Hidden Permissions on Android: A Permission-Based Android Mobile Privacy Risk Model

Saliha Yilmaz and Mastaneh Davis
Kingston University, London, UK

k1939123@kingston.ac.uk
mast@kingston.ac.uk

Abstract: The continuously increasing amount of data input on mobile devices has made collating and monitoring users' data not only uniquely personalised but easier than ever. Along with that, mobile security threats have overtaken with rising numbers in bank fraud and personal information leaks. This suggests that there is a significant lack of awareness of security issues among mobile users. Specifically, permission-based passive content leaks are getting more attention due to the emerging issues in data privacy. One reason for this is that permissions are running in the background collecting and transmitting data between applications within the same permission group, without the user's knowledge. This means, that a supposedly innocent application like the Clock, which is linked with the Calendar to provide the date and time functionality, can have access to any other application within the same Calendar permission group, which is compromising confidentiality. Moreover, this can lead to a violation of data privacy as the user is not aware of which assets are being shared between permissions. Developers of mobile platforms have implemented permission-based models to counteract these issues, however, application designers have shown that they are not necessarily complying with the General Data Protection Regulations (GDPR). For the mobile user, this means that app developers, app providers, and third parties who are included in the applications, can gain access to sensitive data without user consent or awareness. To address this issue, this study examines permissions that are inherent in the Android mobile infrastructure and exemplifies how they can reveal delicate user information, identify user behaviour, and can be shared among other applications - without obviously breaching GDPR guidelines. 10 first-party Android applications were statically analysed by their permissions and manually investigated for their actual purpose and privacy risk. Finally, considering the affected area, these permissions were categorised into four asset groups that form the base of a risk model. With risk levels from low to high, this model provides detection of risks on data privacy in mobile permissions and highlights the difficulty with GDPR compliance, which we therefore named PRAM, a permission-based Android Mobile Privacy Risk Assessment Model.

Keywords: Android Permissions, Mobile Security, Data Privacy, GDPR, Privacy Risk Assessment, Data Intelligence

1. Introduction and Background

In the last decade, mobile devices have become increasingly intertwined with every aspect of our life due to the advancement of smart mobile devices providing functionalities similar to personal computers. As a result, the current mobile devices store and transfer more personal data such as location, personally identifiable information (PII), photos, etc. (Onik, 2018). In particular, the Android platform facilitates first- and third-party applications with application programming interfaces (API) that allow the applications to access information and resources of the system and device hardware (Almomani, 2020). This resulted in increasing the risk of leakage of users' personal information (Kim & Oh & Choi, 2023) and mobile malware attacks (Ashawa & Morris, 2020). Research has shown that the openness of the Android system and the popularity of Android mobile devices have driven the fast development of Android applications, whilst this also led to an increase in personal data leakage (Guaman et al, 2021). One of the main reasons for this is the use of "permissionware" – the end-user knowingly downloads mobile applications and accepts the permissions policies but generally finds it difficult to understand whether the apps access certain information in the background (Guaman et al, 2021). The permission system is fundamental to the operation of mobile apps with the primary purpose to control access to sensitive information and protect system resources. The permission selection relies on the users' ability to decide based on the application's functionality. However, in most cases, users are unaware that some applications may give additional permissions to unrelated parties (Alenezi. M & Almomani. I, 2017). In a recent study Alenezi and Almomani (Alenezi. M & Almomani. I, 2017), conclude that 80.3% of Android apps request more permissions than they need. As the permission system control accesses the sensitive information it creates an environment for attackers to perform malicious activities such as accessing users' confidential information. Furthermore, as part of an Android system, many of the third-party libraries are required access to the Android device information to obtain private data such as geographic information, network connectivity, and MAC address (Zhan et al, 2021). Hence, this may expose users' sensitive information to privacy risks without the users' knowledge or App developers' intentions.

An effective approach to protect mobile data privacy is to monitor, classify the permission behaviour, and assess the privacy risk. However, this process can be overlooked by users as the application permission was introduced for the application's accessibility to device information. In addition, some of the mobile permissions collect sensitive information without fully describing their scope (e.g., custom permissions). According to the official Android site, Android permissions are classified into three protection levels: normal, dangerous, and signature/signatureorsystem permissions. These permissions are set by developers in the manifest.xml file and due to the evolution of the Android operating system resulting in a more complex permission system, the number of normal and signature permissions has increased (Amlomani & Khayer 2020).

Several studies have analysed the privacy risk posed by Android applications (Hamed & Ayed, 2016), (Kim & Oh & Choi, 2023), (Gamba, 2019). The authors Kim & Oh & Choi (2023) have identified privacy indicators and based on them, introduced a model to calculate a privacy score for an Android application. However, the model requires various criteria to evaluate and quantify the objective information i.e., ability, benevolence, and integrity of personal data, and subjective indicators in the interaction between the service and its users, i.e., experience, reputation, and inclination. In another study by Gamba (2019), the privacy of the pre-installed Android applications has been analysed. The study shows that the Android permission system lacks transparency relating to data privacy enhanced by third parties and developers using custom permissions.

The European Union Agency for Network and Information Security (ENISA) addresses this in their privacy and data protection study regarding mobile applications: the European data protection framework defines 'personal data' as identified or identifiable information of a 'natural person', which is very broad (ENISA, 2018). Specific contextual and legal classification in which the information is being processed is needed for an elaborate analysis (ENISA, 2018). This circumstance makes it difficult to determine if mobile application developers' implementation as well as app data collection are following the GDPR guidelines or not. Ideally, the use of an app and processing its sensitive data for a particular specified purpose needs the explicit consent of the user, according to Article 9 of GDPR (ENISA, 2018). Our investigation however has shown that not all mobile permissions comply with this and already fail with the visibility and option to consent to the user. This is not only affecting the more vulnerable third-party applications but also pre-installed first-party apps, our study finds.

To highlight these privacy issues, we considered only pre-installed first-party applications as most research around privacy risk is concerning third-party applications. The thought behind this is that third-party apps are more prone to security vulnerabilities compared to first-party applications, which leaves the pre-installed apps to be overlooked. Therefore, this study focuses on investigating pre-installed Android application permissions as these apps are mostly installed as default on Android mobile devices. Another pivotal reason for us was that first-party applications are more commonly used compared to third-party applications and that regular users are unaware of the apps' possible privacy risks. An example of this is custom permissions, which are often included in applications. With custom permissions, developers can define their own permissions and share resources and capabilities with other apps (Sarkar et al, 2019). This is not only giving developers a leeway to access the functionalities of other applications but raises a privacy issue as upon our permission analysis, most custom permissions' purpose was unidentifiable. In addition, the pre-installed first-party apps may include third-party libraries which can violate user's privacy (Gamba, 2019).

Current research in this area has shown that there is a lack of privacy risk protection regarding mobile applications. This resonates in several areas of the mobile application infrastructure (Sarkar et al, 2019) which leaves the end-users' data vulnerable. While we are highlighting this angle on privacy issues emerging from mobile permissions and remedy the shortcomings of mobile security tools to identify permission vulnerabilities, we also propose a new approach to improve the efforts of application developers to address privacy issues.

2. Data Collection and Methodology

As all Android Operating System devices come with pre-installed applications, 10 of these apps have been used in this study on a mobile device with Google being the main developer (Table 1). These apps were selected on their 'essentiality' and daily use (e.g., Calendar, Clock, Camera). After downloading the applications and their Android Package files (also known as APK), a pen-testing tool named Mobile Security Framework (MobSF) was used to statically analyse the APKs (MobSF Documentation, 2022). We chose MobSF as it is – so far - the closest solution to an automated static analysis tool with a Graphic User Interface (GUI) which makes the data collection of applications easier and more transparent than other pen-testing tools.

The focus of this analysis is the mobile permissions of each application, which were manually collected and categorized into permission groups with a description of their purpose.

The following phases were conducted to disclose the permission misuse and create the variables for the Privacy Risk Model:

1. Display Information

2. Static Analysis
3. Manual Data Evaluation and Data Cleaning
4. Variable Categorization
5. Risk Modelling

The first phase was to investigate what the Android applications from a mobile device with an Android Operating System (e.g., tablet) displayed in the permission settings of the application user interface (see Table 1). The displayed permissions were then compared with the declared permissions in the Android manifest file of each application. For this comparison, the static analysis function of MobSF was utilized. The reason behind this comparison is to see whether the displayed permissions match with what is declared in the manifest file, and if not, to reveal any discrepancies.

We collected with phase A & B data from (so far) 10 applications, cleansed the data from duplicates, and evaluated the findings manually in phase C by investigating the purpose of each of the 351 permissions as these were not clearly described by the Android Developers Documentation. While we included a description of the purpose for each permission, we moved on to phase D and categorised our permission data into 8 variables. It is worth mentioning that our initial categorization of the permission display included 'functional permissions' (permissions, which are necessary to the functionality of certain application activities and somewhat visible to the user) making it a ternary variable but was later changed to the current binary for convenience.

Lastly, we assessed all the collected data and created categories to form a risk model for a permission-based risk assessment proposal in phase E.

3. Permission Analysis

Our initial proposal of the Permission-based Risk Assessment consists of five categories that we declared as the input variables for the privacy risk assessment. These are *Application, Displayed/Not Displayed, Permission Group, Android/MobSF category and Accessed Area.* The *Application* category contains the name of each application, the *Displayed/Not Displayed* are is a binary function (1 for displayed, 0 for not displayed, see figure 3 for an example of permission display), the *Permission Group* (created by us) is showing the affected features of the mobile device, the *Android/MobSF category* reflects the by default declared Android permission protection level and the last category – *Accessed Area* – was introduced by us to determine which data the permission is accessing.

Our initial results show that, after comparing the manifest file declared permissions of each application with the actual display in the permission settings of the mobile device, most permissions are not being displayed or made aware to the mobile user (Figure 1). Moreover, permissions, which are categorized as *dangerous* by Android constitute at least half of the total amount of permissions per analysed app (Figure 2). For a better understanding regarding the Android permission categories: currently Android is classifying its permissions as either *normal, dangerous,* or *signature/signatureorsystem* and denotes these as the *Android Protection Levels* (Android Developers Page, 2023). Normal declared permissions allow access to data and actions that present very little risk to the user's privacy while dangerous permissions give an app additional access to restricted data and allow apps to perform restricted actions that more substantially affect the system and other apps (Android Developers Page, 2023). Signature and SignatureOrSystem permissions can be used between two or more applications that have to share the same declared certificate to ensure privileged access. The reason behind this is that third-party app developers are not granted access to sensitive data. So, if app B wants to access/communicate with the signature permission included in app A, it must share the same signing key (authentication key). However, if two or more applications use the same signature permission these permissions are granted automatically by the system upon installation – without the user's consent (Android Developers Page, 2023).

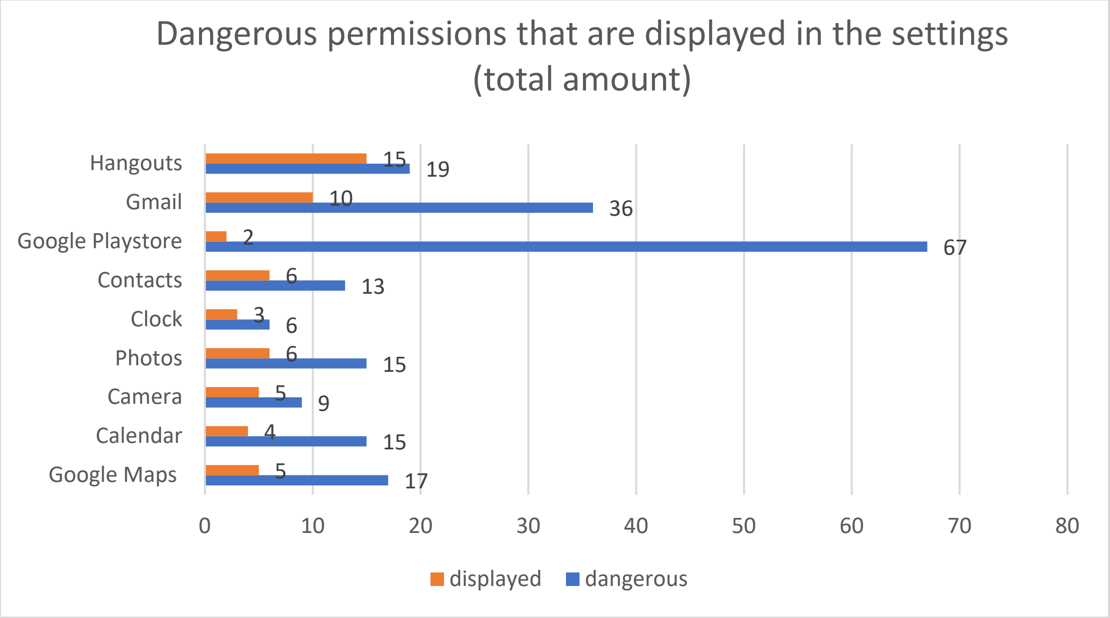


Figure 1: Dangerous permissions displayed in the settings of investigated applications.

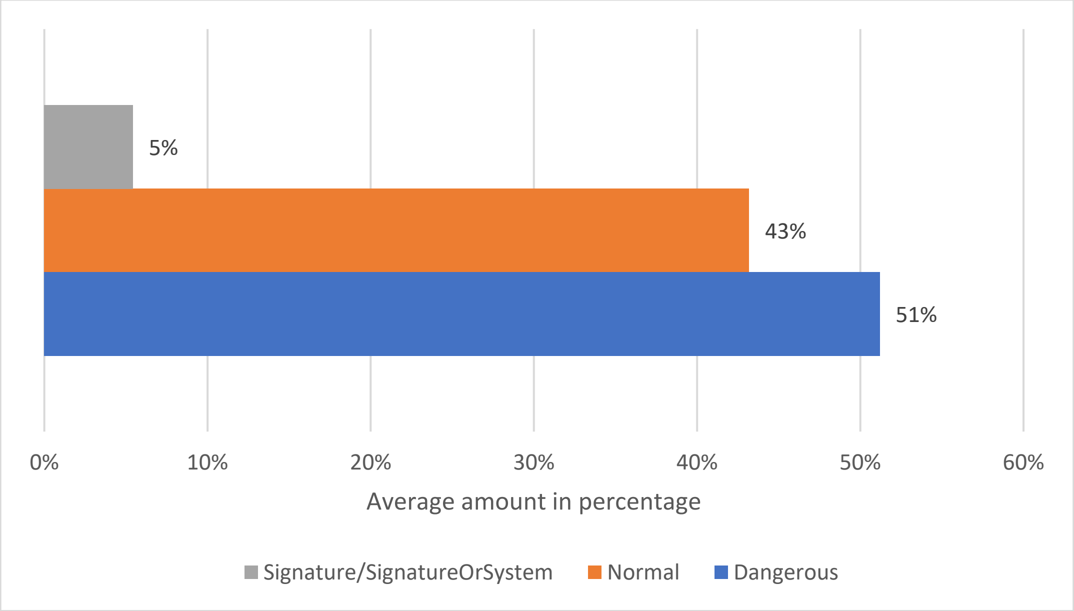


Figure 2: Average amount (%) of signature, normal and dangerous permissions in investigated applications.

When combining these two results, the outcome is that despite having the majority of permissions declared as dangerous (51% of all apps), most are not displayed (72% of all dangerous declared permissions) for the end-user in the mobile device settings. Google Maps for example has only 5 out of 12 dangerous permissions displayed (Table 1).

Table 1: Dangerous and displayed/not displayed permissions of the investigated permissions.

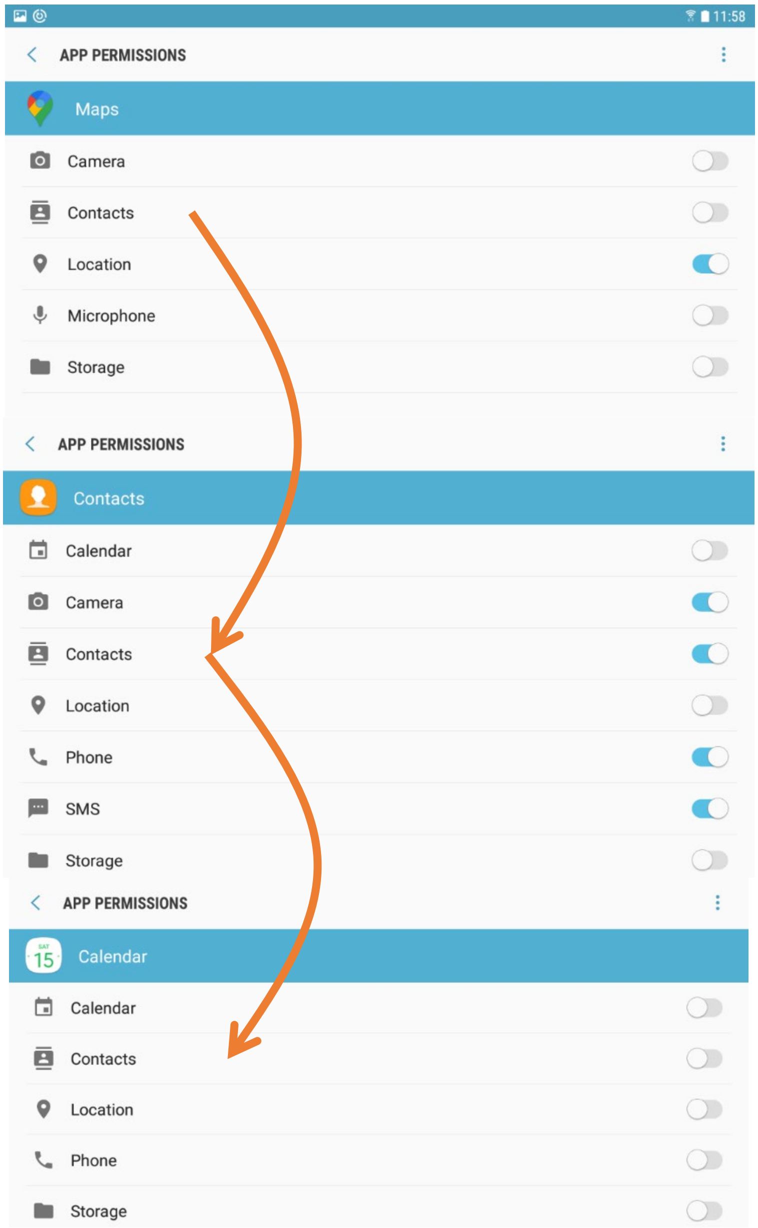
App	dangerou s	displaye d	not displayed	dangerous and not displayed in %	overall dangerous and not displayed
Google Maps	17	5	12	71%	6%
Calendar	15	4	11	73%	6%
Camera	9	5	4	44%	2%
Photos	15	6	9	60%	5%
Clock	6	3	3	50%	2%
Contacts	13	6	7	54%	4%
Google Playstore	67	2	65	97%	33%

App	dangerou s	displaye d	not displayed	dangerous and not displayed in %	overall dangerous and not displayed
Gmail	36	10	26	72%	13%
Hangouts	19	15	4	21%	2%
sum of all dangerous permissions	197	56	141		72%
		28%	72%		

There are also functional permissions, that are mostly categorized as normal according to the Android Protection Level. Although the use of functional permissions is conscious to the user, it is not really a choice since the functionality of many applications relies on allowing those permissions – receiving and sending emails would not work without network permissions nor would any map application without access to the location permission. As normal permissions are usually considered to present very little risk to the user's privacy, they can still raise privacy issues. Our analysis has shown that functional permissions such as `android.permission.INTERNET` (allows applications to open network sockets), which are considered as 'normal' under the Android Protection Level, can still be exploited by other apps either through permission misuse or malicious intent.

Further findings suggest that permissions not only run within one application but can access other applications' permissions in the background. Our research has revealed that these unauthorised hidden permissions run in the background collecting and transmitting data without the user's knowledge. This can be explained by Android's Intents, which are messages allowing application components to request functionality from other components within the Android system (Android Developers Page, 2023). When this happens, applications can request – by using Intents – a certain app (here: Google Maps) to start searching for a location or request directions from one location to another. In this example, Google Maps opens and starts to display the location specified in the Intent. The access to this resource is supposedly protected, apps requesting Google Maps to search for a specified location need to declare the "android.permission.ACCESS_FINE_LOCATION" permission.

The predicament here is, that the Android OS does not check if an app that is accessing a permission-protected resource through another app has itself requested that permission (Muttik et al, 2015). The developer of the app that is exposing access to the permission-protected resource is responsible for this check, which leaves a lack of control that can be used by apps to get access to sensitive resources without users' authorisation (Muttik et al, 2015). An example of this is presented in Figure 3, in which one of the tested applications demonstrates the connection chain to other applications in the permissions settings. This chain of collusion can be exploited and lead to information leaks and unauthorised access.



APP PERMISSIONS

Maps

- Camera ○
- Contacts ○
- Location ●
- Microphone ○
- Storage ○

APP PERMISSIONS

Contacts

- Calendar ○
- Camera ●
- Contacts ●
- Location ○
- Phone ●
- SMS ●
- Storage ○

APP PERMISSIONS

Calendar

- Calendar ○
- Contacts ○
- Location ○
- Phone ○
- Storage ○

Figure 3: Three colluding permissions that have the potential to access and share data.

4. Permission-based Privacy Risk Assessment proposal (in development)

As mentioned in Chapter 3 the five categories form the basis of our risk assessment model. Depending on the composition of the categories/variables the assessment generates a privacy risk score (Figure 4). So far, we have a scoring level of *low, medium, and high risk*, first for each permission then – depending on the outcome of all permissions within an app – a scoring for the whole application. A low privacy risk score generally indicates that the permissions of an application are declared as normal (present very little risk to the user's privacy according to Android), displayed on the device's settings, and have only access to a digital asset that does not comprise personal information of the user. A high privacy risk score would instead show that the mostly dangerous and/or signature permissions are accessing Personal Identifiable Information (PII) but are not made aware to the user in the device's settings (not displayed). In between these two scores, there is also the medium risk, which is composed of parts from both attributes.

In addition to this, we are currently in the process of including intertwined/colluded applications as a variable, which have the potential to access more within the allowed permission group (as explained in Chapter 3/Figure 3).

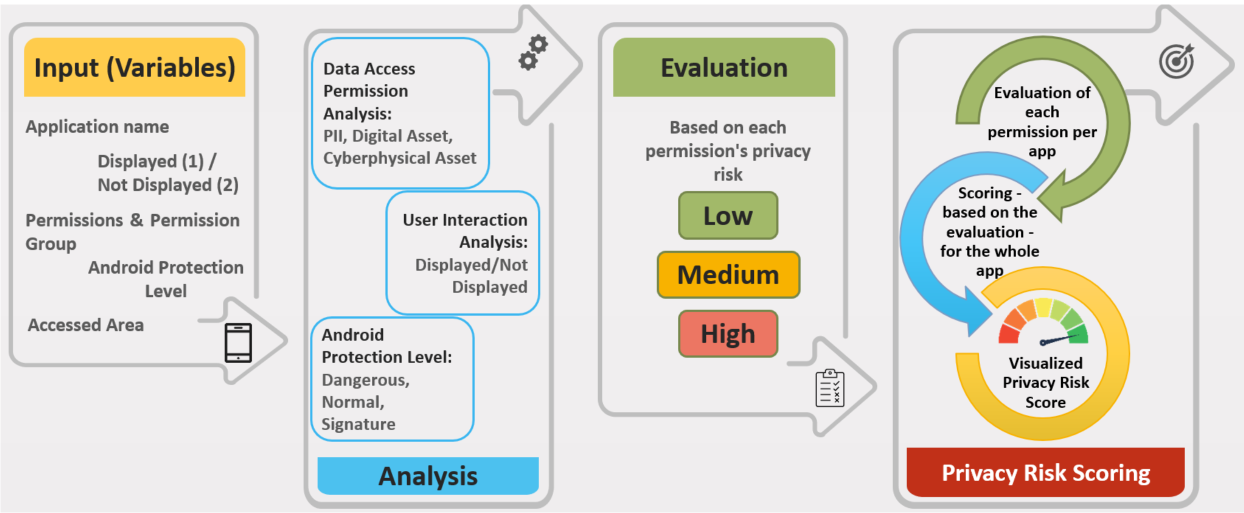

Input (Variables)
Application name
Displayed (1) / Not Displayed (2)
Permissions & Permission Group
Android Protection Level
Accessed Area
Data Access Permission Analysis: PII, Digital Asset, Cyberphysical Asset
User Interaction Analysis: Displayed/Not Displayed
Android Protection Level: Dangerous, Normal, Signature
Analysis
Evaluation
Based on each permission's privacy risk
Low
Medium
High
Evaluation of each permission per app
Scoring - based on the evaluation - for the whole app
Visualized Privacy Risk Score
Privacy Risk Scoring

Figure 4: Process of the proposed Privacy Risk Assessment Model.

5. Conclusion

The shortcomings caused by permission models like the least privilege principle violated by coarse-grained permissions (Bugiel et al, 2013), ignorance by end-users due to lack of permission awareness (Zhang et al, 2014), delegation attacks caused by enforcing access control policies at the level of the individual (Ham et al, 2013), and misusing/overusing permissions (Zarni Aung, 2013) have been broadly discussed in several publications. On the one hand, Android users may grant permissions without fully understanding the implications of their decisions and without the ability to balance functionality and privacy. This can lead to unintended outcomes. On the other hand, Android and application developers increase the number of permissions to improve security and ease the access and inter-communication between the applications but turn a blind eye to the flaws these upgrades have inherent. Furthermore, frequently over-privileged permissions are used by application developers without the user's understanding of their exact nature (e.g., custom permissions) with the tendency to increase with each system update.

This study highlights these flaws and investigates the privacy risks of permissions on individual as well as collective app level. So far, this research has shown that there are privacy concerns regarding custom and hidden permissions that have access to delicate information. This goes further with the colliding permissions between applications, in which one declared permission can access more information than initially allowed in the settings. Particularly, it has been shown that some pre-installed applications collect personal and sensitive information without the user's knowledge. Additionally, this study has demonstrated the lack of transparency of data privacy in current policies.

As a result of our investigation, this study introduces a new way of classifying application permissions, based on their purpose. Along with that, this work proposes a risk assessment model based on the aforementioned classification to provide a manually evaluated scoring that informs users and developers about the lack of privacy security and raises awareness of application permissions. Moreover, this risk model will include visualized results by assessing application permissions and informing developers as well as mobile users through a graph algorithm user interface. As an extension of this study, more testing of this model is needed to provide a better solution for the current permission policies on mobile devices. It is anticipated that this study will contribute to the discussion around mobile data legislation and help to address the privacy issues for policymakers.

References

Alenezi, Mamdouh & Almomani, Iman. (2017). Abusing Android permissions: A security perspective. 1-6. 10.1109/AEECT.2017.8257772.

Almomani, Iman & Al Khayer, Aala. (2020). A Comprehensive Analysis of the Android Permissions System. IEEE Access. 8. 216671 - 216688. 10.1109/ACCESS.2020.3041432.

Android Developers. (n.d.). *Android 6.0 Changes*. Available at: https://developer.android.com/about/versions/marshmallow/android-6.0-changes, [Accessed 29 June 2021].

Android Developers. (n.d). *Intent.* Available at: https://developer.android.com/reference/android/content/Intent, [Accessed 29 June 2021].

Android Developers. (n.d.). *Permissions overview*. Available at: https://developer.android.com/guide/topics/permissions/overview, [Accessed 29 June 2021].

Android Developers. (n.d.). *Privacy best practices*. Available at: https://developer.android.com/privacy/best-practices#permissions, [Accessed 13 May 2022].

Ashawa, Moses & Morris, Sarah. (2021). Android Permission Classifier: a deep learning algorithmic framework based on protection and threat levels. Security and Privacy. 4. 10.1002/spy2.164.

Aung, Zarni & Zaw, Win. (2013). Permission-Based Android Malware Detection. International Journal of Scientific and Technology Research. 2. 228-234.

Bugiel, Sven & Heuser, Stephan & Sadeghi, Ahmad-Reza. (2013). Flexible and fine-grained mandatory access control on Android for diverse security and privacy policies. Proc. 22nd Usenix Security Symp.. 131-146.

Xian, Zhan & Liu, Tianming & Fan, Lingling & Li, Li & Chen, Sen & Luo, Xiapu & Liu, Yang. (2021). Research on Third-Party Libraries in Android Apps: A Taxonomy and Systematic Literature Review. IEEE Transactions on Software Engineering. PP. 1-1. 10.1109/TSE.2021.3114381.

European Union Agency for Cybersecurity (ENISA), (2018). *Privacy and data protection in Mobile applications*. Available at: https://www.enisa.europa.eu/publications/privacy-and-data-protection-in-mobile-applications [Accessed 9 May 2023].

Gamba, Julien & Rashed, Mohammed & Razaghpanah, Abbas & Tapiador, Juan & Vallina-Rodriguez, Narseo. (2019). An Analysis of Pre-installed Android Software.

Guaman, Danny & Del Alamo, Jose & Caiza, Julio. (2021). GDPR Compliance Assessment for Cross-Border Personal Data Transfers in Android Apps. IEEE Access. PP. 1-1. 10.1109/ACCESS.2021.3053130.

Ham, You & Lee, Hyung-Woo & Lim, Jae & Kim, Jeongnyeo. (2013). DroidVulMon -- Android Based Mobile Device Vulnerability Analysis and Monitoring System. 26-31. 10.1109/NGMAST.2013.14.

Hamed, Asma & Kaffel Ben Ayed, Hella. (2016). Privacy risk assessment and users' awareness for mobile apps permissions. 1-8. 10.1109/AICCSA.2016.7945694.

Kim, Nakyoung & Oh, Hyeontaek & Choi, Jun. (2023). A privacy scoring framework: Automation of privacy compliance and risk evaluation with standard indicators. Journal of King Saud University - Computer and Information Sciences. 35. 10.1016/j.jksuci.2022.12.019.

Mobile Security Framework. (2019). MobSF Documentation. Available at: https://mobsf.github.io/docs/, [Accessed 29 June 2021].

Muttik, Igor & Blasco, Jorge & Chen, Tom & Kalutarage, Harsha & Shaikh, Siraj. (2015). Android – Collusion Conspiracy.

Onik, Md Mehedi Hassan & Al-Zaben, Nasr & Yang, Jinhong & Lee, Nam-Yong & Kim, Chul-Soo. (2018). Risk Identification of Personally Identifiable Information from Collective Mobile App Data. 71-76. 10.1109/iCCECOME.2018.8659213.

Sarkar, Anirban & Goyal, Ayush & Hicks, David & Sarkar, Debadrita & Hazra, Saikat. (2019). Android Application Development: A Brief Overview of Android Platforms and Evolution of Security Systems. 73-79. 10.1109/I-SMAC47947.2019.9032440.

Zhang, Yuan & Yang, Min & Xu, Bingquan & Yang, Zhemin & Gu, Guofei & Ning, Peng & Wang, Xiaoyang & Zang, Binyu. (2013). Permission Use Analysis for Vetting Undesirable Behaviors in Android Apps. IEEE Transactions on Information Forensics and Security. 9. 611-622. 10.1145/2508859.2516689.