1

Don't Accept Candy from Strangers: An Analysis of Third-Party Mobile SDKs

ÁLVARO FEAL[1], JULIEN GAMBA[2], JUAN TAPIADOR[3], PRIMAL WIJESEKERA[4], JOEL REARDON[5], SERGE EGELMAN[6] AND NARSEO VALLINA-RODRIGUEZ[7]

Abstract

Mobile application (app) developers often include third-party Software Development Kits (SDKs) in their software to integrate services and features offered by other companies, like online payments or social network integration, or to monetise their apps through advertisements. As a result, SDKs play a key role in the software supply chain. Their integration in apps is practically mandatory if developers aim at producing software that integrates smoothly in the current ecosystem of internet services. Unfortunately, these common software development practices might come at a privacy cost to end users since many third-party library providers implement data-driven business models that allow them to offer their services to developers free of monetary payment. In this chapter, we provide an overview of the third-party library ecosystem for the Android platform and we discuss its privacy implications for mobile end users due to limitations present in the permission system of today's mobile platforms, and the overall lack of transparency in the industry. We apply software analysis techniques and manual analysis to: (1) compare the effectiveness and limitations of state-of-the-art SDK detection tools; (2) manually classify SDKs by their purpose and compare the classification capabilities of current auditing tools; and (3) gain empirical insights about their behaviour and data collection practices. We discuss different ways to tackle the limitations present in current detection tools to increase developers'

[1] IMDEA Networks Institute, Universidad Carlos III de Madrid.
[2] IMDEA Networks Institute, Universidad Carlos III de Madrid.
[3] Universidad Carlos III de Madrid.
[4] ICSI, U.C. Berkeley.
[5] AppCensus Inc., University of Calgary.
[6] ICSI, U.C. Berkeley, AppCensus Inc.
[7] IMDEA Networks Institute, ICSI, AppCensus Inc.

awareness and regulatory enforcement through the design and development of new software analysis tools. We also discuss potential solutions to mitigate the limitations found in the current permission model in order to enhance user awareness and control.

Keywords

Android, Privacy, Third-party SDKs.

I. Introduction

In the last decade, smartphones have evolved from rare gadgets to indispensable and powerful tools ubiquitously carried by billions of users everywhere and nearly at all times. Modern smartphones have a variety of sensors such as the camera, GPS, and microphone that allow application developers to access personal information and details about their environment. These capabilities have enabled a rich ecosystem of innovative and 'smart' mobile apps that help users in all types of online activities, including social networking, banking, shopping, entertainment, and augmented reality applications.

Yet, developing profitable and innovative mobile applications can turn into a complex and costly process. As in the case of web and desktop software development, most app developers rely on already developed components (libraries or SDKs) offered by other companies or organisations (third parties), to integrate desired functionalities in their products, such as online payments, bug reporting, analytics or advertising and graphics support, among many others. The ability to reuse well-tested and well-maintained code in their software – which is often available for free – allows developers to speed up the development process and, ultimately, reduce development costs and time. Moreover, using such libraries constitutes a best practice in the software engineering discipline since these libraries are modular and reusable. This is particularly important in the case of security-critical code, such as cryptographic libraries, which can be implemented once and reused, making the code more robust and bug free.

The use of third-party SDKs may come at a privacy cost for end users, especially when developers integrate proprietary SDKs offered by data-driven organisations like analytics and advertising companies. The ubiquitous nature of smartphones and their capacity to access sensitive and behavioural data, along with the innovations enabled by the 'Big Data' revolution, provide many SDK providers with easy access to an unprecedented volume of high-quality data thanks to developers integrating their components in to millions of apps. Regulatory efforts such as the General Data Protection Regulation (GDPR)[8] and California's Consumer

[8] 'The general data protection regulation', Council of the European Union, www.consilium.europa.eu/en/policies/data-protection-reform/data-protection-regulation/ (last accessed April 2020).

Protection Act (CCPA),[9] enforce transparency by forcing software developers to declare the types of data that general categories of third parties may collect or receive from the app and obtain informed user consent (provided that there is no other legal basis for such collection). Nevertheless, regulation without strict enforcement seems to be insufficient to protect end users' privacy in digital products as previous academic studies have revealed.[10,11,12]

Mobile operating systems such as Android and iOS implement a permission model to enable user control and prevent unwanted or unauthorised access to sensitive data at the application level. However, these security mechanisms are insufficient when SDKs are embedded in an app. This happens because current mobile operating systems allow third-party code to run in the same context and with the same privileges as the app itself. This makes it difficult for users to identify whether a given permission will be used for the primary purposes of the app or for secondary usages such as user profiling or advertising. These inherited privileges, along with the fact that the specific behaviours of these third-party libraries are generally opaque to end-users and developers alike, constitute severe transparency and privacy issues. Users can only rely on the information disclosed by application developers in their privacy policies, which are typically incomplete and inaccurate.[13]

New regulatory frameworks make application developers liable for any personal data collection malpractice incurred by the third-party libraries embedded in their products. Yet, most SDKs do not open their code for inspection, leaving developers with no choice but to trust the SDK providers' claims and disclosures (ie, privacy by trust). Developers should investigate whether their third-party components are (or claim to be) compliant with privacy regulations.[14,15]

In this book, we aim to shed light on the issues and challenges that third-party SDKs bring to the mobile ecosystem from a privacy, transparency, and regulatory compliance standpoint. Our main contributions are:

- We provide an overview and a taxonomy of the SDKs available in the mobile ecosystem.

[9] 'California Consumer Privacy Act (CCPA)', State of California Department of Justice, oag.ca.gov/privacy/ccpa (last accessed April 2020).

[10] Hu, Xuehui, and Nishanth Sastry, 'Characterising Third Party Cookie Usage in the EU after GDPR', *Proceedings of the 10th ACM Conference on Web Science*.

[11] Sanchez-Rola et al, 'Can I Opt Out Yet? GDPR and the Global Illusion of Cookie Control', *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security.*
Jannick Sørensen and Sokol Kosta, 'Before and after gdpr: The changes in third party presence at public and private european websites', *The World Wide Web Conference.*

[12] Janis Wong and Tristan Henderson, 'How Portable is Portable? Exercising the GDPR's Right to Data Portability', *Proceedings of the 2018 ACM International Joint Conference.*

[13] Okoyomon et al., 'On the ridiculousness of notice and consent: Contradictions in app privacy policies', *Workshop on Technology and Consumer Protection (ConPro).*

[14] Razaghpanah et al, 'Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem', *The Network and Distributed System Security Symposium*.

[15] Reardon et al, '50 Ways to Leak Your Data: An Exploration of Apps' Circumvention of the Android Permissions System', *28th USENIX Security Symposium.*

- We discuss existing approaches to detect the presence of SDKs in mobile apps and compare the features provided by state-of-the-art SDK detection tools.
- We combine static, dynamic, and manual analysis to classify the SDKs present in the top 50 most popular apps on Google Play according to their claimed features, and compare the result to the classification provided by current auditing tools.
- We demonstrate the privacy and security risks derived from the data collection practices observed in third-party SDKs embedded in the corpus of 50 apps.
- We conclude by discussing possible solutions to the issues that SDKs bring from a privacy and regulatory compliance standpoint, such as educating developers, joint efforts for auditing SDKs, and improvements to the current permission model.

II. Background

Several studies focusing on code reuse in the Android platform have reported on the rich and diverse ecosystem of SDKs available for app developers and how their use is often perceived as a reflection on software engineering best practices.[16,17,18,19] Most mobile app developers use SDKs to integrate external features, components, and services in their software – eg, integrating game engines, handling online payments – but also for advertisement and analytics purposes.[20]

Previous work has shown the presence of third-party libraries in all kind of applications[21] regardless of their audience,[22,23] origin[24,25] or price.[26] Empirical

[16] Michael Backes, Sven Bugiel and Erik Derr, 'Reliable third-party library detection in android and its security applications', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.*

[17] Theodore Book, Adam Pridgen, and Dan S Wallach, 'Longitudinal analysis of android ad library permissions', *arXiv preprint arXiv:1303.0857.*

[18] Mojica et al, 'A large-scale empirical study on software reuse in mobile apps', *IEEE software.*

[19] Ruiz et al, 'Understanding reuse in the android market', *IEEE International Conference on Program Comprehension.*

[20] Han et al, 'The Price is (Not) Right: Comparing Privacy in Free and Paid Apps', *Proceedings on Privacy Enhancing Technologies Symposium.*

[21] Razaghpanah,' Apps, trackers' (n 14).

[22] Reyes et at, '"Won't somebody think of the children?" Examining COPPA compliance at scale' *Proceedings on Privacy Enhancing Technologies.*

[23] Junia Valente and Alvaro A Cardenas, 'Security and privacy in Smart Toys', *Workshop on Internet of Things Security and Privacy.*

[24] Gamba et al, 'An Analysis of Pre-installed Android Software', *IEEE Symposium on Security and Privacy.*

[25] Wang et al, 'Beyond google play: A large-scale comparative study of chinese android app markets', *Proceedings of the Internet Measurement Conference.*

[26] Han, 'Comparing Free and Paid apps' (n 20).

evidence shows that mobile applications contact on average six domains related to advertisement and tracking SDKs,[27] typically offered by different companies. The use of SDKs in the mobile software supply chain is so extended that even apps that come preinstalled on the phone are packaged with third-party advertising and analytics libraries.[28]

Despite their central role in the app development process, we barely understand the mobile SDK ecosystem and their privacy risks. An app developer might include third-party code in their software without realising that this is potentially harmful for users' privacy. Developers making SDK choices based on the service offered by the third-party library might render ineffective. The boundaries between many SDK categories are unclear, and SDK providers tend to offer more than one product to app developers. For instance, analytics and advertising services have become extremely entangled since most AdTech companies integrate both functionalities in the same SDK, potentially using the data gathered by the analytics service for user profiling or advertising.[29] Unfortunately, there are no reliable methods to accurately quantify their personal data collection practices and their purpose at scale.

The lack of information and transparency in the use of SDKs by mobile app developers has consumer protection implications, too. Marketplaces and app stores allow users to download apps and understand what type of feature they provide, whether they cost money, the number of users that they have and the legal documents in which data collection is explained. However, there is not enough transparency about the presence and data collection practices of SDKs embedded in the apps. Developers, users, and regulators could benefit from the existence of technologies to study the functioning and purpose of third-party libraries, ideally in a publicly available observatory.

A. The Mobile SDK Landscape

There are a broad range of third-party SDK providers that specialise in offering one or multiple features, services, or technologies to application developers. The type of services they offer range from SDKs offering UI support, to SDKs that collect user data in order to generate revenue. To illuminate this ecosystem, we rely on public information from hundreds of SDKs detected by previous research[30,31]

[27] Razaghpanah, 'Apps, Trackers' (n 14).

[28] Gamba, 'Pre-installed Android Software' (n 24).

[29] Ruiz, 'Reuse on Android' (n 19).

[30] Razaghpanah,' Apps, trackers' (n 14).

[31] Ma et al, 'LibRadar: fast and accurate detection of third-party libraries in Android apps', *Proceedings of the 38th international conference on software engineering companion.*

to create a more comprehensive classification of mobile SDKs by their offered functionality:

i. Development Support

These are libraries which help developers adding support features to their code, such as widgets, UI features or JavaScript Object Notation (JSON) and XML formatters. Examples of this category include the Android Support Library and GSON (Google's JSON implementation). These libraries are expected to be found in many applications and, assuming that they have not been tampered with to include malicious code, they should be harmless to users' privacy since they do not collect personal data. Therefore, a-priori they do not need to be included in documents such as the privacy policy. Nevertheless, some development SDKs might engage in personal data collection, like Google's Firebase[32] and Unity3D,[33] a library that supports the development of games but also includes analytics and advertisement capabilities. In this case, their ability to collect sensitive data will vary from one application to another, depending on how application developers integrate these services in their mobile products. It is possible to identify multiple subcategories of development support libraries, depending on their intended purpose:

Networking and protocol support: These libraries offer support for implementing network protocols such as HTTP/HTTPS or Google's QUIC.

Database support: These SDKs provide developers with code to manage and store data, implementing well known database solutions like SQL.

Cryptography support: These libraries help developers implementing cryptographic solutions for data storage or secure communications.

Cloud integration and support: The SDKs in this group allow for the integration of cloud services capabilities into applications, for instance Amazon Web Services[34] or Google's Firebase.[35]

Browser support: These SDKs provide functionalities to open web content, such as Android's WebView which allows applications to render webpages.

Cross-platform development: While application code in Android is developed using one of the two languages supported by the platform (Kotlin and Java), there are several SDKs that allow to include code in other languages for cross-platform development. One example is Facebook Hermes,[36] which allows to include React

[32] 'Google Mobile Services', Android, www.android.com/gms/ (last accessed April 2020).

[33] 'Unity for all', Unity, https://unity.com/ (last accessed April 2020).

[34] 'Getting Started with Android', Amazon AWS, accessed April 2020, https://aws.amazon.com/developers/getting-started/android/ (last accessed April 2020).

[35] 'Firebase', Google, https://firebase.google.com/ (last accessed April 2020).

[36] 'GitHub: Hermes', Facebook, https://github.com/facebook/hermes (last accessed April 2020).

code in Android and iOS apps, or Apache Cordova,[37] which allows using web development techniques to build mobile apps.

ii. Push Notifications/Consumer Engagement

Push notifications are small server-to-client messages used to reach mobile audiences anywhere and anytime. This technology is at the core of companies offering 'customer engagement' services to create a direct communication channel between an external stakeholder (consumer) and an organization (often a company, developer, advertiser, or brand). Many of the companies offering these services also offer analytics and advertisement. This is the case of Google, which offers its own cross-platform service – Firebase Cloud Messaging (FCM)[38] – JPush[39] or airPush.[40]

iii. Online Payments

Several SDK providers like AliPay[41] and Google Pay[42] allow developers to include online payment services. Many mobile applications, especially mobile games, no longer implement advertising-based monetisation models. Solutions like Fortumo[43] allow developers to explore alternative sources of revenues by requesting users to pay a small fee for unlocking premium features or purchasing virtual goods.

iv. Maps and Location Services

SDKs like the Google Maps SDK,[44] Here.com[45] or Baidu Maps[46] allow application developers to add maps, geo-location, and navigation capabilities to their products. The set of features and services offered by maps and location providers is very broad. While some offer pure mapping services, others like Google Maps provide data-driven added value like location-based business searches, geo-coding and even Augmented Reality (AR) services.[47]

[37] 'Apache Cordova', Apache, https://cordova.apache.org/ (last accessed April 2020).

[38] 'Firebase Cloud Messaging', Firebase, https://firebase.google.com/docs/cloud-messaging (last accessed April 2020).

[39] 'Product Introduction of JPush', JiGuang Docs, https://docs.jiguang.cn/en/jpush/guideline/intro/ (last accessed April 2020).

[40] 'The future of Mobile Advertising', airpush, https://airpush.com/ (last accessed April 2020).

[41] 'Trust makes it simple', Alipay, https://intl.alipay.com (last accessed April 2020).

[42] 'Google Pay', Google Developers, https://developers.google.com/pay (last accessed April 2020).

[43] 'Global direct carrier billing platform', Fortumo, https://fortumo.com/ (last accessed April 2020).

[44] 'Overview', Google Maps Platform, https://developers.google.com/maps/documentation/android-sdk/intro (last accessed April 2020).

[45] 'Homepage', HERE Technologies, www.here.com/ (last accessed April 2020).

[46] 'Android SDK', Baidu, lbsyun.baidu.com/index.php?title=androidsdk.

[47] 'Introducing Live View', Google Maps Help, https://support.google.com/maps/thread/11554255?hl=en (last accessed April 2020).

v. Authentication

These are services that allow application developers to protect parts of the application's functionality from unauthorised access using an online identity or two-factor authentication mechanisms. Examples of these SDKs are OAuth and Google's Firebase (Google Login).

vi. Social Networks

These SDKs allow developers to include functionality from social networks, such as login capabilities and the ability to share content with a list of friends. One remarkable example is the Facebook Graph SDK, which also provides analytics and advertisement services. As we will discuss in Section IV, applications integrating these libraries might be able to harvest personal data from the social network profile of the user.

vii. Analytics

Many companies provide analytics tools to understand how users interact with their app, find, and solve bugs and crashes, optimise user engagement, and generate revenue with highly detailed data about customers. Therefore, analytics SDKs could be broken down into several subcategories, with some SDKs providing more than one functionality to the app, including bug reporting (eg, Crashlytics),[48] A/B testing (eg, Firebase A/B testing),[49] and user engagement or CRM (eg, StartApp).[50]

viii. Advertisement

Advertisement SDKs are used by app developers to show ads to users, generating revenue for the developer. Because of targeted advertisement, many of these libraries also collect personal data in order to generate user profiles to better understand the type of content that a given user is interested on. Examples of these libraries are Google's AdMob,[51] Unity3D[52] or Twitter's MoPub.[53]

[48] 'Firebase crashlytics', Firebase, https://firebase.google.com/docs/crashlytics (last accessed April 2020).

[49] 'Firebase A/B Testing', Firebase, https://firebase.google.com/docs/ab-testing (last accessed April 2020).

[50] 'Mobile, Fulfileld', StartApp, www.startapp.com/ (last accessed April 2020).

[51] 'AdMob', Google Developers, https://developers.google.com/admob (last accessed April 2020).

[52] 'Monetize your game', Unity, https://unity.com/solutions/unity-ads (last accessed April 2020).

[53] 'Powerful app monetization', MoPub, www.mopub.com/ (last accessed April 2020).

Table 1 Examples of SDKs for each category

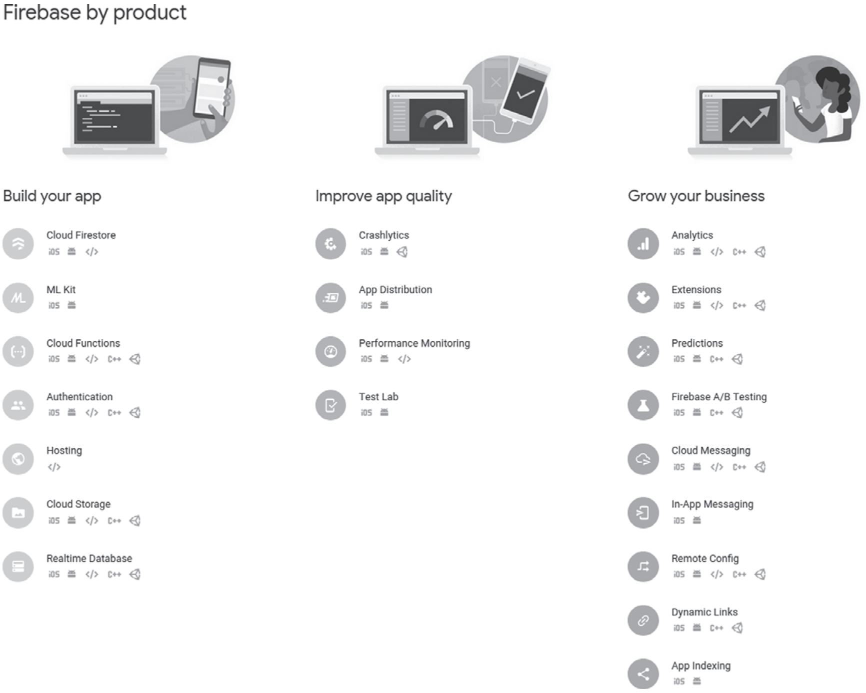
Type	**Examples**
Development support	Android Support, GSON, Unity3D
Network support	OKHTTP, Facebook Fizz, jmDNS
Database support	ORMLite, Android Wire, Firebase
Crypto support	Jasypt, Bouncy Castle
Browser support	HTML TextView, Chromium
Cloud integration and support	Google Firebase (Google Cloud)
Cross Platform Development	Apache Cordova, Facebook Hermes
Push notifications/Consumer engagement	Firebase Cloud Messaging, JPush, airPush
Online payments	AliPay, Fortumo
Social Networks	Facebook, Twitter, VK
Authentication	Google Firebase (2FA)
Maps/Location services	Google Maps, MapsForge, Baidu Maps
Analytics	Firebase, Baidu, Flurry
Advertisement	Unity, Google Ads, Amazon Mobile Ads

As we have seen, many SDKs offer multiple capabilities to app developers in a single library. This impedes attributing a single label in most cases. Table 1 provides examples for each of these types, showing how the same SDK can be labelled differently depending on its behaviour. The screenshot in Figure 1, shows one remarkable example which is Google's Firebase SDK. This SDK unifies analytics services, bug reporting, two-factor-authentication, services for integrating apps with Google cloud, and more. While we acknowledge that this taxonomy might not be complete, we believe that it offers a representative overview of the most common solutions that can be found in today's mobile applications.

B. Privacy Risks: Privilege Piggybacking

Both Android and iOS implement a permission model to protect sensitive system resources and data from abusive, malicious, or deceptive apps. Whenever a user installs an app from the app store, the OS forces the app to declare their access to protected resources. Only when the user consents to this – either at runtime or when installing the app, depending on the sensitivity of the permission and the OS policies – can the app access such protected resources. This app-centric permission model presents fundamental limitations to properly inform users

Figure 1 Screenshot from Firebase documentation page, showing the different products that it provides



Firebase by product

Build your app

- Cloud Firestore
- ML Kit
- Cloud Functions
- Authentication
- Hosting
- Cloud Storage
- Realtime Database

Improve app quality

- Crashlytics
- App Distribution
- Performance Monitoring
- Test Lab

Grow your business

- Analytics
- Extensions
- Predictions
- Firebase A/B Testing
- Cloud Messaging
- In-App Messaging
- Remote Config
- Dynamic Links
- App Indexing

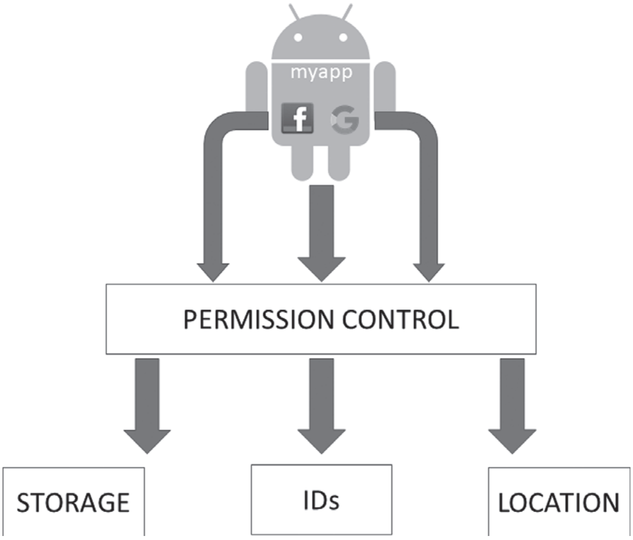
about the access to sensitive data by embedded SDKs. These run with the same privileges and permissions as the host app, as shown in Figure 2.

Many mobile users might trust the app developer when they give the app access to a piece of sensitive data such as location. However, they might not necessarily trust opaque third-party SDKs embedded in the product, particularly when the user is not even aware of their presence or is not familiar with the company, its business, and the way they will process their data.

Unfortunately, mobile operating systems and platforms fail to inform users about the SDKs that might be embedded in an app and whether they access sensitive data (unless the developer voluntarily discloses this list on their privacy policy). According to Google, the inclusion of privacy policies in mobile apps is not mandatory except when the application collects sensitive data or is aimed at children.[54] Furthermore, Google does not detail what review process it is followed to actively look for apps violating such a policy. Despite provisions in relevant privacy regulation like GDPR, CCPA, and others whose purpose is protecting

[54] 'Privacy, Security and Deception', Google Play, https://play.google.com/about/privacy-security-deception/ (last accessed April 2020).

Figure 2 Permission scalation in Android: SDKs can leverage the same permissions as the host application to access protected resources. In this example, the app has access to unique identifiers, location information and the external storage. Both embedded SDKs (Google and Facebook) could access those resources without requesting the appropriate permission to do so

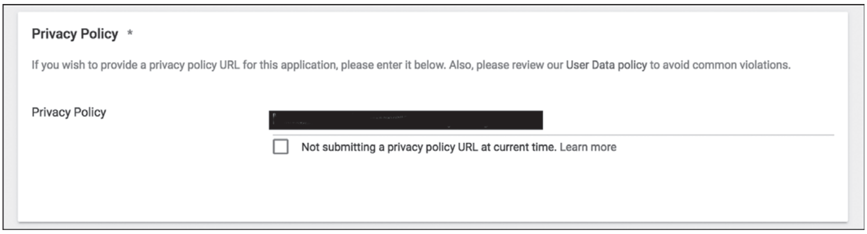


children's privacy like the Children's Online Privacy Protection Act (COPPA) in the US, developers can decide not to add a comprehensive and complete privacy policy when uploading an app to Google Play (see the screenshot of the process shown in Figure 3).

C. Transparency and Privacy Regulation

The GDPR regulates the way in which personal data from European citizens can be accessed, processed, and shared. All European users have the right to

Figure 3 Developers can decide not to include a privacy policy when uploading an app to Google Play



Privacy Policy *

If you wish to provide a privacy policy URL for this application, please enter it below. Also, please review our **User Data policy** to avoid common violations.

Privacy Policy [redacted]

□ **Not submitting a privacy policy URL at current time.** Learn more be informed about data collection practices by online services (such as apps) and – unless grounds such as legitimate interest exist – no data collection should be allowed before the user has granted explicit consent.

In most cases, app publishers (the controllers, according to the rule) are responsible for informing users about the presence of third-party libraries, the type of personal data that they collect and their treatment. Developers could be liable for any privacy malpractice inflicted by third-party SDKs present in their products. However, the SDK itself could be considered as the controller or joint controller of the data if the host application has nothing to do with the data collection process of the SDK. One example of this situation would be a third-party SDK that collects data in an application and uses it for different purposes than those originally intended, thus deciding the objectives and means of processing.[55]

As we discussed previously, Google Play only requires some applications to have a privacy policy link in the app profile in Google Play,[56] namely those falling in the Designed for Families (DFF) program and those that require access to permissions labelled as 'Dangerous' by Android's official documentation.[57] Dangerous permissions are those protecting the access to especially sensitive data, such as location or unique identifiers. On the other hand, DFF apps are those that target children and thus should be complaint with COPPA and GDPR provisions for children data usage.[58] However, there seems to be no control over the completeness and accuracy of the privacy policy content.[59] Moreover, it is possible for apps using non-dangerous permissions to also collect personal data that fall outside of the permission model of Android. This is the case of apps exploiting side channels and covert channels to circumvent the permission model, and those that collect personal data directly introduced by the user in the UI without accessing dangerous permissions.[60]

In the case of iOS, Apple recommends permissions only be requested when they are necessary for the correct functioning of the app, and that the permission request prompt comes with a clear text description of why a permission is needed.[61] Furthermore, in their app store guidelines, Apple states that all applications must include an easy-to-access link to their privacy policy, and that this policy must be complete and define all data collection and sharing practices.[62]

[55] 'Facebook loses Belgian privacy case, faces fine of up to $125 million', Reuters, www.reuters.com/article/us-facebook-belgium/facebook-loses-belgian-privacy-case-faces-fine-of-up-to-125-million-idUSKCN1G01LG (last accessed April 2020).

[56] 'Privacy, Security and Deception', Google Play, https://play.google.com/about/privacy-security-deception/ (last accessed April 2020).

[57] 'Permissions Overview', Android Developers, https://developer.android.com/guide/topics/permissions/overview (last accessed April 2020).

[58] 'Families', Google Play, https://play.google.com/about/families/ (last accessed April 2020).

[59] Okoyomon, 'Ridiculousness of Notice and Consent' (n 13).

[60] Reardon, '50 ways' (n 15).

[61] 'Requesting Permission', Apple Developer, https://developer.apple.com/design/human-interface-guidelines/ios/app-architecture/requesting-permission/ (last accessed April 2020).

[62] 'App Store Review Guidelines', App Store, https://developer.apple.com/app-store/review/guidelines/ (last accessed April 2020).

i. Vulnerable Populations

The GDPR has special provisions for protecting the privacy of children under the age of 16. In the US, the COPPA[63] regulates data collection practices for minors under the age of 13. Both rules require the app to gather verifiable parental consent before collecting any personal or behavioural data from children. There are different ways in which SDKs handle these special provisions. Some libraries directly state in their Terms of Service (ToS) that they are not suitable to be used by apps targeting a children audience,[64] while others integrate switches to adapt their behaviour when the developer states that the application is directed at children. There are several resources available for developers to choose libraries that respect legislation specific to children data. One remarkable example is Google's list of self-certified suitable for children libraries.[65] Unfortunately, it has been proven[66] that self-certification does not guarantee that SDKs are indeed complying with current legislation without external auditing and enforcement. Likewise, Apple provides recommendations for developers of applications that collect, store and process children data. Apple recommends that these applications avoid including third-party analytics and advertisement SDKS. If this were not possible, the developer must ensure that that embedded SDKs comply with any applicable privacy laws.[67]

III. Methods for Detecting the Presence of Third-Party Services

When installing an application, most mobile users do not realise that the app publisher might not be the only actor potentially collecting personal data from them. Previous research has shown that Android apps embed, on average, between six and nine third-party libraries.[68,69] In order to detect SDKs in mobile apps and analyse the behaviour, the research community has primarily used the following two methods.

[63] 'Children's Privacy', Federal Trade Commission, www.ftc.gov/tips-advice/business-center/privacy-and-security/children%27s-privacy (last accessed April 2020).

[64] Reyes, 'Think of the children' (n 22).

[65] 'Participate in the Families Ads Program', Google Support, https://support.google.com/googleplay/android-developer/answer/9283445?hl=en (last accessed April 2020).

[66] Reyes, 'Think of the children' (n 22).

[67] 'App Store Review Guidelines', App Store, https://developer.apple.com/app-store/review/guidelines/ (last accessed April 2020).

[68] Zhang et at, 'Detecting third-party libraries in Android applications with high precision and recall.', *IEEE 25th International Conference on Software Analysis, Evolution and Reengineering.*

[69] Razaghpanah, 'Apps, Trackers' (n 14).

A. Static Analysis

These techniques do not rely on running the software on a system but rather on analysing the code itself. This means that the analysis is generally easier to scale but also prone to failure due to code obfuscation and dynamic code loading.[70,71] Static analysis helps provide a higher bound on the data collection techniques of apps and SDKs, since it identifies every possible behaviour of a piece of software. Nevertheless, when the application is run with real user stimuli, it is possible that not every code path is executed or that pieces of the code are unreachable, thus generating a false positive. This often happens due to unfinished functionalities, dead and legacy code, and snippets copied from the internet and online development fora.[72] Examples of SDK detection tools that rely on static analysis are LibRadar, LibScout and LibPecker.[73,74,75] The problem with this approach is that it will be difficult to identify SDKs in applications using code obfuscation techniques. Many of these tools rely on fingerprinting SDKs to be able to detect them in APKs. Nevertheless, SDKs are in constant evolution[76,77] and, therefore, these fingerprints must be updated and maintained, or else the tool will become obsolete. Another detection tool that uses static analysis is Exodus,[78] which relies on matching code packages names and URLs found with the list of packages names and domains related to a given SDK provider (ie, the package com.crashlytics and the domain crashlytics.com would be matched to the Crashlytics SDK).

B. Dynamic Analysis

Tools based on dynamic analysis rely on running the software on an instrumented device to analyse the tested software's behaviour. Some tools rely on the analysis of information flows,[79] while others rely on intercepting and analysing the traffic generated by the software being tested.[80] While dynamic analysis provides actual

[70] Continella et al, 'Obfuscation-Resilient Privacy Leak Detection for Mobile Apps Through Differential Analysis', *The Network and Distributed System Security Symposium.*

[71] Faruki et al, 'Evaluation of android anti-malware techniques against dalvik bytecode obfuscation', *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications.*

[72] Felt et al, 'Android permissions demystified', *ACM conference on Computer and communications security.*

[73] Ma, 'LibRadar' (n 31).

[74] Backes et al, 'Reliable third-party library detection' (n 16).

[75] Zhang et al, 'Detecting third-party' (n 68).

[76] Calciati et al, 'What did really change with the new release of the app?', *Proceedings of the 15th International Conference on Mining Software Repositories.*

[77] Ren et al, 'Bug fixes, improvements, and privacy leaks', *Proceedings of Network and Distributed Systems Security Symposium.*

[78] 'What Exodus Privacy does', Exodus Privacy, https://exodus-privacy.eu.org/en/page/what/ (last accessed April 2020).

[79] Enck et al, 'TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones', *ACM Transactions on Computer Systems.*

[80] Razaghpanah et al, 'Haystack: In situ mobile traffic analysis in user space', *arXiv preprint arXiv:1510.01419.*

evidence of software behaviour (and, therefore, personal data dissemination), it is harder to automatise and test at scale due to the need to stimulate the device in order to thoroughly test the software.[81,82] Nevertheless, in the case of Android applications, there are tools for simulating user's interaction such as Apium,[83] Culebra Tester,[84] or the Android exerciser Monkey.[85] One instance of dynamic analysis used to detect SDK presence is the Appcensus platform, which runs apps in a highly instrumented version of Android and monitors access to personal data, permission usage and network traffic in order to understand what kind of personal data an app collects and who is responsible for such data collection.[86,87,88] While this solution only reports actual evidence of SDKs present in an app, it will not identify SDKs that do not generate traffic (such as UI, development support or cryptography libraries).

We note that there is a dearth of tools for studying SDKs present in iOS applications. Most of the academic efforts have been directed at understanding privacy aspects of Android applications. This situation is aggravated by the fact that many SDKs offer cross-platform support (Android, iOS, IoT, web), which gives them the ability to monitor users across all their device, and ubiquitously.[89,90]

IV. Comparison of SDK Analysis Tools

We perform a qualitative and quantitative analysis of the capabilities and limitations of four popular SDK analysis tools: three based on static analysis (LibRadar,[91] LibScout,[92] and Exodus[93]) and one based on dynamic analysis (AppCensus[94]). We focus on their ability to detect, classify and characterise SDKs embedded in the 50 most popular mobile applications published in Google Play.[95] Table 2 summarises the four detection tools and the capabilities that they implement.

[81] Reardon, '50 ways' (n 15).

[82] Reyes, 'Think of the children' (n 22).

[83] 'Mobile App Automation Made Awesome', Appium, appium.io/ (last accessed April 2020).

[84] 'Android UI Testing Simplified', CulebraTester, culebra.dtmilano.com/ (last accessed April 2020).

[85] 'UI/Application Exerciser Monkey', Google Developers, https://developer.android.com/studio/test/monkey (last accessed April 2020).

[86] Han, 'Comparing Free and Paid apps' (n 20).

[87] Reyes, 'Think of the children' (n 22).

[88] Reardon, '50 ways' (n 15).

[89] Brookman et al, 'Cross-device tracking: Measurement and disclosures', *Proceedings on Privacy Enhancing Technologies.*

[90] Zimmeck et al, 'A privacy analysis of cross-device tracking', *USENIX Security Symposium.*

[91] Ma, 'LibRadar' (n 31).

[92] Backes et al, 'Reliable third-party library detection' (n 16).

[93] 'What Exodus Privacy does', Exodus Privacy, https://exodus-privacy.eu.org/en/page/what/ (last accessed April 2020).

[94] Reyes, 'Think of the Children' (n 22).

[95] We acquired this list on 25 September 2019.

Table 2 Comparison of features in different SDK detection tools. ATS stands for 'Advertising and Tracking Services', and indicates that these methods group together these services in a single category.

	Analysis method	**SDK Detection**	**SDK Classification**	**Detects personal data dissemination**
LibRadar	Static	Yes	Yes	No
LibScout	Static	Yes	No	No
Exodus	Static	Yes	ATS	No
AppCensus	Dynamic	Yes	ATS	Yes

A. SDK Detection

LibRadar and LibScout detect SDKs at the code-level regardless of their purpose. LibRadar can identify 60 unique SDKs, while LibScout found 29. However, as discussed in Section III, the fact that a given library is detected using static analysis does not imply that it is causing privacy damage to end users. Consequently, in order to understand the privacy risk that a given SDK might pose it is important to distinguish between libraries potentially disseminating sensitive data from those that are simply making the development process of the application easier. It is, therefore, necessary to manually inspect and validate its output to eliminate false positives.

Exodus and AppCensus, instead, report those SDKs with hostnames associated to advertising and tracking services (ATS). More concretely, Exodus reports 67 seven libraries while AppCensus finds network flows attributed to 20 SDKs. This property makes these two options more suitable for privacy and regulatory auditing of Android apps. However, Exodus can still identify libraries that might be present on the code but not necessarily invoked at runtime, thus potentially rendering false positives.

B. SDK Classification

Only LibRadar aims to classify SDKs by their purpose. It considers the following categories: App Market, Development Aid, Development Framework, Game Engine, GUI Components, Map/LBS, Payment, Utility, Advertisement, Digital Identity, Mobile Analytics and Social Network. The classification performed by LibRadar is relatively complete, but it fails to capture the multi-purpose nature of many SDKs as we discussed in Section II B. In fact, most of the libraries that LibRadar can detect are classified as development aid (around 60 per cent), when many of them also provide analytics and advertising services as in the case of Unity 3D and Google's Firebase. Similarly, Facebook Graph's SDK is labelled as Social Networking, when it also allows app developers to integrate Facebook's ads and leverage their analytics services. Exodus and AppCensus do not offer any classification. Instead, they report SDKs associated with advertising, analytics, marketing and tracking services.

To compare the accuracy of the categories offered by LibRadar, we identified and visited the websites of each provider to manually identify their purpose according to the taxonomy introduced in Section II A. Unfortunately, we could not find any information for 20 per cent of libraries found by LibRadar (ie, LibRadar sometimes does not find the whole package name of libraries, making it impossible to match google.com to the appropriate Google SDKs or service included in the app) or we were not able to find the homepage of the library. To minimise human errors, several authors reviewed the output of this process, also putting it into the context of state-of-the-art research on Android privacy and third-party SDKs behaviour.

According to our classification, the majority (43 per cent) of the SDKs detected by LibRadar in our dataset of 50 applications can be classified as development support. This is followed by Analytics SDKs (12 per cent), Social Networking SDKs (7 per cent), Networking SDKs (5 per cent), Advertisement SDKS (4 per cent), and Online Payments (3 per cent). However, as we can see in Table 3 for the most popular SDKs detected by Exodus privacy,[96] LibRadar fails to capture the multi-purpose nature of many SDKs, including those collecting personal data which could be associated with advertising and tracking purposes. The most notable differences are Firebase, which is only labelled as Mobile Analytics by LibRadar, Facebook Ads, which is labelled as Social Network despite also being an advertisement network, and several analytics and advertisement services that are not included in LibRadar's fingerprints.

Table 3 Categories for the 20 most popular SDKs detected by Exodus Privacy across the 50 apps, and according to our manual classification

	Category			
SDK Name	**Manual**	**LibRadar**	**Exodus**	**AppCensus**
Firebase	Analytics, Development support, Database, Cloud, Push notifications, Authentication	Development Aid	Tracker	ATS
Crashlytics	Analytics, Development Support	Mobile Analytics	Tracker	ATS
Facebook SDK	Social network, Authentication	Social Network	Tracker	ATS

(continued)

[96] We use the results for Exodus because they only include tracker related SDKs, which are more relevant from a privacy and regulatory compliance point of view.

Table 3 *(Continued)*

SDK Name	Category			
	Manual	**LibRadar**	**Exodus**	**AppCensus**
Google Ads	Advertisement	Advertisement	Tracker	ATS
Google Analytics	Analytics	Mobile analytics	Tracker	ATS
DoubleClick	Analytics, Advertisement	Not found	Tracker	ATS
Appsflyer	Analytics	Mobile Analytics	Tracker	ATS
Google Tag Manager	Analytics	Mobile Analytics	Tracker	ATS
Facebook Ads SDK	Advertisement	Social Network	Tracker	ATS
Adjust	Analytics	Mobile Analytics	Tracker	ATS
Braze	Analytics	Not found	Tracker	ATS
Amazon Mobile Ads	Advertisement	Advertisement	Tracker	ATS
Appnexus	Analytics	Not found	Tracker	ATS
Moat	Analytics	Not found	Tracker	ATS
ComScore	Analytics, Advertisement	Mobile Analytics	Tracker	ATS
Mapbox	Maps	Not found	Tracker	ATS
Microsoft appcenter crashes	Development support, Analytics	Not found	Tracker	ATS
HelpShift	Analytics	Not found	Tracker	ATS
Demdex	Analytics	Not found	Tracker	ATS
MoPub	Analytics, Advertisement	Advertisement	Tracker	ATS

C. Understanding SDK's Data Collection Practices

In Section II, we discussed that the permission model of both iOS and Android fail to inform users on whether a given permission is requested by secondary purposes related to those of the SDK provider, including advertising and analytics. These two categories of SDK, together with Social Networks, account for 23 per cent of the total third-party components found by LibRadar.

In this section, we look beyond the limitations of mobile system permission models and discuss other potential privacy risks associated with each one of these categories. To do that, we leverage data from AppCensus, which reports the type of data collection by third-party SDKs in each app.

i. Social Networks

These SDKs represent a threat to privacy as they give social networks the ability to monitor users' activities outside of their own mobile applications. This means that social networks can potentially leak user data to the application developer or other third parties present in the app. For example, Facebook, through its own permission model, grants access to data such as the list of friends of the user or the pages that the user has liked on the platform.[97] Likewise, Twitter4J[98] allows developers to interact directly with the user's profile on the platform and Spotify allows gaining access to user data like gender, email account, or age. Cases such as the Cambridge Analytica scandal,[99] in which a political consulting firm got access to data from millions of Facebook users through a third-party app, highlights how dangerous social media data can become if it falls in the wrong hands.

ii. Analytics

Analytics SDKs serve different purposes and, as a result, their privacy risks can vary greatly depending on how app developers integrate them into their solutions. Some analytics libraries are used for user engagement; thus, they collect behavioural data that could be linked to a given user profile. Another example are A/B testing libraries, which rely on showing two different versions of an app component to different users and measuring which of the versions receives more positive interactions, which could reveal cognitive disabilities of the user.[100] All of these uses of analytics tools are legitimate and both apps and users might benefit from them, but the collection of such behavioural data linked to sensitive data (eg, particularly unique identifiers) should be informed to and consented by the user.

Some analytics SDKs allow developers to collect events defined by the application developer, known as 'custom events'. For instance, the developer of a medical app might want to monitor in their analytics dashboard the number of users showing certain symptoms in a geographical area. One library that allows for this kind of behaviour is Firebase, in which developers can register any event that they want to track even if it's not part of the events reported by the SDK by

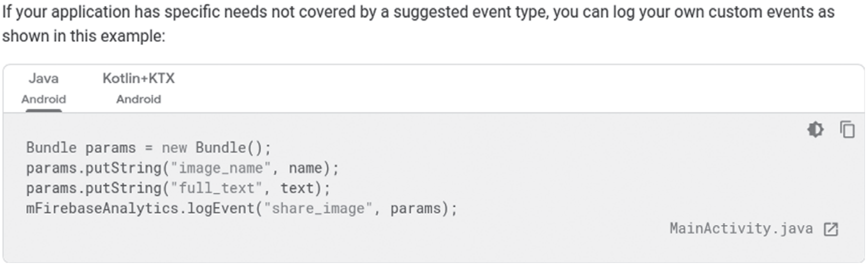
[97] 'Facebook Graph API', Facebook Developers, accessed April 2020, https://developers.facebook.com/docs/graph-api/ (last accessed April 2020).

[98] 'Homepage', Twitter4J, twitter4j.org/en/.

[99] 'Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach', *The Guardian*, www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election.

[100] Dekelver et al, 'Design of mobile applications for people with intellectual disabilities', *Communications in Computer and Information Science.*

Figure 4 Example of a custom event declaration with Firebase



If your application has specific needs not covered by a suggested event type, you can log your own custom events as shown in this example:

```
Bundle params = new Bundle();
params.putString("image_name", name);
params.putString("full_text", text);
mFirebaseAnalytics.logEvent("share_image", params);
```

default (see Figure 4 for a screenshot of this feature). This level of detail gives SDKs the ability to track users' every move and constitute a danger for their privacy, especially when analytics services collect information that can identify users uniquely. Using AppCensus data, we find 11 providers collecting different types of persistent unique identifiers (eg, AppsFlyer, Branch, Facebook, StartApp, Taplytics). While the majority (63 per cent) of them collect the AAID, a resettable user ID recommended by Google's policies, we find that four SDKS collect persistent identifiers like the IMEI or the Hardware ID. This behaviour is against Google's best practices[101] and defeat any privacy purpose of resettable IDs.

iii. Advertisement

Advertisement libraries collect personal data in order to show highly targeted advertisements to users and maximise revenue.[102] This brings severe privacy implications to users, with a high number of mobile SDKs collecting user data to create profiles and with the appearance of companies like data brokers,[103] which specialise in selling these types of profiles. Furthermore, because the advertisement model is highly distributed and dynamic, multiple ad publishers bid for the ability to show an ad to a given user depending on the personal characteristics of such a user.[104] This might result in user data being broadcasted to multiple organisations without the user knowing or consenting to as pointed out by ICO. Using AppCensus data, we observed seven advertisement libraries (including Verizon Ads and MoPub) collecting user identifiers (the AAID and the

[101] 'Best practices for unique identifiers', Android developers, https://developer.android.com/training/articles/user-data-ids (last accessed April 2020).

[102] Michael Plotnick, Charles Eldering, and Douglas Ryder, 'Behavioral targeted advertising', *U.S. Patent Application 10/116,692.*

[103] 'Time to Build a National Data Broker Registry', *The New York Times*, accessed April 2020, https://www.nytimes.com/2019/09/13/opinion/data-broker-registry-privacy.html (last accessed April 2020).

[104] Shuai Yuan, Jun Wang, and Xiaoxue Zhao, 'Real-time bidding for online advertising: measurement and analysis', *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising.*

persistent Android ID) as well as location data. As in the case of mobile analytics, targeted advertisement is not necessarily against the user's privacy if the user has consented to such data collection and if there are appropriate mechanisms in place so that the user can exercise the associated data rights.

V. Mitigating the Privacy Risks of SDKs

We discuss several steps that can be taken in order to mitigate the privacy risks of third-party SDKs discussed in the previous sections.

A. Improving Auditing Tools

As shown in Table 2, the current arsenal of tools for SDK auditing presents several shortcomings. Identifying and classifying SDKs by their purpose, while monitoring their behaviour and data collection practices, is indeed a challenging task. This is particularly challenging when analysing applications at scale. Even in this study, we were unable to match 20.5 per cent of the library code packages found just in 50 apps. This state-of-affairs puts users at a vulnerable position. Those users that want to exercise their data rights lack mechanisms to identify the organisations that have access to their personal data within their apps. Users have to rely on application developers accurately disclosing this information on their privacy policies (assuming that the SDK is fully transparent to the developer). Nevertheless, previous work has shown that these policies are often inaccessible, incomplete, and written in such a way that they are difficult to understand by the average user.[105,106]

There is, indeed, a need for more accurate software analysis tools to catch up with the evolving adTech industry and mobile technologies. These methods must be able to attribute observations to the responsible organisation at scale. Technical tools such as LibRadar,[107] Exodus[108] and Appcensus[109] have made positive steps to provide transparency in this complex and opaque ecosystem. They are important for users, app developers, regulators, and even privacy advocates because they ease the detection and understanding of the SDKs embedded in mobile apps, even for users without much technical expertise. Yet, they present many limitations as discussed in Section IV. We believe that static and dynamic

[105] Carlos Jensen and Colin Potts, 'Privacy policies as decision-making tools: an evaluation of online privacy notices', *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.*

[106] Okoyomon, 'Ridiculousness of Notice and Consent' (n 13).

[107] Ma, 'LibRadar' (n 31).

[108] 'What Exodus Privacy does', Exodus Privacy, https://exodus-privacy.eu.org/en/page/what/ (last accessed April 2020).

[109] Reyes, 'Think of the children' (n 22).

analysis techniques could be combined to develop more comprehensive, accurate and effective analysis tools, capable of overcoming the limitations inherent to each technique when used in isolation.

B. Application Developers and SDK Providers

Intuitively, an app developer will select the third-party SDK that provides the best functionality, but the question remains whether the SDK's resources usage or data sharing practices are ever considered in that decision-making process. It is imperative that developers understand the risks that many SDKs might bring to users' privacy. Application developers are liable for any regulatory violation that occurs within their application, even those inflicted by third-party components. With new legislation such as the GDPR, the fines can add up to €20 million or four per cent of the company's worldwide annual revenue from the preceding financial year, whichever is higher.[110]

It is critical that developers play a more central role in taking responsibility for their decisions to bundle third-party SDKs and that they follow the privacy-by-design principles. For that, it is necessary to also put SDK providers under scrutiny, demanding more transparency about their data collection practices and purposes and about their business models. This could be complemented with contractual agreements between app developers and SDK providers allowing developers to have additional legal guarantees and a better understanding of the privacy implications associated with a given SDK.

C. Stricter App Store Policies

Mobile platform providers should strive to audit applications and SDKs in order to improve and safeguard users' privacy. The Google Play store has checks in place to improve the privacy of applications. In early 2019, Google forced applications requesting call and SMS permission to either be the default messaging or calling app or to submit a special form explaining why such permissions are necessary for the app.[111] Furthermore, in newer Android versions (Android 10) Google has added restrictions for using unique non resettable identifiers (such as the IMEI) and for accessing alternate methods to infer location without requesting the appropriate permission.[112]

[110] 'What are GDPR fines?', GDPR.eu, https://gdpr.eu/fines/ (last accessed April 2020).

[111] 'Reminder SMS/Call Log Policy Changes', Android Developers Blog, https://android-developers.googleblog.com/2019/01/reminder-smscall-log-policy-changes.html (last accessed April 2020).

[112] 'Privacy in Android 10', Android Developer, https://developer.android.com/about/versions/10/privacy (last accessed April 2020).

Google has also published a list of self-certified third-party SDKs suitable for children's applications.[113] This list is a great resource for developers of children-oriented applications, as it reduces the search scope before making the decision to bundle a third-party SDK in their application. In this case, as these SDKs are self-certified, developers must trust that those components do indeed comply with existing regulation. There is no public information on whether Google verifies the claims made by each provider. Similarly, it is still possible to find applications that do not carry a privacy policy and examples of incomplete policies.[114] While we acknowledge that a thorough privacy analysis of all applications submitted to the market is a technically complex and costly task, we believe that these enforcement mechanisms could benefit from including some of the auditing techniques developed by the research community.

These policies are not specific to Android. Apple's App Store provides very exhaustive recommendations for app developers to minimize privacy damage to users.[115] These recommendations are focused on helping developers successfully pass their strict app review process prior to publication. In addition to minimum quality checks and recommendations – eg, releasing bug-free software and offering appropriate content – these guidelines also discuss the need for including complete privacy policies (data access, and third-party SDKs, data minimisation, and access to user data, among many others).

D. Changes in the Permission Model

Current permission models are app-centric by design so there is no separation between SDKs or apps accessing a given permission. Application developers only need to declare the permission in the app manifest file. Therefore, when a user grants the application permission to access a resource, the user has no information about whether an SDK or the app itself will exercise this permission. This goes in the opposite direction of current legislation, which is making strides towards better transparency and informing users about data collection and sharing practices, including the recipients of such data.

Android mentions the use of an explanation before requesting a permission as a best practice but does not enforce it in published apps.[116] If developers had to justify the inclusion of a permission request, users could make a more informed decision on whether to grant such permission or not. Additionally, the operating

[113] 'Participate in the Families Ads Program', Google Support, https://support.google.com/googleplay/android-developer/answer/9283445?hl=en (last accessed April 2020).

[114] Okoyomon, 'Ridiculousness of Notice and Consent' (n 13).

[115] 'App Store Review Guidelines', App Store, https://developer.apple.com/app-store/review/guidelines/ (last accessed April 2020).

[116] 'App permissions best practices', Android developers, https://developer.android.com/training/permissions/usage-notes (last accessed April 2020).

system could monitor and inform users whenever a protected method has been invoked by the actual application or an embedded SDK, and whether this has been disseminated to a server hosted on the internet. Unless users know which company is collecting personal data in an app, they will not be able to exercise their data rights per current legislation.

E. Certifying Bodies and Regulatory Actions

Trusted certification authorities could independently validate and certify the data collection practices of SDK providers. Article 42 and recital 81 and 100 of the GDPR propose ways to ensure compliance by controllers and processors through certification mechanisms and data protection seals or marks.[117] However, previous certification attempts such as COPPA's Safe Harbor have proven ineffective. As revealed by academic research, many applications certified by certification authorities still incur into potential violations of the COPPA law.[118]

It is unclear whether certification mechanisms can be effective in preventing deceptive behaviours and malpractices by third-party SDKs. The success of any certification scheme largely depends on the quality and depth of the certification process – ie, the length for which the process is going to make sure that the SDKs are in compliance with any regulations and for how long they will be able to bring out any potential violations. The use of auditing tools could play a fundamental role in the validation of the claims made by SDK providers from a technical standpoint.

Additionally, regulators must stay diligent and continue investigating any privacy malpractice on mobile applications. The FTC has previously acted towards SDK providers,[119,120] contributing to hold companies accountable when they do not respect users' privacy. These actions also have a valuable educational component. Once a regulatory action shows that a given company's behaviour constitutes privacy malpractice, other companies with similar policies might take additional precautions to protect their brands, reputation, and business, and avoid regulatory scrutiny and fines.

[117] 'Certification', Article 42, GDPR, www.privacy-regulation.eu/en/article-42-certification-GDPR.htm (last accessed April 2020).

[118] Reyes,' Think of the Children' (n 22).

[119] 'Video Social Networking App Musical.ly Agrees to Settle FTC Allegations That it Violated Children's Privacy Law', Federal Trade Commission, www.ftc.gov/news-events/press-releases/2019/02/video-social-networking-app-musically-agrees-settle-ftc (last accessed April 2020).

[120] 'Mobile Advertising Network InMobi Settles FTC Charges It Tracked Hundreds of Millions of Consumers' Locations Without Permission', Federal Trade Commission, www.ftc.gov/news-events/press-releases/2016/06/mobile-advertising-network-inmobi-settles-ftc-charges-it-tracked (last accessed April 2020).

VI. Conclusion

In this chapter, we have discussed the privacy risks and open challenges that SDKs bring to the mobile ecosystem. To illustrate our main points, we compared the SDK detection and classification capabilities of state-of-the-art tools in a group of 50 popular apps from Google Play. We show that, depending on their analysis methodology, auditing tools have different results in terms of the type and number of libraries that they detect. Furthermore, we show that there is a need to manually inspect results in order to better understand the nature and risks of these libraries. We argue that auditing techniques can benefit from the mixing of static and dynamic analysis in order to be more resilient against code obfuscation and attribution problems. We also show that most tools do not focus on classifying these libraries by their purposes and that those that do lack the ability to correctly classify third-party libraries that offer more than one similarity (eg, Unity 3D and Google's Firebase). We also showed empirical evidence of data collection by social networks, analytics, and advertisement third-party SDKs. We find 20 libraries collecting different type of personal and behavioural data (such as unique identifiers and location information). We argue that the collection of personal data by third-party SDKs can often be opaque for the end user. Therefore, we argue that the lack of understanding and awareness around the presence and data sharing practices of SDKS embedded in apps prevents users from making informed decisions. We conclude by discussing open regulatory and technological challenges and propose measures to alleviate this situation. Examples of these measures are modifying current permission models to accommodate SDK resource accesses, educating app developers on the importance of data minimisation, building more complete and robust auditing tools, having better auditing efforts from app stores and creating certifying bodies. These measures could help increasing transparency both for app developers, who would make better informed decisions when bundling third-party SDKs, and for end users, who would have more information at their disposal to decide whether they want to use a given application or not.

Acknowledgements

The authors would like to thank our anonymous reviewers for their valuable feedback that helped us to prepare the final version of this book chapter. This project is partially funded by the US National Science Foundation (grant CNS-1564329), the European Union's Horizon 2020 Innovation Action program (grant Agreement No. 786741, SMOOTH Project), the Spanish MINECO grant (TIN2016-79095-C2-2-R) and the Comunidad de Madrid co-financed by European Structural Funds ESF and FEDER (grant P2018/TCS-4566). The authors would also like to acknowledge the support and feedback provided by the Agencia Española de Protección de Datos (AEPD).

References

Backes, Michael, Sven Bugiel, and Erik Derr. 2016. 'Reliable third-party library detection in android and its security applications.' *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security.* 356–367.

Brookman, Justin, Phoebe Rouge, Aaron Alva, and Christina Yeung. 2017. 'Cross-device tracking: Measurement and disclosures.' *Proceedings on Privacy Enhancing Technologies.* 133–148.

Calciati, Paolo, Konstantin Kuznetsov, Xue Bai, and Alessandra Gorla. 2018. 'What did Really Change with the new Release of the App?' *Proceedings of the 15th International Conference on Mining Software Repositories.* 142–152.

Continella et al, 'Obfuscation-Resilient Privacy Leak Detection for Mobile Apps Through Differential Analysis', *The Network and Distributed System Security Symposium.*

Dekelver, Jan, Marina Kultsova, Olga Shabalina, Julia Borblik, Alexander Pidoprigora, and Roman Romanenko. 2015. 'Design of mobile applications for people with intellectual disabilities.' *Communications in Computer and Information Science.* 823–836.

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

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

Faruki et al, 'Evaluation of android anti-malware techniques against dalvik bytecode obfuscation', *IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications.*

Gamba, J, M Rashed, A Razaghpanah, J Tapiador, and N Vallina-Rodriguez. 2020. 'An Analysis of Pre-installed Android Software.' *S&P.*

Han et al, 'The Price is (Not) Right: Comparing Privacy in Free and Paid Apps', *Proceedings on Privacy Enhancing Technologies Symposium.*

Hu, Xuehui, and Nishanth Sastry. 2019. 'Characterising Third Party Cookie Usage in the EU After GDPR.' *Proceedings of the 10th ACM Conference on Web Science.* New York, NY, USA: ACM. 137–141. doi:10.1145/3292522.3326039.

Jensen, Carlos, and Colin Potts. 2004. 'Privacy policies as decision-making tools: an evaluation of online privacy notices.' *Proceedings of the SIGCHI conference on Human Factors in Computing Systems.* 471–478.

Ma, Ziang, Haoyu Wang, Yao Guo, and Xiangqun Chen. 2016. 'LibRadar: fast and accurate detection of third-party libraries in Android apps.' *Proceedings of the 38th international conference on software engineering companion.* 653–656.

Mojica, Israel J, Bram Adams, Meiyappan Nagappan, Steffen Dienst, Thorsten Berger, and Ahmed E Hassan. 2013. 'A large-scale empirical study on software reuse in mobile apps.' *IEEE software* (IEEE) 31: 78–86.

Okoyomon, Ehimare, Nikita Samarin, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, Irwin Reyes, Álvaro Feal, and Serge Egelman. 2019. 'On The Ridiculousness of Notice and Consent: Contradictions in App Privacy Policies.' *Workshop on Technology and Consumer Protection (ConPro).*

Plotnick, Michael, Charles Eldering, and Douglas Ryder. 2002. 'Behavioral targeted advertising.' *Behavioral targeted advertising.* Google Patents, 11.

Razaghpanah, Abbas, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Phillipa Gill, Mark Allman, and Vern Paxson. 2015. 'Haystack: In situ mobile traffic analysis in user space.' *arXiv preprint arXiv:1510.01419* 1–13.

Razaghpanah, Abbas, 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.'

Reardon, Joel, 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.' *28th {USENIX} Security Symposium ({USENIX} Security 19).* 603–620.

Ren, Jingjing, Ashwin Rao, Martina Lindorfer, Arnaud Legout, and David Choffnes. 2016. 'Recon: Revealing and controlling pii leaks in mobile network traffic.' *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services.*

Ren, Jingjing, Martina Lindorfer, Daniel J. Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. 2018. 'Bug fixes, improvements, … and privacy leaks.'

Reyes, Irwin, 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.' *Proceedings on Privacy Enhancing Technologies* (De Gruyter Open) 2018: 63–83.

Ruiz, Israel J Mojica, Meiyappan Nagappan, Bram Adams, and Ahmed E Hassan. 2012. 'Understanding reuse in the android market.' *2012 20th IEEE International Conference on Program Comprehension (ICPC).* 113–122.

Sanchez-Rola, Iskander, Matteo Dell'Amico, Platon Kotzias, Davide Balzarotti, Leyla Bilge, Pierre-Antoine Vervier, and Igor Santos. 2019. 'Can I Opt Out Yet?: GDPR and the Global Illusion of Cookie Control.' *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security.* New York, NY, USA: ACM. 340–351. doi:10.1145/3321705.3329806.

Sørensen, Jannick, and Sokol Kosta. 2019. 'Before and After GDPR: The Changes in Third Party Presence at Public and Private European Websites.' *The World Wide Web Conference.* New York, NY, USA: ACM. 1590–1600. doi:10.1145/3308558.3313524.

Valente, Junia, and Alvaro A Cardenas. 2017. 'Security & Privacy in Smart Toys.' *IoTS&P '17.* ACM.

Wang, Haoyu, Zhe Liu, Jingyue Liang, Narseo Vallina-Rodriguez, Yao Guo, Li Li, Juan Tapiador, Jingcun Cao, and Guoai Xu. 2018. 'Beyond google play: A large-scale comparative study of chinese android app markets.' *Proceedings of the Internet Measurement Conference 2018.* 293–307.

Wong, Janis, and Tristan Henderson. 2018. 'How Portable is Portable?: Exercising the GDPR's Right to Data Portability.' *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers.* New York, NY, USA: ACM. 911–920. doi:10.1145/3267305.3274152.

Yuan, Shuai, Jun Wang, and Xiaoxue Zhao. 2013. 'Real-time bidding for online advertising: measurement and analysis.' *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising.* 3.

Zhang, Yuan, Jiarun Dai, Xiaohan Zhang, Sirong Huang, Zhemin Yang, Min Yang, and Hao Chen. 2018. 'Detecting third-party libraries in Android applications with high precision and recall.' *25th International Conference on Software Analysis, Evolution and Reengineering (SANER).* IEEE.

Zimmeck, Sebastian, Jie S Li, Hyungtae Kim, Steven M Bellovin, and Tony Jebara. 2017. 'A privacy analysis of cross-device tracking.' *USENIX Security Symposium.* 1391–1408.