
# Exploring decision making with Android's runtime permission dialogs using in-context surveys

**Bram Bonné, *Hasselt University - tUL - imec;* Sai Teja Peddinti, Igor Bilogrevic, and Nina Taft, *Google Inc.***

https://www.usenix.org/conference/soups2017/technical-sessions/presentation/bonne

**This paper is included in the Proceedings of the Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017).**

**July 12–14, 2017 • Santa Clara, CA, USA**

ISBN 978-1-931971-39-3

**Open access to the Proceedings of the Thirteenth Symposium on Usable Privacy and Security is sponsored by USENIX.**

# Exploring decision making with Android's runtime permission dialogs using in-context surveys

Bram Bonné
Hasselt University - tUL - imec
bram.bonne@uhasselt.be

Sai Teja Peddinti Igor Bilogrevic Nina Taft
Google Inc.
{psaiteja, ibilogrevic, ninataft}@google.com

## ABSTRACT

A great deal of research on the management of user data on smartphones via permission systems has revealed significant levels of user discomfort, lack of understanding, and lack of attention. The majority of these studies were conducted on Android devices before runtime permission dialogs were widely deployed. In this paper we explore how users make decisions with runtime dialogs on smartphones with Android 6.0 or higher. We employ an experience sampling methodology in order to ask users the reasons influencing their decisions immediately after they decide. We conducted a longitudinal survey with 157 participants over a 6 week period.

We explore the grant and denial rates of permissions, overall and on a per permission type basis. Overall, our participants accepted 84% of the permission requests. We observe differences in the denial rates across permissions types; these vary from 23% (for microphone) to 10% (calendar). We find that one of the main reasons for granting or denying a permission request depends on users' expectation on whether or not an app should need a permission. A common reason for denying permissions is because users know they can change them later. Among the permissions granted, our participants said they were comfortable with 90% of those decisions - indicating that for 10% of grant decisions users may be consenting reluctantly. Interestingly, we found that women deny permissions twice as often as men.

## 1. INTRODUCTION

Mobile users have an immense choice when searching for an app to install on their devices. Two of the most popular mobile platforms, Google's Android and Apple's iOS, each have more than a million different third-party apps that users can choose from [45], not to mention the additional third-party marketplaces. Users make a number of decisions during the lifecycle of an app on their smartphones, including deciding to install an app, making choices about whether or not to give an app access to personal data, and potentially uninstalling the app. There are many factors that could commingle to bring users to a decision. Part of the thinking around these decisions may involve reasons related to privacy, such as sensitivity to sharing particular types of data, trust in the developer, understanding the value added when personal data is shared, and many more [18, 19, 20, 29]. In order for an app to access personal data, both Android and iOS adopt a runtime permission model, which allows users to decide whether to grant a given permission request at the time when it is first needed within the app. In this paper we explore users' rationales for decision making during these three parts of an app's lifecycle, but with a focus on how users decide about permissions. Importantly, we study users' rationales at the moment they make their decision.

A large body of work has focused on understanding users' attitudes, comfort and their comprehension about permissions [2, 4, 15, 24]. However, almost all prior studies were conducted by using the permission model in which users had to accept or deny all the permissions requested by an app at installation time, without the possibility to grant permissions individually (for versions of Android before 6.0). A series of notable findings by Felt et al. [15] and Kelley et al. [24] showed that few users pay attention to install-time permission dialogs and even fewer understand them. Furthermore, results from other studies [2, 4, 15] indicated that users are often unaware of many permissions they have already granted to their apps. Subsequently, researchers started to advocate for a more contextualized permission model that would allow users to control permissions on a more granular level [13, 34, 48].

Android adopted the runtime permission model starting in version 6.0. There are at least two reasons why runtime dialogs have the potential to improve decision making by providing context. The first is that they often (but not always) clarify to the user why a permission is needed by linking it to the functionality that is triggered, because permissions are requested at the moment the data access is needed. The second is that developers can enrich the information shown in the permission request by providing their rationale[1], which can be considered as additional contextual information. While some developers take advantage of this, many still do not.

Given that most prior results were obtained for the old permission model, it is unclear to what extent they are still applicable to the current runtime model. In this work, we conduct the first study, to the best of our knowledge, that


*Symposium on Usable Privacy and Security (SOUPS)* 2017, July 12–14, 2017, Santa Clara, California.

[1] `https://developer.android.com/training/permissions/requesting.html#perm-request`

examines the reasons why Android users install or remove an app *at the time this happens*, and the motivation behind granting or denying a permission *right after users make their choice*. We are also able to examine users' reasons for each permission group, thus exploring if their reasoning differs when deciding on *location*, *microphone*, *contacts*, and other types of personal data. We capture users' comfort level with their choice both at runtime as well as after the study, which allows us to compare their comfort levels with their decisions both in-context as well as after the fact. Finally, we explore whether other factors, such as demographics, may influence user decision making. Although there exist prior works that studied users' permission choices with the runtime model [29, 30, 31, 49], their goals were not focused on users' rationales.

In order to answer these questions, we employed an open-source Android app called "Paco" [11] (Personal Analytics Companion), which is a platform for ESM (Experience Sampling Method) studies. We extended Paco to be able to query users about the reasons behind the decisions they make on their Android device related to app installs, permission decisions, and uninstalling apps, and made these extensions available to the broader research community. Paco allows us to capture the rationale behind users' actions in-the-moment, when it is fresh in their minds, therefore preserving as much of the actual context as possible. The 157 participants in our study installed Paco on their personal phones and used it for a 6-week period without any interaction with us. We collected over a thousand permission decisions and the associated reasons. Our study is the first, to the best of our knowledge, to collect such data in the wild.

Our main findings include the following. Many of our participants, when deciding about permissions, are indeed thinking about whether or not the permission is needed for the app or for a specific functionality, and whether the app "should" need it. This suggests that the context provided via runtime permissions appears to be helping users make decisions. Our participants accepted 84% of all permission requests, and among those they indicated they were comfortable (right after deciding) with their choice 90% of the time. The remaining 10% of grant decisions have a low comfort score, which suggests that a form of reluctance can occur when granting permissions. When we asked participants at the end of the six week period about some of their decisions, participants were not at all comfortable with 29% of them. We also noticed that the permission denial rates vary across different permissions. For example, microphone permission requests were denied almost twice as often as storage permission requests.

We identify decision rationales for 4 events types (app installation/removal, permission grant/denial) from Android users and rank them according to participant feedback. One of the most common reasons for denying permissions was that users know they can change it later. We further break down the reasons for denials per permission type and find that the dominant rationale for each permission type can differ – sometimes significantly – across permission types.

The remainder of the paper is organized as follows. We discuss related work in Section 2, introduce our methodology in Section 3, and we detail the implementation changes to the Paco app in Section 4. Section 5 presents the results about users' rationales for app installs and removals, and Section 6 discusses the findings about permission grant and deny decisions. We summarize and discuss our findings in Section 7.

## 2. RELATED WORK

Existing research has explored the space of Android permissions and privacy from two perspectives, that of users and developers.

From the user perspective, research has shown that few people actually read application permission requests and even fewer comprehend them [15, 24]. In fact, users were often surprised by the abilities of background applications to collect data [23, 44], and they were concerned when presented with possible risks associated with permissions [14].

To enhance the user experience, some have suggested providing users with more privacy information and personal examples to improve comprehension [18, 25]. Researchers have designed systems to identify privacy violations and to reduce them by recommending applications based on users' security concerns [1, 10, 16, 22, 26, 50, 51, 52]. Resource requests have been categorized into benign and dangerous requests, so that only the dangerous ones require user approval, thereby reducing the number of privacy/security decisions a user needs to take [13]. Some studies employed crowdsourcing to learn user expectations and to create privacy models [28], and others explored creating personalized assistants [30].

The research focused on developer behavior has shown that many developers are not deeply knowledgeable about permissions and often misuse them [42]. Intentionally or unintentionally, they are often making mistakes [39, 40] and are not following the principle of least privilege [47]. To identify this overuse behavior, tools have been developed that employ natural language processing of application descriptions [36], and static and dynamic analysis of Android apps [3, 6, 12, 41]. Further research efforts [10, 17, 37] that design methods to generally identify malicious applications have leveraged permission overuse assessments.

To improve the situation, researchers have suggested reorganizing the permission model with better definitions and hierarchical breakdowns [5], or adding fine-grained access control for better policing [9]. A recent study by Micinski et al. suggests there should be a difference between permission accesses that happen in the background and those that happen interactively (where the access directly corresponds to a user interaction, such as when the user imports their contacts). While the former should be granted explicitly (and regularly notified to the user), the latter should be avoided to prevent user fatigue [33]. Tools have been developed that dynamically block runtime permission requests [38], or that give users the ability to deny data to applications or to substitute user data with fake data [22].

We focus on three existing pieces of research that are closest to our work. In their 2013 work on Android install-time dialogs, Kelley et al. [25] examined the extent to which the design and type of information displayed in the dialogs helps users to choose which apps to install. Both our study and theirs ask participants about factors (such as developer, popularity, reviews, etc.) that influence their choice of which app to install. Interestingly, we find different results in terms

of the ranking of factors (as shown later in Section 5.2). We believe this may come from the different methods of testing, as well as the pre-Marshmallow[2] (theirs) versus post-Marshmallow (ours) permission model. A key difference between their study and ours is that they asked users to choose between pairs of apps for a friend (hypothetical scenario), whereas in our study users choose their own apps, in the wild, on their own devices.

Wijesekera et al. explored permissions in Android in two different studies [48, 49]. These studies explored how a contextualized permission model, based on the principle of *Contextual Integrity* [34] and work by Felt et al. [13], could improve dynamic permission granting. Both these studies rely on a custom version of Android 5.1.1 (pre-Marshmallow) as the study instrument, that logs every sensitive API access that requires a permission. Their first study [48] in 2015 measures how often and under what circumstances smartphone applications access protected resources regulated by permissions. They collected data on phones of 36 people about permission accesses when they happened. At the end of the week, they interviewed people, showed them screenshots of when data had been collected, and asked them if they would have liked to have prevented it (if they had been given the choice). They found that participants wanted to block 1/3 of permission accesses, citing privacy concerns over the data and lack of context about why the app needed the permission to perform its task.

In [49] the authors design a classifier to predict users permission decisions. The prediction takes into account context and generates predictions not only on-first-use, but also on subsequent uses when the context may be different. They postulate that users may not always elect to make the same decision about a permission each time it is used. They also make predictions as to when a user might change their mind, so that they do not ask on each use, but only on key ones where a user's decision may change (e.g. because of a different context). They used their predictor in a user study with 131 people and showed that they can do a far more accurate job of capturing user preferences this way than with the ask-on-first-use model. ("Ask-on-first-use" corresponds to runtime dialogs in versions of Android 6.0 or higher.) This work is very different from ours in that we do not build predictive models, and we are focused on understanding user *rationales* for decision making in the "ask-on-first-use" model. Our study also differs from all of these previous works in that we capture data "in the wild", meaning our participants used their own phones, their own choice of apps and interacted with their apps whenever they normally would.

# 3. METHODOLOGY

To capture users' reasoning when making privacy impacting decisions at the moment these are occurring, we use the Experience Sampling Method (ESM) [21, 27]. This method consists of asking individuals to provide systematic self-reports about their experience at random occasions during the day without the individual expecting it, often aiming to capture candid, in-the-moment experiences. Our methodology consists of surveying users at the time they are making privacy impacting decisions, by surfacing a survey when the participants install or remove an app, or when they change an app's permissions. We use the Android app Paco [11], which is part of an existing platform for ESM studies, and which can be downloaded from the Google Play store, as our study instrument.

In addition to the in-situ questionnaires, we ask participants to fill out an exit survey. This exit survey was used to gauge participants' privacy behaviors and technology savviness, and their awareness about permissions granted to apps on their devices. It also assesses how comfortable participants are with the permission decisions they made in the past.

Similarly to Wijesekera et al. [48], we avoid priming participants beforehand by publicizing the experiment as a study on app interactions, in order to limit response bias. No mention of privacy is made at any point during the study, except in the exit survey.

## 3.1 Designing the Surveys

We now describe the process we followed to design our in-situ and exit surveys (provided in full in Appendix A).

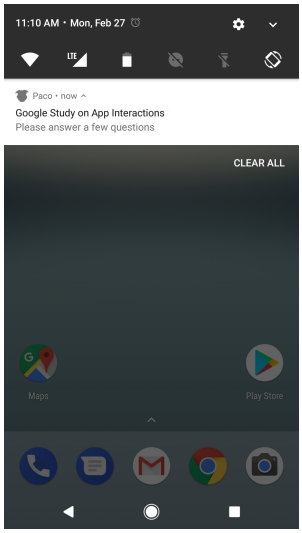
### *3.1.1 In-Situ Surveys*

The in-situ surveys are surfaced when one of the following four events occurs: the participant installs an app, removes an app, grants a permission to an app, or denies a permission to an app. In each of these cases, the participant is asked a question about his/her rationale for performing the action. In two cases, the participant receives a second question. After permission grant events, our second question aims to assess the participant's *reluctance* when allowing the permission, by asking to what extent they agree with the statement "I don't mind giving <app> access to my <permission>". App installation events also cause a second question to surface (after asking about rationales) that asks about the factors - such as app rating or popularity - that influenced their decision to install the app.

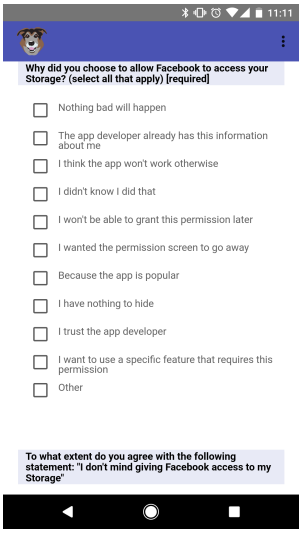
To capture the participant's decision rationales, we designed multiple-choice questions with the option to select multiple reasons, and with an additional "Other" choice allowing a free-form response. To ensure we have an exhaustive list of possible reasons, we first performed a short pre-study through the Google Surveys platform (GS), formerly known as Google Consumer Surveys (GCS). For each of the in-situ questions, we ask a random sample of 1000 participants about their reasons for performing a recent action. For instance, we asked "The last time you <did X>, what were your reasons for <doing X>?". We coded the different responses as follows. Initially two coders each coded half the responses and then cross-checked their responses. With over 90% overlap, they then independently completed the rest. The third coder independently coded responses using labels from the first two. Complete agreement was reached by all coders. Finally, we grouped answers with similar labels, and extracted the most representative answer from each of the top-10 largest groups.

Figure 1a shows how a participant is alerted that there is a question to answer, and Figure 1b shows a sample question for a permission grant request. In order to remove positional bias in the answers, we randomized the order in which the answer options were shown - with the exception of the "Other" option, which is always placed last. In order to reduce participants' response fatigue, we limit the number of questions that are surfaced to at most 3 permission events,

[2]"Marshmallow" refers to Android version 6.0




(a) Notification informing that a survey is available.




(b) Survey question soliciting the reasons for granting the Storage permission to an app.

Figure 1: Example of an in-situ survey in the Paco app.

2 app install events, and 1 app removal event per day, with a maximum total of 5 events per day.

### 3.1.2 *Exit survey*

In the exit survey, we question participants about their privacy behaviors, by asking about which privacy-enhancing practices they have employed in the past (compiled based on a Pew research survey [32]). Additionally, we ask participants to rate themselves on a 5-point scale from early to late adopters of new technology. Apart from these general questions, the exit survey also contains a personalized component. In this part, we ask participants about how comfortable they are with certain apps on their devices having access to a specific permission. These <app, permission> pairs are generated for each participant individually, by inspecting what permissions have already been granted for apps on their devices. These apps are not limited to the ones for which a permission is granted or denied during our study; they also include apps that were installed prior to enrolling in our study.

The personalized questions are worded as hypothetical scenarios, asking for example "How comfortable *would you* be with the <app name> knowing who is calling you". Moreover, the questions do not directly ask about the permissions, but rather about specific data access that this permission entails. For example, instead of asking about how comfortable the participant is with an app having storage access, we ask how comfortable they would be with the app being able to access pictures on their device. When answering such a question, participants are not informed that we selected a <app, permission> pair that exists on their devices. For each of the four permissions – Location, Contacts, Phone and Storage – we select a random app for which the permission was enabled (if available), and generate the corresponding question.

## 3.2 Recruitment and Incentives

Participants were recruited via our company's external U.S.-wide participant database and were sent a screening survey via email. We screened for participants using a device running Android version 6.0 or later, with their device locale set to "English - United States". (The latter requirement is needed because of the way we implemented our changes to Paco, see Section 4.2.) Participant diversity is controlled for gender, age, education and employment. Participant demographics are available in Table 1. After the recruitment phase, participants were informed that they would be required to install the Paco app. They were made aware about the fact that this app monitors their device usage to show survey questions, and were shown a list of all the data collected by Paco. Participants were told that for each of the 6 weeks they participate in our study, they would earn $10 and that submitting the exit survey would earn them an additional $20.

We recruited a total of 193 participants. Of these 193, 34 never finished the setup process and 2 voluntarily dropped out, so they are not included. The other 157 participated for the entire 6 weeks. Thirteen out of the 157 participants did not answer the exit survey, and have been excluded from parts of our analysis relying on exit survey data.

Table 1: Participant demographics

| Gender | Participants |
|---|---|
| Male | 79 |
| Female | 78 |

| Age | Participants |
|---|---|
| 18 - 23 | 29 |
| 24 - 30 | 44 |
| 31 - 40 | 35 |
| 41 - 50 | 23 |
| 51 or over | 26 |

| Education | Participants |
|---|---|
| Up to High school | 15 |
| Some college (1-4 years, no degree) | 40 |
| Associate's degree | 28 |
| Professional school degree | 5 |
| Bachelor's degree | 51 |
| Graduate Degree | 18 |

| Employment | Participants |
|---|---|
| Arts & Entertainment | 8 |
| Business & Finance | 6 |
| Education | 8 |
| Engineering | 12 |
| Health Care | 12 |
| Human Resources | 2 |
| Information Technology | 14 |
| Management | 19 |
| Miscellaneous | 15 |
| Religion | 3 |
| Retail & Sales | 17 |
| Retired | 5 |
| Self-Employed | 6 |
| Student | 18 |
| Undisclosed | 5 |
| Unemployed | 7 |

## 3.3 Ethical Considerations

In compliance with ethical training guidelines in our company, we ensured that participants' anonymity and privacy were respected. We thus carried out the following. First, all

researchers have been trained in ethical user research prior to this study. Second, there was an informed consent process where the participants were informed of all the types of data being collected before they start the experiment. Third, we deleted all the participants' personally identifiable information after the data collection period and thus did not use any of it in our analysis. Fourth, respondents had the option to exit the study at any point in time. Fifth, only the data from participants who completed the entire 6 week study is used in our analysis (data from the 2 who stopped participating is discarded). Lastly, as will be explained in Section 4, we implemented end-to-end encryption on top of Paco to make sure that all gathered data would be available only to the participants and the experiment organizers (and not, for example, to operators of the Paco service or other parties).

## 3.4 Limitations

Our analysis is based on participant self-report data, which is subject to biases such as social desirability and recall. Participation in our study requires installing our study instrument (Paco) and enabling *accessibility* and *app usage* permissions (see Section 4.2), hence our results could be skewed towards participants willing to do so; those unwilling to do so may have characteristics we did not discover. We try to limit such an effect by recruiting a diverse participant pool (controlled for gender, age, education, and employment) and by explaining upfront about all the types of data collected. Only 2 participants, out of 193, voluntarily dropped out of the experiment expressing concerns around the accessibility permission usage, so the effect is indeed limited. In order to limit the leading effect of our in-situ questionnaire towards participants' future actions on permission decisions or app installs, we imposed upper thresholds for the number of such questionnaires, which averaged at only 30 surveyed events per user over a 6-week period.

# 4. TECHNICAL IMPLEMENTATION

Our main survey instrument, the Paco app [11], acts as a behavioral research platform, which allows researchers to survey participants either at predefined intervals or whenever a specific action (such as an app install or permission-related decision) occurs. The advantage of using such an app is that we do not require participants to possess a rooted Android device.

Since Paco did not provide triggers for app installation or permission change events at the time of our study, we extended its code to provide such functionality. Moreover, to ensure that the participants' data is protected while in transit between the device and our servers, we also added end-to-end encryption to Paco. All code changes to Paco were submitted and accepted to the main project, and are now available to other researchers and the general public (Paco GitHub at `https://github.com/google/paco/`).

In addition to extending the Paco platform itself, we also modify the way in which surveys are shown to the participants by making use of Paco's scripting functionality. We discuss these implementations below.

## 4.1 App Installation and Removal Triggers

To identify the moments when a participant installed a new app, or when they removed an app from their phone, we listen for `ACTION_PACKAGE_ADDED` and `ACTION_PACKAGE_REMOVED` intents broadcast by the Android system's package installer, while making sure that these events are not part of a package update (by checking whether the `EXTRA_REPLACING` parameter is set). For both events, we store both the package name of the app and the user-friendly app name (henceforth referred to as app name). The package name is a text string unique to each application on the Google Play store, and is useful for our analysis, whereas the app names are more identifiable and are used in generating survey questions (see Section 4.3). An example package name is `com.rovio.angrybirds` and its app name is *Angry Birds*.

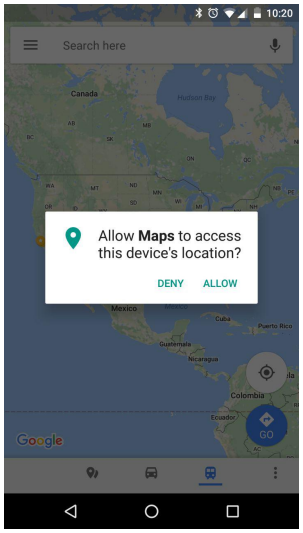
In case of an app installation event, the app name is available by querying the Android's package manager using the package name of the app. Since information about removed packages is no longer available in the package manager after an app is removed, we also manage a separate cache of package names and their corresponding app names. This allows us to access app names even after an app has been removed.

## 4.2 Permission Change Triggers

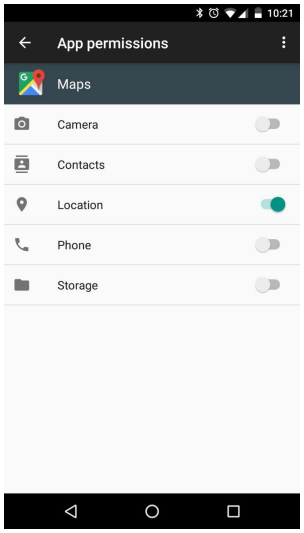
For permission change events, no intent is broadcast by the Android system, requiring us to monitor these permission changes ourselves. One obvious way to perform this would consist in periodically checking which permissions are granted to each of the apps installed on the user's phone, and looking for any changes in this information. This could be done by polling the Android package manager's `getInstalledPackages()` method and passing the `GET_PERMISSIONS` flag. However, a problem with this approach is that we would only detect permission *changes*, missing the case where the user has made a decision to remain in the same state as before. For instance, a user could deny a permission when it hasn't been granted before (permissions are set to deny by default when installing an app).

Because of the previous limitation, the permission change trigger is implemented as an accessibility service, which is used in Android to provide services (such as screen readers or special input devices) to people with disabilities. Because an accessibility service is able to inspect all text and user interface (UI) dialogs that are presented to the user, implementing such a service allows to analyze the text that is currently on the screen. We implement our own accessibility service to listen for events that correspond to the UI elements used for changing permissions. We then extract the text from these dialogs to determine the type of the permission and the app. We limit the accessibility service to only capture events from the `com.google.android.packageinstaller` and `com.android.settings` packages (which covers both the runtime permissions dialogs and the permission screen in the Android settings menu). This makes sure that our service does not needlessly slow down the system, and that it respects the participant's privacy by not collecting data beyond what is needed.

To identify the app for which a permission change event occurred, we query Android's usage statistics manager (this requires the *app usage* permission), determining the last active app that could have triggered a permission dialog to be shown. Because background services in Android are not allowed to request a permission, a permission dialog must always belong to the last active foreground app (if the package installer itself is excluded).




(a) The "Maps" app requesting the Location permission at runtime.




(b) Permission toggles for the "Maps" app in Android's settings.

Figure 2: Android's different methods for modifying an app's permissions.

Two different cases of permission change events are considered. The most common case is the one where an app requests a permission at runtime, either when it is first started or when the user wants to use a specific feature requiring the permission. An example of this case is depicted in Figure 2a, where the "Maps" app requests the Location permission. The second case is where the user actively changes an app's permission, by navigating through the Android's settings menus to either the screen containing all permissions for an app (see Figure 2b), or to the screen containing all apps that request a specific permission.

## 4.3 Generating and Surfacing Surveys

Paco allows to override the way in which surveys are generated and shown to participants, by providing experiment organizers with the ability to write scripts that will be used for generating both the notifications and the actual survey. For this study, we extensively make use of this functionality to dynamically generate questions. First, Paco's scripting functionality is used to comply with the study requirements for the in-situ questions outlined in Section 3.1.1. This includes overriding how often (and for which events) the user is notified, and randomizing the order of all survey responses except the "Other" option.

Furthermore, instead of relying on a predefined set of static questions, we generate them dynamically in order to provide more context to the participant (since the generated survey questions could be answered after a short time gap). For example, instead of asking "Why did you choose to allow the permission just now?", the participant is asked "Why did you choose to allow Maps access to your Location?".

Finally, the exit survey is also offered through Paco. This survey, too, depends heavily on dynamically generated questions. As discussed in Section 3.1, users are asked about how comfortable they are with their apps having access to data associated with a specific permission. These questions are generated for different <app, permission> pairs, where the permissions have already been granted for the app by the participant. For this purpose, the Paco app is extended with the functionality to pass on a list of all apps and their associated permissions to the script that is generating the surveys. This script selects one app for each of the four chosen permissions and generates the questions accordingly.

Table 2: Type and frequency of the different events considered by our study, and the number of events for which a participant was surveyed. See Section 3.1.1 for an explanation on survey limits.

| Event Type | Occurrences | Surveyed |
|---|---|---|
| App Installs | 3118 | 1913 |
| App Removals | 1944 | 775 |
| Permission Grants | 2239 | 1605 |
| Permission Denials | 437 | 272 |
| Total | 7738 | 4565 |

# 5. APP DECISIONS

## 5.1 Data Summary

We track four events in our study: app installs, app removals, permission grants, and permission denials. The total number of events that we recorded in our study are shown in Table 2. As mentioned in Section 3.1.1, we enforce limits on the number of events we survey each day. As a result, not all recorded events are surveyed. Our 157 participants triggered 3118 app install events (of which 1913 are surveyed), and 1944 app removals (of which 775 are surveyed). The apps could have come from either the Google Play store or from other sources. On average each participant installed 20 apps and removed 12 apps during the 6 week period. We note that a participant can install and remove the same app multiple times, and each of these actions would be recorded as a separate event. An app removal event could have occurred for an app that was installed prior to our study, and thus does not necessarily correspond to one of the app install events we observed.

We clarify that the Paco tool recorded all events (not only those surveyed) for all of the 4 event types that occurred on participants' phones during the 6 week period. Based on the complete set of user permission decisions, we observed an overall grant rate of 84% and a denial rate of 16%. Due to our self imposed limits on the number of surveys shown per day, we ended up asking survey questions for 72% of the grant events and 62% of the denial events. For the surveyed responses, we find the grant rate to be 86% (with corresponding denial rate of 14%). Thus the grant and denial rates of our surveyed (i.e., sampled) events is very close to the rates for the total occurrences. Out of the 157 participants, 144 answered the exit surveys. In the rest of the paper, we present results for the surveyed events to ensure consistency with results about participant responses.

In Figure 3, we show the activity level of our participants with our surveys. Most answered at least 10 surveys, and some have answered many more.

## 5.2 App Installs

After installing an app, our participants were asked to select which factors (all that apply) influenced their decision to install the app. These results are shown in Figure 4.

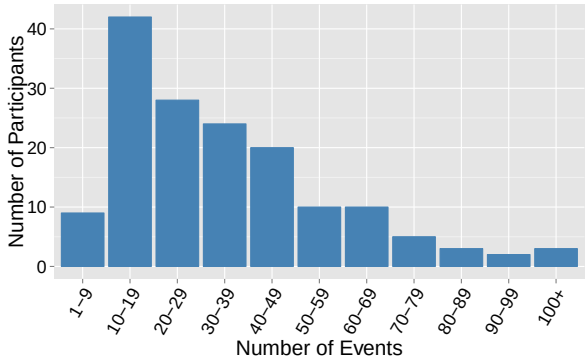



Figure 3: Event distribution across Participants

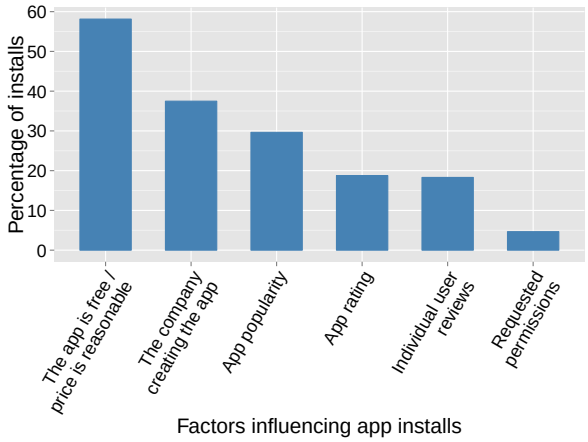



Figure 4: Factors impacting app installation (multiple responses per installation event are possible)

As expected, we observe that price is the dominant factor. What is somewhat surprising is that the company creating the app (i.e. the developer) is the second highest factor, even more important than an app's popularity. Among these six factors, permissions occur the least frequent, and only directly affect 5% of app installation decisions. This is not surprising, because with the runtime permissions model participants do not see the permission requests during the installation flow[3], and thus users are unlikely to think about permissions at that moment. However, these install events – when participants selected permissions as a factor – came from 33% of our participants; this indicates that permissions influenced one third of our participants at least once during app selection. Note that app ratings and reviews can be influenced by privacy concerns around permissions, and thus this 5% metric should actually be treated as a lower bound in terms of its ability to capture the relevance of permissions for app installation.

Our observation about the influence of permissions at installation time corroborates the finding in [25], where permissions ranked 8th out of 11 reasons. However, our findings about the influence of reviews and ratings differ significantly from those in [25] (see Figure 2 therein). They found that ratings, reviews and cost were most important (in that order) and of similar importance, whereas in our study developer and popularity were factors cited more frequently than ratings and reviews. This could be due to different study methods. They asked 366 MTurkers to rate factors on a 5-point importance scale, whereas we asked participants to select all that apply. Moreover, the MTurkers in [25] were asked about their general views, whereas our participants were asked about specific apps right after installation. This suggests that an interesting avenue for future research would be to understand if and why the influence of reviews and ratings are evolving.

[3]Some older apps that do not target an Android API level of 23 (Marshmallow) or above, and that are not yet updated to use the new permissions model, could still show a list of requested permissions at install time.

Table 3: Reasons participants checked for app installation (multiple responses per installation event are possible)

| App Install Reason | Number of Occurrences (% of install events) |
|---|---|
| I want to try it out | 954 (49.9%) |
| The app is useful | 579 (30.3%) |
| The app is part of a product/service that I use | 500 (26.1%) |
| The app is cool or fun to use | 400 (20.9%) |
| I trust the app or the company making the app | 310 (16.2%) |
| My friends/family use it | 276 (14.4%) |
| It was the only app of its kind (no other apps provide the same functionality) | 160 (8.4%) |
| Other | 129 (6.7%) |
| I was required to install it | 126 (6.6%) |
| I was offered something in return (e.g. credits, monetary rewards, discount) | 79 (4.1%) |
| The app has fewer permissions than other apps like it | 34 (1.8%) |
| I don't know | 34 (1.8%) |

Table 3 shows the reasons why users install particular apps. For each reason, the percentages indicate the proportion of install events (total events counts in Table 2) it was selected for. The reason "I want to try it out", that may capture curiosity, dominates the list and is selected in 50% of installations as a reason. The other popular reasons "The app is useful" and "The app is cool or fun to use" stress that the app's functionality plays an important role as well. We found that only 14% of the installs had social influences such as family and friends. Only 34 times (2% of the surveyed installations) did participants indicate that they compared the number of permission requests across apps before installing. However, these 34 instances originated from 15% of our participants. We hypothesize that permissions may not be a key reason at moments of installation because Android users are aware that in the runtime permissions model they can make decisions about permissions later when using the app. In Section 6.1, we see this partly confirmed since for 40% of instances when denials occurred, participants said they did so because they can grant these permissions later.

Table 4: Reasons participants checked for app removal (multiple responses per removal event are possible)

| App Removal Reason | Number of Occurrences (% of removal events) |
|---|---|
| I no longer use the app | 307 (39.6%) |
| To free up space or speed up my device | 216 (27.9%) |
| I didn't like the app | 208 (26.9%) |
| Other | 128 (16.5%) |
| The app is not working as expected | 120 (15.5%) |
| The app is crashing / very slow | 48 (6.2%) |
| Because of advertisements in the app | 42 (5.4%) |
| Because of in-app purchases | 35 (4.5%) |
| The app required permissions I wasn't comfortable granting | 32 (4.1%) |
| I don't know | 16 (2.1%) |

## 5.3 App Removals

The reasons our participants remove apps are shown in Table 4. As expected, the most common reason is that the participant no longer uses the app. The second most common reason, device performance, influenced 28% of app removals. In Section 5.1 we saw that participants are uninstalling apps at an average rate of 2/week. We were surprised by this as we assumed that when users stop using an app, they simply leave it ignored on their device rather than actively bothering to remove it. We see from these rationales that users are often removing apps for performance reasons and this contributes to the removal rate. We note that the "Other" bucket is large. Upon examination of the open ended feedback for the 128 app removal events in the "Other" option, we found that it mostly included additional details clarifying one of the already selected options. Some of the remaining responses suggested issues related to privacy or mismatched expectations. Examples include:

- Permission abuse: "The application is abusing the permission for location that I granted it. Uninstalling for this abuse of GPS." (P7)
- Negative publicity: "Read that the app is stealing private information about the phone and sending it back to China." (P31)
- Expectation mismatch: "It didn't have the information I was expecting it to have according to the description box."(P64)

Not all negative press cycles result in uninstalling apps, but for the participant above (second quote) it did. The reason "App required permissions I wasn't comfortable granting" is among the least influential here, however that option was triggered by 15% of our participants for 32 removal events. Note that if this 15% is extrapolated to the Android user base, that includes over 2 billion active devices, then the order of magnitude for devices uninstalling apps due to permissions would be in the 10s of millions.

In April 2016, the Google Play store started to require all developers to prominently disclose if their app included ads and in-app purchases. Among our participants, we see that only 10% of all uninstall events were influenced by ads or in-app purchases. This low fraction may be due to this extra transparency that helps manage people's expectations.

Table 5: Reasons participants checked for denying a permission to an app (multiple responses per deny event are possible)

| Permission Deny Reason | Number of Occurrences (% of deny events) |
|---|---|
| I think the app shouldn't need this permission | 111 (40.8%) |
| I expect the app will still work without this permission | 110 (40.4%) |
| I can always grant it afterwards if I change my mind | 110 (40.4%) |
| I do not use the specific feature associated with the permission | 95 (34.9%) |
| I consider the permission to be very sensitive | 57 (21%) |
| I don't trust the developer enough to provide this information | 42 (15.4%) |
| I wanted the permission screen to go away | 36 (13.2%) |
| Other | 28 (10.3%) |
| I think something bad might happen if I provide this permission | 15 (5.5%) |
| I didn't know I did that | 7 (2.6%) |
| I don't know | 6 (2.2%) |

# 6. PERMISSION DECISIONS

In this section, we discuss the reasons participants provided when accepting or denying app permission requests. Our participants granted 86% of the surveyed permission requests, indicating that they were 6 times more likely to grant a permission request rather than deny it, on average. It is noteworthy that the 14% of permission requests that were denied came from 49% of our participants. This indicates that nearly half of our participants denied a permission at least once in a 6 week period. We also observed that 95% of all decisions were made via the runtime dialogs as opposed to from inside the Android settings menu. The permission grant ratio for decisions made at runtime is 86%, whereas it is only 71% for decisions made via the settings menu, implying that users are more likely to deny a permission through the settings than when deciding at runtime. One plausible explanation is that users, especially those concerned with privacy, may seek to turn off access to personal data when they are not using an app.

## 6.1 Permission denials

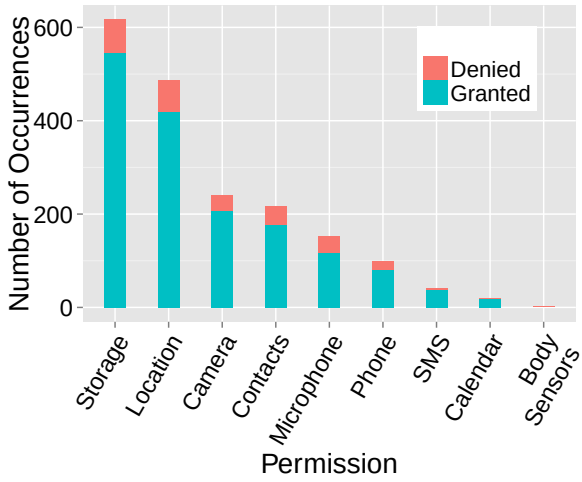
Table 5 shows the reasons participants had for denying permissions. Participants could pick as many reasons as they wanted for each decision, and overall the average number of reasons per denial decision was 2.3. The top two reasons imply that the majority of decisions are being made by focusing on the functionality of the app, and whether or not it really needs the particular permission. This corroborates

previous findings by Wijesekera et al. [48], who observed that relevance to app functionality is a dominant reason for blocking permissions, though we find different fractions of participants who select this reason. Wijesekera et al. found that 53% of their participants wanted to block a permission because it seemed unnecessary for the app functionality. If we use our top two reasons as a proxy for their "unnecessary for app functionality" reason, our data reveals that 34% of our participants fall into this category. A potential explanation for why our study observes fewer participants denying permissions because they felt it was unnecessary is as follows. In [48] the participants were shown (at the end of the study) a handful of permission accesses that had occurred during the prior week and asked if they would have liked to deny them and why. This captures their attitude. In our study, we capture participants actions (i.e., behaviors) and their associated rationale. In essence this gap reflects a type of difference between privacy attitudes and behaviors and thus it is not surprising that the privacy behavior occurs less often than the stated attitude.

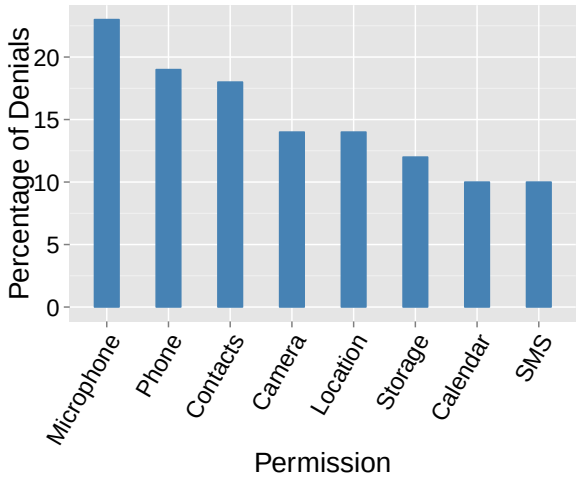
It is interesting to note that the reason "I can always grant it afterwards if I change my mind" is very prevalent among our participants (essentially tied for second place), indicating that users are aware about the fact that permissions for an app can be changed at any time (via Android's settings menu). Providing this answer for a permission denial could indicate that the user is denying the permission initially to see if the app still works, and undoing this decision later if necessary. This may indicate that the participant would prefer to use the app in a more private way and tests that possibility.

There were 57 instances where our participants denied a permission because they explicitly considered it to be very sensitive. It is striking to see that this was a more significant reason than not trusting the developer. Among these 57 instances, only 22 also picked "don't trust the developer" option. This implies that the remaining 35 instances (coming from 18 participants) correspond to scenarios were the participants do not distrust the developer but nevertheless consider the permissions sensitive and do not want to share the data. This suggests that although trust is necessary, it may not be sufficient to convince users to share data. This is of course a complex issue that requires further study because it is hard to know exactly how participants interpreted the "trust" option in our surveys.

We now examine decision making with respect to permission types. In Figure 5a, we see that the largest number of permission decisions occur for Storage and Location permissions. For each permission type, Figure 5b shows the fraction of requests that were denied. As is clear from this plot, the Microphone permission has the highest percentage of denials, followed by the Phone and Contacts permissions. It is interesting that Camera access did not exhibit a similar denial rate as Microphone; we posit that this might occur because the Camera permission sometimes only entails taking still photos (without audio and video). Although Location is perhaps the permission that users are most aware of, it does not appear among the top three most denied permissions. One possible reason is that users might have experienced some sort of habituation effect [7] for the Location permission, where a repeated exposure to such a permission




(a) Number of permission changes per permission.




(b) Percentage of permission requests denied per permission.

Figure 5: Participant Permission Decisions

request could have reduced their level of sensitivity or concern when granting such a permission, similarly to what has been reported in another study on pop-up dialogs [8].

To determine whether some decision rationales are more influential for specific permission types, we broke down our participants' reasons for permission denials according to the permission type. Figure 6 illustrates this via a heatmap. We have removed 2 permission types, SMS and Calendar, because there were fewer than 15 denials for these permissions.

Overall, we observe that the top two or three reasons for each permission type can differ. For example, for Location and Camera the top reason for denying is "I don't trust the developer". This reason has little significance for Phone and Contacts, where the dominant reasons are "I can always grant it afterwards" and "The app will still work without this permission". This shows that users make decisions about each of the permission types according to different rationales. We hypothesize that for Phone and Contacts, our participants

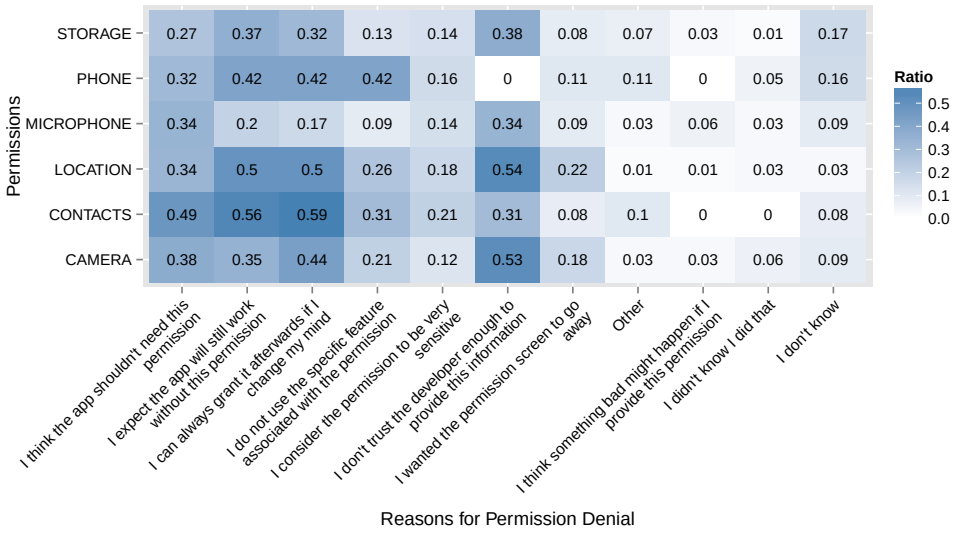



Figure 6: Reasons participants checked for denying each specific permission (multiple responses per deny event are possible). Each entry in the heatmap expresses the ratio of number of times that reason was given for the permission, over the count of all denies for the permission.

might be trying to not share them initially at all (and only doing so later if really needed) - thus issues of functionality are top of mind. However for Location and Camera, it is possible that the reason why the data is needed is often more clear and thus the primary rationale is based on trust.

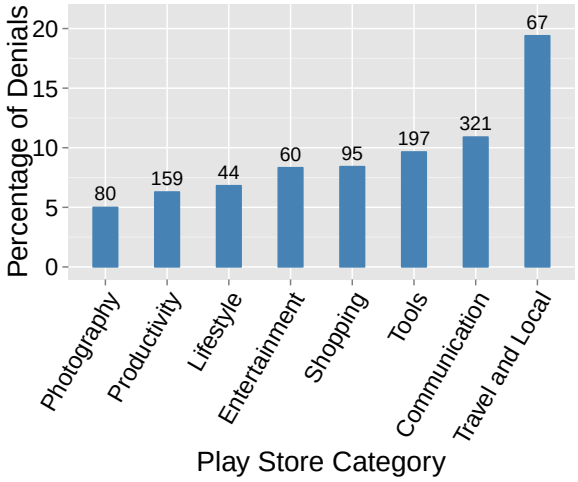



Figure 7: Percentage of permission denials across apps belonging to different Play store categories. The numbers on the bars indicate the total number of permission decisions in each category.

Next, we assess whether the permission denial rates are different across different app categories. For each of the 624 apps that registered a permission grant or denial event in our study, we identified its Play store category and considered it as an indicator of the app's functionality type. We recognize that some Play store categories, such as 'Productivity', are very broad and cover a wide range of app functionalities. However, app category was the only readily available functionality indicator.

Among the 624 apps, 41 did not appear in the Play store and seem to be device manufacturer apps that come pre-installed on the Android device or apps that have been downloaded from other Android app stores. For the remaining 583 Play apps, we aggregated the grants and denials across apps in each Play category. There were just 8 categories that had more than 20 apps, and the denial rates for these categories are shown in Figure 7. We also overlay the number of permission decisions within each category as the number on top of each bar. Denial rates vary between 5% - 19% across these 8 app categories. Moreover, the same permission can have different denial rates across different app categories. For example, 'Travel and Local' had a 43% denial rate for the Location permission, whereas 'Communication' registered only a 11% denial rate for the same permission. This reaffirms the influence of app functionality on users' permission grant or deny decisions.

## 6.2 Permission Grants

We now examine the reasons why users agree to grant permission requests. Table 6 shows that the dominant reason is "I want to use a specific feature that requires this permission", which suggests that users are agreeing because the request is in line with their expectations. As suggested by Felt et al. [13], a goal of using runtime dialogs is to improve the permission decision making and to avoid undermining users' expectations; our results thus indicate progress on that front. The second most important reason is trust in the developer. As discussed earlier, follow up work is necessary to fully understand how trust influences permission choices. Nonetheless, this result underscores how important it is for developers to gain a trustworthy reputation among (potential) users.

Table 6: Reasons participants checked for granting a permission to an app (multiple responses per grant event are possible)

| Permission Grant Reason | Number of Occurrences (% of grant events) |
|---|---|
| I want to use a specific feature that requires this permission | 1095 (68.2%) |
| I trust the app developer | 515 (32.1%) |
| I think the app won't work otherwise | 382 (23.8%) |
| I have nothing to hide | 289 (18%) |
| Nothing bad will happen | 225 (14%) |
| The app developer already has this information about me | 208 (13%) |
| I wanted the permission screen to go away | 164 (10.2%) |
| Because the app is popular | 150 (9.3%) |
| Other | 39 (2.4%) |
| I didn't know I did that | 36 (2.2%) |
| I won't be able to grant this permission later | 22 (1.4%) |

In a similar way as we did for the denials case, we checked whether some reasons are more influential for specific permission types, but found the distribution of reasons to be similar across permission types.

Next we look at the question of whether or not participants grant permissions willingly. Recall that after our participants granted a permission, we asked them to indicate if they agree or disagree (5 pt scale) with the statement "I don't mind giving <app> access to my <permission>" (Q2 in Appendix A.1.3). Surprisingly, we found that 10% of the time, participants indicated that they "Disagree" or "Strongly disagree" with the statement (see Figure 8). This could occur if participants believe an app won't work without the requested permission and so they agree, albeit reluctantly. This can be associated with the phenomenon of "learned helplessness" [46], which covers scenarios when participants convince themselves they agree with something (e.g., data sharing) because they did not really have a choice.

To see whether this comfort level changes over time, we asked participants in the exit survey to rate their comfort level with permissions they had granted to apps on their phones in the past (Q19 – Q22 in Section A.2; we included "I don't know the app" as an additional option). When asking these questions, we made the permissions more specific. For example, if the participant had granted the Storage permission, we ask whether they were comfortable with the app accessing photos on their device storage. These questions were intended not only to revisit comfort with prior decisions, but also to illustrate more explicitly to the participants the implication of their decision. These prior decisions may have occurred any time during our 6 week study or even earlier as explained in Section 3.1.2.

In a surprisingly high number of situations (see Figure 9) participants were not comfortable with their prior decisions. In 29% of scenarios presented to the participants, they indicated they were "Not at all comfortable" with the data access that was allowed to the app. If we include the cases where users were "Slightly comfortable", then we see that in 44% of the cases our participants are not feeling comfortable about their past decisions. These discomfort levels vary based on the permission: on a scale from 1 to 5, where larger numbers indicate a higher discomfort, the Storage permission entails an average discomfort of 3.41, Phone has a discomfort of 3.33, Contacts has a discomfort 3.11, and Location has a discomfort of 2.77.

Participants were not comfortable about permissions they granted in the past and this may be occurring because they do not always understand what a permission entails, and only realize this after it is made explicit. Consider, for example, the Storage permission: this permission might be understood by a user as allowing the app to store data on the device, only to be refuted by our question stating that

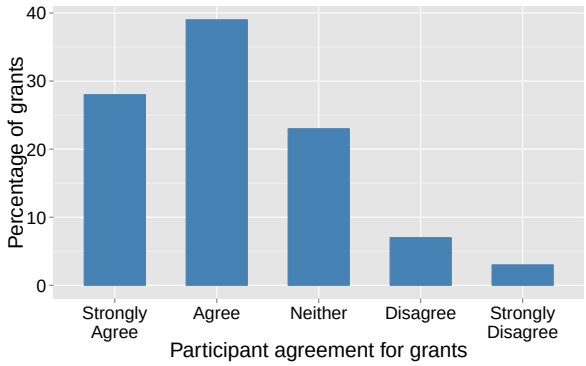



Figure 8: Participant responses to the statement: "I don't mind giving <app> access to my <permission>", right after granting that permission.

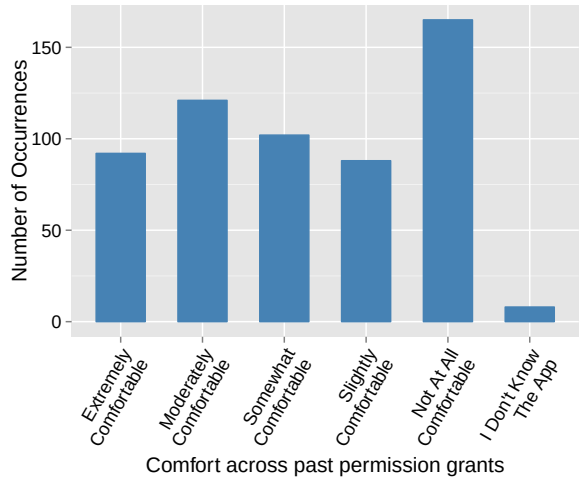



Figure 9: Participant comfort for permissions that were granted in the past, in response to the exit survey question "How comfortable would you be with the <app name> app knowing <information available through the permission>".

the app now has access to pictures on the user's device. This explanation is supported by previous work [18, 43] that has shown how users need to be confronted with a specific scenario before being able to correctly reason about privacy and security.

It is interesting to contrast the 29% discomfort long after decision making, to the 10% reluctance that existed at the moment of decision making. This 29% statistic could be said to capture privacy attitudes; the exit survey captures what people say or think about sharing data when they are being questioned but not making a real life decision. However in practice, in only 10% of grant decisions did users say that they minded sharing the data right after granting. The gap between these numbers approximately captures the difference in participant's attitudes and behaviors, in the context of Android permissions.

## 6.3 Other influences

We check whether the participants' demographics are associated with their grant/denial behavior. We used Pearson's Chi-squared test (with Yates' continuity correction when needed) to check the dependence between participants' age and gender, and their denial behavior. We control for age (gender) when gender (age) is being tested. Due to small sample sizes, we did not test for independence across education and employment demographics. We notice that women across age groups 18-23 ($\tilde{\chi}^2 = 10.7$, df = 1, p-value = 0.001068) and 31-40 ($\tilde{\chi}^2 = 16.3$, df = 1, p-value = 5.396e-05) are three times as likely to deny permissions than men. On average over all age groups, women deny twice as often as men, with a 20% denial rate for women compared to 11% for men ($\tilde{\chi}^2 = 25.6$, df = 1, p-value = 4.11e-07). Comparing men across different age groups, we notice that men's denial rates differ significantly ($\tilde{\chi}^2 = 31.2$, df = 4, p-value = 2.841e-06); participants in age ranges 18-23 and 31-40 have denial rates around 5% whereas the other age groups have denial rates of 15% or higher, about three times higher.

Lastly, we checked associations between participant responses to questions in the exit survey (Q1–Q18 in Section A.2) and their permission denials. We did not find any statistically significant correlations or dependencies.

# 7. DISCUSSION AND CONCLUSION

There are a couple of important takeaways herein for Play store developers. First, we saw that in terms of app installs and uninstalls, permissions were not a dominant reason compared to other reasons. However, 15% of our participants uninstalled apps due to permissions. Extrapolating this statistic to the set of Android devices (over 2 billion), indicates that this could affect tens of millions of devices. This result could motivate developers to reconsider requesting certain permissions at all or to make runtime requests more contextual – for example by only asking for permission access when the user opts to use certain functionality within their app rather than at first run.

Second, the vast majority of rationales for decision making around permissions are related to app functionality, whether the app needs the permission, whether it "should" need it, and whether the user needs the functionality entailed by it. Thus, participants are more willing to grant permissions when the reason for asking is clear. This should motivate developers to provide sufficient and clear explanations for their requests. Android provides a utility method (`shouldShowRequestPermissionRationale()`) to help identify situations where users might need an explanation.

In summary, we observed an overall denial rate of 16%. These denies came from half our participants which indicates that there exists one or some scenarios for many people in which they will deny a permission. The scenarios when participants deny permissions are very varied. This is implied by the findings that i) denial rates vary from 10% to 23% according to permission type, and ii) denial rates vary from 5% to 19% across app genres (Play store categories). Among our participants, we also saw that women denied permissions roughly twice as often as men.

We found that even though the overall grant rate is quite high, at 84%, there is a portion of decisions (10%) in which users grant permissions reluctantly. Moreover, users were surprisingly uncomfortable (29%) when revisiting their prior decisions at the end of our study. This indicates a gap between behaviors and stated attitudes.

Our participants' rationale for denying a permission in 42% of denial instances, was because they knew they could change the permissions afterwards. We hypothesize that this might be happening because participants want to test out whether or not the app will work in a more privacy preserving way (with less user data). Exploring this would be an interesting avenue for future research.

It is interesting albeit hard to understand how users' comfort levels and understanding of permissions have evolved after the introduction of runtime dialogs. In [48] (pre-runtime), the authors state that 80% of their participants wanted to deny at least one permission. In our study, we recorded that 49% of our participants denied permissions at least once. We found that 16% of permission requests were denied. This is about half the rate reported in [48], though the latter study asked participants to allow or deny access many times a permission was used, instead of only on first use as in our study. These two studies differ in their interactions with users, and both involve limited populations, yet these metrics hint that users may be getting more comfortable granting permissions using runtime dialogs. It would be interesting to explore this hypothesis in future research that makes a more direct comparison.

# 8. ACKNOWLEDGMENTS

Many thanks to Bob Evans for collaborating with us on extending Paco; to Rob Reeder, Sunny Consolvo, Tara Matthews, Allison Woodruff, Jeffrey Warshaw, and Manya Sleeper for valuable suggestions on the survey design; to Patrick Gage Kelley for providing feedback on an initial draft; to Clara Sherley-Appel for helping with participant onboarding; to Svetoslav Ganov and Giles Hogben for suggestions on technical implementation.

# 9. REFERENCES

[1] H. M. Almohri, D. D. Yao, and D. Kafura. Droidbarrier: Know what is executing on your android. In *Proceedings of the 4th ACM Conference on Data and Application Security and Privacy*, CODASPY. ACM, 2014.

[2] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal.

Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Proceedings of the 33rd annual ACM conference on Human Factors in Computing Systems*, CHI. ACM, 2015.

[3] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie. Pscout: Analyzing the android permission specification. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS. ACM, 2012.

[4] R. Balebako, J. Jung, W. Lu, L. F. Cranor, and C. Nguyen. Little brothers watching you: Raising awareness of data leaks on smartphones. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS. ACM, 2013.

[5] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji. A methodology for empirical analysis of permission-based security models and its application to android. In *Proceedings of the 17th ACM Conference on Computer and Communications Security*, CCS. ACM, 2010.

[6] E. Bodden. Easily instrumenting android applications for security purposes. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS. ACM, 2013.

[7] M. E. Bouton. *Learning and behavior: A contemporary synthesis*. Sinauer Associates, 2007.

[8] C. Bravo-Lillo, L. Cranor, S. Komanduri, S. Schechter, and M. Sleeper. Harder to ignore? revisiting pop-up fatigue and approaches to prevent it. In *10th Symposium On Usable Privacy and Security*, SOUPS. USENIX Association, 2014.

[9] S. Bugiel, S. Heuser, and A.-R. Sadeghi. Flexible and fine-grained mandatory access control on android for diverse security and privacy policies. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC. USENIX Association, 2013.

[10] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation*, OSDI. USENIX Association, 2010.

[11] B. Evans. Paco – applying computational methods to scale qualitative methods. In *Ethnographic Praxis in Industry Conference Proceedings*. Wiley Online Library, 2016.

[12] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android permissions demystified. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS. ACM, 2011.

[13] A. P. Felt, S. Egelman, M. Finifter, D. Akhawe, and D. Wagner. How to ask for permission. In *Proceedings of 7th Usenix conference on Hot Topics in Security (HotSec)*, 2012.

[14] A. P. Felt, S. Egelman, and D. Wagner. I've got 99 problems, but vibration ain't one: A survey of smartphone users' concerns. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM. ACM, 2012.

[15] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS. ACM, 2012.

[16] C. Gibler, J. Crussell, J. Erickson, and H. Chen. Androidleaks: automatically detecting potential privacy leaks in android applications on a large scale. In *International Conference on Trust and Trustworthy Computing*. Springer, 2012.

[17] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller. Checking app behavior against app descriptions. In *Proceedings of the 36th International Conference on Software Engineering*, ICSE. ACM, 2014.

[18] M. Harbach, M. Hettig, S. Weber, and M. Smith. Using personal examples to improve risk communication for security & privacy decisions. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI. ACM, 2014.

[19] M. A. Harris, R. Brookshire, and A. G. Chin. Identifying factors influencing consumers' intent to install mobile applications. *International Journal of Information Management*, 2016.

[20] M. A. Harris, R. Brookshire, K. Patten, and B. Regan. Mobile application installation influences: have mobile device users become desensitized to excessive permission requests? In *Proceedings of the Twentieth Americas Conference on Information Systems*, AMCIS, 2015.

[21] S. E. Hormuth. The sampling of experiences in situ. *Journal of Personality*, 1986.

[22] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS. ACM, 2011.

[23] J. Jung, S. Han, and D. Wetherall. Short paper: Enhancing mobile application permissions with runtime feedback and constraints. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM. ACM, 2012.

[24] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall. A conundrum of permissions: Installing applications on an android smartphone. In *Proceedings of the 16th International Conference on Financial Cryptography and Data Security*, FC. Springer-Verlag, 2012.

[25] P. G. Kelley, L. F. Cranor, and N. Sadeh. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI. ACM, 2013.

[26] W. Klieber, L. Flynn, A. Bhosale, L. Jia, and L. Bauer. Android taint flow analysis for app sets. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on the State of the Art in Java Program Analysis*, SOAP. ACM, 2014.

[27] R. Larson and M. Csikszentmihalyi. The experience sampling method. *New Directions for Methodology of Social & Behavioral Science*, 1983.

[28] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the*

*2012 ACM Conference on Ubiquitous Computing*, UbiComp. ACM, 2012.

[29] J. Lin, B. Liu, N. Sadeh, and J. I. Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *Symposium on Usable Privacy and Security*, SOUPS, 2014.

[30] B. Liu, M. S. Andersen, F. Schaub, H. Almuhimedi, S. A. Zhang, N. Sadeh, Y. Agarwal, and A. Acquisti. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Twelfth Symposium on Usable Privacy and Security*, SOUPS. USENIX Association, 2016.

[31] B. Liu, J. Lin, and N. Sadeh. Reconciling mobile app privacy and usability on smartphones: Could user privacy profiles help? In *Proceedings of the 23rd International Conference on World wide web*, WWW. ACM, 2014.

[32] M. Madden and L. Rainie. Americans' Attitudes About Privacy, Security and Surveillance. `http://www.pewinternet.org/2015/05/20/americans-attitudes-about-privacy-security-and-surveillance/`.

[33] K. Micinski, D. Votipka, R. Stevens, N. Kofinas, M. L. Mazurek, and J. S. Foster. User interactions and permission use on android. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI. ACM, 2017.

[34] H. Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 2004.

[35] P. A. Norberg, D. R. Horne, and D. A. Horne. The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of Consumer Affairs*, 2007.

[36] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. Whyper: Towards automating risk assessment of mobile applications. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC. USENIX Association, 2013.

[37] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy. Android permissions: A perspective combining risks and benefits. In *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*, SACMAT. ACM, 2012.

[38] B. Shebaro, O. Oluwatimi, D. Midi, and E. Bertino. Identidroid: Android can finally wear its anonymous suit. *Transactions on Data Privacy*, 2014.

[39] I. Shklovski, S. D. Mainwaring, H. H. Skúladóttir, and H. Borgthorsson. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI. ACM, 2014.

[40] M. Smith. Usable Security – The Source Awakens. *Usenix Enigma*, 2016.

[41] M. Spreitzenbarth, F. Freiling, F. Echtler, T. Schreck, and J. Hoffmann. Mobile-sandbox: Having a deeper look into android applications. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, SAC. ACM, 2013.

[42] R. Stevens, J. Ganz, V. Filkov, P. Devanbu, and H. Chen. Asking for (and about) permissions used by android apps. In *2013 10th Working Conference on Mining Software Repositories (MSR)*, 2013.

[43] C. Swanson, R. Urner, and E. Lank. Naïve security in a wi-fi world. In *IFIP International Conference on Trust Management*. Springer, 2010.

[44] C. Thompson, M. Johnson, S. Egelman, D. Wagner, and J. King. When it's better to ask forgiveness than get permission: Attribution mechanisms for smartphone resources. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS. ACM, 2013.

[45] A. Vaidos. Google Play Store vs Apple's App Store - A Comparison. `http://news.softpedia.com/news/google-play-store-vs-apple-s-app-store-a-comparison-512601.shtml`.

[46] J. Warshaw, T. Matthews, S. Whittaker, C. Kau, M. Bengualid, and B. A. Smith. Can an algorithm know the "real you"?: Understanding people's reactions to hyper-personal analytics systems. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI. ACM, 2015.

[47] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos. Permission evolution in the android ecosystem. In *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC. ACM, 2012.

[48] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, and K. Beznosov. Android permissions remystified: A field study on contextual integrity. In *Proceedings of the 24th USENIX Conference on Security Symposium*, SEC. USENIX Association, 2015.

[49] P. Wijesekera, A. Baokar, L. Tsai, J. Reardon, S. Egelman, D. Wagner, and K. Beznosov. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*. IEEE, 2017.

[50] R. Xu, H. Saïdi, and R. Anderson. Aurasium: Practical policy enforcement for android applications. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security. USENIX Association, 2012.

[51] Y. Zhang, M. Yang, B. Xu, Z. Yang, G. Gu, P. Ning, X. S. Wang, and B. Zang. Vetting undesirable behaviors in android apps with permission use analysis. In *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security*, CCS. ACM, 2013.

[52] H. Zhu, H. Xiong, Y. Ge, and E. Chen. Mobile app recommendations with security and privacy awareness. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD. ACM, 2014.

# APPENDIX

# A. SURVEY QUESTIONS

Responses to all questions are required.

## A.1 In-situ questions

### *A.1.1 App installation scenario*

The order of possible responses to the questions for the in-situ survey is always randomized (with the exception of the 'Other' option, which is always placed last).

**Q1: Which factors influenced your decision to install <app>? (select all that apply)**

- App rating
- App popularity
- Individual user reviews
- Requested permissions
- The company creating the app
- The app is free / price is reasonable

**Q2: Why did you install <app>? (select all that apply)**

- The app has fewer permissions than other apps like it
- My friends/family use it
- I want to try it out
- I was required to install it
- The app is part of a product/service that I use
- The app is useful
- The app is cool or fun to use
- I trust the app or the company making the app
- It was the only app of its kind (no other apps provide the same functionality)
- I was offered something in return (e.g. credits, monetary rewards, discount)
- I don't know
- Other: _____

### *A.1.2 App removal scenario*

**Q1: Why did you remove <app>? (select all that apply)**

- The app required permissions I wasn't comfortable with granting
- I no longer use the app
- To free up space or speed up my device
- Because of advertisements in the app
- Because of in-app purchases
- I didn't like the app
- The app is crashing / very slow
- The app is not working as expected
- I don't know
- Other: _____

### *A.1.3 Permission grant scenario*

**Q1: Why did you choose to allow <app> to access your <permission>? (select all that apply)**

- I want to use a specific feature that requires this permission
- I think the app won't work otherwise
- I trust the app developer
- Because the app is popular
- I won't be able to grant this permission later
- I have nothing to hide
- I wanted the permission screen to go away
- Nothing bad will happen
- I didn't know I did that
- I don't know
- The app developer already has this information about me
- Other: _____

**Q2: To what extent do you agree with the following statement: "I don't mind giving <app> access to my <permission>"?**

- Strongly disagree
- Disagree
- Neither agree or disagree
- Agree
- Strongly agree

### *A.1.4 Permission deny scenario*

**Q1: Why did you deny <app> to have access to your <permission>? (select all that apply)**

- I do not use the specific feature associated with the permission
- I think the app shouldn't need this permission
- I expect the app will still work without this permission
- I consider the permission to be very sensitive
- I don't trust the developer enough to provide this information
- I can always grant it afterwards if I change my mind
- I wanted the permission screen to go away
- I think something bad might happen if I provide this permission
- I don't know
- I didn't know I did that
- Other: _____

## A.2 Exit Survey

Each of the questions Q1-Q15 have the same three possible answers:

- Yes
- No
- I don't know what this is / means

**Q1: Have you ever blocked another person on a social network?**

**Q2: Have you ever deleted an online account?**

**Q3: Have you ever downloaded your historical data from an account (e.g. Google Takeout)?**

**Q4: Have you ever changed the privacy settings for any of your accounts?**

**Q5: Have you ever read part or all of an online privacy policy?**

**Q6: Have you ever decided not to install an app on your mobile device because of permissions it requested?**

**Q7: Have you ever uninstalled an app on your mobile device because of permissions it used?**

**Q8: Have you ever declined to give an app permission to do something on your mobile device?**

**Q9: Have you ever declined to use a website because it asked for information you did not want to provide?**

**Q10: Have you ever stopped using an Internet service or website because you were concerned about how it might use your personal information?**

**Q11: Have you ever cleared cookies and/or browser history?**

**Q12: Have you ever installed software to block ads?**

**Q13: Have you ever installed software to stop websites from tracking what you do online?**

**Q14: Have you ever used a password manager?**

**Q15: Have you ever used account settings to limit the data that could be collected or used?**

**Q16: Which of the following best describes the time at which you try new technology?**

- As soon as the technology is available / among the first people to try it
- Sooner than most people, but not among the first
- Once many people are using it
- Once most people are using it
- I don't usually buy or try out new technology

**Q17: When an Internet company collects data about you while you are online, overall how beneficial or harmful is that likely to be for you?**

- Extremely beneficial
- Moderately beneficial
- Slightly beneficial
- Neither beneficial nor harmful
- Slightly harmful
- Moderately harmful
- Extremely harmful

In questions Q18-Q22, we used a 5-pt Likert scale to measure comfort.

**Q18: How comfortable or uncomfortable are you with online companies collecting data about what you do online?**

- Extremely Comfortable
- Moderately Comfortable
- Somewhat Comfortable
- Slightly Comfortable
- Not at all Comfortable

In addition to the 5-pt comfort scale, for questions Q19-Q22 users could also select an option "I don't know the app" if they do not recognize the app in the question. The apps we showed users were ones on their phones, so most of the time apps should be recognized.

**Q19: How comfortable would you be with the <app name> app knowing your home and work address? (only surfaced if an app exists that was given the Location permission)**

- Extremely Comfortable
- Moderately Comfortable
- Somewhat Comfortable
- Slightly Comfortable
- Not at all Comfortable
- I don't know the app

The question answer options for Q20-Q22, were the same as in Q19.

**Q20: How comfortable would you be with the <app name> app knowing the phone numbers of your friends and family? (only surfaced if an app exists that was given the Contacts permission)**

**Q21: How comfortable would you be with the <app name> app knowing who is calling you? (only surfaced if an app exists that was given the Phone permission)**

**Q22: How comfortable would you be with the <app name> app seeing the pictures taken with your camera? (only surfaced if an app exists that was given the Storage permission)**

**Q23: Do you have any feedback for us? Is there anything else you would like to tell us?**
Open ended response