A human oriented Privacy Impact Metric for mobile apps

Amador Aparicio[1], Javier Crespo-Guerrero[1], M. Mercedes Martínez-González[1], and Valentín Cardeñoso-Payo[1]

[1]Departamento de Informática, Universidad de Valladolid,47071, Valladolid, España
{amador,mercedes,valen}@infor.uva.es
javier.crespo.guerrero@estudiantes.uva.es

Abstract. Android is the operating system with the largest presence on mobile devices. The permissions mechanism is used to grant or restrict the access of applications to the device's data and resources. Applications request permission to access them and users decide whether to grant or deny them. Our proposal is to obtain a permissions-based metric, easy to use for device owners, to provide them with guidance on the risk to their privacy that they assume when they install an app on their device and how to minimize this risk. A distinctive feature compared to other proposals is that we use permission groups as one of the parameters. These permission groups express concepts that are more accessible to any type of user than individual permissions and are what users can actually act on. This has the advantage of being easier for users to understand. To facilitate its use, we have developed a service that allows you to consult it, but also to perform simulations to check how granting or denying each group of permissions requested by an application affects before making decisions and taking risks on the device itself. We thus introduce the criterion of usability, which allows us to obtain a more human technology, available to empowered users.

Keywords: Android · Privacy · Permission · Permissions Groups · Malware · Metrics · Security

1 Introduction

Android is the operating system for mobile devices with the largest market share, with three quarters of all smartphones worldwide[1]. In line with the Android security model [32, 18] the applications must request permission to access resources on the user's device [41]. Sometimes applications use more permissions than they actually need, which result in risks to users' privacy. Privacy is a person's right to confidentiality of their private information and identity [23]. As expressed in the General Data Protection Regulation (GDPR) [16], the Spanish Data Protection Agency (AEPD), in its clarification of the concept of personal data [38], states

[1] *Android Statistics (2023).* Available at https://www.businessofapps.com/data/android-statistics/

that personal data are considered to be "any information about an identified or identifiable natural person; an identifiable natural person shall mean any person whose identity can be determined, directly or indirectly". In the digital world, guaranteeing the right to privacy means protecting personal data, much of which is held on users' devices.

Permissions in Android are organized in groups. Each group contains a set of permissions related to accessing some data or resource on the device: camera, location, microphone, etc. If an application requires access to those data or to have control over the device's resources, the user will be explicitly explicitly ask the user for authorization to access those permission [13, 30]. It is at this point that users have the ability to make decisions to protect their privacy by granting or denying these requests. What users actually grant or deny on their devices are groups of permissions [2, 11]. Google provides guidance on what permissions should be included in each group [20], but it is the application developers who make the final decision for the apps they develop. This means that not all applications request the same permissions, even if they request the same groups of permissions. So it is interesting for a user to understand what the actual consequence of granting or denying a request is, and to be able to compare different apps. They could discover that applications on their device may make the same requests, but that these may have different impacts on their privacy.

Although there are proposals and services [36, 3, 37, 17, 33] to show and/or measure the privacy impact of Android applications, permission groups do not appear as a relevant parameter. However, in our opinion, it is the groups that have a direct impact on the usability and ease of understanding of the proposal by end users. That is why we have proposed a metric, and an associated ecosystem, that implements it and offers it as a service to end users. Our hypothesis is that a metric that is easy to understand by users, accompanied by a service that allows them to run simulations without compromising their device, can help them make better decisions about which applications to install and which requests to grant or deny.

This paper presents the metric *Privacy Impact Metric, (PIM)*, the service that facilitates its consultation, and the simulation referred to above. The metric returns a quantitative value, which is easy to understand, to score the privacy impact of an application. Graphical interfaces have been included in the service that implements it to make it easy to understand and for users to easily understand how this value varies depending on the similarity. In addition to the focus on the user interface, we also draw attention to the use of data integration techniques to provide the service with all the data needed to apply the metric. The metric uses app metadata, which are obtained from app repositories, such as the Android marketplace. It also uses data on the use of permissions by malware to find the weight these have in the final score, obtained from malware rankings. However, data are also retrieved from other sources to derive other privacy scores that we use to validate our metrics objectively.

Section 2 reviews other research on privacy and mobile devices. The proposed metrics, the methodology followed, and the service that supports users are presented in section 3. An important part is the validation of the results obtained, in order to improve the proposal. Section 4 compares the results of the proposed metric with other risk assessments done on the same applications. We approach these comparisons as a way to evaluate the quality of the metric objectively. Section 5, we discuss the results obtained, the advantages and limitations of the chosen approach, and the possible avenues of work that arise as a result of this analysis. Finally, we present the conclusions obtained, and the Future Work with which we will continue to advance on this proposal.

The contributions of this research are not limited to the metrics and the service that facilitates its use to users. Other results of this research are an exhaustive listing of Android permissions and their association to groups by default, i.e. when developers do not assign them to any group. This information is not available in the official Android documentation. The xml file with this list can be accessed through the official Android repository at Github[2] and/or through Zenodo[3].

2 Related work

Android has undergone significant changes over the last few years. These changes affect app versions, as they have to be updated to keep up with the constant evolution of the operating system. One consequence of the changes in app versions is that apps increase the number of permissions requested over time [7, 6]. Another consequence of Android's evolution is in its permissions system, where users are only informed an app's permissions when it is executed, but not at the time it is installed. This permissions model does not make users feel more secure compared to the previous model, where users were provided with information about permissions during app installation. [35]. Every time Google reviews the Application Programming Interface (API)[4] used by Android application developers Android apps also reviews the permissions and permission groups that applications can request. In this way, successive revisions of the API have resulted in new permission groups [28].

The permissions system in Android allows users to control the access of applications to certain functions or information on the device [12, 10]. When a user installs an app on their Android device, the app must request permissions to access the device's resources, such as the camera, or the user's personal data, such as contacts, location, and so on. Users have the option to grant or

[2] https://github.com/aosp-mirror/platform_frameworks_base/blob/master/core/res/AndroidManifest.xml

[3] https://zenodo.org/record/8013542

[4] The Android platform provides an API with a set of functions for applications to interact with the Android operating system. The API level is an integer value that identifies the API revision the API revision. https://developer.android.com/guide/topics/manifest/uses-sdk-element?hl=es-419

deny permissions to each app individually. This helps to protect user privacy and prevent applications from accessing sensitive information without the user's knowledg [21]. In addition, app developers must also comply with Google Play's permissions policies, which ensure that apps are secure and respect user privacy.

Starting with Android version 6, permissions requested at runtime are grouped into permission group. This was a significant advancement in Android's security and privacy model, allowing the user to grant or deny access to critical device resources and personal data [2]. When an app needs a specific permission within a permission group, the operating system prompts the user to grant the permission group, not the individual permission (Figure. 1), which has direct implications for user privacy [8]: if a user grants a single permission within a group, the app can silently request more permissions in this group with each update, without having to ask the user. This implies that permission groups can invade the users privacy, as an app can obtain dangerous permissions from the system without the user's consent. [31].

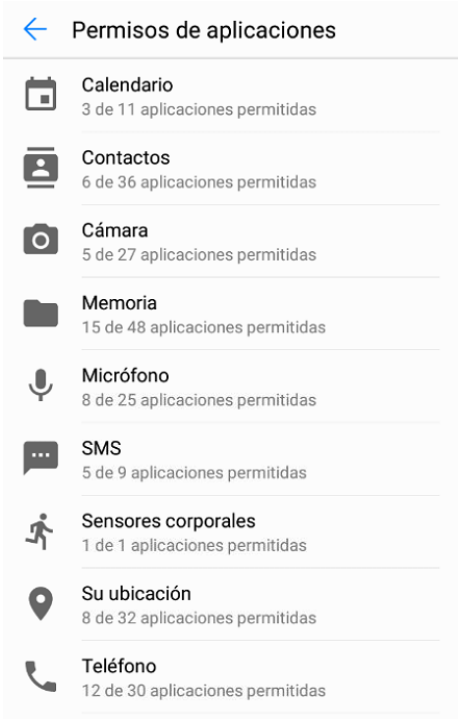

Fig. 1. Screenshot taken from an Android mobile terminal showing the permission groups.

Since Google does not provide official information about which permissions fall within a permission group [12, 19], nor on what level of protection a permit

must have to be part of a permit group [10], Iman M. Almomani and A. Khayer [1] obtained a list of permissions for Android API level 30 that includes 168 permissions defined by Android OS developers, ranging from API 1 (2008) to API level 30 (2020). They categorized the permissions with a level of protection *dangerous* in the permission groups proposed by Google [11]. The work is of interest because, although they do not use permission groups to derive a value indicating how permission groups impact user privacy, they do categorize permissions with a level of protection *dangerous* in the permission groups proposed by Android. The permission groups used have not been updated since API 31 was published, but according to its results, for a permission to be part of a permission group, it has to have a protection level of *dangerous*.

Given that the incorrect use of dangerous permissions can impact the privacy and security of users [40], Yang Wang, Jun Zheng, Chen Sun, and S. Mukkamala conducted a quantitative assessment of the privacy and security risks of Android permissions [44]. Their results show that malware is more likely to request more permissions than benign applications, and also more likely to request dangerous permissions. One result of this work is a ranking of the most frequently used permissions by benign applications and malware, which they use to quantify the risk of the applications, so that those requiring more permissions used by malware are considered more dangerous. To this ranking of permissions are added the permissions used by malware extracted from the source code of the app [15]. This is interesting because we have two sources of permissions: those present in the AndroidManifest.xml file and those present in the source code. It is possible that a permission declared within the code and exploited by the malware is not present in the AndroidManifest.xml file to evade malicious app classification systems used in major app markets. We consider the ranking of the permissions used by the malware useful because it shows that it also exploits normal permissions that the user cannot manage once the app is installed on the device and that affect their privacy.

This is why it is necessary to have privacy metrics in the Android application ecosystem to help assess and understand the risks and protection of users' personal data. Some metrics [27, 9, 20, 44] include the number and type of permissions requested by an application. Others need to install an app on the user's device to access the device's data and resources [34] and be able to determine the permissions used by apps to access sensitive information and critical device resources. From a privacy and information security point of view, we believe that it is not a good idea for users to have to install an app that can access critical system data and resources. Other metrics use data that are dynamically generated when the app is running on the device. [24], such as the network traffic it generates, domain names present in the requests made by the app, the IP addresses of the servers to which it connects, network protocols used and the information sent to it. [42]. Obtaining the information sent by the app is practically impossible when cryptographic protocols are used to protect these data. [4].

Given that mismanagement of permissions by users and that Android app developers do not always understand how permission groups and permissions that request user approval for access to sensitive data work, and declare more permissions than strictly necessary [14][26], there are tools that provide information on why an application may be intrusive or present security issues [43, 39]. However, these tools do not provide the user with the necessary information to know the real impact on the privacy of their data. [25], understand why this impact on privacy is occurring [29], or help the user manage app permission groups correctly so that apps have as little impact as possible on data privacy.

3 Proposal

3.1 Proposed methodology

Figure 2 represents the methodology followed. The rectangles represent actions while the arrows represent the results of the actions.

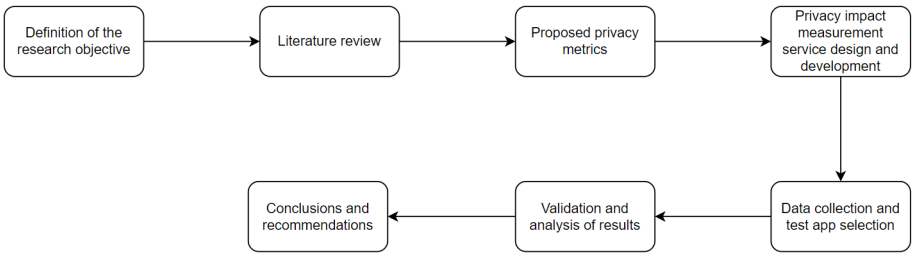

Fig. 2. Proposed methodology.

The steps are as follows:

1. **Definition of the research objective**. Establish the research purpose. In this case, it is to offer a service to end users so that they understand, in a simple and visual way, the impact of apps on privacy and what actions to take to minimize the say impact.
2. **Literature review**. Analyze previous research related to privacy impact measurement in apps and the metrics used. Identify existing privacy measurement tools, techniques and approaches and evaluate their effectiveness and limitations for our purpose.
3. **Proposed privacy metrics**. Establish metrics and evaluation criteria based on the key privacy issues identified in the literature that fit our goal, which is to empower end users.
4. **Privacy impact measurement service design and development**. Define the requirements and functionalities of the service for measuring the privacy impact of Android mobile applications. Design the architecture and

interface of the service, taking into account the collection and analysis of data relevant to measuring the privacy of applications. Implement the service using appropriate technologies and tools, ensuring robustness and scalability.
5. **Data collection and selection of test apps**. Select a representative sample of Android mobile apps as a case study. Collect relevant data, such as permissions used by the apps using static analysis techniques.
6. **Validation and analysis of results**. Validate the results by comparing them with other third-party evaluations.
7. **Conclusions and recommendations**. A determination is made as to whether the metric meets the goal of achieving the objectives. If the objectives are not met, return to the privacy metrics proposal phase.

3.2 Metrics developed

The proposed metric is based on the metadata present in the file *AndroidManifest.xml*. It is based on the following assumptions, which are obtained from the related work.

- The only permissions that users can grant are permissions of type *dangerous*.
- Permission groups are made up of permissions of type *dangerous*.
- The *malware* tends to ask for more permissions than benign applications.

Taking into account the above premises, we propose a metric that relates in a quantitative way the permits of type *dangerous* with the corresponding permission groups. It also takes into account the status for each permission group. This feature will allow users to know how the impact of apps on their privacy varies by activating or deactivating permission groups.

The formulation of the metric is as follows:

$$M(a_i) = \frac{\sum_{j=1}^{m}(e_j \sum_{k=1}^{q} p_{jk})}{I}$$

where:

- A: set of applications of the same category such that $A = \{a_1, a_2, , a_n\}$.
- n: number of applications present in the category A.
- a_i: application within the category such that $a_i \in A$.
- $M(a_i)$: value of the privacy impact of the application a_i.
- m: number of permission groups used by the application a_i.
- G_{ai}: set of permission groups used by the application a_i such that $G_{ai} = \{g_1, g_2, ..., g_m\}$.
- q: number of permissions within a permission group used by the application a_i.
- P_{a_i}: set of permissions used by the application a_i such that $P_{a_i} = \{p_{jk}/j = \{1, ..., m\}, k = \{1, ..., q\}\}$.

- p_{jk}: weight of the permit $k\_ésimo$ within the permissions group $j\_ésimo$ used by the application $a_i \in A$. Weights p_{jk} are assigned with the scores of the *ranking* used in [44]. If the permit is not in the *ranking*, the weight assigned to these permits is $\frac{1}{|P_D|}$, where $|P_D|$ is the number of permits of type *dangerous* proposed by Google [12]. Furthermore, in this way, the weight assigned to these permits will be less than the smallest weight present on the *ranking* of permits operated by the *malware*.
- e_j: permission group status $j\_ésimo \in \{0, 1\}$. 0 indicates that the permission group $j\_ésimo$ is inactive. 1 indicates that the permission group $j\_ésimo$ is active.
- I: maximum impact of an application accessing all available permission groups.

3.3 Proposed system

The proposed metrics will be used within a system that allows users to know how and why mobile apps impact their privacy. Figure. 3 shows the components of the proposed system along with the flow of data. It is divided into two distinct components: the service with which the user interacts and the manager that ensures that the service has all the necessary data (*raw data*) to calculate the metric. This decouples them, facilitating subsequent improvements in one or the other.

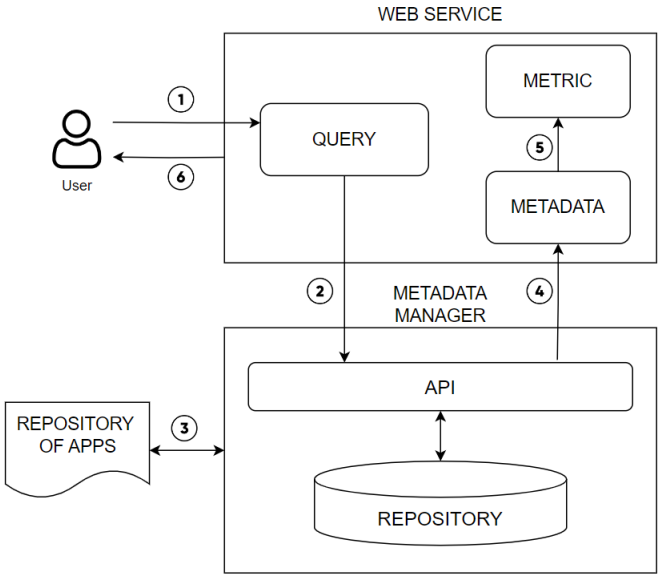

Fig. 3. System data flow.

The system components are as follows:

- **Web service**. Through it, the user informs the system of which app they want information concerning the privacy impact. It is divided into the following subcomponents:
  - **Query manager**. Receives information that identifies the app from which to obtain the metadata and requests it from the metadata manager.
  - **Metadata manager**. Fits the metadata sent by the metadata manager required for the metric.
  - **Metric calculator**. Calculates the value of the privacy impact and provides it to the user.
- **Metadata manager**. This receives from the web service the information concerning the app about which it has to send the metadata, so that the web service can calculate the privacy impact value. In case the metadata related to the requested app is not available, you can access the app repository can be accessed to obtain and store the metadata. It is divided into the following subcomponents:
  - **API**. Allows efficient communication and sharing of information between the web service and the metadata manager. Provides a layer to abstract from the type of repository used.
  - **Repository**. Stores structured app metadata in a lightweight, easy-to-read and write data interchange format.
- **Repository of apps**. External platforms on which the metadata manager can search and download the apps.

3.4 Data flow

The data flow from the time the system is asked to know the privacy impact of an app to the time the system provides these data is as follows:

1. The web service is provided with the app from which it wants to obtain the value of the privacy impact.
2. The web service asks the metadata manager for the metadata of the requested app.
3. The metadata manager returns the app metadata requested by the user to the web service. Optionally, if the metadata are not found in the data repository, the metadata manager accesses the app repository, extracts and stores the app metadata in the repository.
4. The web service applies the metric.
5. The web service provides the user with the app's privacy impact value, informs the user how permission groups impact their privacy so that the user can minimize the impact.

4 Validation and results

4.1 Sources of validation

Table 4.1 shows the three sources used to validate the results provided by the proposed metric.

Source	Type	Category	Access
S1	Paper	HEALTH AND FITNES	Hatamian, Majid, Nurul Momen, Lothar Fritsch and Kai Rannenberg. "A Multilateral Privacy Impact Analysis Method for Android Apps." Annual Privacy Forum (2019).
S2	Paper	PRODUCTIVITY GAME STRATEGY	Barth, Susanne, Menno de Jong, Marianne Junger, Pieter H. Hartel and Janina C. Roppelt. "Putting the privacy paradox to the test: Online privacy and security behaviors among users with technical knowledge, privacy awareness, and financial resources."
S3	Web service	COMMUNICATIONS	Terms of Service, Didn't Read https://tosdr.org/

Table 1. Description of the sources used.

These sources provide us with: references of apps on which to test the metric and values of the privacy impact of the apps already calculated to compare with the values returned by the metric we propose.

4.2 Source 1: HEALTH AND FITNESS

Table 4.2 shows the data provided by the proposed metric (PIM Metric column) when applied to the apps provided by the source S1 [22] (apps columns). It also shows the results related to the privacy impact of the apps it provides (PTaP privacy impact column). It is worth mentioning that the data provided by the metric (PIM Metric column) and the data provided by the source (PTaP Privacy Impact) are normalized between 0 and 1.

Apps	PIM Metric	PTaP Privacy Impact
com.runtastic.android	0.054	0.041
com.endomondo.android	0.325	0.250
com.fitbit.FitbitMobile	0.506	0.556
com.fitnesskeeper.runkeeper.pro	0.158	0.528
com.google.android.apps.fitness	0.047	0.278
com.myfitnesspal.android	0.267	0.250
com.sillens.shapeupclub	0.016	0.361
pedometer.steptracker.calorieburner.stepcounter	0.039	0.361

Table 2. Data provided by source 1 and the proposed metric.

4.3 Source 2: PRODUCTIVITY AND GAME STRATEGY

Table 4.3 shows the data provided by the proposed metric (PIM Metric column) when applied to the apps provided by source S2 [5] (apps column). It also shows the results regarding the privacy impact of the apps it provides (TUDelft column). It is worth mentioning that the data provided by the metric (PIM Metric column) and the data provided by the source (PTaP Privacy Impact) are normalized between 0 and 1.

App	PIM Metric	TUDelft
handsome.com.TodoList	0.008	0
com.iss.tasksplus	0.016	0.500
com.ListAndNote.gen	0.039	0.750
com.challengesinc.tdporject	0	0.250
com.ironhidegames.android.kingdomrush	0.024	0.750

Table 3. Data provided by source 2 and proposed metrics.

4.4 Source 3: COMMUNICATION

Table 4.4 shows the data provided by the proposed metric (PIM Metric column) when applied on the apps provided by the S3 source [**S3**] (apps column). It also shows the results related to the privacy impact of the apps it provides (Tosdr column). It is worth mentioning that the data provided by the metric (PIM Metric column) and the data provided by the source (Tosdr) are normalized between 0 and 1.

App	PIM Metric	Tosdr
whatsapp	0.379	0.500
signal	0.088	0.400
telegram	0.425	0.250

Table 4. Data provided by source 3 and the proposed metrics.

4.5 Validation of results

We have used the *Mann-Whitney Test* to compare two independent samples, consisting of ordinal variables, with a small sample size and unpaired samples

(different groups). We also compare the results obtained by our metric with those obtained by other researchers. The results indicate that they are similar enough to interpret that the scores we obtain with the group-based metric are sufficiently reliable[5].

5 Discussion

An accurate assessment of an app's privacy impact requires the data an app accesses to be compared with the purpose for which it does so, something that can only be done by reading and analyzing the app's privacy policy. This type of comparison usually falls to privacy impact assessment experts. We are therefore considering future collaboration with them to extend the validations with new datasets. However, given that what Android lets you manage are resources–through permissions–and not data, a metric can give an indicative value, useful for users who must make decisions based on hard-to-understand privacy policies. It is a complementary tool, which does not replace the privacy impact assessment that can be done by experts.

It is worth asking whether it would be appropriate to extend the metric using other parameters, from the perspective of a dynamic analysis, such as the observation of data traffic. As indicated in section 2, there are two reasons for not opting for this route. The first, and most fundamental reason, is that users cannot monitor or control the flow of data between external applications and systems. Our goal is to make proposals that help users where they have a decision-making capacity, which is not the case here. Secondly, as indicated in section 2, if data travel in encrypted form, it is impossible to know whether personal data are being exchanged or not.

A final consideration concerns the data from which the metrics are derived. As mentioned above, the service retrieves them from a warehouse that collects metadata from apps, malware and permission rankings, other data used to validate our results, and the results that the service obtains when evaluating an app. One question remains: what happens if the stored data differ from the data the service gets, either because it refers to a previous version, or for some other reason. To avoid integrity issues, it has been decided to consider each version of an app independently, so that collisions due to differences in the set of permissions requested by different versions of an app are avoided. As for the availability of the sources from which metadata are retrieved, this has been approached as an information integration problem, so the possibility of a source becoming unavailable has been foreseen, and can thus be easily replaced with an alternative source.

[5] The data that have been compared and the reports with the results are available at https://doi.org/10.5281/zenodo.8024477

6 Conclusion

The validations performed so far allow us to be optimistic about the direction taken. They indicate that restricting a metric to what users can control does not limit its quality as a support tool for users. Future work will expand the case studies to reinforce this conclusion. It is also planned to include tests with end users to assess their degree of satisfaction, i.e., to know from their experience as users whether the objective of providing a tool that helps them to empower themselves has been achieved.

References

[1] Iman M. Almomani and Aala Al Khayer. "A Comprehensive Analysis of the Android Permissions System". In: *IEEE Access* 8 (2020), pp. 216671–216688.

[2] Panagiotis Andriotis, Gianluca Stringhini, and Martina Angela Sasse. "Studying users' adaptation to Android's run-time fine-grained access control system". In: *J. Inf. Secur. Appl.* 40 (2018), pp. 31–43. DOI: 10.1016/j.jisa.2018.02.004. URL: https://doi.org/10.1016/j.jisa.2018.02.004.

[3] AppCensus. *Mobile App Analysis for Data Privacy*. URL: https://www.appcensus.io/.

[4] Susanne Barth et al. "Putting the privacy paradox to the test: Online privacy and security behaviors among users with technical knowledge, privacy awareness, and financial resources". In: *Telematics Informatics* 41 (2019), pp. 55–69. DOI: 10.1016/j.tele.2019.03.003. URL: https://doi.org/10.1016/j.tele.2019.03.003.

[5] Susanne Barth et al. "Putting the privacy paradox to the test: Online privacy and security behaviors among users with technical knowledge, privacy awareness, and financial resources". In: *Telematics Informatics* 41 (2019), pp. 55–69. DOI: 10.1016/j.tele.2019.03.003. URL: https://doi.org/10.1016/j.tele.2019.03.003.

[6] Paolo Calciati. "Understanding the Evolution of Android Applications". PhD thesis. 2019.

[7] Paolo Calciati and Alessandra Gorla. "How do apps evolve in their permission requests?: a preliminary study". In: *Proceedings of the 14th International Conference on Mining Software Repositories, MSR 2017, Buenos Aires, Argentina, May 20-28, 2017*. Ed. by Jesús M. González-Barahona, Abram Hindle, and Lin Tan. IEEE Computer Society, 2017, pp. 37–41. DOI: 10.1109/MSR.2017.64. URL: https://doi.org/10.1109/MSR.2017.64.

[8] Paolo Calciati et al. "Automatically Granted Permissions in Android apps: An Empirical Study on their Prevalence and on the Potential Threats for Privacy". In: *MSR '20: 17th International Conference on Mining Software Repositories, Seoul, Republic of Korea, 29-30 June, 2020*. Ed. by Sunghun Kim et al. ACM, 2020, pp. 114–124. DOI: 10.1145/3379597.3387469. URL: https://doi.org/10.1145/3379597.3387469.

[9] Chen et al. "Design and Implementation of Privacy Impact Assessment for Android Mobile Devices". In: 2016.

[10] Android Developers. *Android Basic Permission Types*. URL: https://developer.android.com/guide/topics/manifest/permission-element?hl=es-419.

[11] Android Developers. *Android Permission Group*. URL: https://developer.android.com/reference/android/Manifest.permission_group.

[12] Android Developers. *Android Permissions*. URL: https://developer.android.com/reference/android/Manifest.permission.

[13] Android Developers. *Manifest.permission_element*. URL: https://developer.android.com/guide/topics/manifest/permission-element.

[14] Mounika Deverashetti, Ranjitha K., and K. V. Pradeepthi. "Security analysis of menstruation cycle tracking applications using static, dynamic and machine learning techniques". In: *J. Inf. Secur. Appl.* 67 (2022), p. 103171. DOI: 10.1016/j.jisa.2022.103171. URL: https://doi.org/10.1016/j.jisa.2022.103171.

[15] Ibrahim Alper Dogru and Murat Önder. "AppPerm Analyzer: Malware Detection System Based on Android Permissions and Permission Groups". In: *Int. J. Softw. Eng. Knowl. Eng.* 30.3 (2020), pp. 427–450. DOI: 10.1142/s0218194020500175. URL: https://doi.org/10.1142/s0218194020500175.

[16] Unión Europea. *Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos (RGPD).* Disponible en https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/uri=CELEX:32016R0679.

[17] F-Droid. *F-Droid - Free and Open Source Android App Repository.* URL: https://f-droid.org/es/.

[18] Parvez Faruki et al. "Android Security: A Survey of Issues, Malware Penetration, and Defenses". In: *IEEE Communications Surveys & Tutorials* 17 (2015), pp. 998–1022.

[19] Paul Gerber, Melanie Volkamer, and Karen Renaud. "The simpler, the better? Presenting the COPING Android permission-granting interface for better privacy-related decisions". In: *J. Inf. Secur. Appl.* 34 (2017), pp. 8–26. DOI: 10.1016/j.jisa.2016.10.003. URL: https://doi.org/10.1016/j.jisa.2016.10.003.

[20] Google. *Android Open Source Project, AndroidManifest.xml.* Disponible en https://github.com/aosp-mirror/platform_frameworks_base/blob/master/core/res/AndroidManifest.xml (última visita: 9/3/2023).

[21] Google. *Guía de privacidad de Google.* Disponible en https://safety.google/privacy/ads-and-data/ (última visita: 10/3/2023).

[22] Majid Hatamian et al. "A Multilateral Privacy Impact Analysis Method for Android Apps". In: *Privacy Technologies and Policy - 7th Annual Privacy Forum, APF 2019, Rome, Italy, June 13-14, 2019, Proceedings.* Ed. by Maurizio Naldi et al. Vol. 11498. Lecture Notes in Computer Science. Springer, 2019, pp. 87–106. DOI: 10.1007/978-3-030-21752-5_7. URL: https://doi.org/10.1007/978-3-030-21752-5%5C_7.

[23] *International Electrotechnical Vocabulary.* URL: https://www.electropedia.org/iev/iev.nsf/display?openform&ievref=871-%2004-23.

[24] Haojian Jin et al. "Why Are They Collecting My Data?: Inferring the Purposes of Network Traffic in Mobile Apps". In: *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2.4 (2018), 173:1–173:27. DOI: 10.1145/3287051. URL: https://doi.org/10.1145/3287051.

[25] Georgia M. Kapitsaki and Modestos Ioannou. "Report and statistics Preprocessing phase Calculation phase Malware detection Tracker detection VirusTotal analysis storage Score calculation Figure 1 : Steps of Android

App Privacy Analyzer . 3 APPLICATION ANALYSIS PROCESS 3 . 1 Preprocessing Phase". In: 2019.

[26] Reinhold Kesler, Michael E. Kummer, and Patrick Schulte. "Mobile Applications and Access to Private Data: The Supply Side of the Android Ecosystem". In: *Innovation Law & Policy eJournal* (2017).

[27] Asma Khatoon and Peter M. Corcoran. "Android permission system and user privacy — A review of concept and approaches". In: *2017 IEEE 7th International Conference on Consumer Electronics - Berlin (ICCE-Berlin)* (2017), pp. 153–158.

[28] Jin-yung Kim, Yongho Yoon, and Kwangkeun Yi. "Analizador estático para detectar filtraciones de privacidad en aplicaciones Android". In: 2012.

[29] Minkyu Kim et al. "Analysis of Malicious Behavior Towards Android Storage Vulnerability and Defense Technique Based on Trusted Execution Environment". In: 2021.

[30] Li Li et al. "I know what leaked in your pocket: uncovering privacy leaks on Android Apps with Static Taint Analysis". In: *ArXiv* abs/1404.7431 (2014).

[31] R. Li et al. "Android Custom Permissions Demystified: From Privilege Escalation to Design Shortcomings". In: *2021 2021 IEEE Symposium on Security and Privacy (SP)*. Los Alamitos, CA, USA: IEEE Computer Society, May 2021, pp. 70–86. DOI: 10.1109/SP40001.2021.00070. URL: https://doi.ieeecomputersociety.org/10.1109/SP40001.2021.00070.

[32] René Mayrhofer et al. "The Android Platform Security Model". In: *ACM Transactions on Privacy and Security (TOPS)* 24 (2019), pp. 1–35.

[33] Mobsf. *Mobile Security Framework*. URL: https://mobsf.live/.

[34] Kelly E. Orjiude and Chika O. Yinka-Banjo. "A Multilateral Privacy Impact Analysis Method for Android Applications". In: *Annals of Science and Technology* 7.2 (2022), pp. 1–20. DOI: doi:10.2478/ast-2022-0005. URL: https://doi.org/10.2478/ast-2022-0005.

[35] Anthony S Peruma, Jeffrey Palmerino, and Daniel E. Krutz. "Investigating User Perception and Comprehension of Android Permission Models". In: *2018 IEEE/ACM 5th International Conference on Mobile Software Engineering and Systems (MOBILESoft)* (2018), pp. 56–66.

[36] Exodus Privacy. *La plataforma de auditoría de privacidad para aplicaciones Android*. URL: https://exodus-privacy.eu.org/en/.

[37] Privacygrade. *Android Network Tracing*. URL: privacygrade.org.

[38] Agencia Española de Protección de Datos. *Informe jurídico del reglamento general de protección de datos de interés legitimo*. 2019. URL: https://www.aepd.es/sites/default/files/2019-09/informe-juridico-rgpdinteres-legitimo.pdf.

[39] Falcon Sandbox. *Free Automated Malware Analysis Service*. URL: https://www.hybrid-analysis.com/.

[40] Gulshan Shrivastava et al. "Privacy issues of android application permissions: A literature review". In: *Trans. Emerg. Telecommun. Technol.* 31.12

(2020). DOI: 10.1002/ett.3773. URL: https://doi.org/10.1002/ett.3773.

[41] Christoph Stach. "How to Assure Privacy on Android Phones and Devices?" In: *2013 IEEE 14th International Conference on Mobile Data Management* 1 (2013), pp. 350–352.

[42] Android Network Traces. *Android Network Traces*. URL: http://android-network-tracing.herokuapp.com/.

[43] VirusTotal. *VirusTotal*. URL: https://www.virustotal.com/.

[44] Yang Wang et al. "Quantitative Security Risk Assessment of Android Permissions and Applications". In: *Database Security*. 2013.