sciendo

Álvaro Feal*, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, and Alessandra Gorla

Angel or Devil? A Privacy Study of Mobile Parental Control Apps

Abstract: Android parental control applications are used by parents to monitor and limit their children's mobile behaviour (*e.g.,* mobile apps usage, web browsing, calling, and texting). In order to offer this service, parental control apps require privileged access to system resources and access to sensitive data. This may significantly reduce the dangers associated with kids' online activities, but it raises important privacy concerns. These concerns have so far been overlooked by organizations providing recommendations regarding the use of parental control applications to the public.

We conduct the first in-depth study of the Android parental control app's ecosystem from a privacy and regulatory point of view. We exhaustively study 46 apps from 43 developers which have a combined 20M installs in the Google Play Store. Using a combination of static and dynamic analysis we find that: these apps are on average more permissions-hungry than the top 150 apps in the Google Play Store, and tend to request more dangerous permissions with new releases; 11% of the apps transmit personal data in the clear; 34% of the apps gather and send personal information without appropriate consent; and 72% of the apps share data with third parties (including online advertising and analytics services) without mentioning their presence in their privacy policies. In summary, parental control applications lack transparency and lack compliance with regulatory requirements. This holds even for those applications recommended by European and other national security centers.

Keywords: Parental control, Android, mobile apps, static analysis, dynamic analysis

PACS:

DOI 10.2478/popets-2020-0029
Received 2019-08-31; revised 2019-12-15; accepted 2019-12-16.

1 Introduction

The dependency of society on mobile services to perform daily activities has drastically increased in the last decade [81]. Children are no exception to this trend. Just in the UK, 47% of children between 8 and 11 years of age have their own tablet, and 35% own a smartphone [65]. Unfortunately, the Internet is home for a vast amount of content potentially harmful for children, such as uncensored sexual imagery [20], violent content [91], and strong language [51], which is easily accessible to minors. Furthermore, children may (unawarely) expose sensitive data online that could eventually fall in the hands of predators [2], or that could trigger conflicts with their peers [30].

Aiming at safeguarding children's digital life, some parents turn to *parental control applications* to monitor children's activities and to control what they can do on their smartphones [63, 78]. The typical parental control app allows parents to filter, monitor, or restrict communications, content, system features, and app's execution [89]. Other apps provide parents with fine-grained reports about children's usage of the phone, their social interactions, and their physical location. An example of the former is the content discovery platform *Safe Mode with Free Games for Kids* by *KIDOZ*, and of the latter is the *Norton Family* app.

To provide these features, parental control apps rely on the collection and handling of children's behavioral (e.g., location, browsing activities, and phone calls) and personal data (e.g., unique identifiers and contacts), in many cases, using techniques and methods similar to those of spyware [27]. Yet, as with many regular Android applications, parental control software may also integrate data-hungry third-party advertising SDKs— to monetize their software— and analytics SDKs—to monitor the behavior of their users, create bug reports, and build user profiles. As a consequence of Android's

***Corresponding Author: Álvaro Feal:** IMDEA Networks Institute / Universidad Carlos III de Madrid, E-mail: alvaro.feal@imdea.org
Paolo Calciati: IMDEA Software Institute / Universidad Politécnica de Madrid, E-mail: paolo.calciati@imdea.org
Narseo Vallina-Rodriguez: IMDEA Networks Institute / ICSI, E-mail: narseo.vallina@imdea.org
Carmela Troncoso: Spring Lab EPFL, E-mail: carmela.troncoso@epfl.ch
Alessandra Gorla: IMDEA Software Institute, E-mail: alessandra.gorla@imdea.org

permission model, these SDKs enjoy the same set of permissions granted by the user to the host app. Apps also might inadvertently expose data to in-path network observers if they use insecure protocols to transmit sensitive data. These practices involve great privacy risks for minors, *e.g.,* if data is used to profile children's behavior or development, or if the data becomes compromised [64].

To help parents choose among parental control solutions, security centers at the European level [1, 33] have analyzed a number of parental control solutions. Their analysis considers four dimensions: functionality, effectiveness, usability, and security—defined as their effectiveness at deterring children from bypassing the system. However, these reports *do not provide any privacy risk analysis*, nor consider the lack of transparency or other potential violations of relevant privacy laws with specific provisions to safeguard minors' privacy, *e.g.,* the European General Data Protection Regulation (GDPR) [29] and the U.S. Children Online Privacy and Protection Act (COPPA) [35].

In this paper, we present the first comprehensive privacy-oriented analysis of parental control apps for Android from a technical and regulatory point of view. We study 46 apps using both static and dynamic analysis methods to characterize their runtime behavior and their data collection and data sharing practices. We also study the accuracy and completeness of their privacy policies, identifying potential violations of existing regulations to protect minors. We note that during our analysis we do not collect any data from children or any other user. (§ 3.1.4 describes the ethical considerations).

Our main findings are the following:

A. Static taint analysis reveals that both apps and embedded third-party libraries disseminate sensitive information, *e.g.,* the IMEI, location, or MAC address (§ 4). Apps also use *custom permissions* to obtain functionalities exposed by other apps' developers or handset vendors, revealing (commercial) partnerships between them.

B. We find that almost 75% of apps contain data-driven third-party libraries for advertisement, social networks, and analytic services (§ 5). Furthermore, 67% apps share private data without user consent (§ 6.3), even though some of these apps are recommended by public bodies (*e.g.,* SIP-Bench III [1, 33]). Despite processing children's data, 4% of the apps use libraries which claim to not be directed at children and thus do not take extra measures to comply with privacy laws specific to children (*e.g.,* U.S. COPPA rule [35]). Moreover, only two of the seven ad-related libraries found are COPPA compliant according to Google's 2019 list of self-certified libraries suitable for children apps [43].

C. The outgoing flows of 35% these apps are directed to third parties, but 79% of the apps do not name these organizations in their privacy policies (§ 6). We find 67% apps collecting sensitive data without user consent, and 6 apps that do not implement basic security mechanisms such as the use of encryption for Internet traffic.

D. Despite being required by regulations [29, 35], only half of the apps clearly inform users about their data collection and processing practices (§ 7). While 59% of the apps admit third party usage of sensitive data, only 24% disclose the full list of third parties embedded in the software. Furthermore, 18% do not report any data sharing activity even though we find evidence of third-party data collection through embedded SDKs.

2 Parental Control Apps

Android parental control apps offer diverse features to control children's digital activities. In line with previous work [56, 89], we classify apps that enable parents to monitor children's behavior, including location [47], as *monitoring tools*; and we classify apps that enable parents to filter content and to define usage rules to limit the children's actions as *restriction apps*. Some apps offer multiple functionalities simultaneously, and we label them as *monitoring* since it is the most invasive of the two categories.

The way in which parental control apps enforce these functionalities is varied. Common alternatives are replacing Android's Launcher with a custom overlay in which only whitelisted apps are available; installing a web browser where developers have full control to monitor and restrict web traffic; or leveraging Android's VPN permission [13] to gain full visibility and control over network traffic. The most common criteria to customize app's behavior are age and blacklisting/whitelisting. The former restricts the type of content or defines the level of monitoring depending on the children's age. In general, older children are granted access to a larger set of webpages and applications than youngsters. In the latter, parents can either list applications or webpages that they deem inappropriate for their children, or add appropriate content to whitelists.

Most parental control apps have two different components or modes: *i*) the parent app or mode, and *ii*) the children app or mode. The parent app can run either on the children's or on the parent's device enabling access to the children's app data, and to the dashboard for setting monitoring or blocking rules. When the parent mode runs on the children's device, parents' monitoring can be done locally on this device. However, when the parent and children app run on different devices, the children app often uploads information to a central server in order to enable remote access to the children's information to parents. Many apps in our study provide a web-based control panel in which parents can access all information, and change both control and blocking rules. This approach *requires* uploading data to the cloud, and as a results, parents must trust service providers to not disseminate nor process children's data for secondary purposes other than the ones declared in the privacy policy.

To assist developers of children's apps, Google has made public best development practices [5], as well as a list of self-certified third-party advertising and tracking libraries that respect children's privacy [43]. While parental control apps are not necessarily listed in the Designed for Families (DFF) program [42], a Google Play program for publishing applications targeting (and suitable for) children, given their nature we expect them to follow regulation's special provisions and follow best practices for collecting data about minors.

2.1 Regulatory Framework

Minors may be less concerned regarding the risks and consequences of the dissemination of their personal data to the Internet [29, 57]. This has motivated regulators to develop strict privacy laws to protect children privacy such as the Children Online Privacy Protection Act (COPPA) in the US. COPPA dictates that personal data of children under the age of 13 can only be gathered by online services, games, and mobile applications after obtaining explicit and verifiable parental consent [35]. In the EU, the General Data Protection Regulation directive (GDPR) enforces privacy transparency and requires data collectors to obtain consent from European subjects prior to gathering personal data from them, clearly stating the purpose for which it is collected [29]. As in the case of the USA COPPA rule, the GDPR contains additional articles in relation to the privacy of minors —namely Art. 8 [49] and Recital 38 [50] of GDPR—which require organizations to obtain verifiable consent from the parent or legal guardian of a minor under the age of 16 before collecting and processing their personal data. Both regulations explicitly prohibit the collection and processing of minor's data for the purpose of marketing or user profiling, and force developers to implement security best practices (*e.g.,* use of encryption) and to disclose the presence and data collection practices of embedded third-party libraries such as advertising and tracking services [29, 35].

3 Data Collection and Analysis Methodology

Google Play does not provide any specific category for parental control apps. We identify them by searching for the string *"Parental control app"* on Google Play's search engine. As this process can produce false positives, we manually eliminate any app that does not implement parental control functionalities. We repeated this process at different points during the project to add newly published apps to our study. In total, we found 61 parental control apps. Interestingly, the majority of these apps are classified as *Tools* (42% of apps) by developers, despite the presence of the more specific *Parenting* category (15% of apps).

Android app developers often release new versions to include new features to their software or to patch vulnerabilities [26]. Such changes may have an impact on users' privacy [73]. We crawl APKPure [14] —a public dataset which contains historical versions of Android applications—to obtain 429 old versions for the list of 61 apps so that we can study their evolution over time. We discard versions prior to 2016, as we deem them too old for our study. This also results in discarding 15 apps that have not been updated since 2016. Anecdotally, one of the developers only made the parent version available through Google Play, while its counterpart was only available through direct download at the developer's website. We decided to download the children version to avoid bias in our dynamic analysis. Our final dataset contains 419 versions from 46 parental control apps (listed in Table 8 in the Appendix).

For each app we harvest metadata publicly available on the Google Play store: app descriptions, number of downloads, user ratings, privacy policies, and developer details. We use this metadata in our analysis to contextualize our results, showing potential differences across developer type and app popularity, and for analyzing the completeness of privacy policies. [1]

Apps' popularity. The number of installs per app—a metric often used to infer a given app's popularity [86, 92]—varies widely. The dataset contains 22% of apps with over 1M downloads, whereas 37% of apps have less than 100k downloads. When considered together, the 46 apps have been installed by more than 22M users (lower bound). The most downloaded apps typically have a better user rating (on average 3.5 out of 5 for apps over 1M downloads) than those with a lower number of installs (on average 3.2 out of 5 for apps below 100k installs).

App developers. We extract the apps' signing certificates to identify the set of companies (or individuals) behind parental control apps. We find that most developers release one single parental control app, except for 3 developers that have released two apps. 21% of the developers also publish unrelated software in Google Play besides the identified parental control apps. The most remarkable cases are *Yoguesh Dama* [46], an Indian company which has published 38 apps (for volume control, screen locking, wallpaper, and more), and *Kid Control Dev* [45] a Russian company which also builds apps such as flashlights.

One may assume that the use of parental control apps is mostly predicated on trust. Therefore, we investigate whether the identity of the app developer plays a role in the parents' choice. Our initial hypothesis is that parents might prefer software developed by well-known security companies like *McAfee* or *Norton.* However, we find that these developers only account for 9% of total apps, and that they are not the most installed ones – only one of them reaches 1M installs. The most popular parental control apps are those developed by companies specializing in children-related software (*e.g., Kidoware* [54] and *Screen Time Labs* [77]). Most parental control apps seem to monetize their software through in-app purchases (from 1 EUR to 350 EUR) to unlock features and monthly licenses, yet they typically offer a free-trial period.

Delisted apps. We note that 10 applications were removed from Google's app store since our initial app collection (Feb. 2018). The reason why these apps have been delisted is unclear: their removal could be triggered by the developers (*e.g.,* companies no longer operating), or by Google Play's store sanitization process (Play Protect [7]). We still analyze these apps, and report when they cannot be tested because they no longer work.

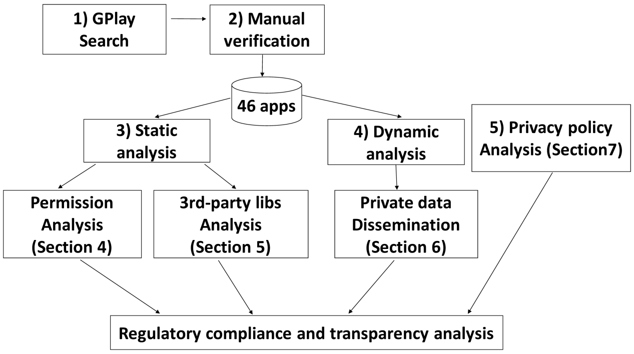
1) GPlay Search
2) Manual verification
46 apps
3) Static analysis
4) Dynamic analysis
5) Privacy policy Analysis (Section7)
Permission Analysis (Section 4)
3rd-party libs Analysis (Section 5)
Private data Dissemination (Section 6)
Regulatory compliance and transparency analysis

Fig. 1. Analysis pipeline to assess the privacy risks and regulatory compliance of parental control apps.

3.1 Analysis Pipeline

To carry out our analysis we take advantage of static and dynamic analysis techniques previously developed by the research community—in some cases extending them to meet our research objectives—as shown in Figure 1. We use both static and dynamic analyses to overcome the limitations they present when used in isolation [72]. § 8 discusses the limitations of our method.

3.1.1 Static Analysis

For each app and version, we first parse its Android Manifest file [11] to understand its high level behavior without analyzing the binary code. Concretely, we search for: *i*) apps with unusual permission requests, and *ii*) apps that request a different number of permissions across versions. We complement this analysis with static taint analysis to investigate potential personal data dissemination by apps and embedded third-party SDKs. Finally, we look at the binary code to identify third-party libraries embedded in the app using LibRadar [59]. This last step is critical to attribute observed behaviors to the relevant stakeholder. We present the results of our static analysis in § 4 and § 5. Static analysis has no visibility into server-side logic, and cannot analyze apps with obfuscated source-code. Further, it may report false positives so we complement it with dynamic analysis, as discussed next.

[1] The set of 46 apps may not contain every available parental control app, but the apps are diverse from different standpoints and can be considered as a representative the dataset.

3.1.2 Dynamic Analysis

The fact that apps declare a given permission or embed a given third-party SDK does not mean that the permission will be used or that the library will be executed. We use dynamic analysis to collect actual evidence of personal data dissemination (§ 6.2), assess the security practices of these apps when regarding network communication (§ 6.4), and identify consent forms rendered to the user at runtime (§ 6.3). We use the Lumen Privacy Monitor app [71], an advanced traffic analysis tool for Android that leverages Android's VPN permission to capture and analyze network traffic – including encrypted flows – locally on the device and in user-space. The use of Lumen only provides a lower bound to all network communications in an app since it can only capture flows triggered by user-, system- or environmental stimuli and codepaths at runtime. Achieving full coverage of parental control apps' actions is complicated and time consuming: we must manually set up a parent account and then exhaustively test the app by mimicking phone usage by a child. The process of testing the applications needs to be manual as the Android exerciser Monkey [8] is not able to either fill login forms or test app features in a non-random way [28]. We provide details on the testing methodology in § 6.2.

3.1.3 Privacy Policy Analysis

We conduct a manual analysis of the latest version of the apps' privacy policies (§ 7) fetched from their Google Play profile. We inspect them to identify: *i*) whether the apps provide understandable and clear terms of use and privacy policies to end users; *ii*) whether they are compliant with the provisions of privacy regulations; and *iii*) whether their disclosed policies match their behavior in terms of access to and dissemination of sensitive data and the presence of third-party libraries.

3.1.4 Ethical Considerations

Previous work has shown the use of parental control applications may have ethical implications [27, 61], therefore **we do not collect any real data from children or any other user**. We perform our data collection on fake accounts owned and operated by ourselves. Furthermore all interaction with the apps is done by the authors of this paper, without any children or end user participation.

We put our focus on understanding the security and privacy practices of these apps to determine to what extent they could be a privacy threat to children. We also stress that some of the privacy issues found in mobile applications might be the result of bad coding habits and lack of awareness—specially when integrating privacy-abusive third-party SDKs—, rather than intentional developer misbehavior. We communicated our findings to the Spanish National Data Protection Agency (AEPD) and other government agencies promoting Internet safety, namely INCIBE's IS4K [1].

4 Permission Analysis

Android implements a permission model to control application's access to personal data (*e.g.,* contacts and email) and sensitive system resources (*e.g.,* sensors like GPS to access location) [10]. The analysis of the permissions requested by an app offers a high level idea of what protected system resources and data the app has access to. Besides Android's official permission list defined in the Android Open Source Project —grouped in different risk levels, the most sensitive ones labeled as "dangerous"—any app can define its own "custom permissions" [40, 83] to expose its resources and capabilities to other apps. We study the use of Android permissions and their evolution across releases to study what data parental control apps access, how it changes across app versions [26, 73], *e.g.,* to adapt to stricter privacy requirements at the platform level [12], whether this data is potentially sent over the Internet, whether these permissions are used by third-party SDKs, the app itself, or by both; and whether there are noticeable differences between monitoring and restriction apps in terms of permissions request.

4.1 Prevalence and Usage of Permissions

Figure 2 shows the permissions requested by each one of the apps in our dataset, as well as the category of the permission. The top part of the plot shows restriction apps and the bottom part lists monitoring apps. Column-wise, we differentiate permissions according to their privacy risk level [6]:

A. Dangerous permissions: (first block from the left, in blue)–*i.e.,* those that could potentially affect the user's privacy or the device's normal operation, and therefore should be explicitly granted by the user.

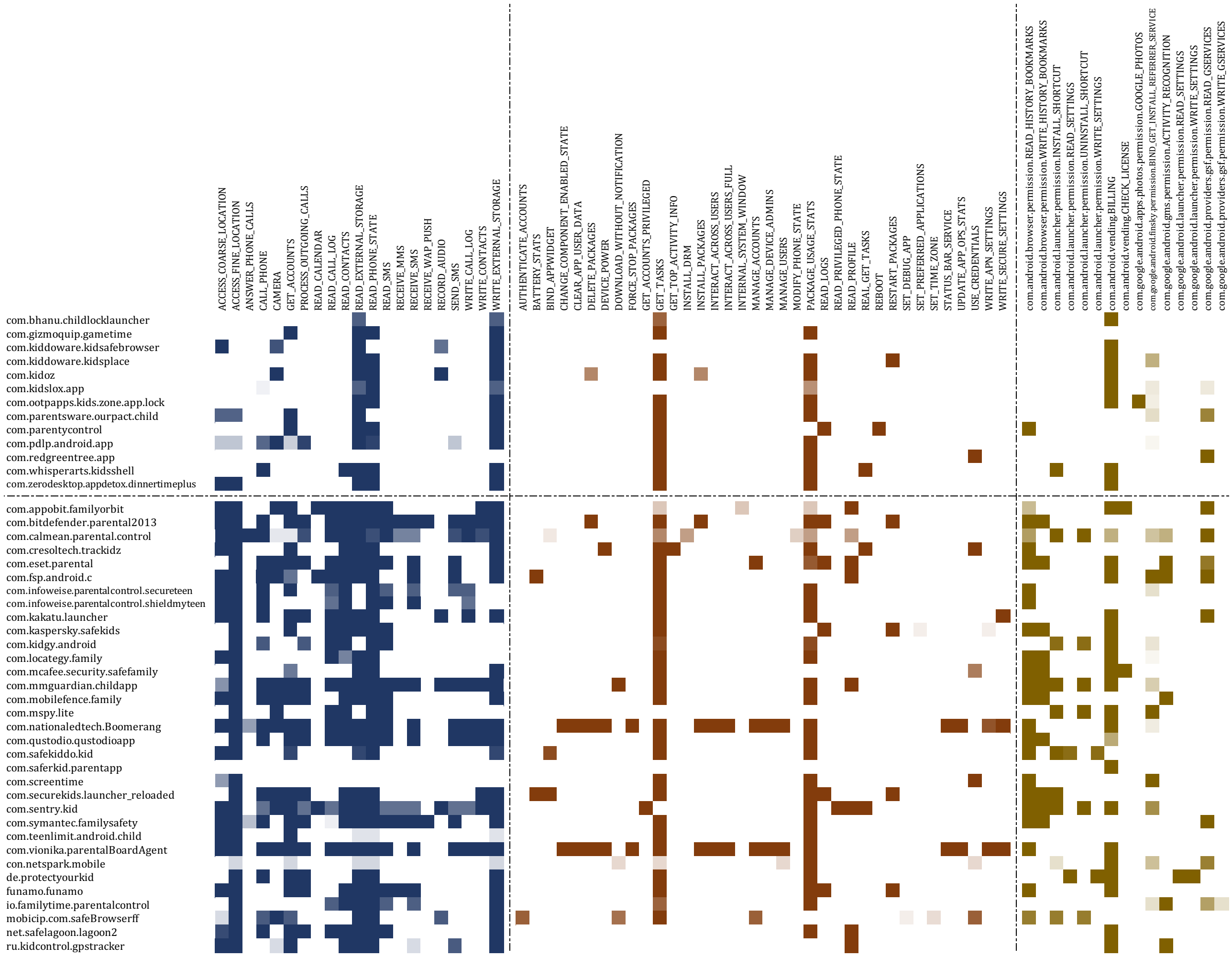

Fig. 2. Permission heatmap. We list one app per row, and we indicate the permissions it requests. We show restriction apps at the top, and monitoring at the bottom. We differentiate dangerous permissions (blue), from signature (orange), and custom (yellow). Gradient shows the percentage of releases requesting the permission, with darker meaning a higher number of releases.

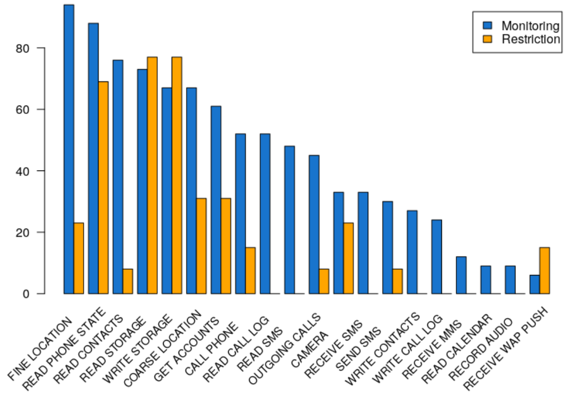

Fig. 3. Comparison of percentage of apps asking for dangerous permissions between Monitoring and Restriction apps.

B. Signature permissions: (second block, in orange) —*i.e.,* those that the system grants only if the requesting application is signed with the same certificate as the application that declared the permission. We also render system permissions in the same category, even those that are deprecated in recent releases of Android, but still work on older releases.

C. Custom permissions (last block, in yellow) —*i.e.,* those that apps define to protect their resources.

Due to space constraints we do not display normal permissions —*i.e.,* those which regulate access to data or resources that pose little risk to the user's privacy, and are automatically granted by the system. However, they are part of our study.

We use color gradients to show apps' permissions request changes over time in Figure 2. Darkest colors show that all releases of the app request the permission (*i.e.,* no changes). Figure 2 shows that, as expected due to their invasive nature, in general, parental control apps need lots of permissions to offer their service. Considering the median value, parental control apps request 27 permissions, 9 of them being labeled as dangerous. For comparison, the top 150 apps from the Google Play Store in May 2019 request 18 permissions, 5 of them labeled as dangerous. In fact, we find that the most extreme parental control app in our dataset (*Boomerang*) requests 91 permissions, of which 16 are dangerous; much more that typically considered privacy invasive non-parental control apps.[2]

We find that over 80% of parental control apps, regardless of their category, request access to location, contacts, and storage permissions. Popular non-dangerous permissions requested by parental control apps are related to Android's package manager (possibly to monitor installed apps), and also to control system settings. Despite the fact that we focus primarily on dangerous permissions, it is important to highlight that non-dangerous ones can be harmful too. One example is the `BIND_ACCESSIBILITY_SERVICE` permission, which is requested by 11 apps of the monitoring category, and 1 of the restriction apps. This permission can be used to monitor and control the UI feedback loop and take over the device [39]. During the installation process many of these apps explain that such permission is necessary to better control the actions of the child. Likewise, despite not being explicitly flagged as "dangerous", some of the deprecated system permissions are also worth noting, since they protect low level functionalities such as the ability to retrieve the list of running processes on the device. Furthermore, the permission model can be exploited to access personal data without user consent via side-channels like using WiFi information to get city level location or the MAC address information as a unique identifier [72].[3] While the type of service provided by these apps might justify the use of such dangerous permissions, the presence of third-party libraries that can piggyback off these privileges to gather children's data is a privacy threat that we will analyze in detail in § 5. [4]

At the other side of the spectrum, we notice that two parental control apps do not request any dangerous permission. One of them (*Parental Control App by Redgreentree*) replaces the Android default launcher to let the child use only apps approved by the parent. The other app (*SaferKid*) is a parent app (which explains the lack of dangerous permissions) present in our dataset because the developers only made the companion app and not the children version of the app available in Google Play.

Anomalous permission requests. While most of the apps requests dangerous permissions, we find that some are rarely used by most parental control apps. We analyze in depth the permissions that are requested by only 10%, or less, of the apps in our dataset,—considering monitoring and restriction apps separately—to identify whether these permissions are justified. We also search for any information on their privacy policy that may justify their use. Table 1 provides a summary of the anomalous permissions for which we found no justification in the app's description or privacy policy. Two restriction apps request permissions to send SMS, process outgoing calls and read contacts. Yet, we could only find a justification for one of them in the app's description. Regarding the monitoring category, we identify seven apps requesting anomalous permissions (*i.e.,* receive WAP push, record audio and read calendar). Three of them are no longer indexed on the Google Play Store (*Kakatu*, *Family Orbit* and *Bitdefender*). The remaining four do not justify why they require access to these permissions in their Google Play description or privacy policy.

Permission requests across app releases. Previous studies show that Android apps tend to increase the number of requested permissions across app releases [25, 82, 87]. While 24% of the apps in our dataset increased the number of permissions requested over time, we find that another 24% of the apps decrease the number of permissions requested over time, *opposite* to the general trend. Part of this decrease is explained by the fact that in early 2019 Google changed its permission policy and disallowed apps to access SMS and calls permissions unless they were the default messaging app [44].

Restriction vs. Monitoring apps. Monitoring apps' goal is to gather behavioral and usage data from children's phone and reporting it to their parents, either via a dedicated web portal or in a parent-specific companion app. Therefore, as Figure 2 reveals, monitoring apps tend to request more dangerous permissions than restriction apps. Figure 3 goes deeper into the specific

[2] For reference, *Facebook* requests 59 permissions, 16 of which are dangerous.

[3] In fact, Android 10 introduced special permissions to avoid the usage of the MAC address as a location surrogate and the access to non resettable identifiers

[4] Because of Android's permission model, third-party libraries inherit the same set of permissions as the host application.

Table 1. Unusual permission requests. We report whether the apps have been unlisted from Google Play (GPlay column).

App	GPlay	Permissions
com.pdlp.android.app		PROCESS_OUTGOING_CALLS SEND_SMS
com.whisperarts.kidsshell		READ_CONTACTS
com.appobit.familyorbit	✓	READ_CALENDAR
com.mmguardian.childapp		RECORD_AUDIO
com.bitdefender.parental2013	✓	RECEIVE_WAP_PUSH
com.sentry.kid		READ_CALENDAR RECORD_AUDIO
com.symantec.familysafety		RECEIVE_WAP_PUSH
com.fsp.android.c		READ_CALENDAR
com.kakatu.launcher	✓	RECORD_AUDIO

cases. Our empirical observations are aligned with our expectations: compared to restriction apps, monitoring apps commonly request permissions necessary to gather children data such as geolocation, read call log or read SMS. The difference between monitoring and restriction apps is notable even for custom permissions (see Figure 2), and in particular regarding browser controlling permissions. Monitoring apps access the browser bookmark history more than restriction apps; however, restriction apps tend to rely more heavily on custom-specific browser permissions to control the browsing activity on the child's device.

Possible information dissemination. The high request rate of dangerous permissions does not imply that parental control apps actually disseminate sensitive information over the network, either to their own servers or to third parties. We apply static taint analysis to estimate to what extent these apps access and disseminate the sensitive information they access and with whom. Using the context and flow sensitive analysis implemented in Flowdroid [16], we observe that the apps in our dataset include 1,034 method invocations to access sensitive data when aggregated, never less that 78, and on average 324. We also observe that 67% of apps also have at least 1 sink—a part of the OS where data can leave the system, such as the network interface—in reachable code, suggesting potential information leaks. Flowdroid reports at least one valid information flow in 14 of these apps, with the extreme case of *Kids Zone*, for which it highlights *72 potential leaks.* While it seems that most of the reported information leaks involve logging user actions (e.g., opening a new activity, logging a failed/successful authentication, etc.), we also observe more critical leaks involving sensitive data like unique identifiers. Specifically, we find that two apps (*SecureTeen* and *ShieldMyTeen*) share the MAC address and SSID through the Internet (they can be used as a side-channel to uniquely identify the device and geolocate the user [72]). The aforementioned apps and *Dinner Time Plus* also disseminate the device IMEI, a unique identifier; and four apps (*MMGuardian*, *Shield My Teen*, *GPS Tracker* and *Mobilefence*) disseminate GPS-level geolocation. We will further investigate in § 5.2 the origin and destination for these information leaks.

Custom Permissions. Android gives developers the opportunity to define custom permissions in order to expose parts of the apps' functionalities to other apps [83]. In the case of custom permissions the user never gets to accept or reject them as they are automatically granted without showing a warning or consent form to the user, as opposed to Android official permissions. While we did not render these permissions in Figure 2 for clarity reasons, we include a plot showing the number and type of custom permissions for each app in our dataset in Figure 5 in the Appendix.

We find 28 custom permissions that—judging by their name—have been declared by parental control app developers. We find examples of custom permissions from companion apps, *e.g.,* the *Spin browser* [21], that substitutes the default user browser, or the companion parent app (*e.g., com.safekiddo.parent*). These permissions are used from the children app to access or control functionalities of the companion apps. We also find apps using custom permissions declared by other developers, such as the app *Parents Around* using custom permissions from *Protect your kid.* This suggests the existence of agreements between app developers to leverage functionality implemented by other apps when both are installed on the same device. We also find several apps using custom permissions related to phone manufacturers, possibly enabled by pre-installed applications [40]. These vendor-declared permissions allow app developers to gain access to other system features not available through the official Android Open Source Project. In some cases, as in the case of the Samsung KNOX API, app developers must become Samsung partners to gain access to their proprietary APIs [76]. Although the parental control app *Parents Around* requires access to custom permissions belonging to five manufacturers, in general apps in our dataset tend to declare permissions for either one vendor or none of them. The most frequent custom permissions are related to Samsung and they are used by various releases in 21 apps of our dataset. The most common vendor permissions are declared by Huawei (10 apps), HTC (8 apps), and Sony and Oppo (4 apps).

5 Third-party Library Analysis

Due to Android's permission model, any SDK embedded in an Android app inherits the privileges and permissions of the host app. This gives many organizations, including third-party providers offering analytics or advertisement services, access to the same data as the parental control app.

Many third-party library providers often prohibit their usage in children-oriented software, such as *Branch* [23] and *Appboy* (now rebranded as *Braze*) [24] due to the strict regulatory frameworks implemented in the USA and the EU to protect minors. Additionally, Google released in May 2019 a list of third-party libraries that self-verify being in compliance with the provisions set by the GDPR and COPPA [43].

In this section, we inspect the parental control apps—across app versions— in our dataset to find third-party libraries and classify them by the functionality provided using LibRadar [59]. We manually analyze its output to sanitize the results and improve LibRadar's coverage, also mapping them to the actual organization providing the service. [5] After this process, we could successfully identify 157 unique libraries (44 could not be classified due to the use of code-obfuscation techniques). Then, we use publicly available information to classify each one of the libraries by its purpose, including their own product descriptions and developer forums.

Table 2. Classification, number and description of third-party libraries found in parental control apps.

Category	Lib #	Description
Social Network	1 (60 pkgs)	Social networks
Advertisement	7 (82 pkgs)	Advertisement
Development	56 (582 pkgs)	Development support tools
Functionality	29 (172 pkgs)	App features (*e.g.*, animations)
Support	43 (443 pkgs)	Support libraries (*e.g.*, DB drivers)
Analytics	21 (220 pkgs)	Analytics services
Unrecognized	44	Unidentified libraries

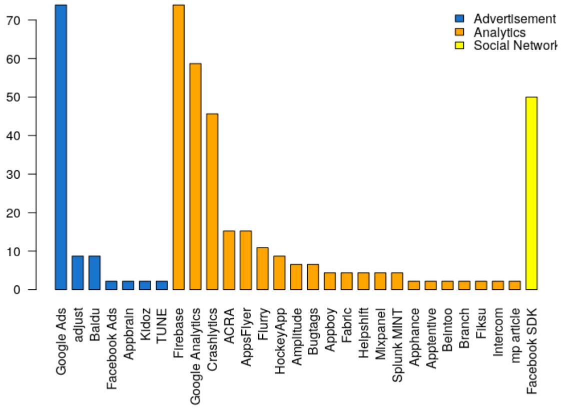

Fig. 4. Percentage of apps using Advertisement, Analytics and Social Network libraries.

Table 2 shows for each resulting category the number of SDKs found, the total number of package names matching each one of these libraries, and a short description of the category. Most third-party libraries embedded in parental control apps are tools for development support—general purpose development libraries, JSON parsers, UI support libraries, etc—followed by SDKs offering analytics services and advertisements — *e.g.,* Flurry and AppsFlyer. The latter category of SDKs are more concerning from a privacy standpoint given their data-driven behavior and business models. Therefore, we focus our analysis on the libraries providing social network integration, online advertisement, and analytics services.

Figure 4 shows that Google Ads, Google Firebase and Google Analytics are present in over 50% of the apps, followed by Facebook SDK at 43%. Some of these libraries belong to the same parent company. For instance, Crashlytics, Firebase, Google Analytics, and Google Ads all belong to Alphabet [3]. Therefore, when we group the SDKs by their parent company, we can observe that Alphabet is present in 93% of the apps in the dataset.

5.1 Apps Violating Libraries' ToS

Finally, we study whether the third-party libraries embedded in each parental control apps allow their use in children-oriented software in their Terms of Service. Reyes *et al.* showed that several apps designed for children and published in Google Play's Designed for

[5] For example, LibRadar can report multiple package names for the same library (*e.g.,* `com/paypal/.../onetouch` and `com/paypal/.../data`) so we cluster those belonging to the same provider: PayPal.

Family program (DFF) used third-party libraries whose Terms of Service (ToS) prohibit their integration in children apps [74]. Following this insight, we use a static analysis pipeline based on Soot and Flowdroid [16, 85] to check whether parental control apps follow a similar behavior. We note that some SDK providers changed their ToS between our study and the previous work by Reyes *et al.* We also study whether these libraries are included in Google's list of self-certified SDKs suitable for children apps (as of May 2019) [43].

While both AppBoy and Branch still indicate that they should not be included in applications directly targeting children, our static analysis pipeline indicates that they are present in children-oriented software as parental control apps. We verify that two apps in our dataset (*GPS tracker* and *Qustodio*) integrate these libraries in the latest version available in our dataset (November 2018). Out of the seven advertising and analytics libraries that we find in our dataset, only two (AdMob and Unity) are present in Google's list. We include the relevant wording extracted from their Terms of Service in the Appendix.

5.2 Who Handles Sensitive Data and What for

Android does not force developers to explain to users whether a given permission is needed by the app's functionality and if it is also accessed by an embedded third-party SDK. Therefore, users cannot make informed decisions on whether to grant a given permission to the application or not based on which software component will gain access to it. To know who is responsible for the data collection enabled by a given permission, we statically analyze the bytecode of the most recent version of each parental control app to search for Android API calls protected by Android permission and then attribute these calls to the host application's code or to an embedded third-party SDK. For that, we updated the mapping between permissions and protected API calls originally made by Au *et al.* to incorporate recent changes in Android's permission model [17, 19]. We rely on LibRadar's package identification to attribute the protected API requests to the actual app, or to any embedded third-party SDK.

Table 3 summarizes the results of this analysis. For each API call protected by dangerous permissions, we report whether it is used only in the application code, only in the library code, or by both. We have an additional column reporting the percentage of uses for which we cannot attribute API calls to each software component for reasons such as: *i*) the permission being requested by the app developer in its Android manifest file but not being invoked in the code; *ii*) our method misses relevant API calls because of incomplete information in Au's mapping [17] [6]; and *iii*) the use of code obfuscation techniques. [7]

Table 3. Attribution of permission usage to code belonging to application or third-party libraries for each dangerous permission. The last column reports uses that we cannot attribute.

Permission	% app only	% lib only	% app & lib	% unclear
ACCESS_COARSE_LOCATION	36	12	52	0
ACCESS_FINE_LOCATION	33	9	55	3
CAMERA	14	7	7	72
GET_ACCOUNTS	25	12	4	59
PROCESS_OUTGOING_CALLS	12	0	0	88
READ_CALENDAR	33	0	0	67
READ_CONTACTS	8	0	0	92
READ_PHONE_STATE	39	26	18	17
READ_SMS	19	0	6	75
RECEIVE_MMS	0	25	0	75
RECORD_AUDIO	20	20	0	60
SEND_SMS	27	27	18	28
WRITE_CALL_LOG	12	12	0	76
WRITE_CONTACTS	22	11	0	67
WRITE_EXTERNAL_STORAGE	6	41	47	6

A significant number of the calls to dangerous permission-protected methods are invoked **only by embedded third-party libraries**. For instance, embedded SDKs access the fine location permission and the read phone state permission, [8]— 9% and 26% of the times respectively—when the host app does not require access to them. In other cases, SDKs piggyback on the set of permissions already requested by the host app: *e.g.,* 55% of the times that fine location is requested by the app, a third-party library also access the permission. This suggests that users' personal data might

[6] Despite our manual efforts to complement the mapping and incorporate recent changes in the Android permission control model, we cannot guarantee its completeness, since some mappings are not yet properly documented in the Android API.

[7] Seven instances of apps reading unique identifiers like the MAC address, SSID, and IMEI were identified in obfuscated code, thus they could not be attributed.

[8] A permission that allows reading unique device identifiers.

be collected and processed for secondary usages beyond those offered by the app.

On a per-app basis, the apps *MMGuardian* and *Mobilefence* access users' geolocation only in application code but, for other apps, only in third-party code. Specifically, we observe that the library *com.proximity* —Google's proximity beacon—is embedded in the app *ShieldMyTeen*. This case might be justified for the need to leverage geo-fencing methods. The app *GPS Tracker* leaks the location in code that belongs to the library *Arity* [15], which is a platform that collects location information to better understand and improve mobility. While we find this library highly unusual for parental control apps, we find that this apps is a "Family GPS tracker" which can be used by parents as well. The developers are very explicit in their privacy policy about sharing data with this company for driving features present in the app.

In the following section we use dynamic analysis to bypass some of the static analysis' limitations, reporting evidence of the collection and dissemination of children's personal data by these third-party libraries.

6 Dynamic analysis

The results of our static analysis show that some parental control apps and embedded third-party SDKs potentially collect and disseminate personal children data to the Internet. In this section, we use dynamic analysis methods to complement our static analysis and collect actual evidence of personal data dissemination to the Internet. To that end, we perform the following user actions to execute and test each app —or pair of apps if there is a parent or companion version—in our database:

A. We install the app and go through the setup process. We create a parent account when necessary, and grant consent to collect sensitive data when asked. In order to perform an apples-to-apples comparison, when prompted, we configure each app to monitor (or block) the activities of a child below 13 years of age. It is also important to note that we run all tests from a European country (Spain) and that when creating the account we are consenting to the privacy policy and terms of services of the app.

B. Due to scalability issues, we cannot test the whole spectrum of children actions that can be blocked or monitored by a parental control app. Therefore, we decided to focus on those that are more likely to be blocked based on their permission requests or advertised features. Specifically, we manually interact with the children device for five minutes as follows: (*i*) we visit a newspaper in the browser (this action is typically allowed for children), but also one pornographic website, and a gambling service (this type of traffic should be typically banned); (*ii*) we open a permitted child game and a potentially blacklisted dating app; (*iii*) we install and uninstall a game; (*iv*) we take a picture with the default camera app and (*v*)—while the test phone does not have a SIM card—we go to the phone and SMS apps and attempt to make a phone call and send a message.

Our analysis estimates a lower bound of all the possible instances of personal data dissemination that might exist in our dataset of parental control apps when compared to our static analysis. Also, since the Android VPN API only allows one app to create a virtual interface at a given time [9], our method does not allow us to fully test 3 parental control apps that require access to the VPN interface to monitor the children's network communications. We still report any data dissemination that might happen prior to or during the VPN interface being setup.

6.1 Third-party Services

Our results confirm the high presence of third-party components in children-oriented apps as summarized in Table 4. We observe network connections to a variety of destinations in the recorded network flows. We find 49 different second-level domains across all apps, 18 of which are associated with third-party Advertisement and Tracking Services services according to the list developed by Razaghpanah *et al.* [70]. The most contacted domain is *Crashlytics*—now Firebase—which is a Google-owned bug reporting and analytics service contacted by 54.8% of apps. Every third-party domain found in our dynamic analysis experiments belongs to SDKs previously reported by our static analysis method. This also verifies some of our claims. For instance, dynamic analysis confirms that the app *GPS Tracker* contacts *Branch* services, an analytics platform to increase revenues through "links built to acquire, engage, and measure across all devices, channels, and platforms" [22] that is not supposed to be used in children-oriented software according to their own ToS.

Table 4. Most popular third parties by apps contacting them

Third party	Type	# apps
Crashlytics	Crash Reporting, Analytics	23
Facebook Graph	Social Network, Ads, Analytics	15
Appsflyer	Analytics	5
Adjust	Ad Fraud, Marketing, Analytics	3
Google ads	Advertising	3
OneSignal	Push Notifications	3

Table 5. Data dissemination without consent

Data Dissemination Type	Count (Unique SLD)	
	1st parties	3rd parties
Android Advertisement ID	2	8
Android ID	4	11
AP MAC address	1	0
WiFi SSID	2	0
IMEI	4	1
Geolocation	0	1
Email	2	0

6.2 Personal Data Collection and Dissemination

Using Lumen we identify 513 flows containing personal data (*i.e.,* resettable and persistent unique identifiers, geolocation, WiFi and Access Point information which can be used as a proxy for geo-location, and list of packages installed) generated by 42 (91%) different apps. We note that the access to the WiFi AP is a known side-channel that has been used as a proxy to access users' geo-location [72].

The fact that we see private data in network traffic does not necessarily imply a privacy violation. This information might be uploaded to the cloud to deliver its intended service, and to inform the companion app or the parent directly through a web dashboard. In fact, 74% of these apps fall in the monitoring category, which are those that request a higher set of permissions. Yet, some types of data like unique IDs and location are extremely sensitive and should not be shared with third-party services, particularly those offering advertising and tracking services that do not comply with child privacy rules. We observe 4 apps disseminating the location to a third-party domain in. Examples of third parties collecting location information are analytics SDKs used for client engagement and app growth (*Amplitude*), and mobile push notification services (*OneSignal*). Other usages might be for providing a service to the parent, as in the case of mapping APIs (*Google Maps*). None of these providers are in Google's list of SDKs suitable for children oriented apps [43].

Google encourages developers to use the Android Advertisement ID (AAID) as the only user identifier [5]. The AAID is non-persistent and users can reset it or opt out of "Ads personalization" in their device. The combination of the AAID with other persistent identifiers without explicit consent from the user is also a violation of Google's Terms of Service [4]. Despite this, we find 24 apps collecting and sharing persistent identifiers such as the IMEI, and 58% apps uploading the AAID along with persistent identifiers, hence defeating the purpose of resettable IDs. This behavior has been previously reported for regular children apps published in the DFF program [74]. Looking further into this issue we find that half of the apps sending the AAID alongside another unique identifier do so to a third-party library.

6.3 User Consent

Both COPPA and GDPR state that companies must obtain verifiable parental consent before gathering data from children below the age limit (13 years of age for COPPA, 16 for GDPR) [29, 35]. App developers should obtain verifiable parental (or legal tutor) consent before collecting sensitive personal data from the minor using techniques such as sending an email, answering a set of questions, or providing an ID document or credit card details at runtime [34]. This legal requirement implies that informing parents or legal tutors about data collection practices on the privacy policy is not enough, especially if the app disseminates sensitive data to third-party services as previously discussed.

In our previous analysis, we consented to data collection in all our experiments by creating an account and operating the app impersonating a child. Nevertheless, we want to determine whether apps collect private data without parental consent. To do so, we rely on the automatic method previously proposed by Reyes *et al.* [74]: we launch each app and run it for five minutes *without* interacting with it. This implies that we *do not* actively consent to data collection and we do not carry out any of the children actions, opting instead to leave the app running with no input. As a result, any sensitive or personal data—particularly unique identifiers and geolocation—uploaded by the app to third parties may be a potential violation of COPPA and GDPR.

Table 6. Apps sending sensitive data without encryption

Application	Data Type	Destination
com.kidoz	AAID	kidoz.net
ru.kidcontrol.gpstracker	IMEI & Location	85.143.223.160
com.parentycontrol	Email	parentycontrol.com
com.safekiddo.kid	IMEI	safekiddo.com
com.kiddoware.kidsplace	Android ID	kiddoware.com
com.kiddoware.kidsafebrowser	Android ID & Hardware ID	kiddoware.com

We find 67% apps disseminating personal data without explicit and verifiable consent. The information shared by these apps includes unique identifiers (*i.e.,* the android serial id, the AAID, or the IMEI) and the location of the user (including alternative location methods such as the SSID or the AP MAC address which have been prosecuted by the U.S. FTC before [36, 72, 74]). 47% of all cases of non-consented data dissemination are going to a third-party. Table 5 summarizes the data types disseminated by the tested apps, grouped by the type of data being disseminated. We note that none of these libraries are in Google's list of certified suitable for children SDKs [43]. We believe that some of these instances may correspond to developers being careless in the way that they integrate third-party libraries, not making sure that the user has read and accepted their data collection methods before starting to collect data.

6.4 (Lack of) Secure Communications

Both the COPPA [35] rule and the GDPR [29] have clear provisions stating that developers must treat data about children with an appropriate degree of security [41, 48]. To assess that this is the case, we study whether parental control apps make use of encryption (*e.g.,* TLS) to upload the data to the cloud. Table 6 summarizes the non-encrypted flows, sorted by pairs of apps and domains, as well as the type of sensitive data that is transmitted in the clear. We find instances of persistent identifiers that enable tracking of users, *e.g.,* the IMEI and the AAID, or geolocation information, being sent in the clear. Finally, we observe one app (*Secure Kids*) uploading without encryption the list of installed packages (used at the server to enable parents to block unwanted apps) alongside an identifier-like string (DEV_ID). The app *com.kiddoware.kidsafebrowser* appears in our results because it is installed as the default web browser by one of the apps in the dataset.

7 Privacy Policy Analysis

Both GDPR and COPPA require app developers to provide clear terms of use and privacy policies to the end user. However, it is known that privacy policies may be difficult to understand—even yielding different interpretations—by average users [53, 67, 79, 88]. We manually inspect the privacy policies indexed by Google Play Store for each app in our corpus to analyze to what extent parental control app developers respect the transparency obligations enforced by regulation. To avoid biases introduced by ambiguous interpretations of the policies, two authors conduct independent manual analysis of each policy, and discuss with a third author in case of disagreement. We note that only 89% of app developers had published a privacy policy on their Google Play profile at the time of conducting our study in February 2018. We discuss the results of our privacy policy analysis along four axis, including examples of wording found in the privacy policies as examples:

General considerations: We first look at whether the policy can be found easily on Google Play, if it provides clear information about the company (*i.e.,* location of company, main purpose, other apps), and if users are notified upon policy changes. We find that 95% of apps provide a direct link to their policies in their Google Play profile, and 10% of apps provide information about the companies. (*e.g., The "Company" or "We/us", with domicile at [...]*) However, despite the fact that changes in privacy policies over time are hard to monitor by users, only 20% of the apps state in their policies that they actively notify users upon policy changes.

Data collection practices: We find that most privacy policies (92.6%) report the kind of data being collected from minors. However, only a few of them (54%) clearly inform users of how these data are processed and for what purpose (*e.g., Your account and contact data are used for managing our relationship with you*). We also see that only roughly half (56%) of the policies portray how long collected data will be stored at the developer's servers.

Data sharing practices: Our static and dynamic analysis show that many parental control mobile applications rely on third-party libraries offering crash reporting, analytics, or advertising services. When any embedded third-party service collects or processes personal data from users, both GDPR and COPPA require developers to list them on the privacy policy of the app. We find that 59% of apps talk about third-party service usage in their privacy policy, but only 24% of the apps list the type of data being sent to them. Finally, companies can also share or sell data to other companies (*e.g.,* data brokers). Even though we see that 78% of policies talk about the possibility of sharing customer's data, they say that this will only happen as a result of company acquisition or legal compulsion.

Third-party service disclosure: We verify whether apps are transparent when they refer to third-party service usage in their policies by cross-checking these policies with our empirical findings regarding the usage of third-party libraries and data dissemination (§ 6). Since GDPR introduced the need to name the third-party companies receiving personal data, we check this in the 25 apps for which we have privacy policies collected from after the GDPR became effective on the 25th of May of 2018. Table 7 shows the number of apps using a third-party service and how many of them actually report this in their privacy policy (*e.g., Google may use the Data collected to contextualize and personalize the ads of its own advertising network*). Only 28.0% name all the third-party services that we find during runtime, while 79% of the studied apps do not name any third parties that they share private data with. We note that the latter apps are potentially in violation of both COPPA and GDPR for not being clear about their data sharing partners. Furthermore, we find one app that shares location data with a third-party service, which can constitute a potential violation of the FTC COPPA rule which prohibits the collection of children's geolocation sufficient to identify a street name and city or town. While this app is not directed only at children, its company name is *Family Safe Productions*, hinting that it can be used for monitoring child locations. Nevertheless, they openly say in their policy that they can share location data with third parties (as reported in § 5.2). We note that we did not find this behavior in our dynamic analysis experiments.

Regulatory compliance: We also study developers' awareness about different related regulation. Only 22% of apps claim COPPA compliance in their policy; and, while only 10% of policies talk about European legislation, 37% mention their compliance with local laws (*e.g., Our Privacy Policy and our privacy practices adhere to the United States Children's Online Privacy Protection Act ("COPPA"), as well as other applicable laws*).

User privacy rights: Finally, we check the rights and choices that users have about their data. To do so we look at the number of apps that allow users to opt out of the data collection practices without having to cease the use of the app, and how many apps give the user a chance to correct, delete, or access their data. We find that in 46% of apps users can opt out of data collection, where 63% of apps give users at least one of the above choices (*e.g., According to European Union applicable data protection legislation (GDPR), data subjects shall have the right to access to data, rectification, erasure, restriction on processing [...]*).

Table 7. Third party presence in apps and on their privacy policy

Third party	Type	# apps	# apps listing them in policy
Crashlytics	Crash Reporting, Analytics	23	5
Facebook Graph	Social Network, Advertising, Analytics	15	3
Appsflyer	Analytics	5	1
Adjust	Ad Fraud, Marketing, Analytics	3	0
Google ads	Advertising	3	1
OneSignal	Push Notifications	3	1
Amplitude	Analytics	2	0
Help Shift	Customer service	1	0
Apptentive	Analytics, User Engagement	1	0
Branch	Analytics	1	0
Splunk	Analytics	1	0

7.1 The GDPR Effect

We first analyzed the privacy policies at the beginning of our study, around three months before the GDPR became effective in May 2018. While we do not revisit the whole analysis, we use the results from our study to evaluate the evolution of the privacy policies for the 25 apps that share data with third parties and that are still available after the 25th of May 2018. Our goal is to learn if companies changed their policies, to understand the impact of the law in the specific case of parental control apps. Comparing the newer policies with the ones of our previous analysis, we find that 2 have major changes in their privacy policy after GDPR. Looking more in detail into the specific cases we find that one app initially did not name its data sharing partners in the privacy policy, and later added that info. Another app did not explain clearly the type of data being collected, and has now changed its policy to be clearer. However, we see that most policies are still unclear as they often omit names of third-party services used by the application. Thus, even when regulatory frameworks push to shed light into the third-party mobile ecosystem, we see the necessity for external auditing of mobile parental control applications beyond usability, and capability aspects.

8 Discussion and Limitations

Our multilateral analysis of parental control apps brings to light a large number of undesirable behaviors regarding children's privacy. First, parental control applications request a large number of invasive permissions. This, combined with the number of analytics and advertisement SDKs embedded in these apps pose a serious threat for children. Not only do these libraries piggyback on the large number of sensitive permissions requested by parental control apps to collect personal data; but there are a number of requested permissions used solely by third-party components. We also find empirical evidence of data sharing practices with third parties: 72% of the apps share data with a third-party SDK, and in 67% apps this sharing happens without explicit and verifiable parental consent. In some cases we observe that the upload of sensitive data to online servers happens without encryption. Finally, our analysis of the apps' privacy policies reveals that they are far from clear about their data collection practices, and usually underreport on what data is shared with other services. Such practices put in question the regulatory compliance of many of these apps.

Given the severe privacy risks revealed by our analysis, we consider that privacy implications should be a fundamental part of any guide designed for parents. We compare our findings with existing recommendations by public bodies to help parents decide whether to use this type of software and which tool is more secure. Our dataset contains 5 of the 10 apps benchmarked in the SIP benchmark [33] by the European commission, which focuses on usability and resistance to children sidestepping attempts. Two of these apps follow privacy-risky practices: *Qustodio* (also mentioned on a large amount of online studies that recommend top parental control apps [68, 75]) shares data with third-parties without consent, and *Parentsaround* does so in addition of sharing unique identifiers with third-party services. We also compare our results to IS4K Cybersecurity, which lists a series of apps, enumerating their functionalities without any judgment on their suitability. Our dataset includes 6 out of the 10 apps in their list. Three of these apps share sensitive data with third parties without appropriate consent, and another app sends data unencrypted.

Furthermore, we argue that app stores should take extra measures for verifying that applications directed at children comply with current legislation and treat children data with extreme care even if they are not in the designed for families program. While a complete and exhaustive analysis of apps could be unfeasible, static analysis showing the presence of analytics and advertisement libraries in apps that will be used by minors—including SDKs that prohibit their use in children apps—should raise concerns. Furthermore, simple dynamic analysis to verify that these apps use the most basic security measures, such as the use of encryption, would already help reducing the risks for children privacy.

Comparison to apps of different nature. In this work we show that parental control applications often misbehave posing privacy threats for children and even parents. However, we cannot say that they are worse in behavior than other Android apps, even children-oriented apps studied in the literature [70, 74]. Previous studies show that security and privacy misbehavior is a common trend in Android applications: they often request more permissions than needed [37], and include a large number of third-party libraries [59, 70]. Regarding the privacy risks of children-oriented applications published in Google Play's Designed for Families program, Irwin *et al.* showed that almost 50% of the apps were in potential violation of the COPPA rule [74].

Much like in the case of children-oriented apps present in Google's DFF program, parental control apps should comply with children-oriented regulation and special provisions. By definition, they are directed at children. However, none of the apps analyzed in this study is listed on Google Play's DFF program. While previous work has shown that Android app's tend to collect plenty of user data [66, 70], in some cases even avoiding the security mechanisms implemented by the platform [62, 72], the fact that parental control apps potentially do not comply with specific regulation is much more worrisome – both for parents and minors.

In fact, there is a big element of trust from parents towards parental control solution developers, which is broken when these services treat children data carelessly, share it with third-parties, or send it across the Internet unencrypted. This misbehavior and parents' inability to audit mobile applications calls for action from researchers and data protection agencies to understand the way in which these apps might be violating current legislation and invoke regulatory actions to fix these issues. Until that moment, some studies proposed the use of non-technical solutions for parental control [58].

Limitations. Our app collection method relies on using the free search capabilities of the Google Play Store, which means that we did not explore every parental control app on the market. Nevertheless, we believe that the set of apps chosen is representative in popularity, developer characteristics, and behavior. Also, as our initial data collection started two years prior to submission, some apps were removed from the marketplace during our analysis and new apps had appeared. We repeated the data collection pipeline later on our study to find newly published apps, and we downloaded historical version of every app that we analyzed to always cover the same time-frame.

Despite combining static and dynamic analysis, we acknowledge that our analysis cannot guarantee full coverage of the code, app features, or the data flows. First, the static analysis will miss code paths implemented in native or obfuscated code. Second, LibRadar uses a database to identify third-party libraries and cannot detect libraries that are not present in this database. Third, Lumen can miss flows that are not triggered by our pre-defined actions. Furthermore, we perform our dynamic analysis on the children app, and it may be that the companion parent app disseminates sensitive data over the network,. Additionally, our Man-in-the-middle proxy is unable to intercept TLS flows for 2 apps that might use techniques like TLS certificate-pinning [69]. Finally, our dynamic analysis has been executed from a testbed geolocated in the European Union. It would be necessary to verify our claims in the U.S. to further investigate potential COPPA violations.

9 Related Work

Researchers resort to static or dynamic analysis –or a combination of both, as in our study– to analyze the behavior of mobile apps. Static analysis techniques have been used to identify possible information leaks in Android [16, 55], to extract behavioral features for malware detection [18, 38], or to study the evolution of the Android ecosystem [25, 26, 72, 82, 87]. Similarly, many dynamic analysis techniques have been used for the same or similar purposes [32, 52, 66, 71, 80], despite the challenges to scale and automate the analysis to large datasets [28, 74]. While we rely on similar static and dynamic techniques for our study, the final goal is quite different, as none of these works focuses on the privacy aspects of parental control apps.

We build on top of previous work on the domain of parental control applications and children privacy. There have been studies on the effectiveness of parental control solutions: Mathiesen argued that such policies are a violation of children's right to privacy [61] while Wisniewski *et al.* presented a comprehensive study of the features offered by parental control apps and concluded that privacy invasive monitoring features are more common than those focusing on teen self regulation [89]. Furthermore, Eastin *et al.* showed that parenting style has an effect on the type of restrictions that parents set on their children's phone usage [31].

We believe this to be the first study taking a look at the privacy implications of parental control apps from a technological point of view. Others have studied privacy implications for children in domains such as social media [60] and smart toys [84, 90]. Reyes *et al.* [74] is the closest work to ours from a methodology standpoint. They analyze mobile apps' compliance with the COPPA regulation. While some of their findings are similar to what we present in this paper, our work is a more thorough analysis of the parental control mobile app ecosystem. Chatterjee *et al.* also studied the parental control ecosystem. However, they focus on assessing how such applications may be used for other purposes (*e.g.,* spyware and partner violence) [27].

10 Conclusion

We have presented the first multi-dimensional study of the parental control apps ecosystem from a privacy perspective. Our findings open a debate about the privacy risks introduced by these apps. Does the potential of parental control apps for protecting children justify the risks regarding the collection and processing of their data?

It is our hope that our study raises awareness in parents and regulatory authorities on the risks brought up by some of these applications. We stress that it is fundamental to complement current benchmarking initiatives [1, 33] with a security and privacy analysis to help parents to choose the best application while taking these aspects into consideration.

Acknowledgment

The authors would like to thank our anonymous reviewers and our shepherd Prof. Micah Sherr (Georgetown

University) for their valuable feedback and their help for preparing the final version of this paper. The authors also thank Julien Gamba for his help in the manual inspection of privacy policies and Juan Caballero for his contribution on the initial version of this work. This project is partially funded by the US National Science Foundation (grant CNS-1564329), the European Union's Horizon 2020 Innovation Action program (grant Agreement No. 786741, SMOOTH Project), the Spanish national projects DEDETIS (TIN2015-70713-R) and SCUM (RTI2018-102043-B-I00), the Spanish grant TIN2017-88749-R (DiscoEdge) and by the Madrid regional project BLOQUES (S2018/TCS-4339).

References

[1] https://www.is4k.es/.

[2] Rodney Alexander. How to protect children from internet predators: a phenomenological study. *ANNUAL REVIEW OF CYBERTHERAPY AND TELEMEDICINE 2015*, page 82, 2016.

[3] Alphabet. Alphabet - Home Page. https://abc.xyz/.

[4] Android. Usage of Android Advertising ID. https://play.google.com/intl/en-GB/about/monetization-ads/ads/ad-id/index.html.

[5] Android. Best practices for unique identifiers. https://developer.android.com/training/articles/user-data-ids.

[6] Android. Permissions overview. https://developer.android.com/guide/topics/permissions/overview.

[7] Android. Play Protect. https://www.android.com/play-protect/.

[8] Android. UI/Application Exerciser Monkey. https://developer.android.com/studio/test/monkey.

[9] Android. VpnService. https://developer.android.com/reference/android/net/VpnService.

[10] Android. Android developer manual: permission model, 2018. https://developer.android.com/guide/topics/permissions/overview.

[11] Android Developers. App Manifest Overview. https://developer.android.com/guide/topics/manifest/manifest-intro.

[12] Android Developers. Privacy changes in Android 10. https://developer.android.com/about/versions/10/privacy/changes.

[13] Android Developers. VPN Service. https://developer.android.com/reference/android/net/VpnService.

[14] APKPure. Homepage. https://apkpure.com/.

[15] Arity. Arity. https://www.arity.com.

[16] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau, and Patrick McDaniel. FlowDroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. In *PLDI 2014*, 2014.

[17] Kathy Wain Yee Au, Yi Fan Zhou, Zhen Huang, and David Lie. Pscout: Analyzing the android permission specification. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12. ACM, 2012.

[18] Vitalii Avdiienko, Konstantin Kuznetsov, Alessandra Gorla, Andreas Zeller, Steven Arzt, Siegfried Rasthofer, and Eric Bodden. Mining apps for abnormal usage of sensitive data. In *ICSE '15*, pages 426–436, 2015.

[19] Michael Backes, Sven Bugiel, Erik Derr, Patrick McDaniel, Damien Octeau, and Sebastian Weisgerber. On demystifying the android application framework: Re-visiting android permission specification analysis. In *USENIX Security 2016*, pages 1101–1118, 2016.

[20] BBC News. Web porn: Just how much is there?, 2013. https://www.bbc.com/news/technology-23030090.

[21] Boomerang. Spin Browser. https://useboomerang.com/spin/.

[22] Branch.io. Homepage. https://branch.io.

[23] Branch.io. Terms of Service. https://branch.io/policies/#terms-and-conditionss.

[24] Braze (formerly AppBoy). Privacy. https://www.braze.com/privacy/.

[25] Paolo Calciati and Alessandra Gorla. How do apps evolve in their permission requests?: A preliminary study. In *MSR '17*. IEEE Press, 2017.

[26] Paolo Calciati, Konstantin Kuznetsov, Xue Bai, and Alessandra Gorla. What did really change with the new release of the app? In *MSR 2018*, 2018.

[27] R. Chatterjee, P. Doerfler, H. Orgad, S. Havron, J. Palmer, D. Freed, K. Levy, N. Dell, D. McCoy, and T. Ristenpart. The spyware used in intimate partner violence. In *2018 IEEE Symposium on Security and Privacy (SP)*, 2018.

[28] Shauvik Roy Choudhary, Alessandra Gorla, and Alessandro Orso. Automated test input generation for android: Are we there yet? *arXiv preprint arXiv:1503.07217*, 2015.

[29] Council of European Union. General Data Protection Regulation 679/2016, 2016. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679&from=EN.

[30] Cyberbullying Research Center. Summary of Our Cyberbullying Research (2004-2016). https://cyberbullying.org/summary-of-our-cyberbullying-research.

[31] Matthew S Eastin, Bradley S Greenberg, and Linda Hofschire. Parenting the internet. *Journal of communication*, 56(3):486–504, 2006.

[32] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *TOCS*, 32(2):5, 2014.

[33] Europen Comission. Benchmarking of parental control tools for the online protection of children , 2017.

https://www.sipbench.eu/index.cfm/secid.1/secid2.3.

[34] Federal Trade Comission. Get Parents' Verifiable Consent Before Collecting Personal Information rom Their Kids. https://www.ftc.gov/tips-advice/business-center/guidance/childrens-online-privacy-protection-rule-six-step-compliance#step4.

[35] Federal Trade Comission. Children's Online Privacy Protection Act, (15 U.S.C. 6501, et seq.,) , 1998. https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule.

[36] Federal Trade Comission. Mobile Advertising Network In-Mobi Settles FTC Charges It Tracked Hundreds of Millions of Consumers' Locations Without Permission, 2016. https://www.ftc.gov/news-events/press-releases/2016/06/mobile-advertising-network-inmobi-settles-ftc-charges-it-tracked.

[37] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 627–638. ACM, 2011.

[38] Yu Feng, Saswat Anand, Isil Dillig, and Alex Aiken. Apposcopy: Semantics-based detection of android malware through static analysis. In *FSE 2014*, pages 576–587, New York, NY, USA, 2014. ACM.

[39] Yanick Fratantonio, Chenxiong Qian, Simon Chung, and Wenke Lee. Cloak and Dagger: From Two Permissions to Complete Control of the UI Feedback Loop. In *S&P*, 2017.

[40] J. Gamba, M. Rashed, A. Razaghpanah, J. Tapiador, and N. Vallina-Rodriguez. An analysis of pre-installed android software. In *S&P*, 2020.

[41] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10. ACM, 2010.

[42] Google. Families. https://play.google.com/about/families/.

[43] Google. Google Play certified ad networks program. https://support.google.com/googleplay/android-developer/answer/9283445.

[44] Google. Providing a safe and secure experience for our users . https://android-developers.googleblog.com/2018/10/providing-safe-and-secure-experience.html.

[45] Google Play. Kid Control Dev profile. https://play.google.com/store/apps/dev?id=6687539553449035845.

[46] Google Play. Yoguesh Dama profile. https://play.google.com/store/apps/dev?id=5586168019301814022.

[47] Google Play Store — FamilySafety Production. GPS Phone Tracker, 2018. https://play.google.com/store/apps/details?id=com.fsp.android.c.

[48] Herald Sun. Police warn photos of kids with geo-tagging being used by paedophiles, 2012. https://www.heraldsun.com.au/technology/news/photograph-uploads-put-kids-at-risk/news-story/9ef00e4105cb1d38d8f5acb77d6c7433.

[49] IAPP. GDPR Article 8. https://iapp.org/resources/article/the-eu-general-data-protection-regulation/#A8.

[50] IAPP. GDPR Recital 38. https://iapp.org/resources/article/the-eu-general-data-protection-regulation/#R38.

[51] Internet Safety 101. Internnet Safety. https://internetsafety101.org/.

[52] Sakshi Jain, Mobin Javed, and Vern Paxson. Towards mining latent client identifiers from network traffic. *Proceedings on Privacy Enhancing Technologies*, 2016(2):100–114, 2016.

[53] Carlos Jensen and Colin Potts. Privacy policies as decision-making tools: an evaluation of online privacy notices. In *SIGCHI*. ACM, 2004.

[54] Kiddoware. Kiddoware homepage. https://kiddoware.com/.

[55] Li Li, Alexandre Bartel, Tegawendé F. Bissyandé, Jacques Klein, Yves Le Traon, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Octeau, and Patrick McDaniel. Iccta: Detecting inter-component privacy leaks in android apps. In *ICSE '15*, pages 280–291, 2015.

[56] Sonia Livingstone, Leslie Haddon, Anke Goerzig, and Kjartan Ólafsson. Risks and safety on the internet: The perspective of european children. full findings. 01 2011.

[57] Sonia Livingstone, Leslie Haddon, Anke Görzig, and Kjartan Ólafsson. Risks and safety for children on the internet: the uk report. *Politics*, 6(1), 2010.

[58] Livingstone, Sonia and Helsper, Ellen. Parental Mediation of Children's Internet Use. *Journal of Broadcasting & Electronic Media - J BROADCAST ELECTRON MEDIA*, 52:581–599, 11 2008.

[59] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. Libradar: Fast and accurate detection of third-party libraries in android apps. In *ICSE 2016*. ACM, 2016.

[60] Mary Madden, Amanda Lenhart, Sandra Cortesi, Urs Gasser, Maeve Duggan, Aaron Smith, and Meredith Beaton. Teens, social media, and privacy. *Pew Research Center*, 21:2–86, 2013.

[61] Kay Mathiesen. The internet, children, and privacy: the case against parental monitoring. *Ethics and Information Technology*, 15(4):263–274, 2013.

[62] Yan Michalevsky, Dan Boneh, and Gabi Nakibly. Gyrophoone: Recognizing speech from gyroscope signals. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 1053–1067, 2014.

[63] Monica Anderson. Parents, Teens and Digital Monitoring. https://stirlab.org/wp-content/uploads/2018/06/2017_Wisniewski_ParentalControl.pdf.

[64] New York Times. Uber hid 2016 breach, paying hackers to delete stolen data, 2017. https://www.nytimes.com/2017/11/21/technology/uber-hack.html.

[65] Ofcom: UK broadband, home phone and mobile services regulator. Children and parents: Media use and attitudes report 2018, 2018. https://www.ofcom.org.uk/__data/assets/pdf_file/0024/134907/Children-and-Parents-Media-Use-and-Attitudes-2018.pdf.

[66] Elleen Pan, Jingjing Ren, Martina Lindorfer, Christo Wilson, and David Choffnes. Panoptispy: Characterizing audio and

video exfiltration from android applications. *Proceedings on Privacy Enhancing Technologies*, 2018.

[67] Harshvardhan J Pandit, Declan O'Sullivan, and Dave Lewis. Queryable provenance metadata for gdpr compliance, 2018.

[68] PCMag. The Best Parental Control Software of 2019. https://uk.pcmag.com/parental-control-monitoring/67305/the-best-parental-control-software.

[69] Abbas Razaghpanah, Arian Akhavan Niaki, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Johanna Amann, and Phillipa Gill. Studying tls usage in android apps. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 350–362. ACM, 2017.

[70] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, Trackers, Privacy and Regulators: A Global Study of the Mobile Tracking Ecosystem. In *Network and Distributed System Security Symposium*, February 2018.

[71] Abbas Razaghpanah, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, and Vern Paxson. Haystack: In situ mobile traffic analysis in user space. *CoRR*, 2015.

[72] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 50 ways to leak your data: An exploration of apps' circumvention of the android permissions system. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pages 603–620, 2019.

[73] Jingjing Ren, Martina Lindorfer, Daniel J Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. Bug fixes, improvements,... and privacy leaks. *NDSS*, 2018.

[74] 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. *Proceedings on Privacy Enhancing Technologies*, 2018(3):63–83, 2018.

[75] SafeWise. Best Parental Control Apps and Software Buyers Guide. https://www.safewise.com/resources/parental-control-filters-buyers-guide/.

[76] Samsung. Knox SDK. https://seap.samsung.com/sdk/knox-android.

[77] Screentime Labs. Screentime homepage. https://screentimelabs.com/.

[78] Benjamin Shmueli and Ayelet Blecher-Prigat. Privacy for children. *Colum. Hum. Rts. L. Rev.*, 42:759, 2010.

[79] 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 *International Conference on Software Engineering*. ACM, 2016.

[80] Sooel Son, Daehyeok Kim, and Vitaly Shmatikov. What mobile ads know about mobile users. In *NDSS*, 2016.

[81] Statista. Mobile Internet, 2018. https://www.statista.com/topics/779/mobile-internet/.

[82] Vincent F. Taylor and Ivan Martinovic. To updae or not to update: Insights from a two-year study of android app evolution. In *ASIA CCS '17*. ACM, 2017.

[83] Güliz Seray Tuncay, Soteris Demetriou, Karan Ganju, and C Gunter. Resolving the predicament of android custom permissions. 2018.

[84] Junia Valente and Alvaro A. Cardenas. Security &#38; privacy in smart toys. In *IoTS&#38;P '17*. ACM, 2017.

[85] Raja Vallée-Rai, Phong Co, Etienne Gagnon, Laurie Hendren, Patrick Lam, and Vijay Sundaresan. Soot – a Java bytecode optimization framework. In *CASCON*. IBM Press, 1999.

[86] Haoyu Wang, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez, Yao Guo, Li Li, Juan Tapiador, Jingcun Cao, and Guoai Xu. Beyond google play: A large-scale comparative study of chinese android app markets. In *IMC '18*. ACM, 2018.

[87] Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu, and Michalis Faloutsos. Permission evolution in the android ecosystem. In *ACSAC '12*. ACM, 2012.

[88] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. The creation and analysis of a website privacy policy corpus. In *Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[89] Pamela Wisniewski, Arup Kumar Ghosh, Heng Xu, Mary Beth Rosson, and John M Carroll. Parental control vs. teen self-regulation: Is there a middle ground for mobile online safety? In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, pages 51–69. ACM, 2017.

[90] Benjamin Yankson, Farkhund Iqbal, and Patrick C. K. Hung. *Privacy Preservation Framework for Smart Connected Toys*. Springer International Publishing, 2017.

[91] Michele L Ybarra, Kimberly J Mitchell, and Josephine D Korchmaros. National trends in exposure to and experiences of violence on the internet among children. *Pediatrics*, 2011.

[92] Nan Zhong and Florian Michahelles. Google play is not a long tail market: An empirical analysis of app adoption on the google play app market. In *SAC*. ACM, 2013.

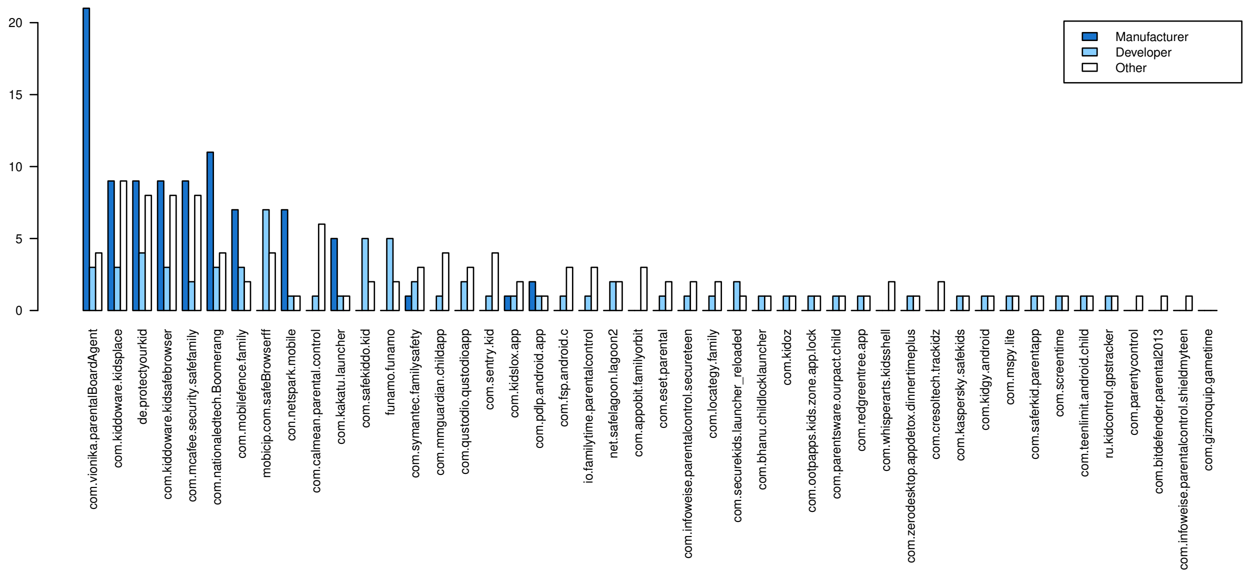

Fig. 5. Number and type of custom permissions found in the apps in our dataset

3rd-party Library ToS

Below, we list the Terms of Service for the third-party libraries prohibiting their usage on children-oriented software:

Branch.io [23]:

You will not use our Services to: {...} (ix) create lists or segments of children under the age of 13 (and in certain jurisdictions under the age of 16), advertise mobile Apps that are directed to children under 13 (and in certain jurisdictions under 16), and/or knowingly market products or services to children under 13 (and in certain jurisdictions under the age of 16), without employing appropriate settings within the Branch SDKs to limit data collection for children under 13 (and in certain jurisdictions under 16), in order to comply with any applicable laws protecting children (including, but not limited to, GDPR and the U.S. Children's Online Privacy Protection Act ("COPPA");

Appboy [24]: (now branded as *Braze*)

Our Services are not directed to individuals under the age of 13. We do not knowingly collect personal information from such individuals without parental consent and require our Customers to fully comply with applicable law in the data collected from children under the age of 13.

Custom Permissions

Figure 5 dives deeper on the number and type of custom permissions for every app in our dataset.

Permission Heatmap

Figure 6 shows a landscape version of Figure 2 to improve readability.

List of Analyzed Apps

Table 8 provides a classification of each app attending to whether it is a monitoring or restriction app, if it is currently available on Google Play and if it is benchmarked by the SIP and IS4K alternatives.

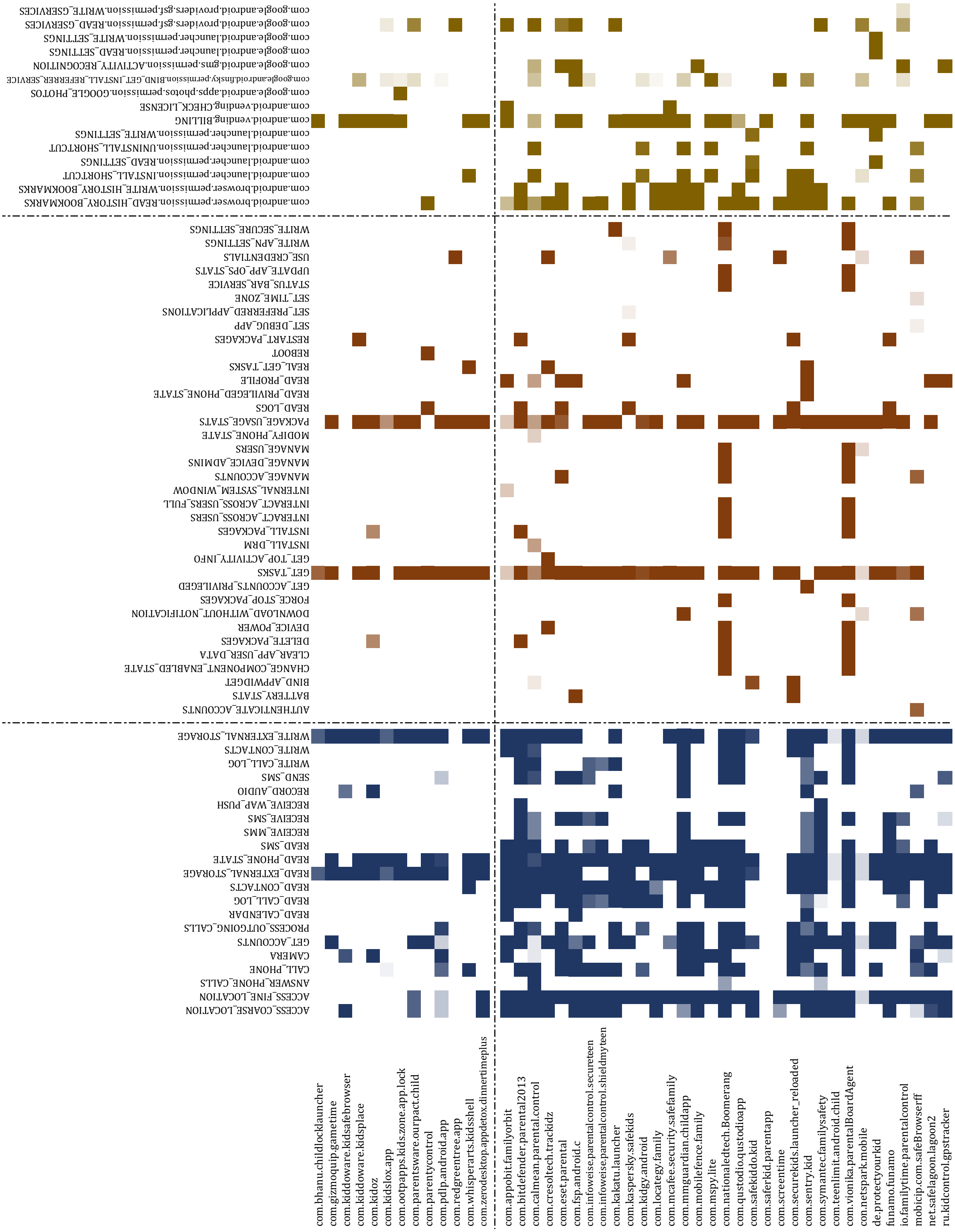

Fig. 6. Landscape version of the permission heatmap shown in Figure 2

Table 8. Summary of our corpus of apps attending to different features. The column "benchmarked" indicates whether the app has been analyzed by [1, 33]. The values "M" and "R" stand for Monitoring and Restriction, respectively

Name	Type	Currently listed (2019/05)	Benchmarked
com.mmguardian.childapp	M	✓	
com.infoweise.parentalcontrol.secureteen.child	M	✓	
com.qustodio.qustodioapp	M	✓	✓
com.kaspersky.safekids	M	✓	
com.kiddoware.kidsafebrowser	R	✓	
com.securekids.launcher_reloaded	M	✓	✓
com.eset.parental	M	✓	✓
com.symantec.familysafety	M	✓	
con.netspark.mobile	M	✓	
com.nationaledtech.Boomerang	M	✓	
com.mobilefence.family	M	✓	
com.infoweise.parentalcontrol.shieldmyteen	M		
com.teenlimit.android.child	M	✓	
com.bhanu.childlocklauncher	R	✓	
com.safekiddo.kid	M	✓	
com.kidoz	R	✓	
com.cresoltech.trackidz	M		
net.safelagoon.lagoon2	M	✓	
com.screentime	M	✓	✓
com.kiddoware.kidsplace	R	✓	
com.mcafee.security.safefamily	M	✓	
io.familytime.parentalcontrol	M	✓	✓
com.vionika.parentalBoardAgent	M		
de.protectyourkid	M	✓	
com.parentsware.ourpact.child	R	✓	
ru.kidcontrol.gpstracker	M	✓	
com.kidslox.app	R	✓	
com.zerodesktop.appdetox.dinnertimeplus	R	✓	
com.pdlp.android.app	R	✓	✓
com.redgreentree.app	R		
com.sentry.kid	M	✓	
com.whisperarts.kidsshell	R	✓	
funamo.funamo	M	✓	
com.bitdefender.parental2013	M		
com.gizmoquip.gametime	R		
com.kakatu.launcher	M		
com.calmean.parental.control	M	✓	
mobicip.com.safeBrowserff	M	✓	✓
com.saferkid.parentapp	M	✓	
com.fsp.android.c	M	✓	
com.locategy.family	M	✓	
com.parentycontrol	R		
com.leapteen.parent	M		
com.kidgy.android	M		
com.mspy.lite	M	✓	
com.appobit.familyorbit	M		