Investigating User Perception and Comprehension of Android Permission Models

Anthony Peruma, Jeffrey Palmerino and Daniel E. Krutz
Department of Software Engineering
Rochester Institute of Technology
Rochester, NY, USA
{axp6201,jrp3143,dxvse}@rit.edu

ABSTRACT

Do you know the permissions your favorite apps use? You probably don't, and you aren't alone. Everyone seemingly talks about how important app security and privacy is to them, but research has shown that users are generally not well informed about the permissions their apps use. This leads to serious ramifications for security, privacy and user perception (rating) of an app. Understanding the current Android permission model and how it can be improved offers significant benefits for both developers and users.

To better understand user perception of the previous, current and a new proposed permission model, we conducted an in-person study involving 185 participants. Our primary findings include I) The current Android runtime model does not make users feel more secure in comparison with the older install-time model. II) Our proposed model is beneficial in helping users feel more secure. III) There is no statistically significant difference between the user ratings given to the apps using the different permissions models. IV) Runtime permission models are significantly beneficial in helping users to recall the requested permissions. V) We found that users were generally well informed about what the requested permissions meant, but age played a significant factor in reducing how informed users were.

CCS CONCEPTS

• **Security and privacy** → *Mobile platform security*; *Software security engineering*; *Privacy protections*;

KEYWORDS

Mobile Permissions, Mobile Privacy, Mobile Security

ACM Reference Format:
Anthony Peruma, Jeffrey Palmerino and Daniel E. Krutz. 2018. Investigating User Perception and Comprehension of Android Permission Models. In *MOBILESoft '18: MOBILESoft '18: 5th IEEE/ACM International Conference on Mobile Software Engineering and Systems , May 27–28, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3197231.3197246

1 INTRODUCTION

Android is the world's most popular mobile platform [1], allowing users to perform a variety of tasks that were previously unachievable in a mobile environment. Android applications (apps) provide the user with a diverse set of functionality allowing them to do everything from post their Facebook status to conducting banking transactions. To perform various sensitive tasks, an app must be granted appropriate permissions to carry out this functionality. For example, if an app requests access to a user's contact list, the app must explicitly ask the user for permission to this information. These app permissions serve as an integral component of the app's security by limiting access to this information and functionality to other potentially malicious apps installed on the user's device, but also in ensuring that a user is adequately able to protect the information on their phone as they desire.

Previous research demonstrated that Android's initial, install-time permission model was not working well, making users uncomfortable and largely uninformed over the permissions their apps were actually using [11]. This lack of comprehension can be very harmful to a user's privacy and comfort level when using the app [11, 30]. To help alleviate these issues, in late 2015 Android implemented a new runtime permission model to provide the user more flexibility over the permissions they wanted to allow their apps to use [2, 3]. Users were provided greater control over what permissions the app had access to.

In our work, we examined if this newer runtime model has met the desired objectives of making users more comfortable and informed in the permissions their apps use. We additionally offer a potentially improved variation of the runtime permission model, one that displays even further app and permission information to the user. We compared these models to understand how effective they are in informing and creating comfort for users with the permissions their apps were using. We also examined if user's recollection of an app's permissions matched reality. To address these questions, we conducted a large-scale in-person user study involving 185 adults. Users were not told that they were participating in a permissions study, but were simply asked to play a simple tic-tac-toe app. They were asked to complete a short pre-survey and were then allowed to play the game which covertly recorded the user's actions. After the game, users were asked to complete a survey which was intended to measure their knowledge and recollection of permissions.

MOBILESoft '18, May 27–28, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5712-8/18/05…$15.00
https://doi.org/10.1145/3197231.3197246

The following research questions guided our work:

RQ1: Does the current Android runtime permission model make users feel more secure with the permissions their apps use in comparison with the previous install-time model? We found that the current Android runtime model does not make users feel more secure with their apps in comparison with the older install-time model.

RQ2: Is our proposed model beneficial for making users feel more secure and be more informed? In comparison with Android's current model, we found that our proposed model makes users feel more secure about the app they were using. However, there was not much difference in our model's ability to help users to recall the permissions requested by their apps.

RQ3: Do users give a different rating to the three permission models? Although not statistically significant, we did find an inverse relationship between the user's rating and how secure they felt using the app.

RQ4: How well do users recall the permissions their apps request? We found that runtime permission models were significantly better in helping users to recall the permissions requested by the apps. We also found that older users recalled the app's requested permissions much more poorly than younger users.

RQ5: How well do users know what the permissions requested by their apps mean? We found that users were well informed of what the definitions are of the permissions the application requested. We did find that age is a significant factor in a user's ability to understand what the requested permissions mean.

The rest of the paper is organized as follows: Section 2 describes related works and Section 3 provides a background on Android permissions. Section 4 discusses the foundation of the study, and Section 5 presents basic results and their implications, while Section 6 provides developer recommendations. Threats to our study are described in Section 7 and Section 8 concludes our work.

2 RELATED WORKS

There has been a substantial amount of work examining mobile user privacy from numerous developer and end-user perspectives [6, 19, 20, 25]. Many works have examined app privacy in a variety of manners. Using crowdsourcing, Lin et al.[17] examined user privacy expectations regarding what an app should and should not do, specifically focusing on instances where the app did not conform to people's expectations. One of their more interesting discoveries found that providing more information about how a specific resource was being used could alleviate some of the user's privacy concerns and make the user more comfortable. This work also found that users generally felt uncomfortable and may even delete applications when they did not understand why it requested a permission they deemed unnecessary.

Gerber et al.[12] conducted a study with 344 participants to test various different permission interfaces and found that people have the ability to read and understand relatively complex interfaces. However, their study suggests that a 'sweet spot' is best for maximizing comprehensiveness and understandability in the request process. Our work differs from this study in that they focused on providing different information to the end user. For example, their work provided information from *privacy experts* and in users selecting the privacy-friendly alternatives.

Using the install-time versions of Android, Felt et al.[11] performed an Internet and laboratory survey to understand user comprehension rates about permissions. They found that participants generally possessed low comprehension and attention rates to the permissions the apps were using. Only 17% of users paid attention to the permissions the app was requesting at install time, and only 3% of Internet participants could adequately answer three permission comprehension questions.

Many works have proposed enhancements to the older install-time permissions model along with the newer runtime model. Several works proposed solutions such as custom frameworks which would provide the user the ability to accept only a subset of the app's requested permissions [8, 15, 22, 26]. While such solutions would have provided the advantage of providing the user more control over their apps, the average Android user would be unlikely to conduct these somewhat complicated technical tasks. Additionally, these recommended changes would often require the user to root their phone, something which could lead to additional security threats [26].

Proposed works have even utilized recommender systems to help users to make important app privacy decisions. RecDroid [24] is a proposed framework which allows users to make resource and privacy-related decisions in real-time using recommendations from expert users of the same apps. Using this framework, users could install apps in a 'probation' model where users would perform real-time resource granting decisions while receiving recommendations from expert users. The app would then run in a trusted mode where the app would be fully trusted, with app permissions being granted. Yang et al.[31] also used crowd-sourcing to help inform users and make them more comfortable with the permissions their apps use. In their approach, groups of users of the same application use the proposed tool to assist one another in permission understanding by sharing permission reviews.

Many enhancements have been proposed to the current runtime permission models as well. Scoccia et al.[27] proposed a more user-centric approach to permissions management known as Android Flexible Permissions (AFP). AFP allows users to specify and customize more granular permissions based on their privacy concerns. AFP allows users to select permissions based on their desired permission levels and the features of the app. While interesting and novel, it does not appear as though the proposed process has yet been thoroughly examined.

Micinski et al.[20] sought to determine if integrating the authorization systems with the app's user interface could improve the user experience. They found that various user interactions such as button clicks could be used as permission authorization, thus reducing the requirement for separate authorizations.

Krutz et al.[16] examined 1,402 open source Android apps to determine what developers were making permissions-related decisions during the app development process. They found that developers with more experience are more likely to make permission-based changes and that permissions are typically added earlier in apps' commit lifetime, but their removal is more sustained throughout the commit lifetime.

In our work, we examined differences in the participant's age in the ability to recall permissions requested by the app. Numerous works have examined the effects of a user's age on their ability to recall [10, 13, 14, 28]. Most have found that older users typically perform much more poorly on recall tasks in comparison with other cognitive actions [7, 9].

3 ANDROID PERMISSIONS

Android apps operate under a privilege-separated system where each app operates with a specific system identify, and each app is isolated from all other apps on the device. More fine-grain functionality enforces permissions on specific operations which the app may carry out. For example, if an app wishes to access the internet or the user's contacts, the app is required to be granted this permission before the app has the ability to execute this functionality. Permissions are separated into several protection levels, with the most important being *normal* and *dangerous* permissions [4].

Prior to Android 6.0, the user was prompted to accept all permissions when attempting to install an app [2, 3]. This all or nothing ordeal was true when updating an app, too. That is, a user had to accept all permissions and update requires to complete the update. Users were also not able to change permissions after installation, and the only way to restrict an app's access to a specific permission was to uninstall the entire app [30]. In the past several years, there has been a substantial amount of research recommending changes to the Android permission model ranging from the creation of privacy profiles [18] to more granular sets of Android permissions [23].

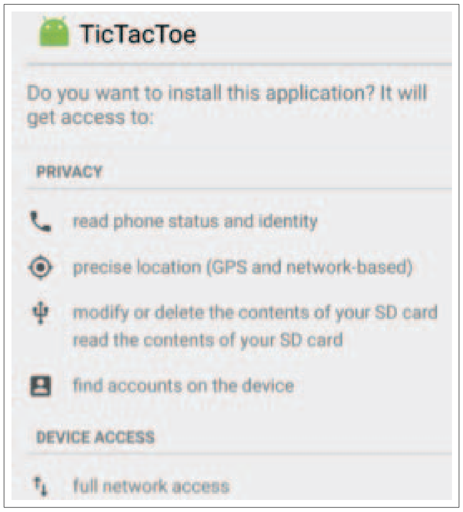
Android 6.0 represented a significant change in how permissions were handled in Android apps. Android 6.0 prompts users about allowing an app's permission requests at runtime, and not upon installing the application. The intention was that this would make installing and updating the app a simpler and easier process while providing the user greater control over the app, and therefore their privacy. Furthermore, Android 6.0 allows the user to use the app while also allowing them to choose the permissions they want the app to have, and even change their permission decisions whenever they please. Developers also have the option of providing a *rationale*, or further explanation, to the user about why the app is asking for a particular permission. An example of the Android 5.0 (install-time) is shown in Figure 1, and Android 6.0 (runtime) permission request is shown in Figure 2.

4 STUDY DESIGN

Our user study was comprised of three primary phases, which are further described later in our work:

(1) **Pre-Survey:** Collect user demographic data and information about their mobile experience.
(2) **Play tic-tac-toe:** User plays the game and is instructed to act like they were using the game on their own phone.
(3) **Post-Survey:** Users provide feedback regarding their experiences and general knowledge of mobile permissions and security.

Overall, 185 participants completed our study. These actions are outlined in Figure 3 and described in the following sections. We conducted a similar study in the Spring of 2016 that we used as a proof of concept for our Spring 2017 user study. We incorporated the lessons learned from this previous study into this work.


TicTacToe

Do you want to install this application? It will get access to:

PRIVACY

read phone status and identity

precise location (GPS and network-based)

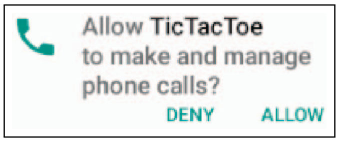
modify or delete the contents of your SD card
read the contents of your SD card

find accounts on the device

DEVICE ACCESS

full network access

Figure 1: Old Android ≤ 5.0 (Install-Time) Permission Model


Allow **TicTacToe** to make and manage phone calls?

DENY ALLOW

Figure 2: Android ≥ 6.0 (Runtime) Permission Request

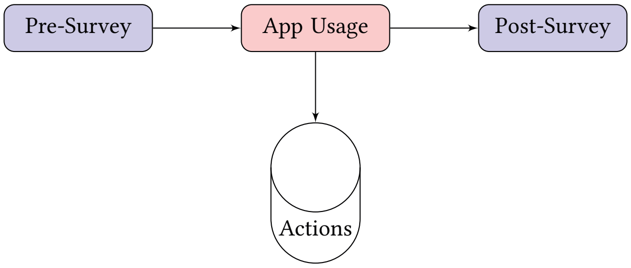


Figure 3: Data Collection Process

4.1 Tic-Tac-Toe App

We developed a simple tic-tac-toe app for our study that was used to evaluate user's permission comprehension, actions and comfort levels. We chose a tic-tac-toe app since we felt that it was a game that most users would easily understand and feel comfortable using. Figure 4 shows a sample screenshot of the app.

We sought to mimic the complete user experience as closely as possible, including installing and using the app. To emulate the install-time permissions model, we had users begin by seeing the pre-installation permission request screen, while users of the runtime permissions models began with an identical screen they would

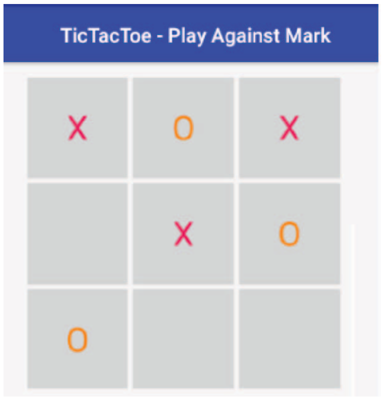


Figure 4: Example of Tic-Tac-Toe Game Used In The Study

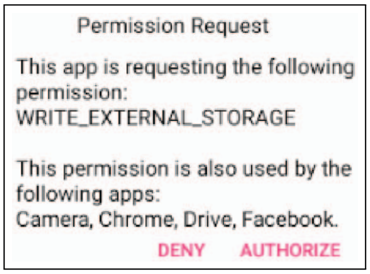
see if they had installed the app on their own device. This meant we had three separate versions of our tic-tac-toe app, each representing a single permissions model.

The app permissions include `ACCESS_FINE_LOCATION`, `WRITE_EXTERNAL_STORAGE`, `READ_PHONE_STATE`, and `GET_ACCOUNTS`. We chose these permissions since they were located in different *permission groups* [3] and would, therefore, be shown to the user. Additionally, a previous study found that these groups represented the most common permission groups in Android apps [21]. The four requested permissions and their stated reasoning within the app are shown below:

- **GET_ACCOUNTS**: Access to the user's account information to share data
- **WRITE_EXTERNAL_STORAGE**: Save the user's score on their device
- **READ_PHONE_STATE**: Added under the guise of managing the user's phone status. This permission was largely added since it can be reasonably assumed to be an outlandish permission for a tic-tac-toe app to request, and we wanted to see how many users would allow this.
- **ACCESS_FINE_LOCATION**: Know the user's location so they could find people to play with around them.

When a user rejected a permission during runtime, they were gracefully made aware that the functionality enabled by allowing the permission would not be enacted. For example, if the user denied the `ACCESS_FINE_LOCATION` permission request, a short message appeared on the screen informing them that they would not be able to find people close by to play against and instead, would be randomly assigned an opponent. This was done to mimic an actual app and user experience as closely as possible.

4.1.1 Proposed Model. To better understand possible improvements to the current Android runtime permission model, a new proposed runtime permissions model was included in our study. This included an extra notification when a permission was requested that included the other fictitious apps on the device that were using the permission. The primary objectives of including this proposed model in our study were to see if including extra information would make users (I) More informed with the permissions their apps were using (II) More comfortable with the permissions their apps were using. Figure 5 shows a sample notification screen.



Permission Request

This app is requesting the following permission:
WRITE_EXTERNAL_STORAGE

This permission is also used by the following apps:
Camera, Chrome, Drive, Facebook.

Figure 5: Example of Proposed Permission Model Showing Other Apps Using Requested Permission(s)

To construct the app with the proposed permission model, we created an Android API 22 (Android 5.0) app that used the install-time permission model. We created custom messages that conveyed the new information to the user whenever a permission occurred. It was necessary to use API 22 since it is impossible to deactivate the permission prompts in Android ≥ 23 (Android 6.0+). While using the app, permission requests were shown to the user with a custom permission request box including the other apps which were using this permission. Other than this extra information shown to the user, their experiences emulated what other Android 6.0+ users would have encountered. During the study, users were not informed that they were using a 'new' model so that we would not bias any of their feedback.

4.2 Survey

Users were asked to complete a 'pre' and 'post' survey. Users were not allowed to use the app until they had completed the initial pre-survey, and were not allowed to use the app while completing the second half of the survey. Users who did not complete both parts of the survey were not included in the study, and their results were discarded. The first half of the survey collected demographic information such as age, gender, length of time using a smartphone, etc. The second half of the survey was completed after the participant had used the app. This survey component measured the user's experiences with the app. While some of the pre-survey questions were used in our analysis, others were collected to merely demonstrate that a diverse set of users were included in our study. The survey and high-level results are shown below:

Pre-Survey

(1) **What Is Your Age?** (optional)
- `<text response>`
  - Range: 18 - 70
  - Mean: 31.5

(2) **What Is Your Gender?** (optional)
  (a) Male `(61.1%)`
  (b) Female `(36.8%)`
  (c) Did not respond `(2.1%)`

(3) **Highest level of *completed* education** (optional)
   (a) High school/GED Degree `(15.6%)`
   (b) Some college `(31.7%)`
   (c) 4-year college `(22.8%)`
   (d) Graduate degree or higher `(16.7%)`
   (e) Prefer not to disclose `(13.2%)`

(4) **How long have you been using smartphone?**
   (a) Never `(5.4%)`
   (b) Less than a year `(6.7%)`
   (c) One year - Three years `(22.4%)`
   (d) More than three years `(65.5%)`

(5) **How long have you been using an Android device?**
   (a) Never `(41.6%)`
   (b) Less than a year `(7.1%)`
   (c) One year - Three years `(20.7%)`
   (d) More than three years `(30.3%)`
   (e) Did not respond `(0.3%)`

Post-Survey

(6) **Check ALL the permissions the App asked for:**
   (a) Record Audio
   (b) External Storage
   (c) Access Location
   (d) Record Audio
   (e) Read SMS Messages
   (f) Read Contacts
   (g) Access Photographs
   (h) Manage Phone Calls

(7) **How easy was it to recall what permissions you *accepted*? (Circle One)**

Not Easy						**Very Easy**
1	**2**	**3**	**4**	**5**	**6**	**7**

Average response: 3.75

(8) **How easy was it to recall what permissions you *rejected*? (Circle One)**

Not Easy						**Very Easy**
1	**2**	**3**	**4**	**5**	**6**	**7**

Average response: 4.26

(9) **To what degree do you agree with the statement "I felt secure using the application (Circle One)? "**

Strongly Disagree						**Strongly Agree**
1	**2**	**3**	**4**	**5**	**6**	**7**

Average response: 4.66

(10) **The overall rating I would give the app: (circle one)**

Lowest						**Highest**
1	**2**	**3**	**4**	**5**	**6**	**7**

Average response: 4.13

(11) **The "Location" permission means the app has the ability to (choose all that apply):**
   (a) Access your approximate location
   (b) View local Wi-Fi connections
   (c) Can access your home location
   (d) Update your Google maps preferences
   (e) Change network connectivity state

(12) **The "Contacts" permission means the app has the ability to: (choose all that apply):**
   (a) Read your list of contacts
   (b) Add & remove users from your contact list
   (c) Know when you communicate with one of your contacts
   (d) Add the app to your list of contacts

(13) **The "SMS" permission means the app has the ability to: (choose all that apply):**
   (a) Send SMS messages
   (b) Delete existing SMS messages
   (c) Read SMS messages
   (d) Disable the ability for the device to send SMS messages
   (e) Share your phone number with 3rd parties

(14) **Is there anything else you want to share with us about your experience with the app? ?** (optional)
   - `text response`

Note: The next question was only asked to participants who used our proposed version of the app

(15) **How helpful was it to see the other apps using the same requested permissions? (Circle One)**

Not Helpful						**Helpful**
1	**2**	**3**	**4**	**5**	**6**	**7**

Average Response: 5.13

4.3 Data Collection Process

Recruitment: Our human study was conducted at Imagine RIT 2017[1], a single day event where thousands of people from the local community visit the Rochester Institute of Technology campus and view a variety of scientific and educational venues including robotics, software projects, and engineering activities. With the assistance of student volunteers, two tables were set up with several MacBook laptops running an Android emulator, with one laptop running each version of the Android app. Participants were

[1] https://www.rit.edu/imagine/

recruited by asking visitors passing by the tables if they would like to play a tic-tac-toe game. Users were only provided vague details about the study being about Android permissions in the event pamphlet (which very few of the participants likely reviewed or recalled information about our exhibit since we were just one of the 100's of exhibitors). Users were told that they should treat the device like their own phone and were only told they were participating in a study if they asked. Importantly, we did not tell the users anything different about the different versions of the apps they were using as well. We simply asked participants to play a game and see if they could win. An approved IRB was obtained prior to beginning our study.

Data Collected We collected a varying number of participants for each app version. While we attempted to balance users evenly, participants ultimately chose the computer (each running a different app version) to sit at. This resulted in a slightly different number of users using each app version. Since our IRB only covered users of at least 18 years of age, we did not retain the results of anyone younger than this age that participated in our survey. A total of 257 people participated in our study. However, after removing participants who were underage and those who did not complete the survey, we were left with 185 users to analyze. We did experience a somewhat disproportionately low number of install-time participants, and although we are unsure why this number was lower, this occurred organically. A breakdown of the collected data is shown in Table 1.

Table 1: Collected Results

Perm-Type	# Participants
Install-time	46
Runtime	73
Proposed	66
Total	185

5 RESEARCH RESULTS

To answer our research questions, we will first provide details about our analysis methods and then examine our research questions along with their results.

5.1 Methods

As with any statistical analysis done on data sets, we had to choose the appropriate methods to accurately analyze our collected data. The purpose of this section is to describe such methods.

Shapiro-Wilk Test After cleaning our data to remove participants who did not complete the survey and were not of 18 years of age, we used the Shapiro-Wilk Test, which is a test of normality, to see if our sample came from a normally distributed population. In practice, the Shapiro-Wilk test is not applied to our entire data set, rather individual questions from the survey. For example, in order to accurately analyze the difference between overall ratings from our proposed model and the old install-time model, two tests would be run: one on the distribution of ratings from our proposed model, and one on the distribution of ratings from the install-time model. If these tests were to return a p-value below our alpha level (i.e., significance level), not only does this tell us that the data is non-normal, but it effects how we analyze our data. In our case, we found that the data we chose to analyze had significantly non-normal distributions. Further details on how this will be dealt with are explained in the next section.

Mann-Whitney U Test Having found that data in our study had significantly non-normal distributions, we chose to use the Mann Whitney U test (MWU), a non-parametric test of the null hypothesis that it is equally likely that a randomly selected value from one sample will be less than or greater than a randomly selected value from a second sample. Unlike the typical t-test used to test differences between two independent samples, MWU does not require the assumption that our data come from normal distributions.

Non-normality of data in our study is not the only reason for choosing the Mann-Whitney U test. The MWU test provides more robust analysis in two distinct ways A) because our study contains ordinal data (the Likert Scales) B) the Mann-Whitney U test is less sensitive to outliers than the t-test is. Meaning, the MWU test is less likely to indicate significance because of the presence of outliers. This allows for more precise, confident analysis.

5.2 Results

Our study examined the following research questions:

RQ1: Does the current Android runtime permission model make users feel more secure with the permissions their apps use in comparison with the previous install-time model? A primary stated goal of converting to a runtime permission model was to make users feel more secure with the permissions their apps were using [2, 3]. To better understand how effective this stated goal is, we asked participants to rate how secure they felt using each version of the app. On average, we found that users of the runtime model felt slightly more secure than those who had used the install-time model. However, the difference was not deemed significant according to the p-value returned by our MWU analysis. The results of this comparison are shown in Table 2.

Table 2: How Secure Users Feel

Install-Time (5.0)	Runtime (6.0)	p-value
4.41	4.55	0.7618

These results are surprising as one of the primary intentions of the Android 6.0 runtime model was to make users feel more secure with the permissions used by their apps. Our results demonstrate that there is no significant difference in how secure users feel, and the type of permission model used.

RQ2: Is our proposed model beneficial for making users feel more secure and be more informed? In a previous research question, we found that the current Android runtime permission model did not make users feel more secure when using the application. To evaluate our proposed model against existing models, we compared the participant questionnaires results for how secure they felt using the app. We found that our model made users feel significantly more secure compared to the older install time model and compared to the current runtime model. The MWU tests reported p-value's of less than 0.05, deeming the differences between average Likert Scale responses for our proposed model compared to the other models as statistically significant. Figure 6 represents the distribution of responses our users gave to the question *How much do you agree with the statement, 'I felt secure using this application'?*.

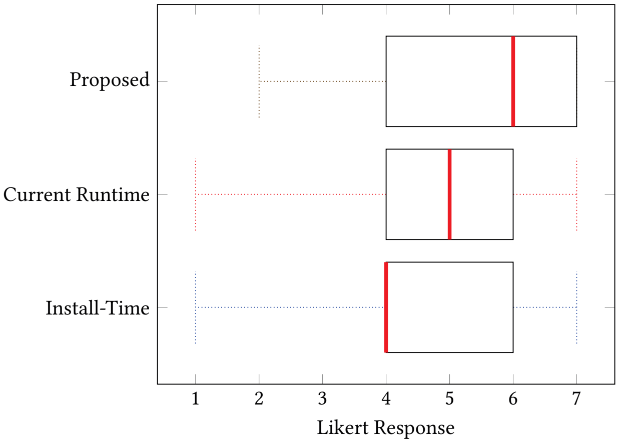


Figure 6: Distributions of How Secure Users Feel

Although our proposed model was able to make users feel more secure, it did not offer a significant difference in helping users to recall the permissions used by the app. The recall of the current model was found to be 61% while the proposed model experienced a modest increase to 67%. This difference according to our MWU test was deemed not statistically significant, and should not be generalized until further studies are conducted.

A primary difference between our proposed model and the current model is that we display extra information to users during their decision-making process. Based on our results, we believe that it is reasonable to surmise that this additional information made users feel more secure while using the application. However, we believe that it is reasonably surprising that it did not make users more informed about the permissions the apps were using.

For users of our proposed model, the questionnaire asked users *How helpful was it to see the other apps using the same requested permissions?* On a Likert scale from 1-7, 7 being the highest rating of helpfulness, we saw an average response of 5.13.

RQ3: Do users give a different rating to the three permission models? We next examined how users perceive the different models from a general ratings perspective. To accomplish this, participants were asked to provide a general rating for each app. Table 3 displays the mean and standard deviation of the collected user ratings for each model.

Table 3: Overall Rating Results

Permission Model	Mean Rating	SD
Install-time	4.61	1.55
Runtime	4.15	1.67
Proposed Model	3.77	1.83

We found that the install-time model had a fairly consistent overall rating, as its SD was lowest with 1.55 (meaning the responses were tightly focused around the mean). However, our proposed model saw more variability with an SD of 1.83, which lead us to investigate the distribution of rating responses for our proposed model. These results are shown in Figure 7.

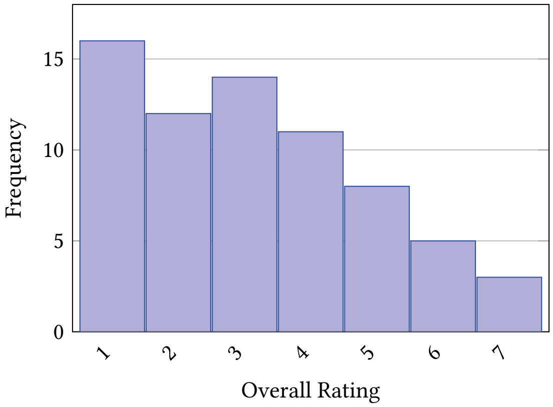


Figure 7: Distribution of Ratings for Proposed Model

The distribution of ratings for our proposed model was found to be fairly right-skewed. This means that the majority of the ratings were on the lower end (between 1-3), and the average rating is being pulled up by the few high-end responses (6-7). These results suggest that users generally had a polarized viewpoint of our proposed permission model. We next compared the 5.0 install time, 6.0+ runtime, and proposed version in a round robin fashion. We also compared the aggregate values of both runtime models against the old 5.0 install-time permission model to evaluate the user's perspective of these methods. We again used the MWU test to assess the differences in ratings between the compared groups. The null hypothesis for each comparison is that both groups come from the same rating distributions, while the alternate hypothesis is that the compared groups come from different distributions. We again used an α-value of 0.05 to determine if the null hypothesis could be rejected. Table 4 displays these results.

The old install-time vs. our proposed version, and the install-time vs. an aggregate of both runtime models were the only comparisons to return significant variations. In both cases, the older

Table 4: MWU User Ratings Results

Group 1	Group 2	p-value	Result
Install-time	Runtime	0.1552	-
Install-time	Proposed Model	0.0143	G1 > G2
Runtime	Proposed Model	0.2044	-
Install-time	Both Runtime Models	0.0321	G1 > G2

install-time model had a higher average rating. This is surprising since the newer runtime model was primarily designed to inform users and make them more comfortable with permissions [2, 3], both things that would likely lead to a higher user rating. However, we found that the opposite was true. Several comments made in the questionnaire represent how this extra information may have been detrimental to the overall user experience.

> *"It was frustrating and annoying having the app ask repeatedly for invasive permissions."*
>
> *"Because the "you are giving permission" statement came up, it made it very apparent that I was giving up a lot of privacy + info."*
>
> *"Requesting permissions "on the fly" is disruptive to the "flow" of the app and kinda creepy."*

This extra permissions feedback can hurt the user's perception of the app. We believe that this extra information in the form of runtime permissions made users rate the app lower. To shed more information on these results, we next used an MWU analysis to compare the relationship between how secure users felt and the rating they gave an app. Our null hypothesis was that all groups had the same distributions, while our alternative hypothesis was that each of the compared groups had a different distribution. Using an α-value of 0.05, we determined that there was no statistically significant relationship between the distributions of these results. However, we did find an interesting correlation between the two values. As demonstrated in Figure 8, there is an inversely proportional relationship between the models that provide more permission information and the user rating.

These results imply that additional permission information shown to the user can hurt their overall experience with the app, leading to a lower overall rating. Whether users find the extra permission information to be intrusive to their experience or merely a nuisance, our findings provide some evidence that making users more informed doesn't always lead to a higher rating for the app, and that our findings warrant future work in this area.

RQ4: How well do users recall the permissions their apps request? In the post activity questionnaire, participants were asked to select the permissions requested by the app they had just used. We then calculated their ability to recall these values. We found that users were able to recall the permissions requested by the application correct about 51% of the time. However, as shown in Figure 9, our study saw significant variability between recall performance when the data was subsetted into groups based on the application model. We found that within this variability, there was a statistically significant difference between the recall statistics of participants using the Android 5.0 model, and the participants using the Android 6.0+ runtime model. More specifically, this difference was in favor of Android 6.0+ runtime as the average recall statistic was 61%, compared to the 16.8% of the install-time model.

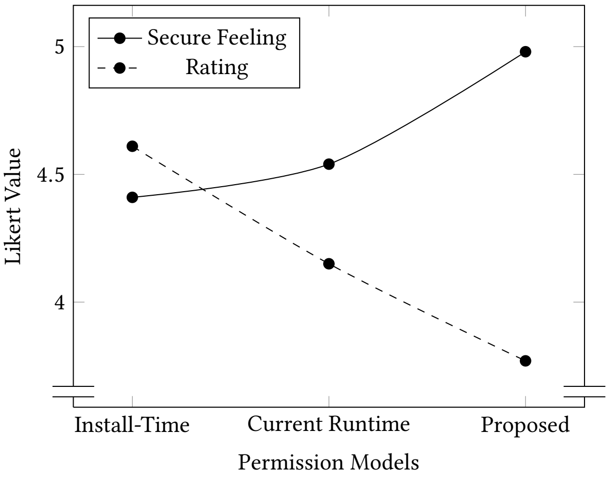


Figure 8: User's Feeling of Security and App Rating

Although the 16.8% recall statistic for Android 5.0 (install-time permissions) seems surprisingly low, from a user standpoint, there are several possible reasons for this. Since users did not know they were going to be quizzed on the permissions of the app, and since Android 5.0 required users to either accept or reject all permissions at install-time, this meant that in order to play the tic-tac-toe game, users had to accept all permissions. Therefore, if users wanted to play the game, it is likely that they blindly accepted all permissions without informing themselves with what was being requested, because all they were thinking about was playing the game. This made recalling the requested permissions that much tougher when asked in the post-activity questionnaire. The time between when the permissions were requested and when the post activity questionnaire was taken may have also played a role in the low recall for Android 5.0, as recalling permissions from install-time meant a longer time period until the questionnaire was taken than from the runtime permissions.

A primary goal of converting to the runtime permission model was to make users more informed about the permissions their apps were using. Our results demonstrate that this model is successful in this goal. However, we also found that there is still a large room for improvement in the existing model for helping users to recall the permissions that their apps are using.

Improving upon the current runtime permission model (6.0+) may ultimately revolve around the target audience for the application. While conducting further analysis about user's ability to recall permissions requested by the app, we found that as age increases, the ability to recall what permissions were requested significantly declined, as indicated by the negative correlation value and p-value in Table 5.

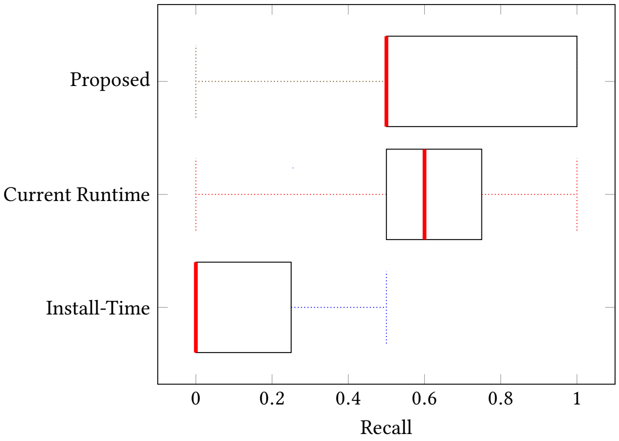


Figure 9: Distribution of Requested Permissions Recall Across Compared App Permission Models

For the above analysis, we chose to use the Pearson Correlation Coefficient (PCC) test over other methods like Spearman's Rank Correlation test. In our context, it was more appropriate to use the PCC test because we were using the raw data to analyze two quantitative variables, and we were interested in looking at the linear relationship between the variables. This "linear" relationship is important because it is one of the main differences between Pearson and Spearman. For example, looking for a linear relationship between two quantitative variables, i.e., using Pearson, means you are interested in seeing constant changes in the dependent variable, with constant changes in the independent variable. However, Spearman's test does not take into account linearity, as it just checks to see if there is a relationship between the independent and dependent variables. Spearman's test would have been more sufficient had we been interested in looking for possible logarithmic, parabolic or other non-linear relationships between the two variables. Spearman would have also been appropriate if we were interested in using ranked values to find significance (i.e., not the raw data), or had we been using ordinal data.

Table 5: Correlation Analysis of Age vs Requested Permissions Recall

X	Y	Correlation	p-value
Age	Recall	-0.206	0.0366

If developers want to take into account user's ability to recall what permissions the app requests, whether it be to make the user more informed etc., our results suggest that they may want to tailor the permissions models around the app's target audience.

RQ5: How well do users know what the permissions requested by their apps mean? In our study, we found that users were well informed, whether it be because of Android or other reasons, of what the definitions are of the permissions the application requested. There was not much variability between each of the permission models that were used, as the proposed model had an average recall of 80%, Android 6.0 (runtime) was slightly lower at 78%, while the lowest was Android 5.0 (install-time) at 74%.

Table 6: MWU Permissions Meaning Recall Results

Group 1	Group 2	p-value
Install-time	Current Runtime	0.2836
Install-time	Proposed Model	0.0930
Current Runtime	Proposed Model	0.4599

Although we did not find any statistically significant differences between application models and their respective recall statistics (results from the MWU analysis shown in Table 6), because of the results we found in *RQ*4, where age played a significant role in determining requested permissions recall, we analyzed the role that age played in the recall statistic of permission meanings. After conducting the same Pearson Correlation Coefficient analysis as in *RQ*4, we once found again that age played a statistically significant role in determining recall, denoted by the resulting negative correlation value and p-value obtained in Table 7. More specifically, we found that as age increases, user's knowledge of the meanings of requested permissions significantly declined.

Table 7: Correlation Analysis of Age vs Permission Meaning Recall

X	Y	Correlation	p-value
Age	Recall	-0.184	0.0488

One possible explanation for the results in Table 7 could be that older people may not use their smartphone devices as frequently as younger users, and therefore may not be as experienced with permission meanings [5, 29]. Our initial findings demonstrate the need for future work to be conducted in this area.

6 RECOMMENDATIONS

Our research has led to several recommendations for app developers, and for the Android OS.

(1) Our proposed changes to the current Android runtime model made users feel more secure and slightly more informed about the permissions their apps use. While we acknowledge that any change to the Android OS is challenging, we believe that this is something that should be considered.
(2) App developers should consider including more information about permissions to make users feel more comfortable and be more informed. This could entail extra popup messages or proper permission rationales. However, our results suggest that this extra information that makes users feel more secure could negatively impact their overall rating of the app.
(3) Since users were typically bad at recalling the app's requested permissions, displaying extra information such as in our

proposed model should be considered. App developers should also keep this poor recall in mind when designing apps and should consider providing additional information such as rationale to the user.

(4) Users generally rated the old, install-time permission model higher than the newer run-time models. One possible explanation of this is that users could have encountered permission fatigue in the run-time versions that showed more information. This supports the notion that users may prefer a smoother user experience as opposed to extra security information. However, more work is needed in this area before a definitive assertion may be made. Developers may want to consider these initial findings when designing the app and when analyzing user feedback.

(5) We found that age plays a large role in a user's ability to recall an app's requested permissions. Additionally, older users were generally less able to correctly state the meaning of requested permissions in comparison to younger users. Therefore, we believe that it is important for developers to keep these findings in mind when developing software typically targeted towards older users.

7 THREATS AND FUTURE WORK

There are several threats and areas of possible improvement for our work. Although users were instructed to use the laptops in the study just like they would their own phone, they were still using a laptop outside of their normal environment. This means that the user feedback and results may not properly represent what would be observed in the real world. Future work can be done to create an app that users install on their own phone and use in their own environment to collect further information for our study.

Although we sought to have a proper representative group participate in our study, our average participant was still reasonably young (31.5 years) and we had a disproportionate number of males and females (113 vs. 68). Future efforts could be made to conduct a study with a more representative user group.

While we had a large number of participants in our study (185), this still represents a very small minority of mobile phone users in the real-world. The vast majority of participants were also local to the Rochester, New York area, not for the entire world. An online study such as one ran on M-Turk could augment our work and provide further information. However, we believe that this would not be likely to yield any new, substantial findings. From our Android/smartphone experience questions in our pre-questionnaire and observations, we can surmise that a substantial portion of participants were not overly tech savvy and comprised a reasonably proper representation of the population. Our proposed model was found to have several benefits. However, future work should be done to see if our findings transcend to other permission models and mobile platforms.

This research lays the groundwork for future work in this area. In subsequent studies, we will use a different software platform to discover if the results transcend outside of mobile. We will also perform a similar study using a non-gaming app. The objective will be to discover if our findings are similar to other categories of apps with different functionality and expectations.

Although we found that our proposed enhancements to the current Android runtime model are helpful in making users feel more secure, future analysis should be done to understand why users felt this way. Are there certain elements of the messages that were more helpful than others? What can be done to make the proposed model even more impactful in terms of making users feel more secure? Additionally, our proposed model was not significantly beneficial in making users recall the permissions requested by the app, and users generally did not give it a higher rating in comparison with the existing model. Before the proposed model could be implemented, refinements need to be made to make it more impactful and beneficial in comparison with the existing model.

We found that participants generally gave a higher rating to the version of the app with the old runtime permission model. An empirical study could be done to examine how the ratings of apps in the GooglePlay store were affected when they transitioned to the runtime permission model. Our smallest group of participants was for the older, Android 5.0 install-time model. Since this group is disproportionately small in comparison to the other evaluated groups, this could be a threat to our results.

In our study, our goal was to emulate the initial app installation and usage process. One should keep in mind that permission notifications only happen the first time the permission is requested, so most of these possible notifications will only be seen by the user once, thus limiting the impact on their overall user experience when the app is used in the real world.

Future work could examine users on a more long-term scale to see the effects the permissions notifications would have on them. This type of study would likely involve consensual users installing and using several versions of the same app, but with different runtime models. Users could be surveyed after a more extensive period of use to further understand their experiences.

To provide more context to our results, an in-person lab study could be conducted. In this study, further user actions and opinions could be recorded and analyzed. However, a study of large magnitude would unlikely be replicable in a lab study.

8 CONCLUSION

We conducted a large in-person study involving 185 people to better understand user comprehension and perception of multiple Android permissions models. The study was conducted using members of the local community and involved participants completing a pre-survey, playing a simple game, and completing a post-survey. Some of our primary findings include that the current Android runtime model does not make users feel more secure than the old install-time model, and that our proposed model is beneficial in making users feel more secure than the current runtime model. We also found non-statistically significant differences between user ratings given to each app when our collected data was grouped by permission model.

REFERENCES

[1] Android, the world's most popular mobile platform. https://developer.android.com/about/index.html.

[2] Get ready for the sweet taste of android 6.0 marshmallow. https://android.googleblog.com/2015/10/get-ready-for-sweet-taste-of-android-60.html.

[3] Requesting permissions at run time. https://developer.android.com/training/permissions/requesting.html.

[4] System permissions. https://developer.android.com/guide/topics/security/permissions.html.

[5] Technology use among seniors. http://www.pewinternet.org/2017/05/17/technology-use-among-seniors/.

[6] P. Andriotis, S. Li, T. Spyridopoulos, and G. Stringhini. *A Comparative Study of Android Users' Privacy Preferences Under the Runtime Permission Model*, pages 604–622. Springer International Publishing, Cham, 2017.

[7] D. Arenberg. The problem of comparing recall and recognition in young and old adults. *Manuscript submitted for publication*, 1985.

[8] M. Conti, V. T. N. Nguyen, and B. Crispo. Crepe: Context-related policy enforcement for android. In *Proceedings of the 13th International Conference on Information Security*, ISC'10, pages 331–345, Berlin, Heidelberg, 2011. Springer-Verlag.

[9] F. I. Craik and J. M. McDowd. Age differences in recall and recognition. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 13(3):474, 1987.

[10] S. Farrell and S. Lewandowsky. Empirical and theoretical limits on lag recency in free recall. *Psychonomic Bulletin & Review*, 15(6):1236–1250, Dec 2008.

[11] 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 '12, pages 3:1–3:14, New York, NY, USA, 2012. ACM.

[12] P. Gerber, M. Volkamer, and K. Renaud. The simpler, the better? presenting the coping android permission-granting interface for better privacy-related decisions. *Journal of Information Security and Applications*, 34:8 – 26, 2017. Human-Centred Cyber Security.

[13] M. J. Kahana, M. W. Howard, and S. M. Polyn. Associative retrieval processes in episodic memory. 2008.

[14] M. J. Kahana, M. W. Howard, F. Zaromb, and A. Wingfield. Age dissociates recency and lag recency effects in free recall. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 28(3):530, 2002.

[15] A. Kaur and D. Upadhyay. Pemo: Modifying application's permissions and preventing information stealing on smartphones. In *Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference -*, pages 905–910, Sept 2014.

[16] D. E. Krutz, N. Munaiah, A. Peruma, and M. W. Mkaouer. Who added that permission to my app?: an analysis of developer permission changes in open source android apps. In *Proceedings of the 4th International Conference on Mobile Software Engineering and Systems*, pages 165–169. IEEE Press, 2017.

[17] 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 '12, pages 501–510, New York, NY, USA, 2012. ACM.

[18] 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 '14, pages 201–212, New York, NY, USA, 2014. ACM.

[19] J. K. MacDuffie and P. A. Morreale. *Comparing Android App Permissions*, pages 57–64. Springer International Publishing, Cham, 2016.

[20] 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 '17, pages 362–373, New York, NY, USA, 2017. ACM.

[21] N. Munaiah, C. Klimkowsky, S. McRae, A. Blaine, S. A. Malachowsky, C. Perez, and D. E. Krutz. Darwin: a static analysis dataset of malicious and benign android apps. In *Proceedings of the International Workshop on App Market Analytics*, pages 26–29. ACM, 2016.

[22] M. Nauman, S. Khan, and X. Zhang. Apex: extending android permission model and enforcement with user-defined runtime constraints. In *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 328–332. ACM, 2010.

[23] G. Paul and J. Irvine. Achieving optional android permissions without operating system modifications. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5, May 2015.

[24] B. Rashidi, C. Fung, and T. Vu. Recdroid: A resource access permission control portal and recommendation service for smartphone users. In *Proceedings of the ACM MobiCom workshop on Security and privacy in mobile environments*, pages 13–18. ACM, 2014.

[25] B. Rashidi, C. Fung, and T. Vu. Android fine-grained permission control system with real-time expert recommendations. *Pervasive and Mobile Computing*, 32:62 – 77, 2016. Mobile Security, Privacy and Forensics.

[26] S. Rasthofer, S. Arzt, E. Lovat, and E. Bodden. Droidforce: Enforcing complex, data-centric, system-wide policies in android. In *Proceedings of the 2014 Ninth International Conference on Availability, Reliability and Security*, ARES '14, pages 40–49, Washington, DC, USA, 2014. IEEE Computer Society.

[27] G. L. Scoccia, I. Malavolta, M. Autili, A. Di Salle, and P. Inverardi. User-centric android flexible permissions. In *Proceedings of the 39th International Conference on Software Engineering Companion*, ICSE-C '17, pages 365–367, Piscataway, NJ, USA, 2017. IEEE Press.

[28] J. Spaniol, D. J. Madden, and A. Voss. A diffusion model analysis of adult age differences in episodic and semantic long-term memory retrieval. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(1):101, 2006.

[29] A. van Deursen, C. Bolle, S. Hegner, S. Hegner, and P. Kommers. Modeling habitual and addictive smartphone behavior: The role of smartphone usage types, emotional intelligence, social stress, self-regulation, age, and gender. *Computers in human behavior*, 45:411–420, 2015.

[30] 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'15, pages 499–514, Berkeley, CA, USA, 2015. USENIX Association.

[31] L. Yang, N. Boushehrinejadmoradi, P. Roy, V. Ganapathy, and L. Iftode. Short paper: Enhancing users' comprehension of android permissions. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, pages 21–26, New York, NY, USA, 2012. ACM.