library does not necessarily mean actual (or proper) usage.

5.4 Lessons Learned

Overall, only a small fraction of apps use on of our CMPs. All of the detected CMPS were IAB compliant but with 5.1% of all apps too low to use for our purposes. As our analysis was static, we do not know the amount of apps actually using the included CMPs and we expect that the amount of actively used CMPs is even lower. Consequently, any analysis done based on CMPs would not yield meaningful results and miss a large fraction of applications.

6 Analyzing Mobile Consent

We have established that focusing our analysis of privacy consent dialogs by leveraging CMPs does not work, thus, a more complex analysis tool chain is required. Our analysis methodology dynamically identifies and analyzes existing consent solutions deployed by apps. We run each app, monitor network traffic, and extract and interact with any presented consent dialog to analyze how consent dialogs affect data transmission and what design choices were made with regard to coercing a user to give consent.

We start by detailing our dynamic analysis and how we run, monitor, and interact with each mobile application. Our goal is to run each app, monitor traffic, as well as analyze, and interact with privacy consent dialogs. For this, we first detail how we designed our *app execution and traffic recording* into which we hook our *consent dialog extraction and analysis* leveraging Appium [4].

We are dedicated to open source research and will open source our tools described below[5].

---

[4]The subdirectory is /Payload/<app name>.app/Frameworks/
[5]https://github.com/the-ok-is-not-enough

6.1 App Execution and Traffic Recording

Execution of apps for iOS and Android face different challenges on a first view as either is their own operating system.

For iOS we leverage a jailbroken iPhone, libimobiledevice [15], Frida [9] for automation, and SSL Kill Switch 2 [25] to disable SSL certificate validation including certificate pinning. A pre-configured mitmproxy [17] is used to collect traffic. We mirror this setup for Android using a Galaxy A13 as a device, Frida and objection [19] for automation and disabling of SSL checking and certificate pinning.

Either framework essentially executes four steps:

1. **Install** the apps and give permissions. This uses the approach described by Koch et al. [48] using `ideviceinstaller` and the iPhone internal permission database for iOS or `adb` on Android that also allows granting permissions.
2. **Run** the apps using Open [20] on iOS and objection on Android and perform any required interaction.
3. **Collect** the traffic generated while the app is running via a pre-configured proxy running mitmproxy.
4. **Remove** the app using again `ideviceinstaller` on iOS and `adb` on Android.

6.2 Consent Dialog Extraction and Analysis

We now address the question of how we detect and analyze displayed privacy consent dialog in an open app. Our first step is to introduce a taxonomy, classifying privacy consent dialog into three major types. Based on this taxonomy, we explain how we distinguish between those types and perform further analysis to detect deviation from the design requirements established in Section 3.

6.2.1 Taxonomy of Consent Dialogs

In Section 5, we analyzed the popularity of different CMPs and had to come to the conclusion that no set of CMPs reaches sufficient popularity to base our analysis on. Consequently, we cannot use key elements of different CMPs to identify, extract, and analyze displayed privacy consent dialogs. We, thus, conducted an initial manual study on a quarter of our apps to analyze observable privacy consent dialogs. Based on this analysis, we defined a taxonomy consisting of three different types of privacy consent dialogs we observed being deployed:

Link: A simple link referencing the privacy policy. While this might inform the user about an existing privacy policy a link can clearly not fulfill our stated design requirements. A link differs from a notice or a dialog by only being a link, without any form of related buttons or context information.

Notice: A notice is a text, e.g., in the form of a banner. The text informs the user about the app's privacy practices in some form but does not offer any form of consent or rejection.

Notices commonly only state that by continuing to use the app, the user consents to the privacy policy and data collection, whereas a dialog provides more information. However, the line between a proper dialog using a 'continue' button and a message, informing you that using the app entails agreeing is thin. The difference between a notice and a dialog resides in the purpose of the contained buttons, specified by their label. A notice only offers buttons that (also) serve another purpose than agreeing to the privacy policy, most commonly they are the log in buttons.

Proper Dialog: A dialog is text, e.g., in the form of a banner, that not only informs the user but also actively requires agreement in form of interaction, i.e., buttons. This is the only type of privacy consent dialog that can potentially fulfill our stated design requirements. However, a more detailed analysis is required, e.g., can a user actually reject data collection or if they are nudged towards giving consent.

6.2.2 iOS based Consent Dialogs

Apple introduced privacy improvements for iOS starting with version 14.5 [1]. Apps do not have default access to the IDFA, a phone global identifier, anymore if a user disables the corresponding setting. Furthermore, an app may now ask for permission to track a user for advertisement purposes. A positive response for this request enables the App to get the IDFA. According to the documentation such permission also entails generic tracking such as 'displaying targeted advertisement' or 'sharing device location data or email lists with a data broker'. Operating system based consent dialogs are out of scope and we deny each app this permission individually after installation.

6.2.3 Differentiating Between Consent Dialogs

After having established a taxonomy we proceed to detect, classify, and analyze presented privacy consent dialogs in running mobile apps (i.e., during step 2). This is done in a two step process: (1) we *sort* a detected privacy consent dialog into our taxonomy; (2) if we detected a proper dialog, we continue to *analyze* it to check for any problematic design patterns casting doubt on the validity of the privacy consent dialog.

As we detect dialogs and interactable elements based on text we need to be able to extract text elements of each app. We are using Appium for this [4]. The Appium framework provides a generic interface to elements of both Android and iOS apps. This interface affords extracting attributes of app elements such as the contained text. Our dialog analysis process is solely based on the text content of an element to stay operating system agnostic as attributes such as class names or id patterns differ between Android and iOS.

(1) Dialog Classification Based on our initial manual study, we compiled a set of common phrases such as *we*

care about your privacy or *by continuing to use our app, you acknowledge that we may process your data in line with our data protection statement* and designed regular expressions capturing essential words or phrases but leaving out app-specific wordings. One example for such a regexp would be `/have read( and understood)?[^.]{3,35} (privacy|cookie|data protection|GDPR) (policy|notice|information|statement)/`.

This regexp captures common essential phrases of a privacy policy text, i.e., *have read and understood*, as well as keywords, e.g., *privacy policy*, *GDPR information*, or *GDPR statement*. However, it also leaves space in between those phrases and keywords for app- or vendor-specific text while ensuring that those key elements are within the same sentence. This allows us to detect the presence of any type of privacy consent dialog that contains phrases related to privacy and the GDPR. Our approach catches any dialog that matches our keywords and we do not analyze further why the developer chose to include those keywords, i.e., perform any natural language processing. Thus, we expect that we will encounter a broad spectrum of privacy related consent dialogs.

We distinguish between a proper dialog and a notice by checking for the presence of buttons again using our initial manual study to identify text elements associated with consent dialog buttons such as *okay*, *accept* or *reject* and whose text is not significant longer than the provided matching expression. Furthermore, we ensure that labels such as *I do not consent* are not matched as *consent* by controlling for negating keywords. If an app does not display a detectable proper dialog or notice but contains a link to a privacy policy, again identified with a set of regular expressions, we classify the presented consent dialog as a link. All extracted information are stored for further processing. All regular expressions are listed in the appendix as well as contained in our source code.

(2) Proper Dialog Analysis: The collected data from the previous step now affords us to perform a more in-depth check of the patterns employed in the detected proper dialog against our previously established requirements (ref. Section 3.3):

Ambiguous Button Labels: Requirement 2 states that a button has to have a an unambiguous label as the common interpretation of the GDPR by DPAs requires, that a privacy consent dialog has to clearly communicate that pressing a button entails consenting. Ambiguous labels such as *continue* violate such criteria. Analogously, this also applies to reject buttons. Labels such as *options* or *manage choices* do not suffice.

Accept Button Highlighted: Furthermore requirement 2 states that the button has to be clear and, thus, a dialog may not nudge the user towards consenting, i.e., the choice of the user has to be free of any outside influence. We are able to detect both color- and size- based emphasis of one of the buttons which would run afoul of this requirement.

Accept but No (Working) Reject Button: According to requirement 4 a user must be able to give or reject consent

and requirement 1 states that rejection has to be able within the same number of clicks. We are able to detect whether such a proper reject option is given. Furthermore, we are able to detect whether an application stops working after declining consent, which would also run afoul of requirement 4.

6.3 In-Depth Traffic Analysis

We are also interested in the transmitted personal information to third parties in the context of privacy consent dialogs, that would clearly require prior consent by the user. To be able to analyze the transmitted traffic in-depth we designed a set of tracking endpoint processors for observed popular endpoints to 20 different known tracking companies. Each processor is designed to analyze the traffic directed at the targeted tracking endpoint and parses the contained data in-depth. We identify a targeted endpoint by scheme, host, and path. This affords us the capability to detect any data collection or processing before Consent. Which is an indirectly stated requirement for any privacy consent dialog: processing can only rely on consent as a legal basis if it occurs after consent has been given.

The endpoints were chosen by their overall requests frequency in our data set as well as for being a known tracking endpoint. We chose to focus on specific endpoints as it allows us to analyze the vast amount of information send to an endpoint including information that would normally be overlooked. We selected datapoints to extract, based on the content of the request by analyzing parameters send via query or body to the same endpoint. To understand values we leveraged the field names if available or recognizing values, e.g., IP or Device Name, by manually analyzing requests. A good example for data transmitted that would be lost by performing a generic search for data of already known information would be the language of the device (commonly just a two letter string), or a flag whether the device is already rooted (a simple boolean value). This type of focused analysis also allows us to decode information that is not transmitted in the clear but encoded. An example would be our support for Protobuf [23] used by Firebase.

Even though not every data point is person specific information (e.g., the localization of a device) as soon as it is transmitted in combination with an identifier[6] it is personal information and requires consent. Furthermore, previous work has shown that a set of non personal identifiers can be used to identify a person [35, 36, 43, 44, 64] or a device [38] and thus could be used as an indirect identifier, which again would then be covered by the GDPR [52]. Thus, focusing specific endpoints allows for a novel perspective on the vast amount of information leaked by mobile applications into the internet.

---

[6]We consider the Google Advertisement ID (GAID), the iOS Advertisement Identifier (IDFA,IDFV), and any UUID to be an Identifier.

6.4 Summary of Capabilities and Limitations

In this section, we described our framework to run mobile applications based on previous work and own additions to provide interaction capability. We are able to automatically download, install, run, interact with, collect traffic, and remove apps for both Android and iOS. Furthermore, we are able to extract and analyze displayed consent dialogs while monitoring transmitted traffic. However, our approach does come with limitations:

Choice of System and Configuration: We leverage a rooted Galaxy A13 for Android and a jailbroken iPhone for iOS. Either method of gaining administrative access can be detected by apps potentially leading to different behavior as when run on a vanilla mobile phone, e.g., some banking apps are known to refuse running on a rooted device for security reasons while other apps might alter their data collection behavior. This does impact the amount of successful app analysis and might impact the observed traffic. Furthermore, despite leveraging SSL KillSwitch and Objection and installing our mitmproxy certificate on the phones, apps do deploy SSL certificate checking not affected by those measures. Consequently, not all requests can be successfully intercepted.

Furthermore, during measurement we deny each app the tracking permission on iOS as OS based consent dialogs are out of scope. This may impact the collection behavior of apps on iOS.

Limited Interaction: We only interact with a detected consent dialog to either give or deny consent. This results in two measurements if both an unambigious accept and a reject button are present. This limited interaction may lead to missed dialogs that appear later during app usage or otherwise missed transmission behavior that is triggered by user interaction.

Wrong Dialogs/Violations Detection: Apps may contain text elements and buttons that indicate a dialog to our implementation but a human inspector would overturn such a verdict. We optimized our approach of dialog detection and analysis to rather miss a dialog if it is not clear cut. Due to this optimization, we may miss dialogs that could be detected using more lenient rules.

Limited Legal Clarity: Our stated design requirements are based on our legal understanding of the sources we found. However, as always with legal matters, it is not always clear cut whether something is against the law or not and opposing opinions do exist. Consequently, every classification by our implementation, indicating some form of GDPR violation, should be treated as opinion rather than fact unless substantiated by a court of law.

Our methodology and consequently any analysis based on it has to account for those limitations. The first two limitations limit the apps and app traffic we are able to monitor and potentially reduce the amount of detected leaked information. The third limitation affects the amount of dialogs we are able to extract and analyze, thus, limits the result set. The

	Overall	Android	iOS
Down.	4779 (100%)	3006 (100%)	1773 (100%)
Succ.	3654 (76.5%)	2134 (71.0%)	1520 (85.7%)
None	2823 (77.3%)	1636 (76.7%)	1187 (78.1%)
Link	232 (6.3%)	151 (7.1%)	81 (5.3%)
Notice	165 (4.5%)	97 (4.5%)	68 (4.5%)
Dialog	434 (11.9%)	250 (11.7%)	184 (12.1%)

Table 1: Table summarizing our detected dialog types overall and split by operating system.

last limitation reduces the amount of consent dialogs we are able to identify. Thus, our overall work and results are in favor of the applications we analyze as we consider a missing consent dialog and undetected personal information leakage less severe than detected ones.

7 Analyzed Consent Dialogs in Numbers

In the previous chapters, we discussed motivation (Section 3), and methodology (Section 6), as well as our initial static analysis on the popularity of CMPs (Section 5). Now we present the raw results obtained by our dynamic analysis of Android and iOS Apps. We start by performing a manual spotcheck of our results, then detail the numbers of detected dialogs, dialog types, and contained Dark Patterns. Afterwards, we present how data collection behavior of apps differs before and after agreeing or rejecting consent as well as our detection of TCF-related string properties set in apps.

We were able to analyze 3654 apps. On iOS we successfully analyzed 1520 apps (85.7%), whereas under Android we were able to successfully analyze 2134 apps (71.0%). An unsuccessful measurement can have different reasons. The most trivial reason is that an app refused to start or stops running at some point during the analysis. This behavior could be due to bugs or the app refusing to start, e.g., due to being run on a rooted/jailbroken device (ref. Section 6.4). The client of Appium running on the device was also observed to crash, leading to an unsuccessful and consequently missing measurement. Furthermore, Appium is not always able to extract elements or screenshots. Those Appium related limitations are not necessarily deterministic and it is possible that an App exhibits multiple failures across multiple measurements. Remember that we require multiple measurements to collect traffic and interact with a privacy consent dialog. An iOS specific limitation is the lack of a working jailbreak for the current version. Consequently, we are forced to work with iOS 14.5, and 169 (9%) required a more recent OS.

Each App exhibiting a measurement error is excluded from our analysis and, thus, incomplete measurements and missing elements or screenshots do not impact our results. However, this also means that we do not observe those apps data collection behavior or privacy consent dialogs, this might introduce

	None		Link		Notice		Dialog	
	🤖	🍎	🤖	🍎	🤖	🍎	🤖	🍎
Initial	192	194	23	14	10	11	25	31
None	-	-	0	5	0	1	0	1
Link	2	5	-	-	0	0	1	0
Notice	1	5	8	6	-	-	0	1
Dialog	4	10	8	2	0	2	-	-

Table 2: We visually checked 500 randomly selected apps and counted how many were classified into the wrong category.

	Initial		Wrong		Missed	
	🤖	🍎	🤖	🍎	🤖	🍎
Clear Accept	11	10	2	0	0	0
Ambig. Accept	9	9	0	0	0	0
Highl. Accept	6	9	0	1	1	0
Clear Reject	5	3	0	0	0	0
Ambig. Reject	7	10	1	3	2	1
Interaction	11	13	2	2	0	0

Table 3: We visually checked 40 randomly selected apps presenting a privacy consent dialog and verified that the design patterns we extract match a visual inspection.

a measurement bias as we miss apps that, e.g., refuse to run on a rooted/jailbroken device.

7.1 Manual Validation of Results

Overall our results demonstrate that only a subset of apps display a dialog and even if they do display a dialog they leverage design choices to nudge a user towards consenting. To underline our results we perform a manual inspection of a subset of apps to understand the performance of both our taxonomy as well as our dialog design pattern detection. We inspected 500 apps concerning their taxonomy and 40 concerning the detected design choices.

For our manual inspection we use artifacts collected during our automated app analysis as Appium allows not only to take a screenshot of the whole app but also of individual elements. Thus, we are able to reconstruct the classification and interactions our measurement implementation did.

Overall the results demonstrate that our implementation is in favor of the app developer as we are under reporting the amount of privacy consent dialog and, thus, the amount of apps that are deploying hostile design patterns to nudge users towards giving consent. Furthermore, our analysis on why our implementation made mistakes shows that errors are due to technical limitations and side cases in phrasing or design choices that require a human to make a proper decision.

Finally, our manual inspection of detected dialogs confirmed our expectation of a broad spectrum of privacy consent dialogs. We encountered dialogs that contain short catch-all texts as well as extensive texts with explicit description of the legal basis for data collection.

7.1.1 Taxonomy

The manual inspection of initial app screens demonstrates that we are able to distinguish between our four different types of privacy consent dialog (None, Link, Notice, Dialog) and do not paint a worse picture than actually exists. We look at the initial display content of an app, i.e., the elements we are basing our taxonomy classification on and check, whether a human inspector would perform a similar judgement as our implementation. Finally, we also look for indicators that a consent dialog could be hidden behind initial privacy consent

dialog unrelated interactions, but is not directly accessible to the user after the app is started.

The manual inspection revealed that our implementation is in favor of app developers as we are underreporting the amount of privacy consent dialogs with 53 apps displaying some stronger form of privacy consent dialog than detected by our implementation. Each app that does not have a privacy consent dialog is negligent in their approach towards user privacy whereas an app that uses Dark Patterns in their privacy consent dialogs could be considered malicious.

We also detected 9 privacy consent dialog apps that were displaying a weaker form of privacy consent dialog. However, only 3 of those relate to a proper privacy consent dialog being part of our design requirement analysis: We detected one dialog due to Appium extracting more than was visually displayed including multiple dialog associated keywords and button labels leading to the wrong classification. The identification of a link as a dialog was due to the unusual structure of the displayed app, dedicating a heading to a privacy link, leading to the wrong classification. Finally, the wrongful identification of a notice as a dialog was due to the app opening a keyboard including a dialog associated button label, leading to the wrong classification. In each case the technical detection of a dialog was correct, however, the context of the detection lead to a human inspector to overturn the classification. A summary of the taxonomy classification check are given in Table 2.

During our visual inspection, we detected 6 apps for which the display indicates that further interaction will lead to a consent dialog, either by saying so or due to an dialog partially hiding a privacy consent dialog. Those numbers show that further work on the open research question of in-depth app interaction is required. However, detecting those dialogs is out of scope for this work, as we are focusing on initially displayed consent dialogs.

7.1.2 Dialog Design Patterns

After analyzing how well our sorting into our taxonomy works we now perform a manual inspection of design requirement

analysis. We isolate different failure modes and contextualize them into our overall design analysis. Overall we encountered 7 classifications that can be partially criticized when applying a human eye. We had one interaction failure, due to a root warning preventing Appium from performing an interaction, thus, potentially incorrectly reporting an app that does not close after refusing consent favoring the dialog design. We extracted one dialog that was hidden behind an overlay, i.e., the dialog was correctly recognized however, the extraction of the buttons yielded the wrong colors. We wrongly classified a link as a dialog, due to a button being present on the screen but actually in no relation to the privacy text which requires context intelligence to recognize. Additionally we wrongly classified a consent dialog as being a privacy consent dialog, however, the text the user agrees only relates to the terms and conditions. The wrong button identifications were due to specific keywords being used separately and thus being wrongly recognized as buttons. Details of our classification check are given in Table 3.

7.2 Consent Dialogs and Dark Patterns

While running the apps, we applied our consent dialog detection. Overall, we detected 814 (22.3%) apps displaying a privacy consent dialog on start. 232 (6.3%) displayed a link, 165 (4.5%) displayed a notice, and 434 (11.9%) displayed a proper dialog. Overall, the differences between Android and iOS are minuscule. The results are listed in Table 1, also stating the distribution for Android and iOS individually.

7.2.1 Dark Patterns

After sorting all detected privacy consent dialog into our taxonomy, we are left with 434 (11.9%) proper dialogs.

Those pass the first muster towards fulfilling our minimal privacy consent dialog design criteria and we are interested in a more in-depth analysis of their design elements. As established in Section 3, a consent inquiry has to be explicit, present an actual choice, and must not trick the user into consenting.

We applied our detection facilities to extract elements of detected proper dialogs and determine if our stated design criteria are met as detailed in Section 6. An overview of the results is given by the UpSet plot in Figure 1.

We detected at least one violation of our design requirements in 429 apps (98.8%). On Android, we detected a violation in 246 (98.4%) dialogs, and on iOS in 183 (99.5%).

However, a violation of our design requirements is by itself not necessarily a GDPR violation. They may only invalidate the consent given and, thus, render any personal data collection based on such a consent a possible violation. We, consequently, still have to analyze the data transmission done by those apps.

7.3 Before and After Consent

During our dialog analysis, we collected all transmitted traffic, each time running the app for 60 seconds. Between each app execution, the environment was reset by removing the app. The first collection established the transmitted traffic before any interaction with the consent dialog. The second collected the transmitted traffic after interaction with the consent dialog. While running the apps, we successfully intercepted 128468 requests with 50% of apps making less than 35 and 75% of apps making less than 15 requests. 187 apps made no request at all out of which 149 were on Android and 38 on iOS. We analyzed the obtained traffic and leveraged our tracking endpoint specific processors. Overall 25.2% of all intercepted traffic was directed to one of our monitored tracking domains with 14.0% of traffic being covered by our endpoint parser. We distinguish between anonymously transmitted data and pseudonymous transmitted data, i.e., data that is accompanied or by itself an identifier.

7.3.1 Before Consent

Analyzing the transmitted traffic, 1285 (35.2%) apps sent at least one request containing a unique identifier rendering the contained information at least pseudonymous and thus covered by the GDPR. The majority of the transmitted data was transmitted pseudonymously. The distribution of different data types transmitted anonymously or pseudonymously is presented in Figure 2.

Overall, 17418 requests went to one tracker covered by our endpoint parser before any interaction took place. Mapping the requests back to apps, leads to 3013 (82.5%) of apps contacted one of our covered tracking endpoints before any interaction took place.

7.3.2 After (No) Consent

For each proper dialog, we repeated our collection twice if a dialog had the appropriate buttons. Once for rejecting consent and in a separate run accepting the consent dialog using Appium. Overall we performed 350 accept and 112 reject analysis. We then monitored the transmitted traffic after interaction.

This resulted in 6653 and 839 requests for accept and reject, respectively. Overall, 77 apps transmitted pseudonymous data after accepting the consent dialog out of which 75 where new.

We observed 5 transmitting pseudonymous data after interacting with an unambiguous reject button out of which all 5 were new. Of the traffic intercepted after giving consent 7.2% went to one of our covered endpoints. Meanwhile after rejecting consent, the proportion was 10.8%. Figure 2 shows the amount of different data points transmitted to trackers after giving consent. There is no corresponding figure for transmitted data after declining consent due to the low number of observed requests.
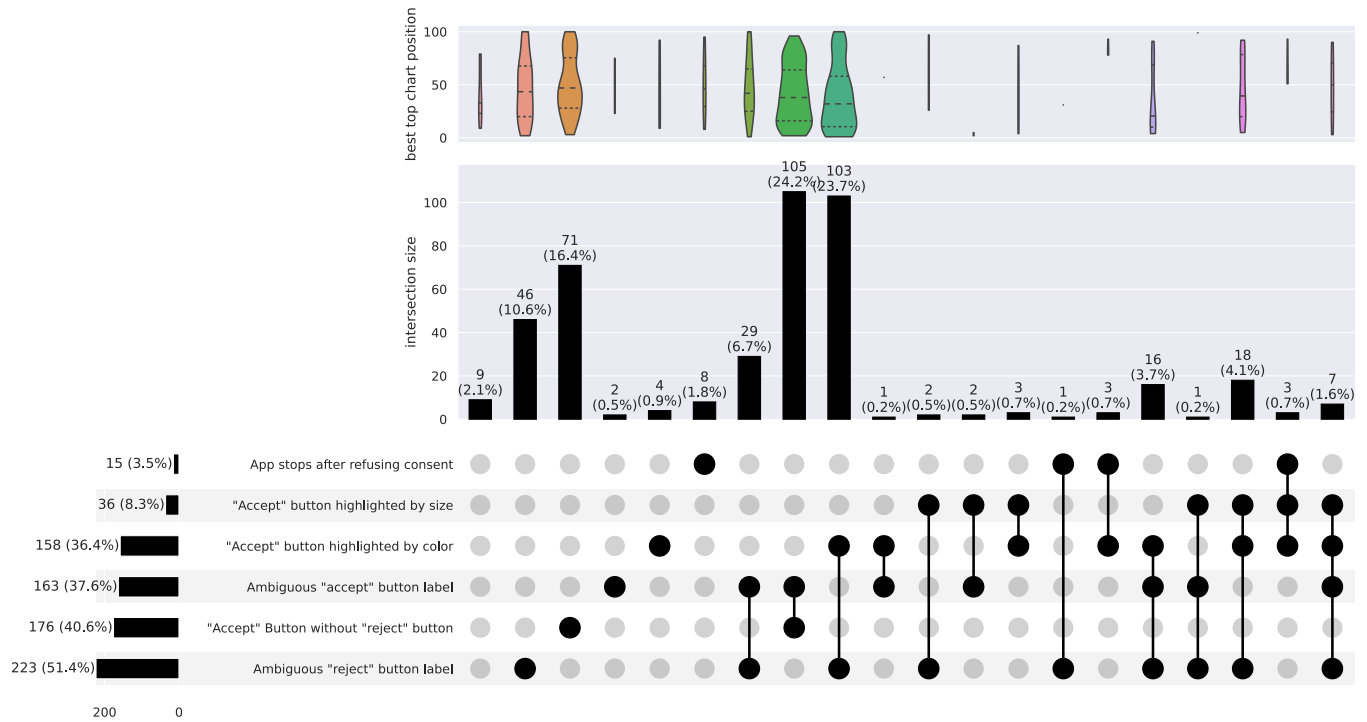
Figure 1: UpSet plot [29, 53] showing the different possible combinations of Dark Patterns we have detected in consent dialogs. The upper violin plot illustrates the distribution of top chart positions among the apps in the respective set.
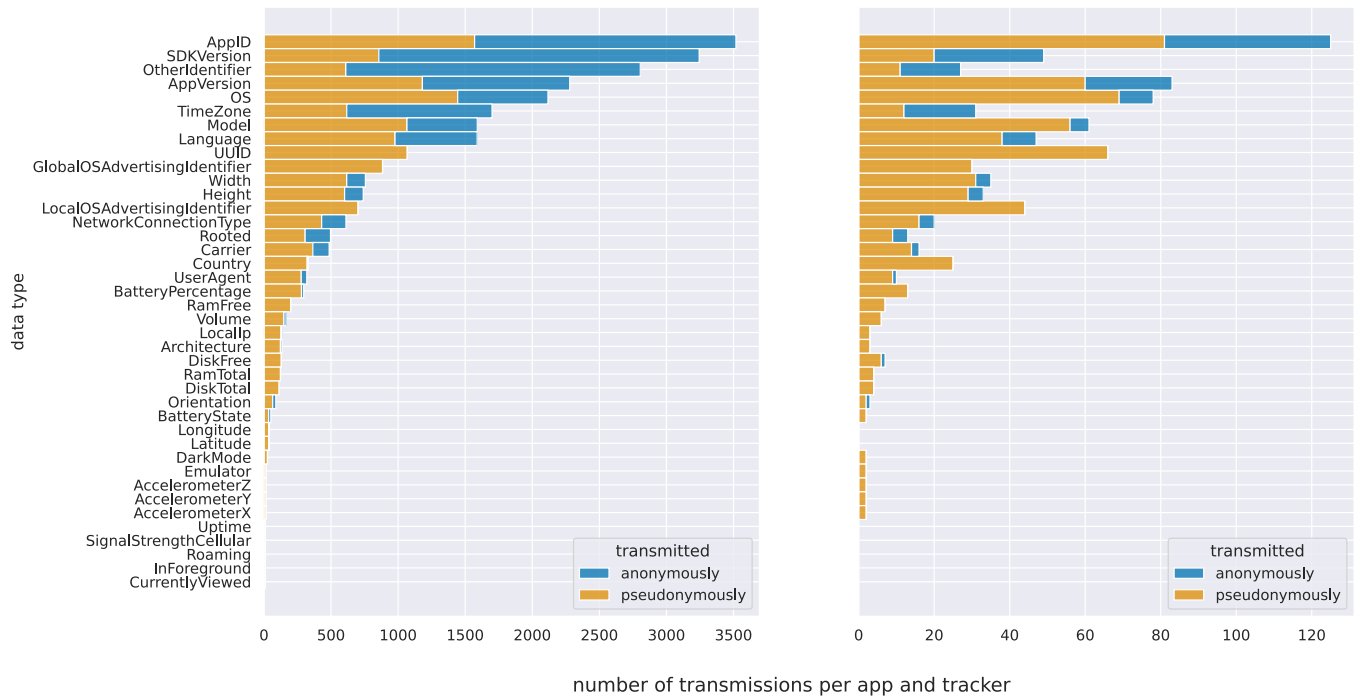

Figure 2: The amount of transmissions of our monitored data types before and after giving consent. The data is grouped by whether they were transmitted anonymously or in combination with a unique identifier. Both IDFA and Google Advertising ID are included in GlobalOSAdvertisingIdentifier.

7.3.3 Relevance of Consent

Overall, we detected pseudonymous data transmission aggregated across our runs with and without interaction for 355 (81.8%) apps with any type of consent dialog. Out of the apps using a dark pattern in their dialog, 187 (43.6%) transmit pseudonymous data in any of our runs. This means that 43.1% of all detected proper dialogs presumably failed to acquire valid consent for the observed data collection.

7.3.4 Detected IAB TCF Property Strings

We already statically inferred that only a small subset of apps are using one CMP out of our selection of possible CMPs. However, we want to verify and augment those numbers with actual observed CMP usage and how sensible retrieved CMP data is. During our runs, we also collected the property strings set by an app during execution and analyzed them to check whether they belong to the IAB TCF [32] by searching for property strings starting with *IABTCF_*. Besides the standardized data storage format this framework also provides the largest amount of covered CMPs in our list and provides a feasible approach to gain insight into the actual usage of CMPs, though, with a selection bias for IAB TCF compliant CMPs.

We detected 146 apps setting such a string (76 on Android, and 70 on iOS). For 57 of those apps, we did not detect a dialog. During manual confirmation, only 17 were actually showing a dialog and thus were missed by us due to highly specific sentences or rendering the display as an image rendering use of Appium to extract text impossible (see Section 7.1). 2 apps were setting the any TCF related string after interaction, the remaining 123 set the values before interaction. We analyzed whether the value for the *applicability of the GDPR* is set, what *CMP is being used*, and the configuration string itself containing *agreed-to collection purposes and vendors*.

Applicability of the GDPR: A total of 129 apps correctly stored that the GDPR applies, 8 apps incorrectly assumed that the GDPR does not apply. This information is stored as a Boolean value labeled '*IABTCF_gdprApplies*'.

Used CMPs: 121 apps set the CMP value, with Sourcepoint (55 apps) and Google (23 apps) being the most popular by far. Followed by OneTrust with only 14 apps. The remaining CMPs only having single digit usage numbers.

Agreed-to Collection Purposes and Vendors: We detected the TC string also containing purposes and vendors in 64 apps. Out of 24 possible different purposes, on average consent was stored for 7.15 with a median value of 10. On average 262.35 vendors were consented to with a median value of 137.

8 No Regard for Consent

In Section 4, we have downloaded 4779 apps, statically analyzed the library usage to detect inclusion of CMPs in Section 5, and run each app to collect traffic while analyzing and interacting with presented consent dialogs in Section 7. We interacted with the dialogs by either accepting or if possible rejecting consent to data collection.

In this section, we discuss the implications of our results. We compare them with corresponding results in previous work on mobile application data collection and web-based consent dialogs. Finally, we discuss consequences and possible solutions to the observed lack of regard for consent and respect of privacy by app developers in a call to action.

8.1 No Real Prevalence of CMPs or the TCF

Overall, we detected only minor usage of CMPs with only 252 (5.1%) apps containing any class or library contained in our curated list of popular CMP identifiers. All of them were IAB TCF compliant. This statically retrieved number was confirmed by our dynamic analysis as we only detected a TCF-related settings in 146 (4.0%) apps. This highly limits any analysis approach based on frameworks and hinders researchers, privacy advocates, and possibly privacy enforcing tools from leveraging the power of a framework to research and improve the state of consent on a fine-grained level, such as the purposes for which consent is given.

When comparing the usage of CMPs with previously reported web usage, we can see a similarly small popularity. Matte et al. [54] report a TCF-compliant cookie banner on 6.2% of crawled Tranco top 1000 sites for different TLDs. They report a high variability between TLDs, ranging from 18.9% on .uk to 0% on .mt. Nouwens et al. [60] detected the usage of a CMP on 6.8% of crawled UK Top 10k domains but reference a no longer accessible survey from 2019 that places CMP usage at around 20%. They also name possible methodological issues that indicate that Nouwens et al. numbers are more accurate concerning actual usage. A newer survey from the same source places the CMP usage across the top 10k US sites at at least 33% as of Q1 2022 [5]. If we assume the academic numbers to be the correct ones, the small popularity of around 6% is mirrored in the mobile space with only 4.0% actively using a TCF-compatible CMP and our static analysis indicating that 5.1% include a CMP from our curated list.

However, even in the few detected usages of the TCF, we detected questionable usage. On average, consent was stored for 262.35 vendors, which is, even assuming a proper consent dialog, questionable under the GDPR as consent needs to be informed. We seriously question the possibility to properly inform a user about that many vendors.

8.2 More Data Transmission Before than After Consent

During our traffic collection, we detected 120976 requests before any interaction took place, and only 6653 and 839 after accepting or rejecting consent, respectively. Overall 14.0% of

requests were analyzed by our endpoint parser. We detected that 75 and 5 apps started sending pseudonymous data in either group, compared to 1285 in our initial traffic observation before any interaction. Based on the difference in magnitude of requests as well as in amount of identifiers send, it is safe to assume that most personal data is leaked before any dialog has been processed and agreed to by the user. This has clear implications for privacy-conscious user as they cannot trust an app to ask for permission prior to data collection, thus, each starting of an app bears a risk of leaking personal information. Furthermore, users cannot expect that they will have a choice free from incentives towards sharing their data if they are presented with a privacy consent dialog and are forced to consciously and carefully interact with privacy consent dialogs to prevent being nudged towards a decision. This demonstrates a clear disregard for consent and user privacy concerning contacting trackers or transmitting personal information. Our results are in line with recent previous work on Android [58], Android compared to iOS [49], and iOS [48]. Based on this repeated demonstration of data leakage by different research groups, it must be considered a fact for the time. The GDPR did not have the intended effect for users of mobile apps and action has to be taken to improve privacy for them.

8.3   The Questionable State of Consent Dialogs

Only a small percentage of apps (22.3%) displayed any form of consent dialog, and even fewer offered the user an actual dialog with at least a button (11.9%) suggesting some form of active choice for the user. Combining this with the high transmission of personal information (Figure 2) and the amount of requests going to our limited set of endpoint trackers already demonstrates that the requirement to inquire active consent or any consent at all is widely ignored.

The state of consent inquiry becomes even worse when looking at the design choices made in the detected proper dialogs. For both Android and iOS, the dialogs that deployed Dark Patterns to nudge a user towards consent were at 98.4% and 99.5%, respectively. This highlights a clear disregard for user choices independent of the operating system.

Another aspect visible from our UpSet diagram (Figure 1) is that Dark Patterns in privacy consent dialogs are not restricted to low or high ranking apps but are present throughout. The diagram includes a violin plot, i.e., a visualization of density and range, displaying the app ranks contained in the sets described by the intersections below. Those violin plots, bearing intersections with a small cardinality, span the whole range or close to the whole range of app ranks considered with the median being close to the center of the range. It is possible that top 100 apps are not sufficiently different a difference becomes visible when analyzing even higher ranks. However, Nguyen et al. performed a long-tail app analysis and found that there is no real difference between high profile apps and long tail apps when it comes to transmitting personal information prior to any consent [58]. It stands to reason that this disregard for privacy transfers to privacy consent dialogs.

Previous work on consent dialog was primarily focused on web consent dialogs. In 2019 Eijk et al. [72] found a consent dialog or notice on 40% of analyzed websites (1500 websites), and Sanchez-Rola et al. [65] on 50% (2000 websites). A year later, Mehrnezhad et al. [56] report observing a consent dialog on 91% of websites (116 websites). Even though the web data sets are individually smaller than our mobile data set, they indicate a high prevalence of dialogs which is mirrored in our subjective observations when browsing the web. This is in contrast to our mobile observations, where we only detected dialogs in 22.3% of all apps.

When comparing privacy consent dialogs on the web with the mobile ecosystem we detected an accept but no reject button in 40.6% of all dialogs, whereas Mehrnezhad et al. [56] reported only 35.4% of all analyzed websites having such a dialog. Nouwens et al. [60] report finding a reject all button on only 12.6% of websites, though this criterion is strictly stronger than ours.

9   Conclusion

In this paper, we presented a tool chain for both Android and iOS to download apps, run, and analyze their traffic, as well as consent dialogs. Furthermore, our tool chain is able to interact with displayed consent dialogs. We used this tool chain to analyze the traffic transmission behavior in the context of the presented consent dialogs. Out of 4779, apps we detected a consent violation in 43.1%. Furthermore, we observed that 94.2% of requests already occur before any interaction with a consent dialog happened, leading us to the conclusion that consent dialogs amount to little more than window dressing and are not respected.

In order to assess our findings in a broader context, we compared the results of our study with similar measurements made on the web. To our surprise, we were able to show that mobile applications are even less privacy-respecting than websites. Our low number of privacy consent dialog conforming to our minimal design requirements paints a clear picture of the app developers' blatant disregard for the user's wishes towards their privacy.

10   Acknowledgements

References

[1] https://developer.apple.com/app-store/user-privacy-and-data-use/. accessed: 2023-02-09.

[2] https://investors.applovin.com/news/news-details/2022/AppLovin-Closes-Acquisition-of-Twitters-MoPub-Business/default.aspx. accessed: 2022-01-06.

[3] App privacy details on the app store. https://developer.apple.com/app-store/app-privacy-details/. accessed: 2022-05-06.

[4] appium automation for apps. https://appium.io/. accessed: 2022-05-15.

[5] Consent management platform (cmp) 2022 tracker. https://www.kevel.com/cmp/. accessed: 2022-05-15.

[6] Data protection commission announces conclusion of two inquiries into meta ireland. https://dataprotection.ie/en/news-media/data-protection-commission-announces-conclusion-two-inquiries-meta-ireland. accessed: 2023-01-05.

[7] Decision on the merits 21/2022 of 2 february 2022. https://www.gegevensbeschermingsautoriteit.be/publications/beslissing-ten-gronde-nr.-21-2022-english.pdf. accessed: 2022-06-04.

[8] Exodus static analysis. https://exodus-privacy.eu.org/en/post/exodus_static_analysis/. accessed: 2022-05-15.

[9] Frida. https://frida.re/docs/android/. accessed: 2022-05-15.

[10] Get more information about your apps in google play. https://blog.google/products/google-play/data-safety/. accessed: 2022-05-06.

[11] 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/default/files/files/file1/edpb_guidelines-art_6-1-b-adopted_after_public_consultation_en.pdf. accessed: 2022-05-07.

[12] Ipatool. https://github.com/majd/ipatool. accessed: 2022-05-15.

[13] Judgment of the court (second chamber) of 19 october 2016. patrick breyer v bundesrepublik deutschland. request for a preliminary ruling from the bundesgerichtshof. reference for a preliminary ruling — processing of personal data — directive 95/46/ec — article 2(a) — article 7(f) — definition of 'personal data' — internet protocol addresses — storage of data by an online media services provider — national legislation not permitting the legitimate interest pursued by the controller to be taken into account. case c-582/14. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:62014CJ0582.

[14] Lg rostock - 3 o 762/19. https://gdprhub.eu/index.php?title=LG_Rostock_-_3_O_762/19&oldid=19832. accessed: 2022-05-7.

[15] libimobiledevice. https://libimobiledevice.org/. accessed: 2022-05-15.

[16] Many more cookie banners to go: Second wave of complaints underway. https://noyb.eu/en/more-cookie-banners-go-second-wave-complaints-underway. accessed: 2022-05-06.

[17] mitmproxy. https://mitmproxy.org/. accessed: 2022-05-15.

[18] noyb aims to end "cookie banner terror" and issues more than 500 gdpr complaints. https://noyb.eu/en/noyb-aims-end-cookie-banner-terror-and-issues-more-500-gdpr-complaints. accessed: 2022-05-06.

[19] objection - runtime mobile exploration. https://github.com/sensepost/objection. accessed: 2022-05-15.

[20] Open 1.1.1-1. http://cydia.saurik.com/package/com.conradkramer.open/. accessed: 2022-05-15.

[21] Orientierungshilfe der aufsichtsbehörden für anbieter:innen von telemedien ab dem 1. dezember 2021. https://www.datenschutzkonferenz-online.de/media/oh/20211220_oh_telemedien.pdf. accessed: 2022-05-07.

[22] Playstoredownloader. https://github.com/ClaudiuGeorgiu/PlaystoreDownloader. accessed: 2022-05-15.

[23] Protocol buffers. https://developers.google.com/protocol-buffers. accessed: 2022-01-10.

[24] 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/ec (general data protection regulation). https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[25] Ssl kill switch 2. https://github.com/nabla-c0d3/ssl-kill-switch2. accessed: 2022-05-15.

[26] Tcf v2.0 cmp service (mobile). https://iabeurope.eu/cmp-list/. accessed: 2022-05-22.

[27] Top 14 consent management tools for mobile apps and games. https://www.blog.udonis.co/mobile-marketing/mobile-games/consent-management-tools. accessed: 2022-05-22.

[28] Top mobile app consent management tools. https://blog.instabug.com/top-mobile-app-consent-management-tools/. accessed: 2022-05-22.

[29] Visualizing intersecting sets. https://upset.app/. accessed: 2022-05-05.

[30] Wecomply! guide for onetrust. https://wecomply.noyb.eu/static/app/pdf/OneTrustGuide.766f4ff956c0.pdf. accessed: 2022-05-15.

[31] What is gdpr, the eu's new data protection law? https://gdpr.eu/what-is-gdpr/. accessed: 2022-05-07.

[32] What is the transparency & consent framework (tcf)? https://iabeurope.eu/transparency-consent-framework/. accessed 2022-05-20.

[33] Yuvraj Agarwal and Malcom Hall. Protectmyprivacy: Detecting and mitigating privacy leaks on ios devices using crowdsourcing. In *The 11th International Conference on Mobile Systems, Applications, and Services (MobiSys'13)*, 2013.

[34] Kevin Allix, Tegawendé F. Bissyandé, Jacques Klein, and Yves Le Traon. Androzoo: Collecting millions of android apps for the research community. In *Proceedings of the 13th International Conference on Mining Software Repositories*, 2016.

[35] Yves-Alexandre de Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the Crowd: The privacy bounds of human mobility. 2013.

[36] Yves-Alexandre De Montjoye, Laura Radaelli, Vivek Kumar Singh, et al. Unique in the shopping mall: On the reidentifiability of credit card metadata. 2015.

[37] Martin Degeling, Christine Utz, Christopher Lentzsch, Henry Hosseini, Florian Schaub, and Thorsten Holz. We value your privacy ... now take some cookies: Measuring the gdpr's impact on web privacy. 2019.

[38] Peter Eckersley. How unique is your web browser? In Mikhail J. Atallah and Nicholas J. Hopper, editors, *Privacy Enhancing Technologies, 10th International Symposium, (PETS)*, 2010.

[39] Manuele Egele, Christopher Kruegel, Engin Kirda, and Giovanni Vigna. Pios: Detecting privacy leaks in ios applications. In *The 18th Annual Network & Distributed System Security Symposium (NDSS'11)*, 2011.

[40] Denzil Ferreira, Vassilis Kostakos, Alastair R. Beresford, Janne Lindqvist, and Anind K. Dey. Securacy: An empirical investigation of android applications' network usage, privacy and security. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '15)*, 2015.

[41] Der Landesbeauftragte für den Datenschutz und die Informationsfreiheit Baden-Württemberg. Faq: Cookies and tracking durch betreiber von webseiten und hersteller von smartphone-apps. https://www.baden-wuerttemberg.datenschutz.de/wp-content/uploads/2022/03/FAQ-Tracking-online.pdf. accessed: 2022-05-07.

[42] Clint Gibler, Jonathan Crussell, Jeremy Erickson, and Hao Chen. Androidleaks: Automatically detecting potential privacy leaks in android applications on a large scale. In *Trust and Trustworthy Computing*, 2012.

[43] Philippe Golle. Revisiting the Uniqueness of Simple Demographics in the US Population. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Societ*, 2006.

[44] José González-Cabañas, Ángel Cuevas, Rubén Cuevas, Juan López-Fernández, and David García. Unique on facebook: Formulation and evidence of (nano)targeting individual users with non-pii data. In *Proceedings of the 21st ACM Internet Measurement Conference*, 2021.

[45] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *USENIX Security Symposium*, 2018.

[46] Maximilian Hils, Daniel W. Woods, and Rainer Böhme. Privacy preference signals: Past, present and future. 2021.

[47] Henry Hosseini, Martin Degeling, Christine Utz, and Thomas Hupperich. Unifying privacy policy detection. 2021.

[48] Simon Koch, Malte Wessel, Benjamin Altpeter, Madita Olvermann, and Martin Johns. Keeping privacy labels honest. In *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2022)*, 2022.

[49] Konrad Kollnig, Anastasia Shuba, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. Are iphones really better for privacy? comparative study of ios and android apps. In *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2022)*, 2022.

[50] Konrad Kollnig, Anastasia Shuba, Max Van Kleek, Reuben Binns, and Nigel Shadbolt. Goodbye tracking? impact of ios app tracking transparency and privacy labels. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, 2022.

[51] Andreas Kurtz, Andreas Weinlein, Christoph Settgast, and Felix Freiling. Dios: Dynamic privacy analysis of ios applications. In *Technical Report*, 2014.

[52] Morgan Lewis, Tess Blair, Patrick Campbell Jr., and Vincent Catanzaro. The edata guide to gdpr: Anonymization and pseudonymization under the gdpr. https://www.jdsupra.com/legalnews/the-edata-guide-to-gdpr-anonymization-95239/#_ftn2. accessed: 2022-05-7.

[53] Alexander Lex, Nils Gehlenborg, Hendrik Strobelt, Romain Vuillemot, and Hanspeter Pfister. Upset: Visualization of intersecting sets. 2014.

[54] C'elestin Matte, Nataliia Bielova, and Cristiana Santos. Do cookie banners respect my choice? : Measuring legal compliance of banners from iab europe's transparency and consent framework. 2020.

[55] Célestin Matte, Cristiana Santos, and Nataliia Bielova. Purposes in iab europe's tcf: Which legal basis and how are they used by advertisers? In *Privacy Technologies and Policy*, 2020.

[56] Maryam Mehrnezhad. A cross-platform evaluation of privacy notices and tracking practices. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, 2020.

[57] Alexios Mylonas, Marianthi Theoharidou, and Dimitris Gritzalis. Assessing privacy risks in android: A user-centric approach. In *Risk Assessment and Risk-Driven Testing*, 2014.

[58] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. Share first, ask later (or never?) studying violations of gdpr's explicit consent in android apps. In *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[59] Trung Tin Nguyen, Michael Backes, and Ben Stock. Freely given consent? studying consent notice of third-party tracking and its violations of gdpr in android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022.

[60] Midas Nouwens, Ilaria Liccardi, Michael Veale, David R. Karger, and Lalana Kagal. Dark patterns after the gdpr: Scraping consent pop-ups and demonstrating their influence. 2020.

[61] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. User tracking in the post-cookie era: How websites bypass gdpr consent to track users. In *Proceedings of the Web Conference 2021*, 2021.

[62] Jingjing Ren, Martina Lindorfer, Daniel Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. Bug fixes, improvements,... and privacy leaks–a longitudinal study of pii leaks across android app versions. In *Proc. of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[63] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. "won't somebody think of the children?" examining coppa compliance at scale. In *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2018)*, 2018.

[64] L. Rocher, J.M. Hendrick, and Y.A. de Montjoye. Estimating the success of re-identifications in incomplete datasets using generative models. In *Nature Communications 10*, 2019.

[65] Iskander Sánchez-Rola, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. Can i opt out yet?: Gdpr and the global illusion of cookie control. *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019.

[66] Ani Karini Muniz Schiebert and Benjamin Altpeter. https://www.datarequests.org/blog/gdpr-territorial-scope/. https://www.datarequests.org/blog/gdpr-territorial-scope/. accessed: 2022-05-07.

[67] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D. Breaux, and Jianwei Niu. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*, 2016.

[68] Ryan Stevens, Clint Gibler, Jon Crussell, Jeremy Erickson, and Hao Chen. Investigating user privacy in android ad libraries. In *Workshop on Mobile Security Technologies (MoST)*, 2012.

[69] Marianthi Theoharidou, Alexios Mylonas, and Dimitris Gritzalis. A risk assessment method for smartphones. In *Information Security and Privacy Research*, 2012.

[70] Martino Trevisan, Stefano Traverso, Eleonora Bassi, and Marco Mellia. 4 years of eu cookie law: Results and lessons learned. 2019.

[71] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. (un)informed consent: Studying gdpr consent notices in the field. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[72] Rob van Eijk, Hadi Asghari, Philipp Winter, and Arvind Narayanan. The impact of user location on cookie notices (inside and outside of the european union). In *Workshop on Technology and Consumer Protection (ConPro '19)*, 2019.

[73] Zhemin Yang, Min Yang, Yuan Zhang, Guofei Gu, Peng Ning, and X. Sean Wang. Appintent: Analyzing sensitive data transmission in android for privacy leakage detection. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*, 2013.

[74] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel Reidenberg, N. Cameron Russell, and Norman Sadeh. Maps: Scaling privacy compliance analysis to a million apps. In *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2019)*, 2019.

Appendix

Targeted Tracker

It is not only interesting to see how much requests is transmitted but also who is on the receiving end. Across all our supported endpoints Firebase was the most popular endpoint for receiving data. Figure 3 shows a scatter plot associating our supported endpoints with the amount of received data points. Overall if an endpoint collected on an operating system, it across a large variety of different data points, however, not all tracker are active on all platforms. Figure 4 shows the data send to our monitored endpoints after giving consent respectively.

Initial observation shows that Facebook, Doubleclick, Mopub, and start.io show anomalies concerning the operating system.

Facebook primarily collects data on Android. However, we still intercept a significant number of requests directed at Facebook, but they do not contain any of our supported personal information. Even after giving consent this pattern does not change. Our hypothesis is that the Facebook associated libraries do not perform any data collection on iOS anymore but are still left in by the developer.

Doubleclick seems to collect only data on iOS. We identified the url `https://googleads.g.doubleclick.net/mads/gma` and `https://googleads.g.doubleclick.net/getconfig/pubsetting` as observed popular endpoints

receiving personal inforamtion. However, those endpoints seem to only be utilized on iOS. Especially the `/mads/gma`, carrying the most information, does not appear in our Android traffic data set.

Mopub is only active on Android. However, Mopub was acquired by AppLovin and has sunset on March 31. 2022 [2]. It is surprising to still see Mopub requests. We explain this artifact by apps still using an old library either due to developers holding off on making the required code changes or the app not having been updated since.

start.io also is only active on Android. We are monitoring three different endpoints and neither is targeted by any requests on iOS.

Regular Expressions

Due to space constraints we cannot include our used regular expressions to identify dialog and dialog elements. However, they are published with our code at `pending publication`.
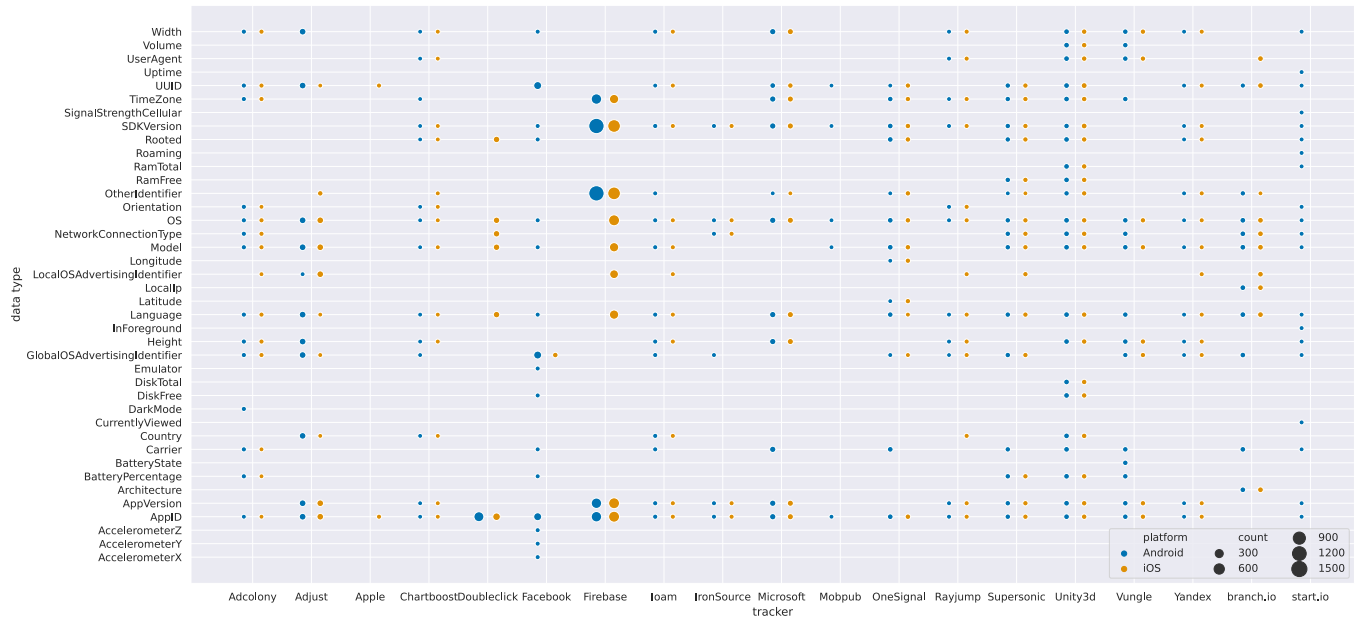
Figure 3: The amount of different datapoints direct at different endpoints prior to giving or refusing consent.
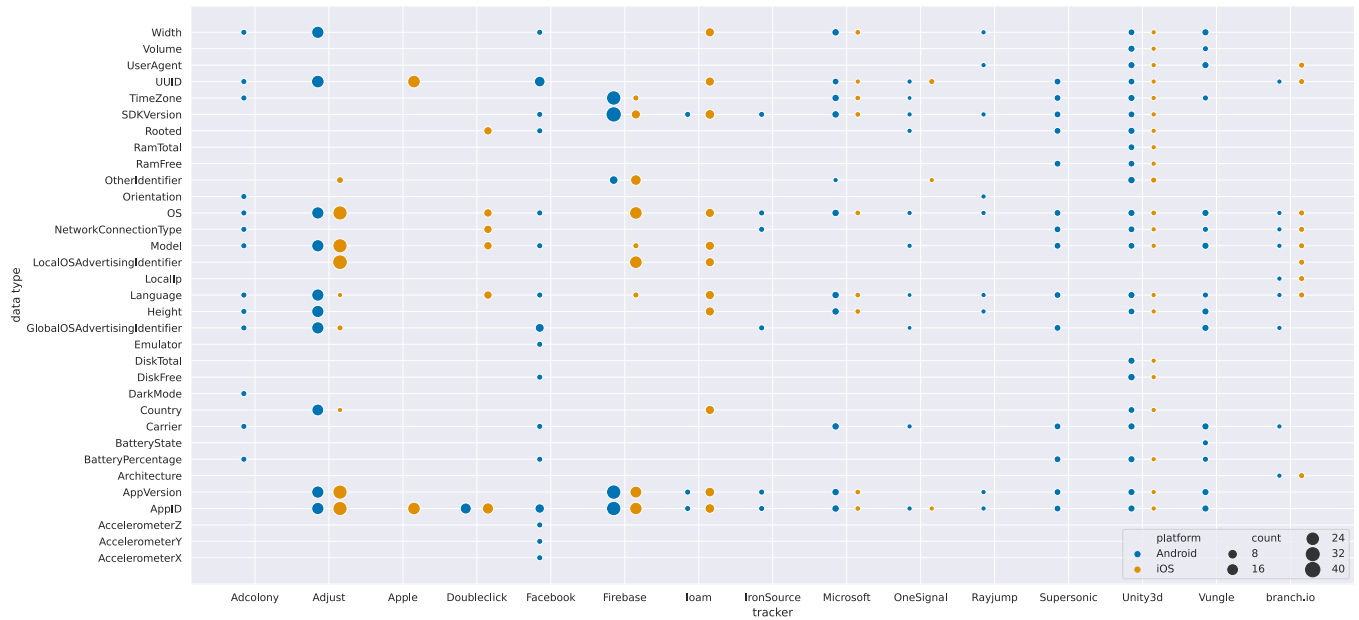

Figure 4: The amount of different datapoints direct at different endpoints after giving consent.