Moritz Gruber*, Christian Höfig, Maximilian Golla, Tobias Urban, and Matteo Große-Kampmann

"*We may share the number of diaper changes*": A Privacy and Security Analysis of Mobile Child Care Applications

Abstract: Mobile *child care management applications* can help child care facilities, preschools, and kindergartens to save time and money by allowing their employees to speed up everyday child care tasks using mobile devices. Such apps often allow child care workers to communicate with parents or guardians, sharing their children's most private data (e. g., activities, photos, location, developmental aspects, and sometimes even medical information). To offer these services, child care apps require access to very sensitive data of minors that should never be shared over insecure channels and are subject to restrictive privacy laws. This work analyzes the privacy and security of 42 Android child care applications and their cloud-backends using a combination of static and dynamic analysis frameworks, configuration scanners, and inspecting their privacy policies. The results of our analysis show that while children do not use these apps, they can leak sensitive data about them. Alarming are the findings that many third-party (tracking) services are embedded in the applications and that adversaries can access personal data by abusing vulnerabilities in the applications. We hope our work will raise awareness about the privacy risks introduced by these applications and that regulatory authorities will focus more on these risks in the future.

Keywords: child care, app, management, Android, privacy, security, COPPA, GDPR

DOI 10.56553/popets-2022-0078
Received 2021-11-30; revised 2022-03-15; accepted 2022-03-16.

***Corresponding Author: Moritz Gruber, Christian Höfig:** AWARE7 GmbH, E-mail: moritz@aware7.de
Maximilian Golla: Max Planck Institute for Security and Privacy, E-mail: maximilian.golla@csp.mpg.de
Tobias Urban: Institute for Internet Security – if(is); secunet Security Networks AG, E-mail: urban@internet-sicherheit.de
Matteo Große-Kampmann: Ruhr University Bochum; AWARE7 GmbH; Institute for Internet Security – if(is); E-mail: matteo@aware7.de

1 Introduction

Our lives become more digital every day, and children are not exempt from this anymore. This shift has introduced various new challenges for parents, guardians, teachers, and child care workers, including threats such as stalking and other child maltreatment that benefits from using the latest technologies [20, 46, 64, 94]. The use of child care facilities differs by country, social insurance system, income level, and other factors. According to the US *National Center for Education Statistics* (NCES), about 63% of children ages three to five that had any non-parental care were enrolled in some child care center program in 2019 [13]. Everyday life in these centers is traditionally analog, where employees mainly use computers and the Internet for administrative tasks. Digital solutions such as mobile devices are only recently coming into these centers and can support the staff in common – often very time-consuming – tasks like documentation [31, 69].

Mobile *child care applications* offer various features to control and manage child care environments such as kindergartens, preschools, or daycare centers. According to their developers, three aspects of child care might benefit from the usage of these applications: (1) documentation of developmental aspects of the children, (2) engagement with the parents, and (3) support for administration and management of the child care center. While child care app solutions may help, they could also introduce severe risks to children's and parents' privacy because of the sensitive nature of the exchanged data. The processed data might include a child's name, birthday, private photos, current location, activities, and sometimes even health data. The child care management applications market is growing steadily, and individual providers are developing applications for separate target groups with different feature sets. Due to the heterogeneity of the market, some features are provider-specific, e. g., interfaces to location-specific backend systems or *COVID-19* temperature monitoring.

Previous works mainly studied how children or teens make use of digital technologies [92] or analyzed apps

BY-NC-ND

that help parents to monitor or control the (online) activities of their children [12, 21, 45, 91]. This work analyzes 42 child care management applications meant to assist child care workers with administrative tasks. We present the first comprehensive privacy-oriented analysis of such mobile applications that enable parents to participate more closely in their children's early development.

To summarize, we make the following key contributions:

1. We analyze the applications' behavior using static and dynamic analysis to find privacy leaks and potential security misconfigurations that can lead to children's privacy violations.
2. We show that these applications use dangerous permissions, rely on open cloud storage, and show abusive tracking behavior in case studies.
3. Finally, we discuss compliance and regulatory requirements of child care applications and find that the stated claims in the privacy policies and terms of services do not match the applications' behavior.

The remainder of the paper is structured as follows: First (Section 2), we provide some background information on the analyzed application family and the regulatory requirements. Afterward, we discuss some related work (Section 3). In Section 4, we provide an overview of the threat model (Section 4.1), the applications we analyzed (Section 4.2), give a detailed description of the used static (Section 4.4) and dynamic (Section 4.5) analysis tools. In Section 5, we present the results. More precisely, we describe the analyzed apps (Section 5.1), discuss their privacy implications in-depth (Section 5.2), and finish the section with an analysis of their security measures (Section 5.3). Finally, we consider the ethical implications of our work (Section 6), discuss our findings (Section 7), list the limitations of our approach (Section 8), and conclude our work (Section 9).

2 Background

This section provides an overview of the application family we analyze. Moreover, we introduce two relevant legislatures to understand why privacy should be an integral part of these applications. Finally, we describe the difference between static and dynamic analysis of mobile apps.

2.1 Mobile Child Care Applications

Mobile child care applications offer various features to control and manage child care environments like kindergartens or daycare centers. Notably, the applications are *not meant to be used by children themselves* but rather by those who either care for them or are legal representatives, i. e., child care workers, educators, parents, and guardians. The applications offer different features, but they generally try to support the staff by easing the mandatory documentation of development and education [14, 89]. They do this by offering features for instance sending out daily reports, writing bills to parents, or managing children's attendance in the center. Some applications offer a chat-like interface where the caregivers and parents can directly communicate and share different media. While this feature is meant to increase parents' engagement by sharing photos, videos, or various information about the children, it can also increase certain risks due to the significantly increased attention [35, 57, 76]. Some applications can also share the location of a group with the parents or other child care workers. In general, all the applications we analyzed offer the following three core functionalities to staff members:

1. **Documentation**: Taking notes, photos, or videos (e. g., child development, activities).
2. **Communication**: Sending messages to parents or co-workers (e. g., announcements, location sharing).
3. **Administration**: Managing the child care center (e. g., billing, group management, shift schedules).

Figure 1 provides an example of a child care app. During our research, some applications also added a health monitoring feature to reduce the likelihood of a COVID-19 outbreak by e. g., logging the temperature of children multiple times during the day and sharing it with parents via the application.

Notably, most applications do not offer a registration form or a "Sign up" button. The child care center must enroll and buy a subscription and invite parents to join the respective group(s). A monthly subscription usually starts at about $15.00 and increases with the number of children. Some applications offer a demo button to explore their features.

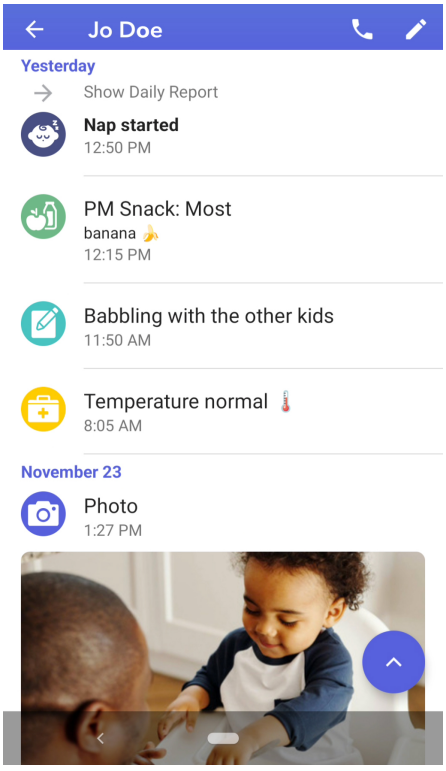

Fig. 1. Screenshot of the activities feature in the *brightwheel* child care app. The staff can document activities such as nap and potty times, meals, incidents, attendance, medications, and health checks, share photos/videos, give praise, or create custom notes.

2.2 Regulatory Requirements

Children are specially protected by regulatory bodies because they are legally not allowed to decide certain things for themselves. Therefore, parents can inadvertently compromise the privacy of children and other adults by sharing information on the Web [58]. To better protect children, regulators developed restrictive privacy laws. Two notable laws are the *US Children Online Privacy Protection Act* (COPPA) and the *General Data Protection Regulation* (GDPR) by the European Union.

If the data is used for marketing or user profiling, the collection and processing of minors' data are explicitly prohibited by both regulations. Moreover, both regulations require developers and companies to follow security best practices and mandate to disclose any third-party data collection happening, such as the use of third-party tracking services. In the following, we describe the two major regulatory frameworks that aim to protect minors from disseminating their data on the Internet.

2.2.1 COPPA

COPPA regulates how online services must handle data gathered from children under 13 years. COPPA is a US-only regulation, so it only applies to US citizens and to platforms in the US. It states that online services, mobile applications, games, and other services can only gather data after obtaining the explicit and verifiable consent of parents or legal guardians [22]. For example, disclosing information via payment systems or phone calls requires parental consent. The *Federal Trade Commission* (FTC), which protects consumer rights in the US, offers a six-step plan for data processors to verify compliance [23].

2.2.2 GDPR

The GDPR changes the way data processors deal with data for residents in the European Union. GDPR is one of the first regulations that exports the marketplace principle. It is applicable for every European whose data is processed on digital platforms no matter where it is used. Different works analyzed the impact of the GDPR on the online ecosystem [51, 61, 62, 83, 84]. Furthermore, it specifies strict requirements on the lawfulness of processing child-related data in Art. 6 (1) f [17]. If a processor collects data, transparency and consent are enforced before the information is collected [86]. Art. 8 GDPR [55] and Recital 38 GDPR [18] force data processors to require consent from a parent or legal guardian if data is processed or collected about an under 16-year-old person. Under Art. 8(2) GDPR, the controller is also required to make "*reasonable efforts*" to verify that consent has been given or authorized by the holder of parental responsibility regarding available technology. In the case of child care applications, data about the child is indirectly processed, so Art. 8 should be stated in the privacy policies, and parents should be made aware that their children's data is processed.

2.3 Analyzing Mobile Apps

Researchers often analyze mobile applications in static, dynamic, or hybrid settings [67, 81]. For example, in the past, they used static analysis to identify data leaks [43, 75], detect malware [25, 56, 81], or identify several other security and privacy issues [44]. *Static analysis* aims to analyze an application by extracting and analyzing features extracted from the applications, on

Android called an apk, i. e., the source code of the application. While *dynamic analysis* can be used for similar purposes [10, 30, 70], it focuses on the actual communication of the application at runtime, as well as dynamically loaded code and content by the application [72].

3 Related Work

This section provides an overview of common privacy issues in childcare and describes related work on privacy issues of parental control apps.

3.1 Child Care and Children's Privacy

Outside of the digital space, privacy is an integral factor for the well-being and development of children. Zeegers et al. found that 58 of 100 three to five-year-old children said they had a special place at the daycare center that belongs only to them [93]. child care spaces are designed to offer privacy and a child-like environment [52, 82]. This is because privacy is essential in child care centers and kindergartens [26, 40]. There is various work that focuses on privacy aspects of different types of devices [11, 87], and applications used by children [2, 21, 54]. When it comes to the perception of online privacy risks, children understand what privacy is and why they need it [41, 95]. Different works tried to define a framework and guideline to protect children's privacy online [36, 47]. In the mobile application space, Meyer et al. analyzed the advertisement behavior of applications for children [54]. Reyes et al. examined the 5,855 most popular free children's apps. They found that most of the analyzed apps potentially violate COPPA and that 19% of the applications send personally identifiable information over the network [77]. We extend this area of research by analyzing the privacy and security of online environments that companies do not directly design for kids but process children's data nonetheless.

3.2 Parental Control Apps

Similar to our analysis, previous work studied privacy aspects of so-called parental control applications. Feal et al. analyzed Android parental control apps that can monitor and limit mobile app usage e. g., gaming, web browsing, or texting [21]. They found that such applications lack transparency and compliance with regulatory requirements. Ali et al. found that pervasive security and privacy issues are prevalent in most parental control applications and concluded that they seem to aid cyberbullying and child maltreatment directly [2]. Our work differs by focusing on the security and privacy aspects of child care applications, which children do not use but store and process much of their data.

4 Method

This section provides an overview of the threat model, the applications we analyze, the used analysis framework, and its static and dynamic components.

4.1 Threat Model

This work starts with the premise of four different threat models. In all models, the adversary aims to exfiltrate personal data via different means. The models are: (1) a person that has (unauthorized) access to the phone and tries to access sensitive data processed or stored by the application, without proper permissions to do so; (2) a benign user of the app that attempts to access data of other users illicitly; (3) an adversary aiming to access or manipulate data transited between the application and the server ("man-in-the-middle attack"); and (4) an attacker that tries to access sensitive resources stored and processed on the server (e. g., a database or file storage). These models exclude malicious app providers, developers, or other external entities. We assume that the service providers, the used third-party services, and their respective employees do not have any malicious intentions. However, using a third-party library that collects children's data might unintentionally break GDPR or COPPA compliance. Furthermore, developers might unknowingly implement features in an insecure or incompliant way.

4.2 Application Selection

We aim to analyze the privacy and security attributes of mobile child care applications used by parents and child care workers for documentation, communication, and administration. We limit ourselves to Android applications present in the *Google Play Store*. However, all of the analyzed apps were present in other stores, and we assume that results are comparable – at least to some extent. The first step in our experiment is to select and download the applications, including all metadata of interest.

Since child care apps are just becoming popular, there is no distinct category on the Play Store, and no clear market leaders exist. Hence, to find them, we used different search queries (e. g., "child care app" or "preschool management") and took applications into account that Google recommended. We analyzed all identified applications ($n = 46$) manually if they support at least the following criteria: (1) the app has to be intended for the child care sector (e. g., we excluded games in which one takes care of a child), (2) they should facilitate communication between child care center workers and parents, and (3) they provide a management interface for the children and store individual data related to a child. Overall, we identified 42 applications that fit our criteria (the complete list can be found in Section 5.1).

4.3 Analysis Framework

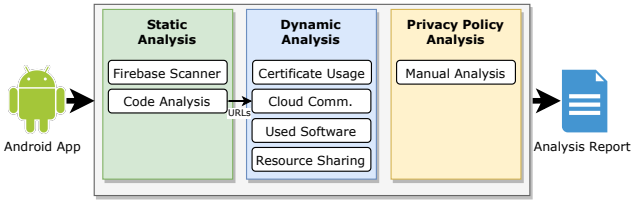

Fig. 2. Analysis pipeline used for the child care applications.

To analyze the identified apps, we use a framework compounded by different existing analysis tools. Generally, we analyze the applications utilizing three general mechanisms: (1) static analysis, (2) dynamic analysis, and (3) a manual analysis of the privacy policies. We rely on existing tools that provide a rich set of information on each application for each of these. The framework works as follows: an application of interest is analyzed by each tool automatically, and the results are combined into a single report. Only the analysis of the privacy policies is done manually to avoid any (potential) misinterpretation. Figure 2 summarizes our approach. In the following, we describe each analysis step in detail.

4.4 Static Analysis

Our static analysis approach uses three components: (1) a *Firebase* scanner, (2) a tool to perform static code analysis, and (3) a combination of tools to analyze potential issues with cloud communication.

4.4.1 Firebase Analysis

FireBase [28] is a development platform provided by Google that supports developers in building and running mobile applications. Due to its rich feature set, Firebase is often used and popular among developers [15]. Applications often suffer from vulnerabilities due to wrongly configured instances of the Firebase service [34, 80]. Hence, we utilize software called *FireBase Scanner* [78] to analyze whether an app uses an insecure Firebase instance. The scanner checks if an app uses Firebase and verifies if the instance can be accessed without authentication, i. e., if the database (the *.json* file) of the Firebase instance can be downloaded using an HTTP GET request.

4.4.2 Static Code Analysis

To better understand the inner workings of an application, we use the *Mobile Security Framework* (MoBSF) [1], an all-in-one mobile application security assessment tool. *MoBSF* is a framework to statically and dynamically analyze Android and iOS applications. In our static analysis, we use the framework to inspect: (1) permissions, (2) libraries, and (3) *Software Development Kit* (SDK) levels used by an application. The framework also provides a list of known tracking codes, prints all encountered URLs, and identifies any potential hard-coded secrets (e. g., passwords or keys) present in the code.

Permissions. By default, the framework prints all permissions an app needs and provides an internal (heuristically) classification on the critically of them. We utilize the permissions to understand which data an application can access (e. g., email address or name) and which system resources (e. g., camera) are available. This allows us to cross-compare different apps and assess their impact on users' privacy.

Libraries. The third-party libraries used by an app provide insights into parts of an application's functionality (e. g., an included ad library hints that the app might show ads). MoBSF provides a list of used libraries and checks if they were misconfigured. However, some libraries actively protect themselves (e. g., via obfuscation) from a deeper analysis. Furthermore, we use LibRadar++ [48, 49] to automatically detect third-party libraries and the permissions that they add to the appli-

cation. We manually analyze the output of LibRadar++ and sanitize it to improve the clarity of the results.

Android SDK API Levels. In the final step of the MoBSF analysis, we look at the supported SDK versions of an application. This information is helpful because outdated Android versions might pose additional security risks due to known exploitable vulnerabilities in the operating system.

4.5 Dynamic Analysis

The second pillar of or analysis framework is the dynamic analysis component (cf. Figure 2). Within the dynamic analysis, we focus on three parts: (1) certificate usage, (2) used software, and (3) resource sharing. We rely on a rooted Android smartphone running Android Oreo (8.1) to conduct the analysis. We instrumented the device to record all network traffic utilizing the HTTP man-in-the-middle proxy *Burp Suite Professional* [71]. The Burp Suite root certificate was stored in the system's certificate store for this purpose. In this way, the tool can create trusted certificates for all requested domains. All tested apps were initially tested without valid credentials. We created accounts in different roles (e. g., "parent" or "educator") to test the 42 apps. Experienced security researchers analyzed each app manually. They performed *non-invasive* manual tests described in the following sections. Where possible, we created accounts in different roles (e. g., "parent" or "educator") and interacted with each of the 42 applications for 25–30 minutes by performing typical user actions (e. g., reading and writing messages, sharing photos, and viewing reports on meals, naps, and other activities). If we were able to register as an educator, we performed typical actions like creating new groups, adding and checking-in children, recording activities, creating announcements, and scheduling events. We chose not to perform any *invasive* tests, e. g., performing Denial of Service attacks, to ensure the safe and secure operation of the services.

4.5.1 Certificate Usage

The correct usage and validation of certificates are essential to ensure that it transmits all data confidentially and trustworthy. Hence, missing or lax certification validation (e. g., accepting all certificates [9]) might void the anticipated security guarantees of the services that rely on them. A common way to ensure that a TLS connection uses only the desired certificates is *certificate pinning*, which ensures that the client only accepts a set (or one) of previously known certificates for a TLS connection. For this, the application stores the known certificates' hash values locally and checks if the presented certificates' hashes match these values. If an app uses this mechanism, it will reject all certificates issued by our HTTP proxy because the hash values of the certificates do not correspond to the ones stored in the application. Apps that do not rely on the mechanism will accept any certificates trusted by the operating system. Hence, we can test which apps utilize certificate pinning in our dynamic analysis. More precisely, if an application does not establish a TLS connection to an endpoint despite the system recognizing the certificate to be trusted, we consider that the application uses certificate pinning correctly. Our approach does not try to circumvent certificate pinning, e. g., by trying to dynamically patch the application. Thus, our setup cannot break any application. Instead, we only intercept the traffic by replacing a certificate and performing a man-in-the-middle attack. However, in two cases, certificate pinning was enabled that prevented our attack. As we decided not to patch the applications, we were not able to look into the network traffic in these two cases.

4.5.2 Deployed Software Components

An essential step to creating secure software is always to use up-to-date components (e. g., libraries). Furthermore, no components should be used that suffer from known vulnerabilities. Hence, we scan if an application uses outdated and vulnerable (third-party) software. We use the results from MoBSF and check the versions ourselves for outdated versions. Additionally, we analyzed the responses of all observed HTTP communication when possible. The returned headers can provide hints on the systems and software used in the backend of the app or library. Finally, we logged the IP addresses of communication endpoints based on DNS name resolution. These IP addresses were then examined for known vulnerabilities using the *Shodan Search Engine* [50]. We consider a backend system outdated or vulnerable if a known vulnerability for a deployed software version could be identified in either the returned headers or the Shodan search. We did not evaluate if any identified vulnerability was exploitable for ethical reasons. Hence, our results have to be interpreted as an upper bound.

4.5.3 Resource Sharing

Cross-Origin Resource Sharing (CORS) is an HTTP mechanism that defines how an application can access resources from another origin. It is used to define specific exceptions to the *Same-Origin Policy* (SOP). A misconfigured cross-origin resource sharing policy allows an attacker to send an arbitrary origin header in an HTTP request to the server and receive an access-control-allow-origin header with the origin domain sent in response. In order to determine the extent to which the backends of the apps examined are vulnerable, we sent a request with a manipulated origin header to the login function in each case. It was considered a vulnerability if the URL sent in the Origin header was returned in the Access-Control-Allow-Origin header. Such behavior allows to send requests to an API from arbitrary websites and read the server's response. This can be used to bypass the security guarantees provided by the same-origin policy. For example, a misconfigured SOP can lead to the leakage of API keys compromising the whole integrity of an app or leakage of user data.

4.5.4 Injection and Access Control Vulnerabilities

Finally, we examine if the analyzed applications suffer from two common vulnerability families [68]: (1) broken access control and (2) injection vulnerabilities. Starting with the latter, we check the login mask of the application for these types of issues. For ethical reasons, we only use a simple test (e. g., `' OR 1=1--`) on a single instance (i. e., the login form) to avoid the risk of disrupting the application's backend in any meaningful way. Additionally, we evaluate how the applications implement their access control. To do so, after logging in, we capture a valid 'authorized' request that contains some authentication token (e. g., a cookie). We then resend the request without the token. If the server accepts and processes the request, we assume that the implemented access control is broken. Finally, we check if we can access data from other users by direct object reference. If object referencing was implemented using numeric IDs, we check to access the following and previous ID. If we could access data in this fashion, we assume that the application is vulnerable to such attacks. Again for ethical reasons, we did not investigate further than this. The described approach does not exhaustively test if an application is vulnerable but instead checks if the developers are aware of the potential security issue and have implemented some protection mechanism.

4.5.5 Cloud Communication

In the last step of our dynamic analysis, we look at an app's communication, which includes first-party (e. g., the server endpoint of the application) and third-party communication (e. g., a web service to store data like an *Amazon S3* bucket). To do so, we resort to the following two open-source tools: *sslscan* [74] and *cloud_enum* [37].

sslscan. This tool queries any TLS service to analyze the used protocol versions, cipher suites, and certificates. Our analysis uses this to understand if applications use invalid or outdated security parameters. Naturally, the tool needs an endpoint (i. e., an URL) to test. As previously mentioned, MoBSF provides a list of identified hard-coded URLs. We use them as input in this step. Thus, we can analyze which TLS protocol version is active, whether TLS fallback and compression are used, and if the application uses any insecure ciphers.

cloud_enum. This tool tries to find any publicly accessible cloud storage resources at *Amazon Web Services* (AWS), *Microsoft Azure*, and *Google Cloud Platform* based on given keywords. To compile a list of keywords for a given application, we use the name of the application and domain names (eTLD+1) that we found in the code that is directly related to the application. We check all identified keywords manually before starting the scan to ensure that they are related to the given application. Publicly accessible cloud storage resources are a severe privacy problem. Given the correct URL, anyone can access all data in a storage bucket and exfiltrate all – potentially sensitive – data.

4.6 Privacy Policies

In the last step, we manually inspect all privacy policies of the analyzed apps. While the apps are not directly intended to be used by children, they most certainly will process data related to children where specific consent must be given by the legal guardian and special lawfulness requirements must be met e. g., Art. 8 GDPR or Art. 6 (1) f. Hence, we are interested in how application providers aim to protect this vulnerable group. For the analysis, we use the linked policies in the Google Play store to provide a working link. The analysis focuses on how data related to children is collected and processed. More specifically, we look at how each company aims to comply with their respective legislation (i. e., COPPA and the GDPR—Section 2).

One challenge we face with our policy corpus is that it is multilingual, as 12 (29%) policies are not available in English. Thus, we follow a manual analysis approach, which is still practical given the size of our dataset. In addition, we argue that our approach is less error-prone than using automated frameworks such as Polisis [33], PolicyLint [4], or PoliCheck [5] that require a trained NLP model for each language and use case.

5 Results

In this section, we describe the applications (Section 5.1), then we present their privacy issues (Section 5.2), and finally, we highlight some security problems (Section 5.3).

5.1 Analyzed Applications

To get a better overview of the analyzed apps (cf. Table 1), we looked at the metadata provided in the Google Play Store. One meaningful indicator is the number of installations per app, which one can use to determine the application's popularity. The applications in our corpus show a wide range of installation numbers. Two (5%) apps have more than one million installations, 11 (26%) more than 100,000 installations, and 29 (69%) of them have less than 100,000 installations. Compared to other application categories (e. g., communication or games [38, 42]), the analyzed apps are less popular, which is probably because the application family is relatively young. For example, the two most significant services *brightwheel* and *Bloomz* was founded around 2014 and secured their first significant funding in 2016 [24, 39]. If combined, the analyzed apps are still used by more than three million users. Looking at the rating of the apps, we see an average rating (★) of 4.0 out of five stars.

5.2 Privacy Implications

In the first part of our analysis, we assess the privacy implications based on the trackers present in the analyzed applications and elaborate on the requested permissions. Furthermore, we analyze the applications' privacy policies to understand how they comply with current legislation and their transparency on third-party usage.

5.2.1 Third-Party Libraries

The LibRadar++ tool was used to statically scan the 42 apps analyzed for third-party libraries. The results can be found in Table 1. Three (7%) of the apps examined are XAPKs, which are not supported by LibRadar++. They were therefore not evaluated further for their third-party library usage. Apart from this, we can confirm that the remaining 39 applications contain at least 214 unique third-party libraries. However, LibRadar++ could not classify 103 libraries because of the application's code-obfuscation techniques. Thus, we were forced to exclude them from further analysis and analyze the remaining 111 libraries. In total, we found 997 occurrences of third-party libraries. 643 (64%) of these libraries are tools for development and functionality e. g., JSON parsers, Google services, or QR code processing. We also identified 213 (21%) libraries that are used by social networks e. g., Facebook and Instagram. Following this, 107 (11%) third-party libraries offer analytics, e. g., *CrashLytics*, *Google Analytics*, or *Amazon* in-app purchasing. Further 34 (3%) libraries are used for advertising. These findings are particularly concerning from a privacy point of view, because of the business model behind them [73]. For example, the applications *OWNA*, *Kangarootime*, and *LiveKid* implement and contact the *OneSignal* tracker. They all send data to this analytics platform that "*crafts unique messages based on your users in-app and real-world behaviors*," which also states that "*COPPA is the responsibility of the publisher to maintain*" [66].

5.2.2 Tracking Libraries

Across the 42 applications, we identified 27 distinct tracking libraries. Following their subscription-based business model (Section 2), non of the analyzed applications include libraries used to serve ads, which is desirable from a privacy perspective. However, not only ad libraries collect personal data. We observe a long-tailed distribution with some popular trackers and several used by only one application (combined as "Other").

There are apps with no trackers and apps with up to seven trackers. Unsurprisingly, the most popular tracking libraries are provided by Google. The most used tracker is the Google Firebase Analytics tracker, with 40 out of 42 possible occurrences. This is most likely since the majority of apps make use of a Firebase instance. The second most used tracker is Google CrashLytics (18 apps). Leaving aside these two frequently occurring

App	Category	Installs	★	3P Libs.	Dangerous	Trackers	Lowest SDK	Pinning	TLS	Vulnerabilities
1Core Family	Prod.	1,000+	4.2	25	4	3	Level 21 (A5.0)	○	TLSv1.0	
Bloomz	Education	1,000,000+	5.0	21	5	3	Level 16 (A4.1)	○	TLSv1.0	
brightwheel	Education	1,000,000+	4.8	81	3	6	Level 16 (A4.1)	○	TLSv1.0	
CARE Kita App	Parenting	10,000+	4.0	XAPK	2	1	Level 21 (A5.0)	○	TLSv1.0	Cloud, GraphQLi
Cheqdin	Education	1,000+	3.0	10	3	2	Level 16 (A4.1)	○	TLSv1.0	
Child Journal	Education	5,000+	4.8	20	4	3	Level 17 (A4.2)	○	TLSv1.0	
Daily Connect	Education	50,000+	5.0	XAPK	5	1	Level 21 (A5.0)	○	TLSv1.0	
Educa Touch	Education	50,000+	1.8	18	4	2	Level 21 (A5.0)	○	TLSv1.0	Cloud
Famly	Social	100,000+	3.8	10	6	1	Level 22 (A5.1)	○	TLSv1.0	
HiMama	Education	100,000+	5.0	19	3	3	Level 27 (A8.1)	○	TLSv1.0	
HOKITA-Eltern	Comm.	500+	4.8	9	7	1	Level 21 (A5.0)	○	TLSv1.2	
Illumine	Education	5,000+	4.4	19	10	3	Level 21 (A5.0)	○	TLSv1.0	
Isy Kita	Education	1,000+	4.7	6	3	3	Level 21 (A5.0)	○	TLSv1.0	
Kangarootime Parent	Education	10,000+	4.1	54	3	5	Level 16 (A4.1)	○	TLSv1.0	
Kaymbu	Education	5,000+	4.5	46	4	4	Level 16 (A4.1)	○	TLSv1.0	
Kidling Kita-App	Parenting	500+	5.0	12	3	2	Level 22 (A5.1)	○	TLSv1.0	
KidReports	Lifestyle	100,000+	1.9	9	3	2	Level 21 (A5.0)	○	TLSv1.0	
KigaRoo für Eltern	Parenting	5,000+	5.0	5	0	1	Level 22 (A5.1)	○	TLSv1.0	
KiKom Kita App	Parenting	10,000+	4.4	4	5	0	Level 22 (A5.1)	○	TLSv1.0	
Kinderly Together	Education	500+	4.8	26	3	1	Level 21 (A5.0)	○	TLSv1.0	
KinderPass	Education	1,000+	4.6	46	5	2	Level 22 (A5.1)	○	TLSv1.0	
Kindy – Die Kita-App	Parenting	5,000+	4.0	8	3	4	Level 19 (A4.4)	○	TLSv1.0	
Kita-Info-App	News Mag.	100,000+	3.3	1	2	1	Level 21 (A5.0)	○	TLSv1.0	
Kitaportfolio	Comm.	500+	4.8	14	2	2	Level 19 (A4.4)	○	TLSv1.0	
Leandoo Eltern	Comm.	10,000+	3.9	6	4	1	Level 23 (A6.0)	○	TLSv1.0	
LifeCubby Family	Education	10,000+	3.1	32	5	3	Level 26 (A8.0)	○	TLSv1.0	SQLi
LittleLives	Education	100,000+	3.3	31	8	7	Level 21 (A5.0)	○	TLSv1.0	
LiveKid	Education	100,000+	5.0	XAPK	7	4	Level 16 (A4.1)	○	TLSv1.0	
nemBørn	Social	1,000+	2.3	43	5	2	Level 19 (A4.4)	○	TLSv1.0	Cloud
OWNA Childcare App	Education	10,000+	4.3	7	6	4	Level 23 (A6.0)	○	SSLv3	Cloud
Parent: Child Care App	Social	10,000+	1.7	75	5	3	Level 21 (A5.0)	○	TLSv1.0	
Parent Portal	Parenting	10,000+	4.7	3	2	0	Level 22 (A5.1)	○	TLSv1.0	Cloud
ParentZone	Parenting	50,000+	5.0	24	4	3	Level 21 (A5.0)	○	TLSv1.0	
PREto3	Education	1,000+	4.7	86	4	2	Level 21 (A5.0)	○	TLSv1.0	
Procare: Childcare App	Education	100,000+	5.0	77	5	4	Level 21 (A5.0)	○	TLSv1.0	
Sandbox Parent App	Parenting	10,000+	2.9	6	2	1	Level 19 (A4.4)	●	TLSv1.0	
Sdui	Education	100,000+	4.1	3	1	1	Level 21 (A5.0)	○	TLSv1.0	IDOR
Smartcare for Parents	Education	100,000+	2.5	9	3	2	Level 21 (A5.0)	○	TLSv1.0	
Storypark for Families	Education	100,000+	3.0	37	3	4	Level 21 (A5.0)	○	TLSv1.0	
Stramplerbande	Social	5,000+	2.8	27	1	1	Level 21 (A5.0)	○	TLSv1.0	IDOR
Tadpoles Parents	Education	100,000+	5.0	28	4	7	Level 22 (A5.1)	○	TLSv1.0	
Zaycare	Business	10+	4.8	40	3	2	Level 21 (A5.0)	●	TLSv1.0	

Table 1. Application analysis results: The first four columns show general information about the applications, such as the reported installation numbers and ratings based on the Google Play Store as of November 2021 (cf. Section 5.1). The following three columns summarize the privacy-related findings such as the number of identified dangerous permissions and trackers (cf. Section 5.2). Three apps are XAPKs, which cannot be analyzed for third-party libraries (3P Libs.) using LibRadar++. The last four columns show the security-related issues we identified in the applications (cf. Section 5.3).

trackers, we notice that 19 of the 42 apps (45%) use more than two trackers. We found the PII using our dynamic network analysis tools (i. e., Burp Suite, MoBSF). To identify the PII, we searched for 'plain' text occurrences of the information in the (decrypted) network traffic. However, we did not check for hashes or other obfuscation methods (e. g., different encoding or HTTP compression standards). Thus, our results can be seen as a lower bound. One particularly concerning example that we found was from *LiveKid*, which shares user data directly as a *Slack* message with the developers.

Our analysis revealed that the data that is shared via these third-party tracking services includes:

1. **User Data**: User ids, types, e. g., "teacher," phone number, email address, username, advertising ids.
2. **User Interaction**: Session duration, button clicks e. g., "screen attendance students".
3. **Device Data**: Manufacturer, model, OS, API level, IP address, screen, battery, cellular carrier, free memory/disk, language, timezone, orientation, etc.
4. **App Data**: Namespace, version, build data, type, package names, libraries.

5.2.3 Requested Permissions

Permissions of an application are implemented using a permission model which restricts or grants access to personal data (e. g., photos or contacts) in Android. Overall, we observed 96 distinct permissions used by the analyzed applications. Of those permission 49 (51%) are general permissions provided by Android with the remaining 47 (49%) are either app-specific (e. g., `com.famly.famly.permission.C2D_MESSAGE`) or defined by the device manufacture (e. g., `com.htc.launcher.permission.UPDATE_SHORTCUT`). Moreover, we find a wide range of used permissions for the apps. On average, an app uses 19 permissions (Min: 3; Max: 40; SD: 9.7). 30 (71%) of the examined apps can access all files on the device or an SD memory card. 28 (66%) apps request permission to take photos and videos. 14 (33%) of the examined apps can record sound by requesting access to the microphone. One app uses permission that allows access to the user's contacts, five (11%) can read and write to the calendar. 41 (97%) apps use the permission for push notifications. three (7%) apps can set a timer, and five (11%) can make phone calls. Critically from a security perspective, four (9%) apps can download and install additional software.

Definition of Different Permission Types. We distinguish between four general types of permissions, following the definitions made by Google [27] and the work of Feal et al. [21]:

1. **Normal:** Provide limited risk to other applications, the system, or the users. Granted to the app at the time of installation without prompting the user.
2. **Signature:** Enables communication between multiple apps of the same developer. Only granted if the requesting app is signed with the same certificate.
3. **Dangerous:** Grants the application (1) access to personal user data (e. g., user's location) or (2) control over the user's device. Only granted after explicit consent from the user.
4. **Proprietary:** Defined by an application itself or proprietary to a device manufacture.

Figure 3 in Appendix A provides an overview of the requested permissions. In the following, we will take a closer look at the identified dangerous permissions.

Requested Dangerous Permissions. During the analysis of the dangerous permission, it was evaluated which permissions are frequently used together. It was noticed that all apps that use the permission `android.permission.READ_EXTERNAL_STORAGE` also set write access with the permission `android.permission.WRITE_EXTERNAL_STORAGE`. A total of 71% of the apps requested read and write access. The situation is similar for audio recording permissions, which are also combined with read (28%) and write (33%) permissions. These permissions show that they often occur together and are functionally related. Overall, 92% of the examined apps use the permission to write to external storage. The apps use these permissions to exchange data with parents and other parties. It is more noticeable that one of the examined apps requires access to the smartphone's contact book. Another dangerous permission used by three (7%) of the apps enables the installation of additional software. The permission to read the stored access data used by one of the examined apps is also questionable.

Rarely Used Permissions. We define rarely used permissions as permissions that are used by less than five apps. Of the 96 permissions identified in the apps examined, 62 could be classified as Rarely Used Permission. This implies that less than five apps use 65% of the identified permissions. Of the 62 rarely used permissions, 33 (53%) are Android permissions, 24 (39%) are app-specific permissions, two (3%) are mobile manufacturer permissions, and three (5%) are miscellaneous permissions that cannot be categorized automatically. To find out when the permissions were introduced, the official Android documentation can be used as the API level is indicated within the documentation. For 13 (39%) of the 33 permissions, API Level 1 is indicated, which means these permissions have existed since the initial release.

Additionally, eight (24%) of the official Android permissions are not listed in the Android permission documentation. In the case of the app-specific permissions, the user never accepts or rejects them because they automatically accept them. There is no warning or consent popup for these custom permissions. Judging by the name of these permissions, 24 (39%) of the permissions have probably been declared by the app developers. 13 (54%) of those permissions want access to `permission.C2D_Message`, which has been deprecated since June 2012 and shut down as of July 2015. Seven (29%) of the app-specific permissions are related to push notifications of particular providers such as *Amazon Device Messaging* or *Meizu by Tencent Cloud*. Four (17%) of the found app-specific permissions could not be categorized.

5.2.4 Privacy Policies

Next, we analyzed the privacy policies of the evaluated child care apps. When analyzing the policies of the individual apps, we verified whether the policies explicitly mention the protection of the children's data. Table 2 shows the results of our multidimensional analysis.

General Policy Overview. In total, we analyzed 42 privacy policies, one for each app. The vendors of the applications originate from 12 countries ("Origin"). 14 (33%) apps are developed in the US and 12 (29%) in Germany. Unfortunately, 12 (29%) of the analyzed policies are not available in English, which is why we resorted to manual analysis of the policies (cf. Section 4.6). 34 (81%) apps are available in both tested Google Play Stores (USA and EU). The dates when the privacy policies were last updated range from 2013 to 2021 (i. e., before the introduction of the GDPR, which came into effect in 2016). Six (14%) of the privacy policies do not specify the exact day, month, or year they came into effect. 24 (57%) companies do not name a Data Protection Officer ("DPO") in their policies. Companies must appoint a DPO according to the GDPR if a company processes sensitive data or systematically monitors individuals at large scale [19].

Data Handling and Tracking. According to the GDPR, every person under the age of 16 years is considered a child. It is then essential that the child is informed about the processing of their data or that the holders of parental responsibility have consented or agreed to the processing of their child's data. Since the GDPR only applies in Europe, we also looked at the Children's Online Privacy Protection Act (COPPA) in our analysis. This act states that all persons under the age of 13 years are children. First, we checked whether the linked privacy policy from the app store is applicable ("Applicable"). In 31 (74%) cases, the policy refers to the mobile child care app. However, in the other 11 (26%) cases, the policy is about another component, such as, a website or a more general policy of the company (i. e., referring to job applicants and their payroll and health insurance data). Thus, they are not tailored for the specific use case and are partly concerned with subjects not commonly associated with mobile applications (e. g., usage of cookies).

In terms of processing sensitive data of children i. e., minors, under the age of 13 years ("Processing of Children Data"), the results show that 18 (43%) apps do not mention such processing at all. Thus, it seems that the apps make no difference between children's data and other data, which can be a violation of COPPA and GDPR. Two (5%) of the analyzed apps claim that they are not responsible for such processing. This only outsources the problem to the child care center, which is then required to provide a privacy policy and obtain consent to process the data. Only roughly half of the services in our corpus, 22 (52%), mention that they have protection measures in place to process data of minors.

According to our findings, 40 (95%) apps in our corpus use some (third-party) tracking service. The vendors should mention the usage of trackers in their privacy policies if they utilize them. Thus, we checked whether the policies mentioned trackers or listed the third-party tracking services. Concerningly, we find that only 13 app vendors (31%) mention the use of trackers in their respective policy. Hence, the vast majority, 29 (69%) of the policies, lack this information.

Transparency on Data Processing. For companies that want to be transparent about their data processing practices, it is inevitable to name the data their application collects. Of the analyzed apps, 28 (67%) include a (partial) list of which data is obtained, stored, and processed ("Data Stored"). Concerningly, 13 companies do not specify the data they collect. Thus, the majority of apps transparently discuss the sensitive data they use to run their service, e. g., *Daily Connect* states in their policy: "*For example, we may share aggregated or de-identified information* [...]*, such as calculating the average number of diaper change*[s] *per day* [...]").

Another critical aspect to understanding how a company uses personal data is to list the third parties that access the collected information (i. e., sharing of data). 16 (38%) companies list the used third parties transparently in their policy, and ten (24%) of the companies claim that they do not share any data with third parties. However, we found that seven out of these ten applications do not comply with their privacy policy by sharing data with third parties even though their policy states they do not (cf. Section 5.2.1).

Finally, the retention time of the collected data and how users can access such data needs to be specified. Starting with the latter, if companies provide ways for users to access their data, 18 providers (43%) rely on email. Often, this is a manual, slow, and unsuccessful process [83]. Only one company provides a web form to process such requests. 23 (55%) companies do not provide any information on subject access requests. Regarding data retention, 16 (38%) companies do not specify any timeframes. However, one-third of the analyzed applications specify that they are aware of the legal obliga-

App	Origin	Play Store	Version	English	Applicable	Processing of Children Data	Tracking Mentioned	Data Stored	Data Sharing	Data Retention	Data Access	DPO
1Core Family	USA	USA	unsp.	●	○	○	●	unsp.	yes, but no list	unsp.	unsp.	○
Bloomz	USA	Both	Aug. 21	●	●	●	●	listed	yes, listed	legal obligation	unsp.	●
brightwheel	USA	Both	Feb. 15	●	●	○	●	unsp.	yes, but no list	unsp.	Email	○
CARE Kita App	DEU	Both	unsp.	○	●	Not resp.	○	unsp.	no	unsp.	unsp.	●
Cheqdin	GBR	Both	Mar. 20	●	●	○	●	listed	yes, listed	custom	Email	○
Child Journal	USA	Both	Nov. 19	●	●	●	○	listed	no	custom	Webform	○
Daily Connect	USA	Both	unsp.	●	●	●	●	listed	yes, listed	custom	unsp.	○
Educa Touch	NZL	Both	Jan. 21	●	●	●	●	listed	no	legal obligation	Email	○
Famly	DNK	Both	Jan. 21	●	●	○	●	listed	yes, but no list	custom	Email	●
HiMama	CAN	Both	unsp.	●	●	○	●	listed	no	unsp.	Email	○
HOKITA-Eltern	DEU	EU	May 18	○	○	○	○	N/A	N/A	N/A	N/A	●
Illumine	IND	Both	Aug. 21	●	●	●	●	listed	yes, listed	legal obligation	Email	●
Isy Kita	DEU	EU	unsp.	○	●	●	●	listed	no	custom	Email	○
Kangarootime Parent	USA	USA	Jul. 20	●	●	●	○	unsp.	unsp.	unsp.	unsp.	●
Kaymbu	USA	Both	Oct. 13	●	●	●	●	listed	no	unsp.	Email	●
Kidling Kita-App	DEU	Both	Aug. 21	●	●	●	○	listed	yes, but no list	legal obligation	unsp.	●
KidReports	USA	Both	Sep. 18	●	○	○	●	unsp.	unsp.	unsp.	unsp.	●
KigaRoo für Eltern	DEU	Both	Nov. 21	○	○	○	●	listed	yes, listed	custom	unsp.	●
KiKom Kita App	DEU	EU	Jun. 21	○	●	○	N/A	listed	yes, but no list	legal obligation	unsp.	○
Kinderly Together	GBR	Both	May 18	●	●	●	●	listed	yes, listed	legal obligation	unsp.	○
KinderPass	ARE	Both	Aug. 21	●	●	●	●	listed	yes, listed	custom	Email	○
Kindy – Die Kita-App	DEU	Both	Aug. 21	○	●	○	●	unsp.	yes, listed	legal obligation	Email	○
Kita-Info-App	DEU	EU	Feb. 21	○	●	●	○	listed	yes, listed	custom	unsp.	○
Kitaportfolio	DEU	Both	May 18	○	○	○	●	listed	yes, but no list	legal obligation	unsp.	●
Leandoo Eltern	DEU	Both	unsp.	○	○	●	●	listed	yes, listed	legal obligation	unsp.	●
LifeCubby Family	USA	Both	Jun. 18	●	●	●	○	listed	unsp.	unsp.	unsp.	○
LittleLives	SGP	Both	Apr. 20	●	●	●	●	listed	yes, listed	custom	Email	●
LiveKid	CHE	Both	Sep. 21	○	●	●	●	listed	yes, listed	unsp.	unsp.	○
nemBørn	DNK	Both	Jan. 21	●	●	○	●	unsp.	unsp.	unsp.	Email	○
OWNA Childcare App	AUS	Both	Aug. 21	●	●	●	●	unsp.	yes, listed	unsp.	Email	●
Parent: Child Care App	DNK	Both	Dec. 21	●	●	●	●	listed	yes, listed	unsp.	Email	●
Parent Portal	GBR	Both	Mar. 21	●	○	○	N/A	unsp.	no	legal obligation	unsp.	○
ParentZone	GBR	Both	Dec. 18	●	○	○	○	listed	yes, but no list	unsp.	unsp.	○
PREto3	USA	Both	Jun. 19	●	●	●	○	unsp.	yes, but no list	custom	unsp.	○
Procare: Childcare App	USA	Both	Sep. 18	●	○	○	●	unsp.	unsp.	unsp.	unsp.	●
Sandbox Parent App	USA	USA	Jul. 18	●	○	○	○	listed	yes, but no list	unsp.	unsp.	○
Sdui	DEU	Both	Dec. 21	●	●	Not resp.	○	unsp.	unsp.	legal obligation	unsp.	○
Smartcare for Parents	USA	USA	Mar. 21	●	●	●	●	listed	yes, listed	unsp.	Email	○
Storypark for Families	NZL	Both	Dec. 20	●	●	●	●	listed	yes, listed	legal obligation	Email	●
Stramplerbande	DEU	Both	May 21	○	○	○	○	unsp.	unsp.	legal obligation	Email	●
Tadpoles Parents	USA	Both	Sep. 20	●	●	●	○	listed	no	custom	unsp.	○
Zaycare	NLD	Both	Feb. 21	○	●	○	●	listed	yes, but no list	legal obligation	Email	○

Table 2. Privacy policy analysis results: The first six columns offer general information about the policy such as its language and whether it is applicable by referring to the analyzed child care application. The next two columns focus on the requirement to mention the processing of children data and use of trackers. The remaining columns describe other common GDPR-related requirements. All of the results are discussed in detail in Section 5.2.4. ●: Policy fulfills the requirement; ○: Policy does not fulfill the requirement.

tions; however, they do not specify them. Eleven (26%) of the analyzed applications define a custom retention period and state this in their policy.

Consent Mechanisms. A vital tool to transparently inform about data collection and processing practices are consent mechanisms. Of the analyzed apps, 30 (71%) require an invitation code from a child care center to register. For those apps, we could not check if (1) a question asking for consent appears after successful registration or (2) if the consent mechanism happens prior to the installation of the app, e. g., when enrolling at the center. For the remaining 12 apps: seven (17%) do not ask for consent, five (12%) ask for consent, but only four (10%) of them mention tracking in their policy. Moreover, two applications request consent and track users even though their respective privacy policy states they do not share data (cf. Section 5.2.1).

ToS Compliance. Finally, we looked at the respective terms and conditions of the applications and found that 13 (31%) of the analyzed apps did not have any terms and conditions. However, even if they would offer such a document, their usefulness for users to understand how the application is earning money and what services are provided remains questionable [7, 63].

5.3 Security Analysis

We analyze the security of the applications along two dimensions: (1) the attributes of the mobile application itself and (2) the attributes of the backend systems. We

extend our analysis in this direction and report security issues because both COPPA with its six-step compliance plan and GDPR with Article 32 mandate to "implement reasonable procedures to protect the security of kids' information" [18, 23].

5.3.1 Mobile Applications

In the following, we focus on the attributes of the mobile application itself. We report the SDK API levels, which are minimally needed to run an application, the utilized third-party libraries, and further findings. The analysis was conducted between September and October 2021 using the most recent version of the apps.

Android SDK API Levels. First, we take a look at the *target* and *minimum* API level required by the mobile applications; none of the applications defined a *maximum* API level. Most apps were designed for Android 11 (API level 30; 38%) or Android 10 (API level 29; 57%). At the time of writing, Android 12 was the most recent SDK platform but was only released one month before the analysis started. Only one application targeted an Android version that no longer receives security updates (Android 7.1; API level 25). We see a different picture when looking at the minimum supported API levels. In this case, only one application requires an API level that still receives security updates (Android 8.1; API level 27). All other applications (98%) still support versions that are no longer officially supported ("end of life"). For the minimum supported API level (Android 4.1; API level 16), the official support ended nine years ago. From a security perspective, this is very critical. For most of these versions, several exploitable vulnerabilities are known that might lead to the corruption of the entire system and, consequently, the child care application.

Utilized Libraries. As described in Section 4, we use MobSF to conduct a binary analysis of the used third-party libraries of the mobile applications. We could not identify third-party libraries in 16 (38%) of the applications. In total, we identified 468 different libraries in the remaining applications (Max: 161; Min: 0 Mean: 28, SD: 43). The developers of the applications (wrongly) configured 141 (30%) of those libraries in a way that they pose a security risk for the application e. g., *lib/arm64-v8a/libjsc.so* does not enforce complete relocation read-only (RELRO), which could lead to unintentional overwriting of restricted memory. Overall, these risks affect 23 (55%) of the analyzed applications. Similar findings are reported in the related literature [88] and pose a threat to the security of our analyzed applications. These misconfigurations potentially increase the attack surfaces of the applications. However, it is hard for developers to keep third-party libraries up-to-date, which leads to more vulnerabilities in mobile applications [6, 16].

Further Application Findings. We found several other security issues, which we summarize below.

Hard-Coded Secrets: We found not a single hard-coded secret (e. g., passwords or API keys) in the applications. In nine (21%) applications, we found tracking IDs that are not security-critical and are only used to identify an application.

Certificate Pinning: We tested if the mobile applications utilize certificate pinning to prevent man-in-the-middle attacks on the encrypted TLS channel. Overall, two (5%) of the apps used certificate pinning, 40 (95%) of them did not, thus, allowed us to read their encrypted communication with their respective backend.

5.3.2 Application Backends

In this section, we focus on the security properties of the backends of the individual mobile applications. We look at the usage of cloud storage, elaborate on the configuration of TLS, and discuss potential vulnerabilities from which the backends might suffer.

Cloud Storage. During our analysis, we aimed to identify the (cloud) data sinks of each application (e. g., AWS or Firebase – Section 4). Five (12%) of the tested applications rely on open and insecure cloud storage that anyone can access. Since the identification of the cloud storage is based on a heuristic, we manually validated if the storage buckets belong to the analyzed applications. Unfortunately, we could confirm this in all cases by accessing our own shared data. We could access all processed data for these applications ranging from activities about the children (e. g., when and where the parents can pick up the children) to photos and videos taken of the children. Due to the high sensitivity of this data and the catastrophic consequences in case an adversary obtains a copy, we immediately notified the companies in a responsible disclosure process (Section 6).

Transport Layer Security. To assess the confidentiality of the data when sent to the application backend, we inspected the HTTP traffic of each application. The applications established the observed HTTP connections

via a TLS protected channel (i. e., HTTPS connections). Hence, none of the analyzed apps sends or receives data over an unencrypted channel.

Nevertheless, a TLS connection does not automatically guarantee fully protected communication [8, 65]. Based on the URLs that we identified in our static analysis, we extracted those that correspond to the analyzed apps based on their domain (eTLD+1). In total, the applications established TLS connections to 162 endpoints. We established a connection to each endpoint and examined the versions of TLS and the used cipher suites. Of the tested systems, 89 (55%) relied on TLS 1.0, and 91 (56%) only supported TLS 1.1 connections, which are considered to be out-of-date [53] and insecure. One system even required an SSLv3 connection. 98% of the analyzed applications relied on such insecure TLS connections. A large fraction of this is owed to old Google Firebase instances, which still support TLS 1.0. 38 (24%) of all analyzed systems support TLS 1.3, the latest protocol version, and 142 (88%) support TLS 1.2. As of the time of writing, both of these TLS versions are considered secure if configured correctly.

One way to undermine the security of a TLS connection is so-called "downgrade" attacks [59]. One can use the TLS fallback `SCSV` option to prevent such attacks. Of the analyzed systems, 128 (79%) enabled this option. Finally, we analyzed if any of the services use cipher suites that the *NIST* considers insecure [53]. Overall, 59 systems (37%) still relied on such suites (e. g., `TLS_RSA_WITH_3DES_EDE_CBC_SHA` was supported by 53 backend systems). The usage of such cipher suites affects 35 (83%) of the analyzed mobile applications. Hence, a motivated adversary might be able to manipulate or read the seemingly encrypted traffic.

Outdated and Vulnerable Software. To get an understanding of the software used in the backends, we analyzed the HTTP response headers (e. g., *Apache/2.4.29 Ubuntu*). To analyze the systems, we utilize the *Shodan Search Engine* (Section 4). One limitation of analyzing HTTP response headers is that the server can control the content of each header (e. g., insert dummy values) or omit a header altogether. Our experiment could not identify any software headers for 22 (52%) of the analyzed application backends, but 20 systems reported at least one software they use.

Surprisingly, 14 (70%) of them used software for which vulnerabilities are publicly known (e. g., *Common Vulnerabilities and Exposures* (CVE)), which is a reference method for publicly known vulnerabilities). Hence, such services (openly) provide information that an attacker can utilize in the early stages of an attack. For example, one of the applications uses a vulnerable web server (*Apache Tomcat v7.0.42*), for which in total eight vulnerabilities are publicly reported by Shodan. For another service, Shodan could identify 16 vulnerabilities, one of them would allow bypassing the server's authentication. In these exemplary cases, an adversary can profit from a rich set of *nice-to-have* background information.

Further Backend Findings. Finally, we want to address further security-related findings that potentially allow an adversary to undermine security fundamentally.

Injection and Access Control: Two common vulnerabilities are insufficient access control and injection attacks [68]. In our analysis of the backends, we made a simple test to understand if the developers of the applications are aware of these problems (Section 4). Even these simple tests showed that four (10%) of the mobile applications suffer from critical vulnerabilities. A SQL injection in the login field allows the attacker to gain full access to the entire database for one application. Another application suffered from a *GraphQL* [32] injection vulnerability, which would allow an adversary to get full access to all stored data. The two other applications were vulnerable to direct object referencing attacks that allowed anyone with a valid account to access the data of other users by altering the user id. All these attacks allow (full) exfiltration of the data.

Cross-Origin Resource Sharing: Finally, we turn to the usage of cross-origin resources or, more precisely, how applications limit the use of such. Cross-origin resource sharing is an extension of the same-origin policy, which prohibits access to data of a web application from other domains (Section 4). Of the analyzed backends, 11 (26%) used no or an insecure HTTP cross-origin resource sharing header. For those applications, an adversary might be able to read data of logged-in users via malicious JavaScript code that is executed if the user visits some website that is under the attacker's control.

6 Ethical Consideration

While there is no ethics committee covering this type of work at the organizations involved in this research, strict laws and privacy regulations are in place, and we discussed our analysis plan with peers to ensure proper design. We conducted our work according to ethical best practices detailed in the *Menlo Report* [85]. For example, we used artificial information for the user accounts and handled the required data securely in access control

and (encrypted) storage. Only the researchers that conducted the static and dynamic analysis had access to the test systems. We always used a minimally invasive approach and accessed only our data. Most importantly, we never accessed, altered, or in any way interacted with children's data. Professional penetration testers, part of the team, performed the static and dynamic analysis. These team members were professionally trained and adhered to strict guidelines for "ethical hacking". Finally, we did not offer the tested applications at a courtesy rate.

Responsible Disclosure. We responsibly disclosed the issues to the vendors on November 12, 2021. We sent a complete report of our findings to each company. Each report contained all findings for the specific application, including hints on how to fix each issue. Furthermore, we assist those developers that get back to us to ensure that the applications are appropriately hardened.

7 Discussion

Our multifactorial analysis of child care applications leads to two broader research discussions from our findings. One is the direct threat to privacy by tracking mechanisms, and the other is the inherent threat of information leakage through security-related issues. Moreover, we like to emphasize the need for different stakeholders in this ecosystem (especially developers and regulatory authorities) to protect children's privacy more carefully.

7.1 Tracking

This work is the first multi-dimensional analysis of child care management applications privacy. Our multilateral analysis uncovered several unwanted practices regarding the privacy of children. For example, we found that child care management apps generally request several dangerous permissions, which is even higher when third-party (tracking) libraries are used. The analyzed apps share user data, user interaction, device data, and app data with third-party services and, in one case, even via *Slack* (Section 5.2.2). Moreover, we found that the developers of the apps do not even mention the use of such third-party tracking services in their privacy policies. We think that the identified issues relate to technical and organizational problems within the child care applications since sharing often happens without explicit and verifiable parental consent, and the applications do not mention the processing of children's data in their privacy policy (Section 5.2). Overall, these practices are a potential threat to the privacy of all parties involved – especially the parents and children – and put the regulatory compliance of the applications in question.

7.2 Misconfigured, Outdated, and Insecure

Security-related issues such as misconfigured applications or outdated backends affect users' privacy. For example, 98% of the analyzed applications support no longer maintained Android versions, which are no longer supplied with security updates. We assume the developers support these 'old' operating systems (Android 8.0 or lower) to increase their user base, which still accounts for approx. 15–20% of all mobile Android devices [79]. Our results show that misconfigured libraries and unprotected cloud storages increase the attack surface of multiple applications. Most applications rely on out-of-date and "end of life" technology within the application and their backends, opening the door for privacy leaks. In contrast to the supported Android devices, the backends and used libraries are presumably under the developers' control (i. e., a third party does not maintain them). Therefore, a secure operation of the services should be part of the company's quality and information security management.

7.3 Call to Action

Our work shows that applications that process data without children's knowledge or consent pose a significant threat to their privacy. Children cannot be expected and are not capable of making an informed decision. Hence, it is the job of their parents, the child care workers, and the operators of these facilities to act with caution when they try to meet their responsibilities. We hope our work will raise awareness about two things, (1) the privacy risks introduced by these applications and (2) the layered problem of parents or employees of child care institutions deciding for children in terms of digital privacy. This indirect processing of children's data is an essential distinction to previous work (e. g., risks from parental control application), and we hope that regulatory authorities will focus on these risks. Parents' inability to audit such applications is a call to action for different stakeholders. The decision processes

of child care centers and parents should include their children's online privacy and the security maturity of the platforms they use to share data about their children [3]. Data protection agencies and regulators must understand how child care applications exploit the privacy of parents and children and mandate the adaption of the current legislation. Developers could benefit from access to security experts' consultancy, training, support from code analysis tools, and clarification of security risks in official documentation [60, 65, 90]. Finally, the identified threats such as the leakage of sensitive data must be addressed to be safe from child maltreatment and other risks. As using child care apps to share activities and photos in real-time can also increase the risk of "helicopter parenting" [35, 76], future work could analyze this phenomenon from a human-centric point of view (e. g., how do parents decide which information on their children they share?). It is also unclear whether alternative approaches to direct communication, e. g., via instant messaging such as WhatsApp, are better regarding children's privacy and how low-income households are affected by this mobile child care app trend. However, we consider such investigations out of scope for this work.

8 Limitations

Our application corpus only consists of 42 apps, which is a subset of all available applications. Due to a lack of support by LibRadar++ for XAPKs (e. g., three applications use XAPKs) and code-obfuscation techniques (e. g., 103 third-party libraries use obfuscation), we could not analyze all data. Moreover, we only analyzed Android applications and did not analyze their respective iOS counterparts. We used a rooted Android 8.1 phone for our dynamic analysis. However, newer Android versions introduce multiple changes to different aspects of the operating system, e. g., permissions and data. Some of these changes also propagate downwards for applications that target older builds, e. g., all updates to the permission model [29]. We argue that using an Android 8.1 phone is valid because it is the shared possible lower bound for the applications we analyzed in our corpus (cf. Table 1). Our study aimed to assess the privacy implications of child care apps and does not analyze the difference in implementations of specific apps. Furthermore, the vulnerabilities on the backends apply to all mobile applications. Another limitation of our application corpus is that we only analyzed applications available in the country-specific version of the store of the authors. Hence, we might have missed applications only in specific locations. However, our corpus includes applications from vendors in the US, Europe, and Asia.

Our analysis utilizes different established tools that allow a (semi-) automatic analysis of mobile applications. However, like all studies that scale their experiment in such a way, these tools will not find all vulnerabilities and potential privacy issues. Furthermore, some findings can be false positives. In cases where it has been ethically possible, we verified reported results to avoid over-reporting and minimize the false-positive rate.

Another limitation is a fluent transition between child care and pre-/elementary school apps. Some of the analyzed applications might be used in these contexts in addition to child care. However, child care was a central aspect of the application (e. g., activities in all-day elementary schools), and the children do not actively participate in digital communication. Therefore, we argue that this distinction is not an issue in our analysis.

9 Conclusion

In this work, we analyzed 42 Android child care management applications using a framework that combines static and dynamic analysis tools. These applications assist child care workers with daily tasks and address a growing demand for fast and easy communication, which parents generally expect. Unfortunately, most of the analyzed apps require a large number (Mean: 19) of potentially *dangerous* permissions. We also found 107 tracking and third-party libraries in 40 of the applications. Moreover, our results show that the privacy policies are unclear about the data collection practices, do not state how the companies protect the children's data, and underreport on the scope of data shared with third-party services. Concerningly, some of the tested applications relied on misconfigured cloud storage that allowed anyone to access and download data ranging from all children's activities, over messages, to personal photos. Developers of such insecure and privacy-invasive applications need to be made aware of their responsibilities, and regulatory authorities need to focus on the risks posed by such applications. Unfortunately, until then, it will be the job of the parents and the operators of the child care facilities to act with caution not to risk their children's privacy.

Acknowledgments

The authors would like to thank their shepherd, Anastasia Shuba, and the anonymous reviewers for their helpful comments. This research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 *CASA* – 390781972.

References

[1] Ajin Abraham, "Magaofei", Matan Dobrushin, and Vincent Nadal. 2021. Mobile Security Framework (MobSF) – Version v3.4.3. https://github.com/MobSF/Mobile-Security-Framework-MobSF, as of March 15, 2022.

[2] Suzan Ali, Mounir Elgharabawy, Quentin Duchaussoy, Mohammad Mannan, and Amr Youssef. 2020. Betrayed by the Guardian: Security and Privacy Risks of Parental Control Solutions. In *Annual Conference on Computer Security Applications (ACSAC '20)*. ACM, Austin, Texas, USA, 69–83.

[3] Tawfiq Ammari, Priya Kumar, Cliff Lampe, and Sarita Schoenebeck. 2015. Managing Children's Online Identities: How Parents Decide What to Disclose about Their Children Online. In *ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, Seoul, Republic of Korea, 1895–1904.

[4] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. 2019. PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play. In *USENIX Security Symposium (SSYM '19)*. USENIX, Santa Clara, California, USA, 585–602.

[5] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. 2020. Actions Speak Louder than Words: Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck. In *USENIX Security Symposium (SSYM '20)*. USENIX, Virtual Conference, 985–1002.

[6] Michael Backes, Sven Bugiel, and Erik Derr. 2016. Reliable Third-Party Library Detection in Android and Its Security Applications. In *ACM Conference on Computer and Communications Security (CCS '16)*. ACM, Vienna, Austria, 356–367.

[7] Simon Bradshaw, Christopher Millard, and Ian Walden. 2011. Contracts for Clouds: Comparison and Analysis of the Terms and Conditions of Cloud Computing Services. *Journal of Law and Information Technology* 19, 3 (July 2011), 187–223.

[8] Marcus Brinkmann, Christian Dresen, Robert Merget, Damian Poddebniak, Jens Müller, Juraj Somorovsky, Jörg Schwenk, and Sebastian Schinzel. 2021. ALPACA: Application Layer Protocol Confusion – Analyzing and Mitigating Cracks in TLS Authentication. In *USENIX Security Symposium (SSYM '21)*. USENIX, Virtual Conference, 4293–4310.

[9] Chad Brubaker, Suman Jana, Baishakhi Ray, Sarfraz Khurshid, and Vitaly Shmatikov. 2014. Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations. In *IEEE Symposium on Security and Privacy (SP '14)*. IEEE, San Jose, California, USA, 114–129.

[10] Kevin Zhijie Chen, Noah M. Johnson, Vijay D'Silva, Shuaifu Dai, Kyle MacNamara, Thomas R. Magrino, Edward Xue-Jun Wu, Martin Rinard, and Dawn Xiaodong Song. 2013. Contextual Policy Enforcement in Android Applications with Permission Event Graphs. In *Symposium on Network and Distributed System Security (NDSS '13)*. ISOC, San Diego, California, USA.

[11] Gordon Chu, Noah Apthorpe, and Nick Feamster. 2018. Security and Privacy Analyses of Internet of Things Children's Toys. *IEEE Internet of Things Journal* 6, 1 (Aug. 2018), 978–985.

[12] Phoebe K. Chua and Melissa Mazmanian. 2021. What Are You Doing With Your Phone? How Social Class Frames Parent-Teen Tensions around Teens' Smartphone Use. In *ACM Conference on Human Factors in Computing Systems (CHI '21)*. ACM, Yokohama, Japan, 353:1–353:12.

[13] Jiashan Cui and Luke Natzke. 2021. NCES: Early Childhood Program Participation – 2019. https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2020075REV, as of March 15, 2022.

[14] Sonja Damen, Nadine Madeira Firmino, and Kirsten Fuchs-Rechlin. 2021. Guidebook: Observation and Documentation in Child Day Care Facilities in North Rhine-Westphalia (Germany). https://www.kita.nrw.de/sites/default/files/documents/2021-09/bedo-orientierungsleitfaden_2021_webversion.pdf, as of March 15, 2022.

[15] Biniam Fisseha Demissie and Silvio Ranise. 2021. Assessing the Effectiveness of the Shared Responsibility Model for Cloud Databases: The Case of Google's Firebase. In *IEEE International Conference on Smart Data Services (SMDS '21)*. IEEE, Virtual Conference, 121–131.

[16] Erik Derr, Sven Bugiel, Sascha Fahl, Yasemin Acar, and Michael Backes. 2017. Keep Me Updated: An Empirical Study of Third-Party Library Updatability on Android. In *ACM Conference on Computer and Communications Security (CCS '17)*. ACM, Dallas, Texas, USA, 2187–2200.

[17] Sheila Donovan. 2020. 'Sharenting': The Forgotten Children of the GDPR. *Peace Human Rights Governance* 4, 1 (March 2020), 35–59.

[18] The European Parliament and the Council of the European Union. 2016. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27. April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). http://data.europa.eu/eli/reg/2016/679/oj, as of March 15, 2022.

[19] The European Parliament and the Council of the European Union. 2022. Does My Company/Organisation Need to Have a Data Protection Officer (DPO)? https://ec.europa.eu/info/law/law-topic/data-protection/reform/rules-business-and-organisations/obligations/data-protection-officers/does-my-company-organisation-need-have-data-protection-officer-dpo_en, as of March 15, 2022.

[20] Lesley-Anne Ey and C. Glenn Cupit. 2011. Exploring Young Children's Understanding of Risks Associated with Internet

Usage and Their Concepts of Management Strategies. *Journal of Early Childhood Research* 9, 1 (Feb. 2011), 53–65.

[21] Álvaro Feal, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, and Alessandra Gorla. 2020. Angel or Devil? A Privacy Study of Mobile Parental Control Apps. In *Privacy Enhancing Technologies Symposium (PETS '20)*. Sciendo, Virtual Conference, 314–335.

[22] Federal Trade Commission. 1998. Children's Online Privacy Protection Rule ("COPPA"). https://www.ftc.gov/enforcement/rules/rulemaking-regulatory-reform-proceedings/childrens-online-privacy-protection-rule, as of March 15, 2022.

[23] Federal Trade Commission. 2017. Children's Online Privacy Protection Rule: A Six-Step Compliance Plan for Your Business. https://www.ftc.gov/tips-advice/business-center/guidance/childrens-online-privacy-protection-rule-six-step-compliance, as of March 15, 2022.

[24] Richard Feloni. 2016. How the Founder of a Preschool Management App Cleverly Negotiated a $600,000 Deal with Mark Cuban and Chris Sacca. https://www.businessinsider.com/brightwheel-gets-shark-tank-investment-2016-5, as of March 15, 2022.

[25] Yu Feng, Saswat Anand, Isil Dillig, and Alex Aiken. 2014. Apposcopy: Semantics-based Detection of Android Malware through Static Analysis. In *ACM Symposium on Foundations of Software Engineering (FSE '14)*. ACM, Hong Kong, China, 576–587.

[26] Manuela Ferreira. 2013. "If We Twirl Too Much Will Our Panties Be Seen?": The Social Construction of Intimacy and Bodily Privacy Among Girls in a Kindergarten. *Issues in Early Education* 9, 2 (21) (April 2013), 43–52.

[27] Google, Inc. 2022. Android Developers: Base Permission Types. https://developer.android.com/guide/topics/manifest/permission-element, as of March 15, 2022.

[28] Google, Inc. 2022. Firebase. https://firebase.google.com, as of March 15, 2022.

[29] Google, Inc. 2022. Privacy Changes in Android 10. https://developer.android.com/about/versions/10/privacy/changes, as of March 15, 2022.

[30] Mariem Graa, Nora Cuppens-Boulahia, Frédéric Cuppens, and Ana Cavalli. 2012. Detecting Control Flow in Smartphones: Combining Static and Dynamic Analyses. In *Symposium on Cyberspace Safety and Security (CSS '12)*. Springer, Melbourne, Australia, 33–47.

[31] Sandra Grant, Susan Danby, Karen Thorpe, and Maryanne Theobald. 2016. Early Childhood Teachers' Work in a Time of Change. *Australasian Journal of Early Childhood* 41, 3 (Sept. 2016), 38–45.

[32] GraphQL Foundation, Inc. 2022. GraphQL: A Query Language for Your API. https://graphql.org, as of March 15, 2022.

[33] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G. Shin, and Karl Aberer. 2018. Polisis: Automated Analysis and Presentation of Privacy Policies Using Deep Learning. In *USENIX Security Symposium (SSYM '18)*. USENIX, Baltimore, Maryland, USA, 531–548.

[34] Mario Heiderich, Jörg Schwenk, Tilman Frosch, Jonas Magazinius, and Edward Z. Yang. 2013. mXSS Attacks: Attacking Well-Secured Web-Applications by Using innerHTML Mutations. In *ACM Conference on Computer and Communications Security (CCS '13)*. ACM, Berlin, Germany, 777–788.

[35] David Gauvey Herbert. 2016. A Much Closer Look at Nap Time: America's Day Care Industry Goes Gaga for Digital Oversight. https://bloom.bg/1WaMhF3, as of March 15, 2022.

[36] 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 *ACM International on Conference on Information and Knowledge Management (CIKM '15)*. ACM, Melbourne, Australia, 1111–1120.

[37] "initstring". 2020. cloud_enum: Enumerate Public Resources in AWS, Azure, and Google Cloud – Version v0.6. https://github.com/initstring/cloud_enum, as of March 15, 2022.

[38] Kuyju Kim, Taeyun Kim, Seungjin Lee, Soolin Kim, and Hyoungshick Kim. 2018. When Harry Met Tinder: Security Analysis of Dating Apps on Android. In *Nordic Conference on Secure IT Systems (NordSec '18)*. Springer, Oslo, Norway, 454–467.

[39] Lora Kolodny. 2016. Bloomz Raises $2.3 Million to Connect Teachers and Students' Families. https://techcrunch.com/2016/05/24/bloomz-raises-2-3-million-to-connect-teachers-and-students-families/, as of March 15, 2022.

[40] Priya C. Kumar, Marshini Chetty, Tamara L. Clegg, and Jessica Vitak. 2019. Privacy and Security Considerations for Digital Technology Use in Elementary Schools. In *ACM Conference on Human Factors in Computing Systems (CHI '19)*. ACM, Glasgow, Scotland, United Kingdom, 307:1–307:13.

[41] Priya C. Kumar, Shalmali Milind Naik, Utkarsha Ramesh Devkar, Marshini Chetty, Tamara L. Clegg, and Jessica Vitak. 2017. "No Telling Passcodes Out Because They're Private": Understanding Children's Mental Models of Privacy and Security Online. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, Portland, Oregon, USA, 46:1–46:21.

[42] Renuka Kumar, Sreesh Kishore, Hao Lu, and Atul Prakash. 2020. Security Analysis of Unified Payments Interface and Payment Apps in India. In *USENIX Security Symposium (SSYM '20)*. USENIX, Virtual Conference, 1499–1516.

[43] Li Li, Alexandre Bartel, Tegawendé F. Bissyandé, Jacques Klein, Yves Le Traon, Steven Arzt, Siegfried Rasthofer, Eric Bodden, Damien Octeau, and Patrick McDaniel. 2015. IccTA: Detecting Inter-Component Privacy Leaks in Android Apps. In *IEEE International Conference on Software Engineering (ICSE '15)*. IEEE, Florence, Italy, 280–291.

[44] Li Li, Tegawendé F Bissyandé, Mike Papadakis, Siegfried Rasthofer, Alexandre Bartel, Damien Octeau, Jacques Klein, and Le Traon. 2017. Static Analysis of Android Apps: A Systematic Literature Review. *Information and Software Technology* 88, 1 (Aug. 2017), 67–95.

[45] Sonia Livingstone, Leslie Haddon, Anke Görzig, and Kjartan Ólafsson. 2012. Risks and Safety on the Internet: The Perspective of European Children. https://eprints.lse.ac.uk/33731/, as of March 15, 2022.

[46] Sonia Livingstone and Ellen J. Helsper. 2012. Children, Internet and Risk in Comparative Perspective. *Journal of Children and Media* 7, 1 (Nov. 2012), 1–8.

[47] Qian Luo, Jiajia Liu, Jiadai Wang, Yawen Tan, Yurui Cao, and Nei Kato. 2020. Automatic Content Inspection and Forensics for Children Android Apps. *IEEE Internet of Things*

Journal 7, 8 (Aug. 2020), 7123–7134.

[48] Ziang Ma. 2017. LibRadar: A Detecting Tool for 3rd-Party Libraries in Android Apps – Version v1.5.0. https://github.com/pkumza/LibRadar, as of March 15, 2022.

[49] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. 2016. LibRadar: Fast and Accurate Detection of Third-Party Libraries in Android Apps. In *International Conference on Software Engineering Companion (ICSE-C '16)*. IEEE, Austin, Texas, USA, 653–656.

[50] John Matherly. 2022. Shodan Search Engine. https://www.shodan.io, as of March 15, 2022.

[51] Célestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do Cookie Banners Respect my Choice?: Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *IEEE Symposium on Security and Privacy (SP '20)*. IEEE, Virtual Conference, 791–809.

[52] Lorraine E. Maxwell. 2007. Competency in Child Care Settings. *Environment and Behavior* 39, 2 (March 2007), 229–245.

[53] Kerry A. McKay and David A. Cooper. 2019. Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations: NIST Special Publication 800-52 Revision 2.

[54] Marisa Meyer, Victoria Adkins, Nalingna Yuan, Heidi M. Weeks, Yung-Ju Chang, and Jenny Radesky. 2019. Advertising in Young Children's Apps: A Content Analysis. *Journal of Developmental and Behavioral Pediatrics* 40, 1 (Jan. 2019), 32–39.

[55] Ingrida Milkaite and Eva Lievens. 2019. Status Quo Regarding the Child's Article 8 GDPR Age of Consent for Data Processing Across the EU. https://www.betterinternetforkids.eu/practice/awareness/article?id=3017751, as of March 15, 2022.

[56] Stuart Millar, Niall McLaughlin, Jesus Martinez del Rincon, Paul Miller, and Ziming Zhao. 2020. DANdroid: A Multi-View Discriminative Adversarial Network for Obfuscated Android Malware Detection. In *ACM Conference on Data and Application Security and Privacy (CODASPY '20)*. ACM, New Orleans, Louisiana, USA, 353–364.

[57] Claire Cain Miller. 2018. The Relentlessness of Modern Parenting. https://www.nytimes.com/2018/12/25/upshot/the-relentlessness-of-modern-parenting.html, as of March 15, 2022.

[58] Tehila Minkus, Kelvin Liu, and Keith W. Ross. 2015. Children Seen But Not Heard: When Parents Compromise Children's Online Privacy. In *The World Wide Web Conference (WWW '15)*. ACM, Florence, Italy, 776–786.

[59] Bodo Möller and Adam Langley. 2015. *TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks*. RFC 7507. RFC Editor. 1–7 pages. https://tools.ietf.org/html/rfc7507

[60] Duc Cuong Nguyen, Dominik Wermke, Yasemin Acar, Michael Backes, Charles Weir, and Sascha Fahl. 2017. A Stitch in Time: Supporting Android Developers in Writing Secure Code. In *ACM Conference on Computer and Communications Security (CCS '17)*. ACM, Dallas, Texas, USA, 1065–1077.

[61] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. 2021. Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps. In *USENIX Security Symposium (SSYM '21)*. USENIX, Virtual Conference, 3667–3684.

[62] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. In *ACM Conference on Human Factors in Computing Systems (CHI '20)*. ACM, Honolulu, Hawaii, USA, 1–13.

[63] Jonathan A. Obar and Anne Oeldorf-Hirsch. 2018. The Biggest Lie on the Internet: Ignoring the Privacy Policies and Terms of Service Policies of Social Networking Services. *Information, Communication & Society* 23, 1 (July 2018), 128–147.

[64] Gwenn Schurgin O'Keeffe and Kathleen Clarke-Pearson. 2011. The Impact of Social Media on Children, Adolescents, and Families. *Pediatrics* 127, 4 (April 2011), 800–804.

[65] Marten Oltrogge, Nicolas Huaman, Sabrina Amft, Yasemin Acar, Michael Backes, and Sascha Fahl. 2021. Why Eve and Mallory Still Love Android: Revisiting TLS (In)Security in Android Applications. In *USENIX Security Symposium (SSYM '21)*. USENIX, Virtual Conference, 4347–4364.

[66] OneSignal, Inc. 2022. Guides: Common Questions About Onesignal's Data Handling and Security. https://documentation.onesignal.com/docs/data-questions, as of March 15, 2022.

[67] Lucky Onwuzurike, Mário Almeida, Enrico Mariconti, Jeremy Blackburn, Stringhinim Gianluca, and Emiliano De Cristofaro. 2018. A Family of Droids-Android Malware Detection via Behavioral Modeling: Static vs Dynamic Analysis. In *IEEE Conference on Privacy, Security and Trust (PST '18)*. IEEE, Belfast, Ireland, 1–10.

[68] OWASP Foundation, Inc. 2021. Open Web Application Security Project (OWASP) Top 10. https://owasp.org/Top10/, as of March 15, 2022.

[69] Will Parnell and Jackie Bartlett. 2012. iDocument: How Smartphones and Tablets Are Changing Documentation in Preschool and Primary Classrooms. *YC Young Children* 67, 3 (May 2012), 50–57.

[70] Sebastian Poeplau, Yanick Fratantonio, Antonio Bianchi, Christopher Kruegel, and Giovanni Vigna. 2014. Execute This! Analyzing Unsafe and Malicious Dynamic Code Loading in Android Applications. In *Symposium on Network and Distributed System Security (NDSS '14)*. ISOC, San Diego, California, USA.

[71] PortSwigger, Inc. 2021. Burp Suite Professional – Version v2021.10.1. https://portswigger.net/burp/pro, as of March 15, 2022.

[72] Zhengyang Qu, Shahid Alam, Yan Chen, Xiaoyong Zhou, Wangjun Hong, and Ryan Riley. 2017. DyDroid: Measuring Dynamic Code Loading and Its Security Implications in Android Applications. In *Conference on Dependable Systems and Networks (DSN '17)*. IEEE, Denver, Colorado, USA, 415–426.

[73] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. 2018. Apps, Trackers, Privacy, and Regulators: A Global Study of the Mobile Tracking Ecosystem. In *Symposium on Network and Distributed System Security (NDSS '18)*. ISOC, San Diego, California, USA.

[74] "rbsec". 2021. sslscan2: Tests SSL/TLS Enabled Services – Version v2.0.10. https://github.com/rbsec/sslscan, as of March 15, 2022.

[75] 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 *USENIX Security Symposium (SSYM '19)*. USENIX, Santa Clara, California, USA, 603–620.

[76] Natalie Reilly. 2019. The Problem With Helicopter Parenting Is That It Works. https://www.smh.com.au/lifestyle/life-and-relationships/the-problem-with-helicopter-parenting-is-that-it-works-20190410-p51csx.html, as of March 15, 2022.

[77] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. 2018. "Won't Somebody Think of the Children?" Examining COPPA Compliance at Scale. In *Privacy Enhancing Technologies Symposium (PETS '18)*. Sciendo, Barcelona, Spain, 63–83.

[78] Shiv Sahni. 2019. FireBase Scanner. https://github.com/shivsahni/FireBaseScanner, as of March 15, 2022.

[79] StatCounter. 2021. Mobile Android Version Market Share Worldwide 2018-2021. https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide, as of March 15, 2022.

[80] Marius Steffens, Christian Rossow, Martin Johns, and Ben Stock. 2019. Don't Trust The Locals: Investigating the Prevalence of Persistent Client-Side Cross-Site Scripting in the Wild. In *Symposium on Network and Distributed System Security (NDSS '19)*. ISOC, San Diego, California, USA.

[81] Kimberly Tam, Ali Feizollah, Nor Badrul Anuar, Rosli Salleh, and Lorenzo Cavallaro. 2017. The Evolution of Android Malware and Android Analysis Techniques. *Comput. Surveys* 49, 4 (Feb. 2017), 76:1–76:41.

[82] Anika M. Trancik and Gary W. Evans. 1995. Spaces Fit for Children: Competency in the Design of Daycare Center Environments. *Children's Environments* 12, 3 (Sept. 1995), 311–319.

[83] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2019. A Study on Subject Data Access in Online Advertising After the GDPR. In *International Workshop on Data Privacy Management (DPM '19)*. Springer, Luxembourg City, Luxembourg, 61–79.

[84] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Measuring the Impact of the GDPR on Data Sharing in Ad Networks. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '20)*. ACM, Taipei, Taiwan, 222–235.

[85] US Department of Homeland Security. 2012. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. https://www.caida.org/publications/papers/2012/menlo_report_actual_formatted/, as of March 15, 2022.

[86] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (Un)informed Consent: Studying GDPR Consent Notices in the Field. In *ACM Conference on Computer and Communications Security (CCS '19)*. ACM, London, United Kingdom, 973–990.

[87] Junia Valente and Alvaro A. Cardenas. 2017. Security & Privacy in Smart Toys. In *Workshop on Internet of Things Security and Privacy (IoT S&P '17)*. ACM, Dallas, Texas, USA, 19–240.

[88] Rui Wang, Yuchen Zhou, Shuo Chen, Shaz Qadeer, David Evans, and Yuri Gurevich. 2013. Explicating SDKs: Uncovering Assumptions Underlying Secure Authentication and Authorization. In *USENIX Security Symposium (SSYM '13)*. USENIX, Washington, District of Columbia, USA, 399–414.

[89] Washington State Department of Children, Youth, and Families. 2020. Guidebook: Foundational Quality Standards for Early Learning Programs in Washington State (USA). https://www.dcyf.wa.gov/sites/default/files/pdf/FoundationalQualityStandardsAwarenessGuide.pdf, as of March 15, 2022.

[90] Charles Weir, Ben Hermann, and Sascha Fahl. 2020. From Needs to Actions to Secure Apps? The Effect of Requirements and Developer Practices on App Security. In *USENIX Security Symposium (SSYM '20)*. USENIX, Virtual Conference, 289–305.

[91] Pamela Wisniewski, Arup Kumar Ghosh, Heng Xu, Mary Beth Rosson, and John M. Carroll. 2017. Parental Control vs. Teen Self-Regulation: Is There a Middle Ground for Mobile Online Safety?. In *ACM Conference on Computer-Supported Cooperative Work and Social Computing (CSCW '17)*. ACM, Portland, Oregon, USA, 51–69.

[92] Sarita Yardi and Amy Bruckman. 2011. Social and Technical Challenges in Parenting Teens' Social Media Use. In *ACM Conference on Human Factors in Computing Systems (CHI '11)*. ACM, Vancouver, British Columbia, Canada, 3237–3246.

[93] Sarah K. Zeegers, Christine A. Readdick, and Sally Hansen-Gandy. 1994. Daycare Children's Establishment of Territory to Experience Privacy. *Children's Environments* 11, 4 (Dec. 1994), 265–271.

[94] Leah Zhang-Kennedy, Christine Mekhail, Yomna Abdelaziz, and Sonia Chiasson. 2016. From Nosy Little Brothers to Stranger-Danger: Children and Parents' Perception of Mobile Threats. In *ACM Conference on Interaction Design and Children (IDC '16)*. ACM, Manchester, United Kingdom, 388–399.

[95] Jun Zhao, Ge Wang, Carys Dally, Petr Slovak, Julian Edbrooke-Childs, Max Van Kleek, and Nigel Shadbolt. 2019. 'I Make up a Silly Name': Understanding Children's Perception of Privacy Risks Online. In *ACM Conference on Human Factors in Computing Systems (CHI '19)*. ACM, Glasgow, Scotland, United Kingdom, 106:1–106:13.

A Requested Permissions

Figure 3 shows the identified permissions by category, as discussed in Section 5.2.3. To increase the readability of the plot, we excluded *normal* permissions. We find that 59% of the requested permissions belong to the *normal* category, 21% are classified as *dangerous*, 15% as *proprietary*, and 5% as *signature* permissions. Overall, the normal permissions, which do not require user consent, are requested 457 times (on average, 11 per app). More importantly, the dangerous permissions, which require explicit consent by the user, are requested 166 times (on average, four per app). Finally, we see that proprietary permissions are requested 116 times and signature permissions 42 times. The cross-comparison of the permissions is interesting as well: For example, the permissions for reading and writing to the SD card are often used together (reading: 23 (54%), writing: 28 (66%)). Similarly, permissions for accessing the microphone and the camera are frequently combined and are used by 14 (33%) apps. Eleven (26%) of the examined apps use the permission to access the camera in addition to the location of the mobile device. All these permissions are functionally related and allow the examined apps to perform certain functions, such as sharing photos or voice messages with parents or sending data and documents.

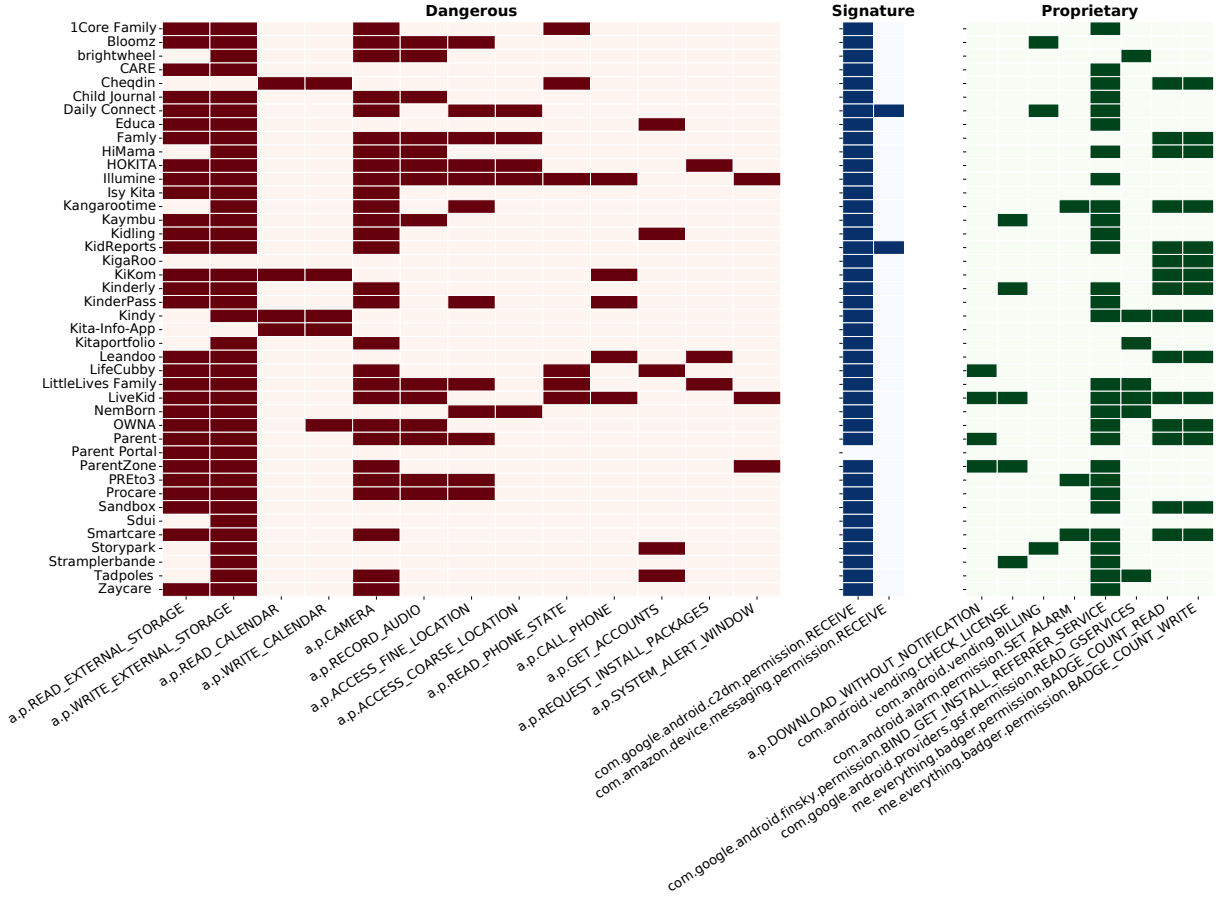

Fig. 3. Identified permissions by category. *Normal* permissions and permissions that occurred less than three times are omitted. "a.p." is an abbreviation for "android.permission."