

# Not Seen, Not Heard in the Digital World!

## Measuring Privacy Practices in Children’s Apps

Ruoxi Sun<sup>\*†</sup>, Minhui Xue<sup>†‡</sup>, Gareth Tyson<sup>§</sup>, Shuo Wang<sup>†‡</sup>, Seyit Camtepe<sup>†‡</sup>, Surya Nepal<sup>†‡</sup>

<sup>\*</sup>The University of Adelaide, Australia

<sup>†</sup>CSIRO’s Data61, Australia

<sup>‡</sup>Cybersecurity CRC, Australia

<sup>§</sup>Hong Kong University of Science and Technology (GZ), China

### ABSTRACT

The digital age has brought a world of opportunity to children. Connectivity can be a game-changer for some of the world’s most marginalized children. However, while legislatures around the world have enacted regulations to protect children’s online privacy, and app stores have instituted various protections, privacy in mobile apps remains a growing concern for parents and wider society. In this paper, we explore the potential privacy issues and threats that exist in these apps. We investigate 20,195 mobile apps from the Google Play store that are designed particularly for children (Family apps) or include children in their target user groups (Normal apps). Using both static and dynamic analysis, we find that 4.47% of Family apps request location permissions, even though collecting location information from children is forbidden by the Play store, and 81.25% of Family apps use trackers (which are not allowed in children’s apps). Even major developers with 40+ kids apps on the Play store use ad trackers. Furthermore, we find that most permission request notifications are not well designed for children, and 19.25% apps have inconsistent content age ratings across the different protection authorities. Our findings suggest that, despite significant attention to children’s privacy, a large gap between regulatory provisions, app store policies, and actual development practices exist. Our research sheds light for government policymakers, app stores, and developers.

### 1 INTRODUCTION

The last decades have seen a dramatic increase in our reliance on mobile services. Particularly due to COVID-19, more and more young people are spending time using mobile applications for entertainment, remote work, online learning, and day-to-day tasks [61]. Unfortunately, the Internet is home to a vast amount of content, which is potentially harmful to children, including uncensored sexual imagery, violent content, and strong language [9, 27]. 46% of parents say their children, aged 11 or younger, who uses YouTube have encountered videos that were inappropriate for their age [8]. Furthermore, children may expose sensitive data online, with more than 1/3 of young people in 30 countries reporting being a victim of online bullying [26].

Due to these concerns, many international jurisdictions have enacted privacy laws and regulations to promote and protect the privacy of children. These include the Children’s Online Privacy Protection Act (COPPA) [19] and its implementation, the Children’s Online Privacy Protection Rule (COPPR) [6], which “imposes certain requirements on operators of websites or online services directed to children under 13 years of age”, the European General Data Protection Regulation (GDPR) [23], and the Privacy Act 1988 [1]

in Australia. These regulations restrict the behavior of apps and define the obligations of app developers. For example, GDPR Art. 8, COPPA, and COPPR §312 requires that applications obtain verifiable parental consent prior to collecting personal information from children. Apps must make reasonable efforts to ensure that the notification is received by a parent of the child. Although various privacy assertions are required in app stores (such as the permission list and the privacy policies), it is usually difficult for regular users to understand the potential threats an app may pose, let alone identify undesired or malicious application behaviors.

To help parents determine age-appropriate mobile apps for their children, app stores have released strict developer policies, along with inspection and vetting procedures before app publishing. Critical app information is provided to help users understand the app before using it (such as the number of installs, requested permissions, a rating score, the name of developer, and the comments by other users). Furthermore, every app that is sold through the Google Play store is rated for age-appropriateness. These content age rating systems recognize that mobile apps now run the entire gamut from interactive picture books for toddlers, through to graphic adult content. Hence, parents can use ratings to help with making app purchasing decisions. In 2015, Google Play launched the “Designed for Families” program [49], which allows app publishers to opt into an additional review in order to have their apps labeled as being family-friendly (aiming at highlighting pre-approved, child-safe apps). Further, in 2020, Google Play added a “Teacher Approved” section, in which the Play store consults with teachers and specialists. They rate these apps based on design, appeal, age appropriateness and the appropriateness of ads [12]. However, considering that most information is provided by app developers’ self-reporting (e.g., by filling a form or answering questionnaires), a centralized age appropriateness rating system is still missing.

We argue that the complexity of these different policies and systems challenge the ability of many parents (and children) to make informed decisions. To gain an understanding of this complexity, we provide the first comprehensive measurement study of privacy practices in children’s mobile apps. We inspect apps from both a technical and user-available information perspective, with the aim to expose improper children’s app development practices and privacy threats. We particularly focus on apps that are (i) designed primarily for children under 13 and listed on the Children tab (Family apps, *i.e.*, can be found at <https://play.google.com/store/apps/category/FAMILY>); and (ii) designed for everyone, *including* children (Normal apps). From the technical side, we conduct static and dynamic analysis of 20,195 apps, including 3,627 Family apps and 16,568 Normal apps. From the users’ perspective, we collect and

analyze content age ratings from 5 different children’s app rating authorities and 13,132,577 comments from 11,831 apps. Our main findings are as follows:

- Through static and dynamic analysis, we find that 4.47% of Family apps request location permissions (which is forbidden by the Play store); 81.25% of Family apps use trackers that are not allowed to be used in children’s apps. Even major players (having 40+ children’s apps published in the Play store) do not follow the Play store policies.
- We compare the content age ratings given by various agencies (these tag the suitability of apps for different age groups). We identify significant inconsistency among these different agencies. 19.25% apps have such issues, with 13.25% of Family apps and 8.99% of Normal apps having severe inconsistencies across authorities. We conclude that greater transparency should be introduced to help inform parents.
- From users’ comments, we find that Family app users complain more about app content, but less about privacy and security. Highly-rated apps have a larger number of complaints. We conclude that a more efficient mechanism to report and check inappropriate app content should be established.

Our findings suggest that despite significant attention paid to children’s privacy by legislatures, app stores, and society, there is still a large gap between regulatory provisions, app store policies, and actual development practices. We argue that app stores should establish more effective supervision mechanisms, provide more detailed information, reduce their reliance on developers’ self-certified information, and respond more actively to user feedback. We believe our study can provide useful insights for government policymakers, app stores, developers, and researchers to build privacy-preserving apps for children. Our source code is publicly available at <https://github.com/children-privacy/children-privacy>. **Ethical considerations.** All the apps and user comments in this research are collected from publicly available resources. The app and developer names mentioned in this paper are anonymized. All personal information (usernames, timestamps) is removed from the dataset. We have disclosed the findings to Google Play store and related entities.

## 2 LEGAL BACKGROUND AND PLAY STORE POLICIES

This section describes the legal background for online children’s apps that are subject to regulations or policies, such as COPPA, GDPR, and the Google Play Store policies. We further briefly outline our legal analysis of potential violations in Android children’s apps. **Consent from parents.** Children merit specific protection with regard to their personal data, as they may be less aware of the risks related to the processing of personal data [24]. According to COPPA §312.4 [6], apps with target users under 13 must make reasonable efforts, taking into account available technology, to ensure that a parent of a child receives *direct notice* of the apps’ practices with regard to the collection, use, or disclosure of personal information from children. As regulated in GDPR Art. 8, the processing of personal information is lawful only if the consent is given by the holder of parental responsibility over the child, meaning that users who

are 15 years or younger need parental consent where applicable (member states can choose a younger age down to 13).

Therefore, companies must obtain verifiable parental consent before gathering data from children below the age limit (13 years of age for COPPA, 16 for GDPR). This legal requirement implies that informing parents or legal tutors about data collection practices via the privacy policy is not sufficient, especially if the app disseminates sensitive data to third-party services. With this in mind, we wish to determine whether apps collect private data without user consent. As a result, any sensitive or personal data, particularly unique identifiers or geolocations, uploaded by the app to third parties without a user consent may be a violation of COPPA and GDPR. Note, in the Play store policy, apps that solely target children are not allowed to access location permissions.

**Families policy requirements.** To better serve users, the Play store requires developers to provide accurate information about their apps. In addition to filling out the age rating questionnaire, developers also must provide details about their app’s target audience and content. Depending on the target audience selections the developers make, the app will be subject to additional Google Play policies, to ensure that apps for children have appropriate content, show suitable ads, and handle personal and sensitive information correctly. For apps that are designed primarily for children under 13, they must participate in the Designed for Families [49] and comply with Google Play’s Families Policy Requirements [50]. For any apps that have at least one target audience age group that includes children, developers must comply with Google Play’s Families Policy Requirements. In short, for any apps that have target users that include children, Google Play’s Families Policy Requirements are compulsory. Developers are responsible for ensuring their apps are appropriate for children and compliant with all relevant laws. Failure to satisfy the requirements may result in an app’s removal or suspension.

According to the Designing Apps for Children and Families policy by the Play Store [49], apps designed specifically for children must participate in the Designed for Families program. This requires that apps can only use Google Play certified ad SDKs listed in Table 1. In order for an ad SDK to be included on the list, the ad SDK must self-certify that they are compliant with Play’s Families Ad Program policies and all applicable local laws and regulations. Apps that include children and adults in their target audience, but are not in the Designed for Families program, are allowed to use non-certified ad SDKs for serving ads only to users above the age of 13. Therefore, any non-certified ad SDK used in Family apps violates the Play Store policy. Normal apps that have non-certified ad SDKs but do not implement a neutral age screen may also violate the policy.

**Age ratings.** Age ratings (also known as maturity or content age ratings) rate the suitability of TV broadcasts, movies, comic books, or video games for its audience [22, 34]. This usually places a media source into one of a number of categories, to show which age group is suitable to view the media. In the Google Play store, the app developers are responsible for completing a rating questionnaire about the nature of the apps’ content. The ratings assigned to the app, displayed on Google Play, are determined by the questionnaire responses. Misrepresentation of an app’s content may result in removal or suspension [52]

**Table 1: Ad SDKs that participate in Play’s Families Ads Program**

Ad SDKs	Code Signature	Network Signature
AdColony	com/adcolony/ com/jirbo/adcolony/	adcolony.com
AppLovin	com/applovin	applovin.com, applvn.com
Chartboost	com/chartboost/sdk/	chartboost.com
Google AdMob	com/google/ads/ com/google/android/gms/ads/ com/google/android/ads/ com/google/unity/ads/ com/google/android/gms/admob	2mdn.net, google.com, dmtry.com, doubleclick.com, doubleclick.net, mng-ads.com
InMobi	com/inmobi, in/inmobi/	inmobi.com, inmobicdn.net, inmobi.cn
ironSource	com/ironsource/	ironsrc.co
Kidoz	com/kidoz/sdk	kidoz.net/kidoz-sdk
SuperAwesome	tv/superawesome/sdk, tv/superawesome/lib/	superawesome.com
Unity Ads	com/unity3d/services, com/unity3d/ads	unity3d.com
Vungle	com/vungle/publisher/ com/vungle/warren/	vungle.com

The ratings are intended to help consumers (especially parents) identify potentially objectionable content that exists within an app. Considering that, in different territories, rating standards can have differences and each rating authority uses its own methodology, an app can earn different ratings. In Figure 1, we list a few rating authorities, including the Entertainment Software Rating Board (ESRB) [22] in Americas, the Pan European Game Information (PEGI) [34] in Europe and Middle East, Unterhaltungssoftware Selbstkontrolle (USK) [60] in Germany, the Australian Classification Board (ACB) [20] in Australia, and the International Age Rating Coalition (IARC) [18]. For some apps, their age ratings across different territories are inconsistent and confusing. For example, the app, sg\*\*\*ve, earns a content age rating of “PEGI 12” in Australia, while in Germany, it is rated as “USK 16+”. Note that, in the requirements of “PEGI 12”, an app could contain “slight violence towards fantasy characters”, “non-graphic violence towards human-looking characters”, and “mild bad language and no sexual expletives”. However, an “USK 16+” app is allowed to contain “realistic violence”, “shock and horror elements”, “consistently explicit language”, and “erotic or sexual focus”, which obviously cannot fall into the category “PEGI 12”, even the gap between suitable age ranges of the two ratings is only 1 year (“PEGI 12” is suitable for age group 12 to 15 and “USK 16+” is suitable for age group 16 to 17).

### 3 ANALYSIS PIPELINE

We present an overview of our analysis pipeline in Figure 2. We combine the advantages of both static and dynamic analysis, as well as performing content (age) rating and user comment analysis. Our goal is to triage suspicious apps and analyze their behaviors in depth. Specifically, we evaluate apps (with targeted users including children) to find evidence of violations against privacy regulations

Rating Authorities	Territories	Suitable Age Groups																			
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
ACB	Australia	G														15+		R18+			
		PG																	M15+		
ESRB	Americas	E								E10+		Teen			M	Adults					
PEGI	Europe & Middle East	N/A	3				7				12				16		18				
			PG																		
USK	Germany	All						6+				12+				16+		18+			
IARC	Genric	N/A	3+				7+				12+				16+		18+				

**Figure 1: Age groups of content age ratings from different rating authorities.**

and app store policies. Our pipeline covers 20,195 different Android apps. We check for improper use of trackers in children’s apps, privacy leakage without consent, inconsistent age ratings from different authorities, and complaints by users. All of the apps are downloaded from the Google Play Store using a Google Play scraper.

Our pipeline performs several tasks. We first decompile the APK file of the Android app under-study, and conduct a static analysis to determine what permissions have been requested and what trackers are used in the app. Next, we execute each app version individually on a physical mobile phone with a network monitor, which allows us to observe apps’ run-time behaviors. We further collect the age ratings using different country settings and identify improper ratings according to inconsistent levels across rating agencies. Finally, we analyze users’ comments with both NLP-based topic modeling and rule-based detection techniques to identify undesired app behaviors from users’ complaints.

#### 3.1 Data Collection

We collect apps and their age ratings, as well as the user comments from Google Play store. Here we describe how we collect the data. **App collection.** We wrote a Google Play Store scraper to download the most popular children’s apps (English version) under each category. We collect 3,627 apps that participate in the Design for Families program. We refer to this group as “Family apps” as their target users are purely children and families (categorized as “Children” in the Play Store). We also collect 16,568 apps that are not particularly designed for children, but include children as target users. We refer to these as “Normal apps” (with age ratings of “Everyone”, “Everyone 10+”, and “Teen”). Note, we download all apps from the U.S. Google Play Store. Because the popularity distribution of apps is long tailed, our analysis of the 20,195 most popular apps is likely to cover most of the apps that people currently use (from Mar 2021 to Dec 2021). We obtain executable APK files and corresponding metadata (e.g., category and age rating) from Google Play store.

**Age ratings collection.** We collect the age ratings of each app in the dataset from different countries by changing the country setting in the URL of each app when visiting the Play Store. We gather data for Australia, France, Germany, UK, and USA. For example, the URL [https://play.google.com/store/apps/details?id=\[package\\_name\]&hl=en&gl=US](https://play.google.com/store/apps/details?id=[package_name]&hl=en&gl=US) will lead us to the Play Store in the USA and if we change `gl=US` to `gl=FR`, it will present the app information to users from France. The reason we select these five countries is that they use different age rating standards that cover the main choices around the world, including ACB (Australia - games only),

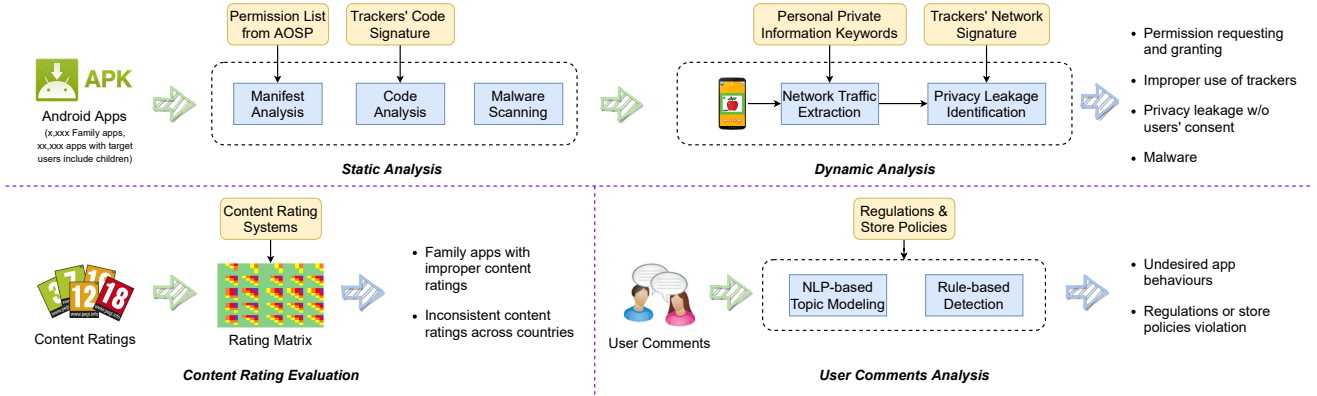


Figure 2: An overview of the analysis pipeline

ESRB (North & South America), PEGI (Europe & Middle East), USK (Germany), and IARC (other countries).

For a fair comparison, we only keep the age ratings of an app if it is available in all five countries. However, as we download all the apps from the U.S. Play Store, there is no Normal App rated as “Mature” and “Adults Only” in ESRB in our dataset. Finally, we collect age ratings for 9,453 apps.

**User comments collection.** We collect the 3,000 most recent comments for each app. Note, as we aim to find undesired app behaviors and policy violations from user comments, we only collect the comments with rating stars below 2. This focuses our analysis on negative feedback. We further remove comments with fewer than 5 words. In total, we collect 13,132,577 comments for 11,831 apps.

### 3.2 Static Analysis

Our static analysis focuses on three parts: Manifest Analysis, Code Analysis, and manual inspection. We use these to measure the use of dangerous and signature permissions, as well as trackers in children’s apps. To perform static analysis on the Android Package (APK) binary files, we first decompile the APK of each app to its corresponding *class* and *xml* files using AndroGuard [2]. The de-compiled *AndroidManifest.xml* file is parsed to extract the permissions requested by the app. We then compare the Family apps and Normal apps with respect to the requisition of location related permissions (*i.e.*, *ACCESS\_COARSE\_LOCATION* and *ACCESS\_FINE\_LOCATION*).

In the tracker analysis, we extract all the class names in the de-compiled source code and match them with trackers’ code signatures. We adopt a list of trackers from Exodus Privacy with 693 known trackers [5] and extend the list with 32 more trackers reported in other online resources. We then search for class names that contain any code signatures from the tracker list. For example, an app that contains *com/adcolony* in its source code will be detected as containing tracker “AdColony”, which matches its signature *com/adcolony/*, *com/jirbo/adcolony/*.

Note, our static approach may report false positives. It cannot determine whether the trackers are active or not during run-time. However, we argue that such static analysis still reflects potential tracking of users, especially when an app contains a large number of trackers. We further compare the use of trackers in Family and

Normal apps. Finally, we manually review the Normal apps with most trackers to check whether there are age screens implemented.

### 3.3 Dynamic Analysis

Our static analysis focuses on detecting embedded third-party SDKs that potentially collect and disseminate personal children data to the Internet. We complement this with dynamic analysis, to collect evidence of personal data dissemination.

**App execution.** To analyze whether personally sensitive information is leaked to third-parties without user consent, we rely on the automatic method previously proposed by Feal *et al.* [25]. We launch each app and run it for 5 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.

For this, we implement the dynamic analysis described in Figure 2, which consists of 4 Xiaomi Notebook and 9 Android phones running a rooted Android 10. This allows us to monitor the network traffic of each of the 20,195 Android apps. We automatically run each app using the Android Automator Monkey [4] without human intervention. Monkey is a UI fuzzer which simulates user input events, such as clicks, touches, or gestures, into the app. Concretely, we start each app with the Android Debug Bridge (ADB) command, *adb shell monkey -p [package name] n*, where the “-p” parameter specifies the package to run, and *n* indicates the number of events. Here, we set *n* as 1. After 5 minutes, we force stop and uninstall the app. The logs are then cleared and the device is ready to be used for the next test. We store the resulting network traffic in a database for offline analysis, which we discuss later.

**Network monitoring.** We monitor all network traffic, including TLS-secured flows, using a network monitoring tool, Lumen Privacy Monitor [56]. This has shown to be effective in several prior research activities [25, 28, 30, 57]. The network monitoring module leverages Android’s VPN API to redirect all the device’s network traffic through a localhost service that inspects the traffic, regardless of the protocol used. The network streams are reconstructed and linked back to the original app through mapping the UID obtained from *proc filesystem* to the socket owned by app. In addition, TLS interception is also enabled through the system trusted root certificate installed into the system, which allows Lumen to decrypt TLS

network traffic [56]. Note that, due to the limitation of dynamic testing (*i.e.*, we may miss catching network traffics), our findings can only provide a lower bound of traffic monitoring.

**Personal information in network flows.** We define “personal information” as any piece of data that relates to an identified or identifiable individual or data subject and could distinguish them from another. It is well known that third-party advertisement networks, online service providers, and mobile app developers track users’ personal information across kinds of interfaces, such as devices, websites, and mobile apps, to better target ads or potential customers. For this reason, we look into apps’ access to the persistent identifiers that enable long-term tracking, such as the geolocation information.

We focus on detecting apps that access specific types of sensitive data without user consent. Notably, the unauthorized collection of geolocation information in Android has been the subject of prior regulatory action [50]. In Table 2, we list the summary of personal identifiable information that is checked in the network traffic data, as well as examples of corresponding keywords/values we search for in the traffic. Note, some of the PII (*e.g.*, device model and brand) may be needed for proper app functionality and is generally not considered sensitive within the industry. Therefore, we consider them as low risk.

### 3.4 Content Age Rating Evaluation

**Measuring rating inconsistency.** To investigate inconsistent age ratings, we measure the inconsistency with levels from 0 to 4, depending on the distance of suitable age groups between each category. Specifically, we consider the inconsistency between two ratings from authorities  $A$  and  $B$  in the following manner. (i) We define the suitable age group for rating  $A$  within range  $[a_1, a_2]$ , where  $a_1$  is the minimal allowed age of rating  $A$  and  $a_2$  is the minimal allowed age of the next rating level minus 1. For example, the suitable age group for “USK 6+” is  $[6, 11]$ , as the next rating level is “USK 12+” for which the minimal allowed age is 12. (ii) We determine the inconsistency level between two ratings  $A$  and  $B$ , according to the gap between two age groups, *i.e.*,  $b_1 - a_2$ , as shown in Equation 1.

$$\text{Inconsistency Level} = \begin{cases} 0, & \text{if } b_1 - a_2 \leq 0, \\ \lfloor \frac{(b_1 - a_2)}{T} \rfloor + 1, & \text{if } 9 \geq b_1 - a_2 > 0, \\ 4, & \text{otherwise.} \end{cases} \quad (1)$$

where  $T$  is the threshold between two inconsistency levels (we increase the level by 1 for each  $T$  years gap). We set  $T$  as 3 as for most content age ratings the interval between neighboring levels is around 3. For example, if the two age groups overlap (*i.e.*,  $b_1 - a_2 < 0$ ), we determine the inconsistency level as 0; while for a 9 years gap, we determine the inconsistent level as 4. To summarize the above, we present a matrix in Figure 3, showing the level assigned to each combination of ratings. Using this, we later compare the age ratings of each app. Further, for any that has inconsistency levels larger than 3, we manually review them (including their description and Android packages provided across different territories).

**Table 2: Summary of personal identifiable information checked in network transmission data**

PII	Description	Example words	Key-	Risk
Device Model	Identifies device model and manufacturer.	Redmi Note9 Pro		Low
Brand	Identifies phone brand. When combined with other information, it can be used to identify a user uniquely.	xiaomi		Low
Board Info	Identifies hardware and the phone model.	miatoll		Low
Build number	Identifies uniquely the Android OS and the version.	QQ3A.200905.001		Low
MAC Address	Identifies uniquely the WiFi AP users are connecting to, leaking users’ activities and location.			Mid
Private IP	By leaking the private IP address of your device, an ad network, tracker or application developer can better identify unique users.	129.127.146.***		Mid
Device Fingerprint	A fingerprint of user device which can be used by analytics and ad services to track user.	google/ walleye/ walleye:8.1.0 OPM1.171019.011/ 4448085:user/ release-keys		High
Location	The location data of a user.	"\$country", "\$city"		High
Timezone	Identifies the current timezone.	America/New_York		High
IMEI	The IMEI (International Mobile Station Equipment Identity) identifies the device uniquely, which could be used to track user’s traffic and online behavior.	866400053132507		High
Serial number	Allows ad networks and online trackers to identify a user uniquely for tracking, surveillance or advertising purposes.	3a9eb795		High
Advertising ID	A unique string of characters that identifies the user’s device, for purposes like measuring app usage and ad personalization.	7cba4b19-3ec3-4c14-9ec8-10ca1ad1abe1		High

### 3.5 User Comments Analysis

Beyond the static and dynamic analysis of apps, the comments left by users may also highlight the violation of regulations or policies, including user concerns. Recent research [33] relies on user comments to identify violations against mobile application market policies, using a semi-automated rule-based process. Figure 4 presents an overview of our comment analysis pipeline. We adopt natural language processing (NLP) techniques to interpret the comments and train machine learning models to categorize and identify informative comments. This comment analysis complements the static and dynamic analysis results (since some application behaviors may not be identified by technical analysis, but could be better identified by users). The rules will be further used in the recognition of regulation violations or undesired app behaviors.

**Pre-processing and embedding.** We filter out comments with fewer than 5 words. We further translate emojis into words using the `emoji` Python package [3]. We utilize the pre-trained RoBERTa [42]



		Rating from Authority B																	
		ACB				ESRB				PEGI				USK				IARC	
		G	PG	M	R18+	E 10+	Teen	M 17+	Adults	3	7	12	16	18	All	6+	12+	16+	18+
Rating from Authority A	ACB	G	0	0	1	1	2	0	0	0	0	0	1	2	0	0	0	1	2
		PG	0	0	1	1	2	0	0	0	0	0	1	2	0	0	0	1	2
		M	0	0	0	0	1	1	0	0	0	0	0	1	1	0	0	0	1
		M15+	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1
		R18+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ESRB	E	0	0	3	3	4	0	1	2	3	4	0	0	0	2	3	4	0
		E 10+	0	0	2	2	3	0	0	1	2	3	0	0	0	2	3	0	0
		Teen	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0	0
		M 17+	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0
		Adults	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0
	PEGI	3	0	0	4	4	4	0	2	3	4	4	0	0	0	3	4	4	0
		PG	0	0	0	0	2	0	0	0	1	2	0	0	0	0	1	2	0
		7	0	0	2	2	3	0	0	1	3	3	0	0	0	1	2	3	0
		12	0	0	0	1	2	0	0	0	1	2	0	0	0	0	1	2	0
		16	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
	USK	All	0	0	4	4	4	0	2	3	4	4	0	0	0	3	4	4	0
		6+	0	0	2	2	3	0	0	0	1	2	0	0	0	0	1	2	0
		12+	0	0	0	1	2	0	0	0	1	2	0	0	0	0	1	2	0
		16+	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1	0
		18+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	IARC	3+	0	0	4	4	4	0	2	3	4	4	0	0	0	3	4	4	0
		7+	0	0	2	2	3	0	0	0	1	2	0	0	0	0	1	2	0
		12+	0	0	0	1	2	0	0	0	0	1	2	0	0	0	0	1	2
		16+	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	1	0
		18+	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3: Inconsistency level matrix for age rating evaluation

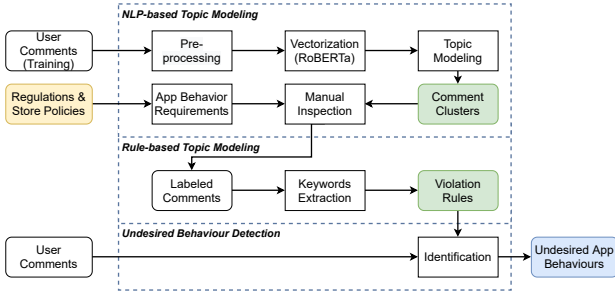


Figure 4: Summary of user comment analysis pipeline

embedding to vectorize each comment into a 300-dimensional vector. RoBERTa (Robustly optimized BERT approach) is a retraining of BERT [21] with an improved training methodology and 10x more training data to improve the BERT performance. It achieves state-of-the-art results (2% to 20% improvement over BERT) [63].

**Comment clustering.** The next step of user comment analysis is to cluster the comments to tease out the different concerns that users describe. These include complaints about functionality, performance, advertisement, personal data collection, vulgar content, violence, and payment deception. We refer to these as *undesired behavior topics*. As the Play Store user comments do not have fine-grained labels for these topics, we use unsupervised learning to cluster comments into types.

We rely on  $k$ -means clustering using Cosine distance to identify the user concerns. We use  $k$ -means clustering as our experiments expose good results (we leave exploration of other clustering solutions to future work). Without knowing how many distinct topics users may write about, the challenge of applying  $k$ -means in our clustering task is how to determine a proper  $k$  value (*i.e.*, the number of target clusters). Recent work by Nema *et al.* [45] proposes a Summarization Metric as shown in Equation 2, aiming for a  $k$  that results in well separated clusters and a high number of compact clusters.

$$\text{Summarization Metric} = \text{dist}_k * M_k, \quad (2)$$

where  $\text{dist}_k$  is the minimum of the cosine distance between all pairs of  $k$  cluster centers, and  $M_k$  is the number of compact clusters. A compact cluster is a cluster in which at least 30% of the samples have silhouette scores higher than the average silhouette score in this cluster. We iterate through  $k = 5, 10, 15, \dots, 100$  and chose  $k$  for which the summarization score is highest, as we want to increase both  $\text{dist}_k$  (distance between cluster centers) and  $M_k$  (number of compact clusters). A larger  $k$  may include more undesired behavior topics in the clustering results, but could also lead to a too heavy workload in the later manual inspection. Finally, we select  $k$  as 40.

To summarize the topic of each cluster, we further conduct a manual inspection on the clustering results. Specifically, we select the top 20 (per cluster) representative comments that are nearest to the corresponding cluster centers and can be considered as representative of the entire cluster. We rank a comment's representativity using their silhouette scores. We carefully analyze these representatives manually across all clusters and determine the topic of each cluster. Considering that not all user comments focus on expression of concerns or complaints, we only keep the 18 clusters with topics related to undesired app behaviors, and drop the remaining 22 clusters.

**Semantic rules extraction.** If we directly use  $k$ -means to recognize users' concerns, it may lead to a high false positive rate. This is because we can only classify each comment based on its distances to the cluster centers, and we have to assign a label even if it may not be relevant. Therefore, based on the clustering results, we propose a rule-based approach that assigns topics to comments according to rules, *i.e.*, if a comment is matched to a semantic rule, the comment will be recognized as belonging to the corresponding topic. Specifically, for each cluster, we first remove stopwords (according to the NLTK English stopwords list [11]) and sort the remaining words in descending order based on TF-IDF weighting [55] to generate a word list. We then manually select keywords for each word list and merge the word lists that usually appear in the same topics. We finally obtain 15 topics in 4 categories as shown in Table 3.

For each topic, we obtain a keyword set containing one or more representative keywords. Finally, we generate semantic rules in a form of  $\{w_1, w_2, d, t\}$  for each keyword set, where  $w_1$  and  $w_2$  are two keywords,  $d$  is the distance constrain between  $w_1$  and  $w_2$ ,  $t$  is the topic that a matched comment will be assigned to. Concretely, we first manually select 50 representative comments from each topic as labeled samples. Then we traverse each keyword set by calculating F1-scores for each single keyword (set  $w_2$  as null and  $d$  as 0) and each pair of keywords under different distance constraints (from 1 to 20), and select the rules with F1-score larger than 0.8. For example, a semantic rule could be  $\{kid, improper, 2, not\_proper\_for\_kids\}$ , which detects any comments that contains the keywords *kid* and *improper*, and the distance between the two keywords are smaller than 2 words. Manual inspections are further conducted on a pilot set of 5,000 labeled comments and rules with an error rate larger than 10% are removed. Finally, we obtained 19 rules for user comment categorization.

We note that, as the semantic rules are extracted from representative comments in clusters, this approach may still focus more on

**Table 3: List of user comment topics.**

Categories	Topic Descriptions
<b>Content</b>	<ul style="list-style-type: none"> <li>Containing violence, blood or scaring contents that are not proper for kids.</li> <li>Containing sexual content that is not allowed for kids.</li> <li>The app content encourages the use of tobacco or drugs.</li> <li>Exposing inappropriate language to children.</li> <li>Presenting depicting criminal activities to kids.</li> </ul>
<b>Ads</b>	<ul style="list-style-type: none"> <li>Providing too many advertisements.</li> <li>The users are disrupted by ads.</li> <li>Ads shortcuts in launching menu or notification bar.</li> <li>Redirection or drive-by download by ads.</li> </ul>
<b>Privacy</b>	<ul style="list-style-type: none"> <li>Leaking or stealing users’ private information.</li> <li>Abusing the permissions (e.g., requesting unnecessary permissions).</li> <li>Collecting unnecessary private data.</li> <li>Sharing data with third-parties or other users without user’s consent.</li> </ul>
<b>Security</b>	<ul style="list-style-type: none"> <li>Containing virus or malware.</li> <li>Being suspicious to payment fraud.</li> </ul>

the representative comments and can only provide a lower bound of the detection of users’ complaints on undesired app behaviors or regulation violations.

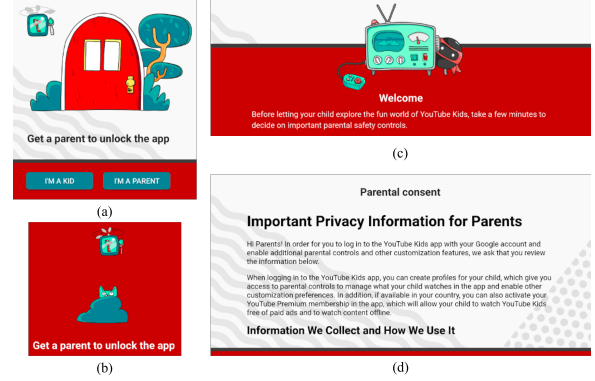
## 4 RESULTS

In this section, we present the measurement results from applying our analysis pipeline to 3,627 children’s apps that participate in the Google Family project (the “Family Apps”), and 16,568 normal apps with kids included in the target user group (the “Normal Apps”). We conduct a regression test and find that 2,857 (77.8%) of detected apps are still listed in Google Play (for more than 1 year since we collected them).

### 4.1 The usage of Location Permissions

Through the static analysis, we measure the use of location related permissions, including ACCESS\_COARSE\_LOCATION and ACCESS\_FINE\_LOCATION permissions, in the Family and Normal apps. We find that 162 (4.47%) Family apps request at least one of the location related permissions, asking for the access to user’s coarse or fine location information (3.58% for ACCESS\_FINE\_LOCATION and 2.32% for both), which is potentially violating the Design for Family policy. We further report the permissions usage in normal apps primarily as a comparison point to show that children’s apps are requesting fewer permissions. 39.83% of Normal apps request location permissions (36.03% for ACCESS\_FINE\_LOCATION and 28.31% for both). However, note that all the Normal apps in our research have target users that include children (less than 13 years old). We argue that developers should be equally cautious about using location permissions when the target users include children.

**Case study: YouTube Kids.** The YouTube Kids app [73], a Google product built with children in-mind, aims to make it safer and easier for children to explore videos. This provides a good example of permission request notifications for children. As shown in Figure 5(a), when the app is opened for the first time, a simple and clear notification is presented to users, noting that a parent is required to unlock the app. If the user selects “I’M A KID”, the app



**Figure 5: Examples of YouTube Kids notifications**

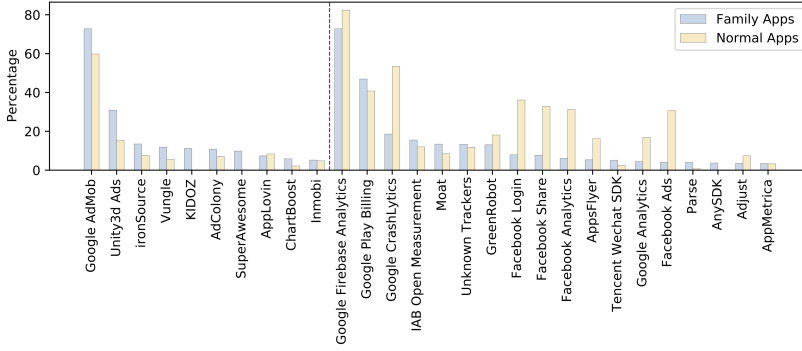
requires the child to get their parent, and the only choice is going back to the previous screen (Figure 5(b)). Only after the “I’M A PARENT” button has been clicked, the app starts showing a introduction video (Figure 5(c)) and obtaining permissions from parent user (Figure 5(d)). Although a child may fool this process and login with their parent’s account, the app tries its best to ask consent from guardians. Through our manual checks on 200 randomly selected Family and Normal apps, we did not find other permission notification designed for children users in other apps.

#### Finding summary:

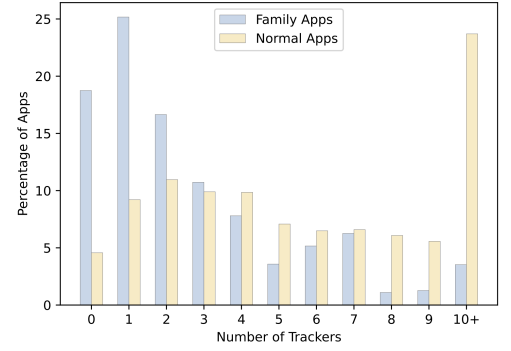
- 4.47% of Family apps request location-related permissions, which is potentially violating the Design for Family policy.
- 36.03% of Normal apps request the access of fine location information, where children users are not particularly unprotected.

### 4.2 Third-party Trackers

To determine the usage and the potential privacy leakage through third-party trackers, we next leverage static and dynamic analysis. **The use of third-party trackers.** The static analysis results on the use of third-party trackers in both Family and Normal apps are presented in Figure 6(a). The trackers on the left side of the vertical red dash-line are allowed by the Design for Family program. Google AdMob is the most frequently used ad SDK in both Family and Normal apps. Not surprisingly, Family apps use the kids-allowed SDKs more frequently, especially for SDKs such as KID0Z, SuperAwesome, and ChartBoost. This could indicate that the Play store’s Design for Family program contributes positively to app development — more developers consider using the recommended SDKs when their target users are children. However, as shown by the right side of the vertical red dash-line, we find that quite a few (81.25%) Family apps use SDKs that are not allowed for children’s apps, i.e., not in the Google Play certified ad SDKs. For example, 72.8% of Family apps use Google Firebase Analytics and the number for Google Play Billing is 46.91%, while other tracker SDKs such as Google CrashLytics, Facebook Login, and Appsflyer are more frequently used in Normal apps. Even we exclude ads and tracker SDKs from Google, there are still 13.08% Family apps use at least one tracker that is not allowed for children’s apps.



(a) Percentages of apps that use top-15 DFF trackers and top-15 non-DFF trackers.



(b) Distribution of apps w.r.t. number of non-kids trackers used.

Figure 6: Trackers analysis results.

Considering that some tracker SDKs are very popular among both Normal apps and Family apps, and they may provide irreplaceable functionalities to apps, we argue that a list of kids-allowed tracker SDKs is essential to protect the privacy of children users. This will likely also benefit app developers.

Figure 6(b) shows the distribution of the number of trackers per-app. Worryingly, only 18.75% of Family apps do not use a disallowed ad or tracker SDK, whereas 95.43% of Normal apps use SDKs that are disallowed when the target users include children. Even when we exclude SDKs from Google and Facebook, the numbers are still high: 38.49% of Family apps and 28.25% of Normal apps use at least one non-kids SDK. In fact, 1.63% of Family apps and 5.54% of Normal apps use over 10 non-kids SDKs. From these results, we confirm that the Design for Family requirements are *not* followed by all app developers. We note that, through static analysis, we cannot determine which ad or tracker SDKs are active during run-time. However, dynamic analysis may still be incomplete as it is hard to catch all tracking. We posit if the trackers are not necessary, the app developers should remove the SDKs from a children’s app.

**Privacy leakage.** We further analyze the specific privacy leakages via both tracker SDKs and first-party APIs. We identify that 3.79% of Family apps leak the device model, brand, builder, and other PII to third-party trackers without any consent from users. We observe that 7 Family apps sent Private IP data to Google, and Supersonic Ads obtains Advertisement ID (from 3 Family apps), Location Information (1 app), and Android Serial (from 3 Family apps) without users’ consents. Furthermore, Location Information is sent to Facebook (from 4 Family apps), and Babybus collects the Android Serial (from 4 Family apps). We cannot check how the leaked PII is used, but emphasize that the leaking of PII has already violated GDPR and other privacy regulations (which is forbidden no matter if the target users are children or not). We note again that our results only present a lower-bound of the PII leakage without users’ consent.

**Case study: excessive tracking.** During the experiments, we found an interesting phenomenon: apps from the same developer tend to have similar trackers, even when the apps’ functionalities or content are different. For example, we find that 6 apps from Tu\*\*\*ns, a large developer in the children’s game market (publishing over 100 games with 950+ million downloads according to its official

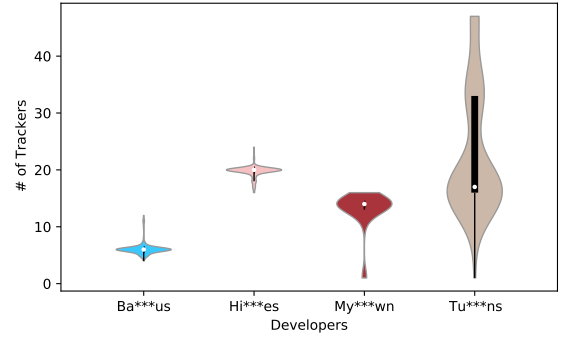


Figure 7: Number of trackers used in children’s apps from 4 big players.

website), has 47 trackers. Although Tu\*\*\*ns has a privacy policy which lists 17 ad or tracker SDKs, there are still 30 trackers that have not been mentioned. It is therefore unclear if parents or children know that their personal information could be shared with 17 (or even 47) third-party trackers.

We investigate more apps to see if developers who have a large number (40+) of children’s apps, embed a large numbers of trackers. In Figure 7, we select four developers from our static results and list the distribution of numbers of trackers in their products. 142 out of the 150 apps (94.7%) from Ba\*\*\*us have 5 to 6 trackers; 105 out of 133 apps (78.9%) from Hi\*\*\*es have 20 trackers, and 29 out of 43 apps (67.4%) have 14 trackers. The distribution of the number of trackers is less concentrated among apps from Tu\*\*\*ns: 6/62 Tu\*\*\*ns apps have 47 trackers; 14 apps have 34 trackers, and 16 trackers are detected in the remaining 41 apps. The reason behind this phenomenon could be that developers directly apply the same tracker configurations while developing different apps, instead of setting trackers for each app according to what personal information should be collected and shared. Further, we conclude that most apps from these 4 big players contain trackers not allowed in children’s apps.



#### Finding summary:

- 81.25% of Family apps use trackers that are not allowed for children; disallowed trackers are also frequently used in 95.43% of Normal apps whose target user includes children.
- 1.63% of Family apps and 5.54% of Normal apps use over 10 ad or tracker SDKs that are not allowed to be used in children’s apps.
- Even big players do not follow the Play store policy. We find 4 developers who have 5–47 disallowed trackers in each of the 43 to 150 children’s apps they listed on Play store.

### 4.3 Inconsistent Content Age Ratings

We next investigate the age ratings of the 9,453 apps across all 5 rating authorities. These are ACB for Australia, ESRB for Americas, PEGI for Europe and the Middle East, USK for Germany, and IARC (which is not country specific). We seek to check if the age ratings given by these different agencies are inconsistent. We note that there could be demographic or cultural bias from parents. The purpose of releasing age ratings across different authorities is to provide a general reference message for parents’ decision-making. Such information could be limited to only present highly inconsistent ratings. Indeed, we find inconsistent ratings in 19.25% apps, and 9.99% apps have an inconsistency level above 3. This indicates that age rating inconsistencies among the various rating authorities are prevalent.

Figure 8 presents the age rating results across the top-10 categories of Normal apps vs. Family apps. Note, as one app may have multiple inconsistent rating pairs across different levels detected, the sum of apps could be higher than 100%. Although Family apps have a lower ratio of inconsistent ratings than the other 10 Normal app categories, at least 6% Family apps have an inconsistency level higher than 3. 26% of COMICS and 15% of ENTERTAINMENT apps have inconsistency levels above 3, and at least 55% of SOCIAL apps have a rating inconsistency level 2. We conjecture that the reason for highly inconsistent ratings in these categories may be because the rules to flag sensitive content could be different across the rating authorities. However, we argue it is still not reasonable to have an app rated as 18+ in one territory and 3+ in another (level 4). This will confuse parents and increase the risk to children (e.g., allowing a child to play a game rated 18+ in another country). Therefore, we argue that, rather than only displaying the age ratings in the user’s current territory, the app store should also provide ratings from other authorities as an option. Briefly listing the reason why the app is rated will also help parents and children to make a decision.

**Case study: highly inconsistent ratings.** Here, we showcase some inconsistent age ratings to highlight key concerns. For example, `com.my***in`.app, a plant identification app with professional care guides (1,000,000+ installs), has been rated as “PEGI 3” and “Rated for 3+” in IARC. However, when we visit the Play store using `gl=de` and `gl=us`, the ratings change to “USK 12+” and “ESRB Mature 17+”. The only explanation we can find is “Drug use”. If the app truly contains information related to drugs, it is unsuitable for 3-year old children. However, we manually check the app and only find drug-related information for plant diseases. Thus, the labels in PEGI 3 and IARC are unnecessary.

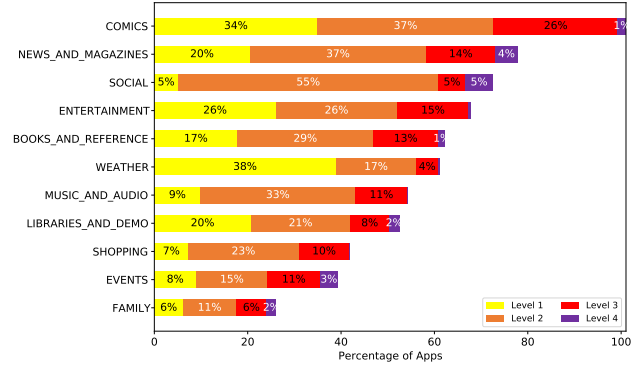


Figure 8: Percentage of apps (per category) with inconsistent age ratings.

Another example is `com.di****rd`, an instant messaging and digital distribution platform. As of 2021, the service has over 350 million registered users and over 150 million monthly active users (over 100M installs from the Play store). As a social app, it is quite reasonable to be rated as “Parental guidance” in PEGI (as such apps may not always have predefined content that can be classified beforehand). It is rated as “Teen” in ESRB, yet in USK, the app is rated as “18+”, which hints that the app contains “drug use, realistic and explicit violence”. However, there is no explanation on the app page why the app is rated “18+”. We argue there should be better descriptions that explain why the app is rated, at least for apps not recommended for children.

#### Finding summary:

- 21.69% of apps have inconsistent content age ratings across different rating authorities.
- There are many examples of confusing and inconsistent ratings, including among Family apps. 6% of Family apps have an inconsistency level above 3.
- The app store should provide ratings from other authorities as references for users, and explanations of ratings should be more transparent.

### 4.4 User Complaints

Finally, we analyze the 13,132,577 comments downloaded from 11,831 apps in the Play store. We strive to measure users’ complaints across 4 categories: app content (where some content within is not suitable for children), advertisement, privacy, and security.

Table 4 presents the percentage of comments that have been detected as complaints in each category, alongside the percentage of corresponding apps that they refer to. In 2.12% of the comments for Family apps, we recognize at least one user complaint relates to the app content, while the number is just 0.92% for Normal apps. This indicates that the users of Family apps complain more about the content. This reflects that the developers of Family app may not apply special controls on app content when the users are children. The ratios of comments related to advertisement complaints are similar in both Family and Normal apps, although this impacts more Normal apps than Family apps (34.81% vs. 25.77%). Further, fewer Family apps are complained about with respect to privacy

Table 4: Comment analysis results.

Categories	Family Apps		Normal Apps	
	% Comments	% Apps	% Comments	% Apps
Content	2.12%	30.62%	0.92%	33.86%
Ads	1.01%	25.77%	0.97%	34.81%
Privacy	0.09%	6.91%	0.52%	29.05%
Security	0.16%	12.89%	0.43%	23.76%

and security than the other two categories. A possible explanation might be that non-technical users presumably cannot tell which trackers are embedded in an SDK, so they are unlikely to complain in an app comment about that particular privacy threat, but focus more on app content and advertisements in children’s apps, which resonates with the contribution of our study. According to our measurements, 45 of the top-100 apps have quality rating scores above 4 stars and over 10M installations.

After we disclosed our findings to the Google privacy team, Play store updated the FAMILY category requirement, and explicitly required Teacher Approved badge for each kid-friendly app, where teachers and specialists rate apps based on criteria such as age and ads appropriateness, rather than self-certified [51]. The Families policy has also been updated to declare that apps will be available only to users in regions where content within that app is deemed appropriate [53]. We provide more details of Play store policy updates in Appendix.

**Finding summary:**

- 30.62% and 25.77% of Family apps have content-based and ad-based complaints, respectively. Users report privacy and security issues on Family apps less often though (6.91% and 12.89%, respectively).
- Even highly popular and well rated apps accumulate many complaints — among the top 100 apps that have the most complaint comments, 45 apps have app rating scores above 4 and over 10M installs.

## 5 RELATED WORK

In this section, we survey the recent research related to the evaluation of children’s apps, static and dynamic analysis, and user comments analysis.

**Evaluation of children’s apps.** Previous efforts have studied COPPA and children’s apps from various perspectives. Previous work examined the risks posed by third-party components bundled in children’s apps, with a focus on targeted advertisements [10, 39]. Several studies examined privacy policies [44, 58, 59, 66] according to the requirements of regulations. Other research has focused on methods aiding developers to make their apps more child-friendly in terms of content and privacy [31, 38]. Several works focus on the evaluation of app content age rating. For example, a study by Chen *et al.* [17] looks into the inappropriate age rating of mobile apps, exposing its potential risk for the children and adolescents. However, this research focuses on the comparison of age ratings between iOS app store and Google Play store, rather than exposing undesired application behaviors.

**Static and dynamic analysis.** Static code analysis techniques are widely used in the assessment of mobile apps [15, 16, 32, 36, 37, 41, 64, 65, 67, 71, 72]. For example, FLOWDROID [7] statically computes data flows in apps to understand which parts of the code that data may be exposed to. Notably, these off-the-shelf tools only utilize syntax-based scanning and data-flows, which leads to false positives that are not relevant to personal identifiable information. In contrast, dynamic analysis executes the code. Whereas static analysis often suffers from false positives, dynamic analysis is limited by the execution coverage [57]. Most work in this category focuses on analyzing apps’ network traffic [29, 35, 56, 62, 68]. In our work, we utilize static analysis for identifying permission requests and third-party trackers, while we trace privacy leakage using network traffic through a dynamic network monitoring tool [56].

**User comments analysis.** User comments of mobile apps have been extensively studied from several perspectives, including mining user opinions [14, 47, 48, 69, 70], app comment filtering [14, 43], and exploring other concerns [13, 46]. Chen *et al.* [13] conducted a study on fraudulent campaigns to falsely boost apps’ rankings, which result in inappropriate risk exposure for children and adolescents. Other work looks into research domains related to app descriptions, privacy policies, and other mobile app meta text, adopting NLP techniques [54]. PPChecker [74] identifies the inconsistencies between the sensitive behaviors of apps and their privacy policies. Recent research by Liu *et al.* [40] tries to solve the problem of compliance analysis between GDPR and privacy policies, utilizing a combination of sentence classification and rule-based analysis. However, the corpus suffers from an imbalanced data problem, which negatively affects the classification accuracy. We build on these techniques to investigate children’s app behavior through comments analysis.

## 6 CONCLUSION

This paper has focused on the privacy practices of apps that are designed for children or have target users that include children. We have measured the use of permissions and trackers, investigated inconsistency in content age ratings, and analyzed user comment feedback. Our measurement results illustrate that, despite many privacy protection regulations and the strict requirements imposed by the app store, children still experience privacy threats. This is caused by things like permission requests without child-friendly notifications, abuse of ad trackers, confusing and inconsistent content age ratings, as well as privacy leakage without users consent. Ultimately, we conclude that the existing self-certification-based content age rating mechanism must be improved immediately.

## ACKNOWLEDGEMENT

The work has been supported by the Cyber Security Research Centre Limited whose activities are partially funded by the Australian Government’s Cooperative Research Centres Program. We thank Sai Teja Peddinti for useful discussion and guidance on the verification of our results.

## REFERENCES

- [1] 1988. Privacy Act 1988. <https://www.legislation.gov.au/Series/C2004A03712/Compilations>.
- [2] 2012. Andro Guard. <https://github.com/androguard/androguard>.

- [3] 2021. Emoji Python package. <https://github.com/carpdm20/emoji.git>.
- [4] 2022. Application exerciser Monkey. <https://developer.android.com/studio/test/monkey>.
- [5] 2022. Exodus tracker investigation platform. <https://etip.exodus-privacy.eu.org trackers/all>.
- [6] National Archives. 2022. Part 312 - Children's online privacy protection rule. <https://www.ecfr.gov/current/title-16/part-312>.
- [7] Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Ochteau, and Patrick McDaniel. 2014. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. *Acm Sigplan Notices* (2014), 259–269.
- [8] Brooke Auxier, Monica Anderson, Andrew Perrin, and Erica Turner. 2020. Parenting children in the age of screens. <https://www.pewresearch.org/internet/2020/07/28/parenting-children-in-the-age-of-screens/>.
- [9] Joshua Barrie. 2017. This is how much of the internet is porn. <https://www.mirror.co.uk/tech/how-much-internet-porn-9784557>.
- [10] Ravi Bhaskar, Seungyeop Han, Jinseong Jeon, Tanzirul Azim, Shuo Chen, Jaeyoon Jung, Suman Nath, Rui Wang, and David Wetherall. 2014. Brahmastra: Driving apps to test the security of third-party components. In *23rd USENIX Security Symposium (USENIX Security 14)*, 1021–1036.
- [11] Sean Bleier. 2009. NLTK's list of English stopwords. <https://gist.github.com/sebleir/554280>.
- [12] Mindy Brooks. 2020. Find high-quality apps for kids on Google Play. <https://blog.google/products/google-play/teacher-approved-apps/>.
- [13] Hao Chen, Daojing He, Sencun Zhu, and Jingshun Yang. 2017. Toward detecting collusive ranking manipulation attackers in mobile app markets. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, 58–70.
- [14] Ning Chen, Jialiu Lin, Steven CH Hoi, Xiaokui Xiao, and Boshen Zhang. 2014. ARminer: Mining informative reviews for developers from mobile app marketplace. In *Proceedings of the 36th international conference on software engineering*, 767–778.
- [15] Sen Chen, Lingling Fan, Guozhu Meng, Ting Su, Minhui Xue, Yinxing Xue, Yang Liu, and Lihua Xu. 2020. An empirical assessment of security risks of global Android banking apps. In *Proceedings of the 42st International Conference on Software Engineering*. IEEE Press, 596–607.
- [16] Sen Chen, Ting Su, Lingling Fan, Guozhu Meng, Minhui Xue, Yang Liu, and Lihua Xu. 2018. Are mobile banking apps secure? What can be improved?. In *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 797–802.
- [17] Ying Chen, Heng Xu, Yilu Zhou, and Sencun Zhu. 2013. Is this app safe for children? A comparison study of maturity ratings on Android and iOS applications. In *Proceedings of the 22nd international conference on World Wide Web*, 201–212.
- [18] International Age Rating Coalition. 2022. How IARC works. <https://www.globalratings.com/how-iarc-works.aspx>.
- [19] Federal Trade Commission. 1998. Children's Online Privacy Protection Act (COPPA). <http://www.ftc.gov/ogc/coppa1.htm>.
- [20] Regional Development Department of Infrastructure, Transport and Communication. 2022. The advisory categories for films and computer games. <https://www.classification.gov.au/classification-ratings/what-do-ratings-mean>.
- [21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 4171–4186.
- [22] Entertainment Software Association. 2022. ESRB ratings guide. <https://www.esrb.org/ratings-guide/>.
- [23] European Parliament and Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament. *Official Journal of the European Union* 119 (2016), 1–88.
- [24] European Parliament and Council of the European Union. 2022. Special protection of children's personal data. <https://gdpr-info.eu/recitals/no-38/>.
- [25] Álvaro Feal, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, Alessandra Gorla, et al. 2020. Angel or devil? A privacy study of mobile parental control apps. *Proceedings of Privacy Enhancing Technologies (PoPETs)* (2020).
- [26] The United Nations Children's Fund. 2019. UNICEF poll: More than a third of young people in 30 countries report being a victim of online bullying. <https://www.unicef.org/press-releases/unicef-poll-more-third-young-people-30-countries-report-being-victim-online-bullying>.
- [27] The United Nations Children's Fund. 2020. Child protection advocacy brief. <https://www.unicef.org/media/96691/file/Online-Protection-Advocacy-Brief-2021.pdf>.
- [28] Julien Gamba, Mohammed Rashed, Abbas Razaghpanah, Juan Tapiador, and Narseo Vallina-Rodriguez. 2020. An analysis of pre-installed android software. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1039–1055.
- [29] Catherine Han, Irwin Reyes, Amit Elazar Bar On, Joel Reardon, Álvaro Feal, Serge Egelman, Narseo Vallina-Rodriguez, et al. 2019. Do you get what you pay for? Comparing the privacy behaviors of free vs. paid apps. In *Workshop on Technology and Consumer Protection (ConPro 2019)*.
- [30] Catherine Han, Irwin Reyes, Álvaro Feal, Joel Reardon, Primal Wijesekera, Narseo Vallina-Rodriguez, Amit Elazar, Kenneth A Bamberger, and Serge Egelman. 2020. The price is (not) right: Comparing privacy in free and paid apps. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (2020).
- [31] Bing Hu, Bin Liu, Neil Zhenqiang Gong, Deguang Kong, and Hongxia Jin. 2015. Protecting your children from inappropriate content in mobile apps: An automatic maturity rating framework. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 1111–1120.
- [32] Yangyu Hu, Haoyu Wang, Ren He, Li Li, Gareth Tyson, Ignacio Castro, Yao Guo, Lei Wu, and Guoai Xu. 2020. Mobile app squatting. In *Proceedings of The Web Conference 2020*, 1727–1738.
- [33] Yangyu Hu, Haoyu Wang, Tiantong Ji, Xusheng Xiao, Xiapu Luo, Peng Gao, and Yao Guo. 2021. Champ: Characterizing undesired app behaviors from user comments based on market policies. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*, 933–945.
- [34] Pan European Game Information. 2022. PEGI age ratings. <https://pegi.info/page/pegi-age-ratings>.
- [35] Anh Le, Janus Varmarken, Simon Langhoff, Anastasia Shuba, Minas Gjoka, and Athina Markopoulou. 2015. AntMonitor: A system for monitoring from mobile devices. In *Proceedings of the 2015 ACM SIGCOMM Workshop on Crowdsourcing and Crowdsourcing of Big (Internet) Data*, 15–20.
- [36] Chaoran Li, Xiao Chen, Ruoxi Sun, Minhui Xue, Sheng Wen, Muhammad Ejaz Ahmed, Seyit Camtepe, and Yang Xiang. 2022. Cross-language Android permission specification. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 772–783.
- [37] Li Li, Alexandre Bartel, Tegawendé F Bissyandé, Jacques Klein, Yves Le Traon, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Ochteau, and Patrick McDaniel. 2015. Icta: Detecting inter-component privacy leaks in android apps. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 280–291.
- [38] Ilaria Liccardi, Monica Bulger, Hal Abelson, Daniel J Weitzner, and Wendy Mackay. 2014. Can apps play by the COPPA Rules?. In *2014 Twelfth Annual International Conference on Privacy, Security and Trust*. IEEE, 1–9.
- [39] Minxing Liu, Haoyu Wang, Yao Guo, and Jason Hong. 2016. Identifying and analyzing the privacy of apps for kids. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, 105–110.
- [40] Shuang Liu, Baiyang Zhao, Renjie Guo, Guozhu Meng, Fan Zhang, and Meishan Zhang. 2021. Have you been properly notified? Automatic compliance analysis of privacy policy text with GDPR Article 13. In *Proceedings of the Web Conference 2021*, 2154–2164.
- [41] Tianming Liu, Haoyu Wang, Li Li, Xiapu Luo, Feng Dong, Yao Guo, Liu Wang, Tegawendé Bissyandé, and Jacques Klein. 2020. MadDroid: Characterizing and detecting devious Ad contents for Android apps. In *Proceedings of The Web Conference 2020*, 1715–1726.
- [42] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [43] Washington Luiz, Felipe Viegas, Rafael Alencar, Fernando Mourão, Thiago Salles, Dárlinton Carvalho, Marcos Andre Gonçalves, and Leonardo Rocha. 2018. A feature-oriented sentiment rating for mobile app reviews. In *Proceedings of the 2018 World Wide Web Conference*, 1909–1918.
- [44] Aaron K Massey, Jacob Eisenstein, Annie I Antón, and Peter P Swire. 2013. Automated text mining for requirements analysis of policy documents. In *2013 21st IEEE international requirements engineering conference (RE)*. IEEE, 4–13.
- [45] Preksha Nema, Pauline Anthonysamy, Nina Taft, and Sai Teja Peddinti. 2022. Analyzing user perspectives on mobile app privacy at scale. In *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*.
- [46] Duc Cuong Nguyen, Erik Derr, Michael Backes, and Sven Bugiel. 2019. Short text, large effect: Measuring the impact of user reviews on android app security & privacy. In *2019 IEEE symposium on Security and Privacy (SP)*. IEEE, 555–569.
- [47] Ehsan Noei, Feng Zhang, and Ying Zou. 2019. Too many user-reviews! What should app developers look at first? *IEEE Transactions on Software Engineering* 47, 2 (2019), 367–378.
- [48] Sebastiano Panichella, Andrea Di Sorbo, Emtiza Guzman, Corrado A Visaggio, Gerardo Canfora, and Harald C Gall. 2016. Ardoc: App reviews development oriented classifier. In *Proceedings of the 2016 24th ACM SIGSOFT international symposium on foundations of software engineering*, 1023–1027.
- [49] Play Store. 2015. Creating apps and games for children and families. <https://play.google.com/console/about/families/>.
- [50] Play Store. 2015. Designing apps for children and families. <https://support.google.com/googleplay/android-developer/answer/9893335>.
- [51] Play Store. 2022. Build teacher approved apps. <https://play.google.com/console/about/programs/teacherapproved/>.
- [52] Play Store. 2022. Content ratings. <https://support.google.com/googleplay/android-developer/answer/9898843>.

- [53] Play Store. 2023. Enforcement process. <https://support.google.com/googleplay/android-developer/answer/9899234>.
- [54] Zhengyang Qu, Vaibhav Rastogi, Xinyi Zhang, Yan Chen, Tiantian Zhu, and Zhong Chen. 2014. Autocog: Measuring the description-to-permission fidelity in android applications. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 1354–1365.
- [55] Anand Rajaraman and Jeffrey David Ullman. 2011. *Mining of massive datasets*. Cambridge University Press.
- [56] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, Phillipa Gill, et al. 2018. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS 2018)*.
- [57] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 2019. 50 ways to leak your data: An exploration of apps’ circumvention of the android permissions system. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 603–620.
- [58] Irwin Reyes, Primal Wijesekera, Abbas Razaghpanah, Joel Reardon, Narseo Vallina-Rodriguez, Serge Egelman, Christian Kreibich, et al. 2017. “Is our children’s apps learning?” Automatically detecting COPPA violations. In *Workshop on Technology and Consumer Protection (ConPro 2017)*.
- [59] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, Serge Egelman, et al. 2018. “Won’t somebody think of the children?” Examining COPPA compliance at scale. In *The 18th Privacy Enhancing Technologies Symposium (PETS 2018)*.
- [60] Unterhaltungssoftware Selbstkontrolle. 2022. USK age categories. <https://usk.de/en/the-usk/faqs/age-categories/>.
- [61] Gregorio Serra, Lucia Lo Scalzo, Mario Giuffrè, Pietro Ferrara, and Giovanni Corsello. 2021. Smartphone use and addiction during the coronavirus disease 2019 (COVID-19) pandemic: Cohort study on 184 Italian children and adolescents. *Italian Journal of Pediatrics* 47, 1 (2021), 1–10.
- [62] Anastasia Shuba, Athina Markopoulou, and Zubair Shafiq. 2018. NoMoAds: Effective and efficient cross-app mobile Ad-blocking. In *Proceedings of the Privacy Enhancing Technologies Symposium (PETS)*, Vol. 2018.
- [63] Aastha Singh. 2021. Evolving with BERT: Introduction to RoBERTa. <https://medium.com/analytics-vidhya/evolving-with-bert-introduction-to-roberta-5174ec0e7c82>.
- [64] Ruoxi Sun, Wei Wang, Tian Dong, Shaofeng Li, Minhui Xue, Gareth Tyson, Haojin Zhu, Mingyu Guo, and Surya Nepal. 2022. Measuring vulnerabilities of malware detectors with explainability-guided evasion attacks. *arXiv preprint arXiv:2111.10085* (2022).
- [65] Ruoxi Sun, Wei Wang, Minhui Xue, Gareth Tyson, Seyit Camtepe, and Damith C Ranasinghe. 2021. An empirical assessment of global COVID-19 contact tracing applications. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1085–1097.
- [66] Ruoxi Sun and Minhui Xue. 2020. Quality assessment of online automated privacy policy generators: An empirical study. In *Proceedings of the Evaluation and Assessment in Software Engineering*. 270–275.
- [67] Zhushou Tang, Ke Tang, Minhui Xue, Yuan Tian, Sen Chen, Muhammad Ikram, Tielei Wang, and Haojin Zhu. 2020. iOS, your OS, everybody’s OS: Vetting and analyzing network services of iOS applications. In *29th USENIX Security Symposium*. 2415–2432.
- [68] Max Van Kleek, Ilaria Liccardi, Reuben Binns, Jun Zhao, Daniel J Weitzner, and Nigel Shadbolt. 2017. Better the devil you know: Exposing the data sharing practices of smartphone apps. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 5208–5220.
- [69] Lorenzo Villaruel, Gabriele Bavota, Barbara Russo, Rocco Oliveto, and Massimiliano Di Penta. 2016. Release planning of mobile apps based on user reviews. In *2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE)*. IEEE, 14–24.
- [70] Phong Minh Vu, Hung Viet Pham, Tam The Nguyen, and Tung Thanh Nguyen. 2016. Phrase-based extraction of user opinions in mobile app reviews. In *Proceedings of the 31st IEEE/ACM international conference on automated software engineering*. 726–731.
- [71] Wei Wang, Ruoxi Sun, Minhui Xue, and Damith C Ranasinghe. 2020. An automated assessment of Android clipboards. In *2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE)*.
- [72] Minhui Xue, Cameron Ballard, Kelvin Liu, Carson Nemelka, Yanqiu Wu, Keith Ross, and Haifeng Qian. 2016. You can yak but you can’t hide: Localizing anonymous social network users. In *Proceedings of the 2016 Internet Measurement Conference*. 25–31.
- [73] YouTube. 2022. YouTube Kids. [com.google.android.apps.youtube.kids](https://www.google.com/android/apps/youtube/kids).
- [74] Le Yu, Xiapu Luo, Xule Liu, and Tao Zhang. 2016. Can we trust the privacy policies of android apps?. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 538–549.

## APPENDIX

### 1 UPDATES TO GOOGLE PLAY POLICIES

After we disclosed our findings to the Google privacy team, Play store announced several updates to Google Play Policies. Here we list some of the announcements and updates from <https://support.google.com/googleplay/android-developer/answer/9934569> that are related to our findings.

- “Effective May 11, 2022: We’re updating our Families policy to explain that if an app contains content that is not globally appropriate, we may make the app available only to users in regions where content within that app is deemed appropriate.”
- Announced on July 27, 2022: “We’re clarifying our Families Data Practices policy to state that apps that solely target children must not transmit Android advertising identifier (AAID), SIM Serial, Build Serial, BSSID, MAC, SSID, IMEI, and/or IMSI. Apps that target both children and older audiences must not transmit AAID, SIM Serial, Build Serial, BSSID, MAC, SSID, IMEI, and/or IMSI from children or users of unknown age.”
- “Effective November 1, 2022: We’re updating our Families Self-Certified Ads SDK Program to require that self-certified ads SDK providers must submit new policy-compliant, self-certified versions and a test app to remain in the Families Self-Certified Ads SDK Program.”
- Announced on November 16, 2022: “To better align with the existing policy language, we’re renaming the Designing Apps for Children and Families policy page to Google Play Families Policies. We’re also consolidating the page formerly known as Families Ads & Monetization onto the newly titled Google Play Families Policies page.”
- “Effective May 31, 2023: We’re updating our Families Self-Certified Ads SDK policy to require developers with apps in the Families Program to only use self-certified versions of SDKs when serving ads to children or users of unknown age.”