On a (Per)Mission: Building Privacy Into the App Marketplace

Hannah Quay-de la Vallee
Brown University
hannahqd@cs.brown.edu

Paige Selby*
Brown University

Shriram Krishnamurthi
Brown University

ABSTRACT

App-based systems are typically supported by marketplaces that provide easy discovery and installation of third-party apps. To mitigate risks to user privacy, many app systems use permissions to control apps' access to user data. It then falls to users to decide which apps to install and how to manage their permissions, which many users lack the expertise to do in a meaningful way. Marketplaces are ideally positioned to inform users about privacy, but they do not take advantage of this. This lack of privacy guidance makes it difficult for users to make informed privacy decisions.

We present both an app marketplace and a permission management assistant that incorporate privacy information as a key element, in the form of permission ratings. We discuss gathering this rating information from both human and automated sources, presenting the ratings in a way that users can understand, and using this information to promote privacy-respecting apps and help users manage permissions.

1. INTRODUCTION

App-based devices have become pervasive in consumers' lives [1], due in part to apps' easy installation model. Most app ecosystems are supported by a central marketplace that enables users to easily search for, investigate, and install apps, allowing users of all levels of technical ability to customize their devices. However, the amount of user information associated with these devices makes third-party apps a threat to user security and privacy. The expansion of the app model beyond smartphones to platforms such as desktops, cars, and the Internet of Things exacerbates this concern.

Many platforms try to mitigate these risks by requiring users to grant permission before apps can access certain hardware resources and user data. Unfortunately, such systems force users, even technical novices, to manage their own privacy without assistance. Furthermore, most systems ask for consent either at or after installation time, when users have already chosen an app, making it onerous for them to switch apps if they dislike an app's permission requests.

App stores, as a primary source of app information, are ideally positioned to act as a fulcrum to assist users in managing their privacy. In fact, marketplaces *already* influence users' decisions by ranking apps and thus filtering which apps users see. Unfortunately, marketplaces are not incentivized to put user privacy first. It is even possible, given the profit model of many app stores, that apps that use more ad libraries are put first.

Apart from baseline protection like malware detection, most app marketplaces do not use their position to better inform or protect users. Google Play, Android's proprietary marketplace, allows users to search by price and star rating but does not provide privacy-based search options, nor any privacy guidance past simply listing apps' permissions. As a result, many users can only give *uninformed consent* to permission requests, as they are ill-equipped to judge whether an app's permissions are appropriate for its purpose.

Worse, users who *do* have judgements about apps' permissions have no good way to express themselves to other users or to developers. Some try to communicate their opinions via app reviews, but these are difficult to find amongst the myriad reviews. Worse still, developers who *want* to explain their app's permission requests also lack a dedicated forum to do so. Some developers use their app's description page but, since this is not standard practice, it is easy for users to miss, especially in Google Play, where long descriptions are hidden by default. Other apps, like Pinterest, explain their permission requirements on their websites [3], where only a very motivated user is likely to find it.

Despite this quagmire, user reviews *have* been a useful privacy tool, harnessed by researchers to inform users about the consequences of updating their existing apps [23], and by developers to read user opinion and guide app development. For instance, an update of the Avis car rental app added the "retrieve running apps" permission. This led users to leave a spate of negative reviews, spurring the app's developers to remove the permission. These examples show that reviews can be a valuable source of information about app permissions, but current marketplaces limit their effectiveness by making them difficult for users and developers to find.

We have built two Android apps[1] that leverage privacy

*This author is now at Google.

SPSM'16, October 24 2016, Vienna, Austria

ISBN 978-1-4503-4564-4/16/10…$15.00

DOI: http://dx.doi.org/10.1145/2994459.2994466

[1]Although our apps are built for Android, the concepts apply to a wide range of app platforms, such as Chrome browser extensions and Internet of Things apps. Ideally, our apps' features would be built into these platforms' proprietary marketplaces.

Do you use ⟨app⟩? ▷ Yes: *No follow-up question* ▷ No: Do you use a similar app?
Do you think there are other apps that could be used in place of ⟨app⟩? ▷ Yes: Can you think of any examples of apps that could be used in place of ⟨app⟩? ▷ No: Why do you think that ⟨app⟩ is unique?

Table 1: Classification survey questions for each app, with ⟨app⟩ replaced by the app's name. Workers were given the description of each app from the Play store.

rating information to help users make informed privacy decisions. We also allow developers to respond to these ratings, thereby providing a channel for communication between users and developers. The first app is a privacy-conscious marketplace, which helps users to find privacy-respecting apps. The second is a permission management assistant to help users regulate their apps' permissions after they are installed.

This paper makes several contributions. First, in Section 2, we show that users face privacy decisions both when selecting which apps to install and when managing their apps after installation. Second, in Section 4 we use crowdsourcing and automated tools to collect ratings of apps' permissions to assist users with their privacy decisions, and in Section 5 we show how these ratings can be used to promote privacy-respecting apps in a marketplace. In Section 6 we discuss the crowd feedback-based method we used to develop an interface for presenting rating information to users, including some unexpected subtleties in the design of such an interface. All of these features are incorporated into our two apps, which are discussed in Section 3.

2. PRIVACY DECISIONS FACING USERS

To better help users with privacy decisions, we needed to understand what types of choices users actually make. At first blush, users face two types of privacy decisions: which apps to install and how to manage their apps' permissions after installation. If users need a *specific* app, managing permissions after installation is the only way for users to protect their privacy, and so they could benefit from a tool to manage the permissions of installed apps, which would require privacy ratings for each permission. However, there may also be times where users can choose between similar apps, in which case a privacy-conscious store, with overall ratings for each app, would be helpful. To determine which tools to build, we studied whether users ever have a meaningful choice between different apps.

We posted Mechanical Turk surveys for 66 Android apps. For each app, we showed workers the app's description from Google Play, and asked whether workers thought that app was replaceable. If they thought it was, we asked if they could name an example substitute. If they thought the app could not be replaced, we asked why they felt it was unique. These questions are shown in Table 1. We then asked several demographic questions.

To select the 66 apps, we used the MarketBot scraper [2] to collect the descriptions of the top five apps in 11 of Google Play's categories, along with five white noise apps. We also chose six apps that were closely tied to a service external to the app, such as the Stop and Shop app, which is only useful at a physical Stop and Shop store. All of the apps had at least 100,000 installs, and only eight apps had less than 1M installs, suggesting that all the apps were interesting to a broad range of users. Table 2 in the appendix shows the complete list of apps.

Each survey asked about three to five apps, and no survey contained two apps from the same category. We gathered 10 to 12 responses for each survey. Our workers were 61% male and 39% female, had an average age of 29, and 84% were from the United States and 16% were from India.

Apps varied significantly in their substitutability, (ANOVA, $p < 0.001$), indicating that some apps are interchangeable, while other apps provide unique functionality, tying users to that app. Rather than dividing clearly into replaceable or unique, however, we found that apps fall along a spectrum of substitutability. On one end are *single-source* apps, which offer unique functionality that cannot be replicated by a different app. Instagram is an example of a single-source app, as less than 20% of workers felt it could be replaced. On the other end of the spectrum are *generic* apps, such as Waze, which 100% of workers felt was replaceable. In the middle are *mixed-mode* apps, which can be either single-source or generic depending on the user. For example, consider Strava, an app that allows users to track their physical activity and compete with friends. For users who only use the tracking features, it could be replaced by a similar app, such as MapMyRide. Other users might care deeply about the social features of Strava, and so other apps would not be an acceptable substitute.

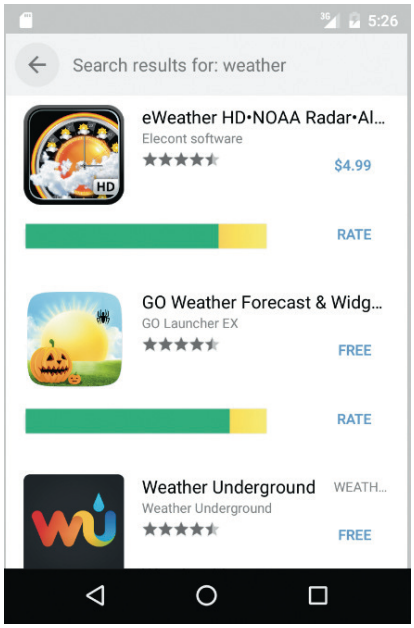
Although there were not clear groupings of apps, some categories were more substitutable than others. For example, apps in the "social" category were considered, perhaps unsurprisingly, significantly less substitutable than apps in the "travel_and_local" category (Tukey's HSD, $p < 0.01$).

Ultimately, whether a given app is replaceable depends on the user, and therefore apps cannot be classified a priori. Overall, however, 30% of apps were considered "substitutable" by at least 75% of our workers, and 77% of apps were considered substitutable by at least 50% of workers. This indicates that users, whether they are aware of it or not, are making two distinct types of privacy choices: which apps to install (for generic apps), and how to manage apps' permissions after installation (for all apps, but most importantly single-source apps).

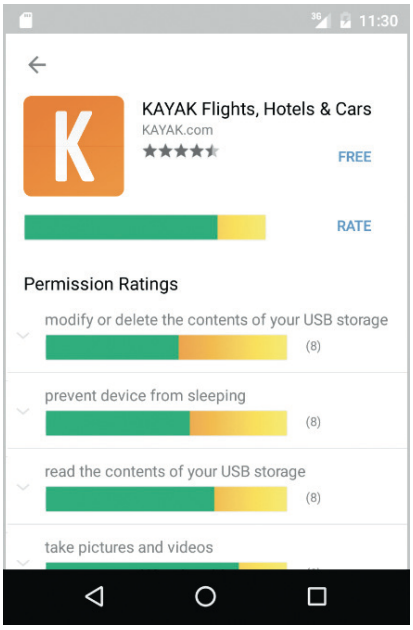
3. THE APPS

The two types of privacy decisions discussed in Section 2 require two approaches to assisting users. A privacy-aware marketplace would aid users with installation decisions by helping them find more privacy-respecting apps. A privacy assistant could help users manage their apps' permissions after they are installed on users' devices. We split these two approaches into two separate apps, the PerMission Store, and the PerMission Assistant[2]. Dividing the functionality into separate apps means that users who are only interested in one app are not required to accept the risks of both. In particular, the Assistant needs to access the list of apps the user has downloaded, information the Store does not need. Both apps already contain information for approximately 1500 Android apps from Google Play leaderboards, and are continuing to collect information for more apps.

[2]Both apps are available from `OnAPermission.org`.



(a) The search results page.



(b) The Kayak app page.

Figure 1: Screenshots of the PerMission Store.

3.1 The PerMission Store

The PerMission Store (shown in Figure 1) is designed to be a comprehensive app store, so, in addition to privacy ratings, it includes apps' description, screenshots, icon image, star rating, developer, category, and price from Google Play[3] and allows users to search and browse through apps, and rate permissions. There is one notable feature our store does not provide: it relies on the Play store to actually install apps. When users click to install an app in the PerMission Store, they are taken to that app's page in the Play store, where they can then install the app. Ideally, users would complete the entire process within our marketplace, but this would expose users to insecurity by requiring third-party downloads and by bypassing the malware protections in place in the Play store.

The PerMission Store displays privacy ratings at two levels: the permission-level and the app-level. Both levels of rating are represented with percentage bars developed via a series user interface design studies (Section 6). The permission-level ratings are comprised of both automated and human ratings as described in Section 4.3 and provide users with detailed information they can use to make privacy decisions. These ratings are unique to a given app-permission combination, and so the same permission may have a different rating on different apps.

App-level ratings are calculated from permission-level ratings (see Section 4.3), and serve several purposes. First, they are incorporated into the PerMission Store's ranking mechanism (discussed in Section 5), which is used to sort responses to user search queries, thus allowing the PerMission Store to promote more privacy-respecting apps. They also provide a broad privacy overview, making it easier for users to compare apps. Throughout the marketplace, an app's app-level privacy ratings are displayed next to its star rating from the Play store so that users can weigh both when choosing apps.

When users search or browse apps, they are shown tiles that display the apps' general information, like name, developer, app-level privacy rating, star rating, and price (see Figure 1(a)), as well as links to rate or install the app. If a user clicks one of these tiles they are taken to the app's page (an example of which is shown in Figure 1(b)), which has more detailed information like permission-level ratings and comments, and the app's description. The permissions are ordered worst-rated to best to ensure that users see the most worrisome permissions.

3.2 The PerMission Assistant

The PerMission Assistant (shown in Figure 2) helps users manage permissions for apps they have already installed. Because user time and attention is limited, the Assistant sorts a user's installed apps by their worst-rated permissions, which allows users to address the most concerning permissions first. It is thus useful for apps the user installed before the PerMission Store was available, and for single-source apps where the user cannot switch to a more privacy-respecting alternative. The Assistant allows users to run these apps within their own privacy limits. Because it relies on the ability to turn individual permissions off, the PerMission Assistant requires Android Marshmallow, while the PerMission Store can be used with any Android version.

The PerMission Assistant uses the same interface elements as the PerMission Store to display an app's permission ratings and provides a link to manage a given app's permissions. Because we cannot actually edit other apps' settings, this link takes them to the app's page in their device's settings. This is, of course, a security necessity, because Android should not allow apps to adjust each others' permissions. However, it does mean that we cannot display privacy ratings on the actual adjustment screen in settings. This is an issue that could be solved if these ratings were incorporated into the Android infrastructure.

[3]Scraping the Play store, while not explicitly prohibited in the letter of the Terms of Service, is somewhat counter to their spirit. Integrating our store into Google Play would render this step unnecessary.

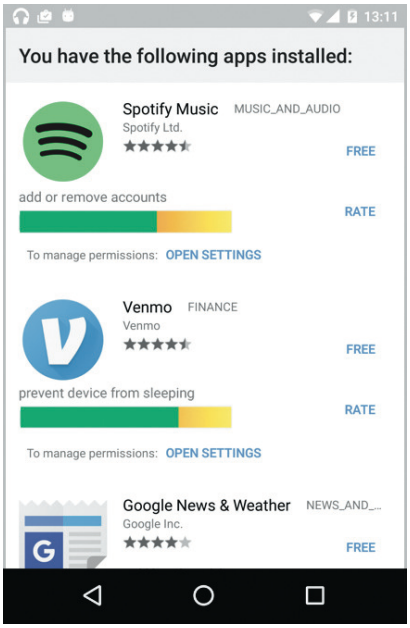


Figure 2: The home page of the PerMission Assistant.

4. POPULATING PRIVACY INFORMATION

The essential feature of our apps is privacy information, which we gather from two sources: an automated tool and human raters. As discussed in Section 3, our apps use both permission-level and app-level ratings. Since we cannot know, for a given app, whether a user will need permission-level ratings (to manage permissions) or app-level ratings (to choose between apps), we collect ratings for all apps at the permission level and compute an app-level rating from the permission-level ratings. Section 4.1 and Section 4.2 discuss collecting ratings from automated and human sources, and advantages and disadvantages of each. Section 4.3 discusses combining the human and automated ratings and calculating the app-level rating.

4.1 Automated Ratings

The research community has developed a number of systems that use automated techniques to provide privacy and security information about Android apps. Some attempt to identify malware apps [29, 30], while others detect worrisome permissions or suspicious handling of user data [10, 13, 26]. Section 7 offers further discussion of such systems.

These automated tools can provide objective, quantitative privacy information for a large number of apps at low cost. We use ratings from the DroidRisk system [25], which analyzes permission request patterns in both malware and benign apps to assign a risk score to each permission. (Since Android has added new permissions since the development of DroidRisk, it does not provide scores for all the current permissions.) It should be noted that we are repurposing DroidRisk, which was designed as a malware-detection tool rather than as a rating system for legitimate apps.

While these ratings are useful, automated tools suffer from many shortcomings. They are often difficult to use, even for sophisticated users (the authors of this paper were unable to get many of these tools to run). They provide little-to-no qualitative feedback, such as discomfort or confusion about permissions. Finally, many of these tools cannot consider the *context* of a permission (accessing contact data may be worrisome for a flashlight app, but not a messaging app). Because context and qualitative information like how users feel about a certain permission are important elements in user decisions, our apps use the DroidRisk ratings primarily as a complement to the human ratings.

4.2 Human Ratings

To capture the full range of users' concerns and to provide an on-going feedback mechanism for developers, our apps incorporate human ratings and reviews, similar to the star ratings and text reviews in Google Play, along with the DroidRisk ratings.[4] However, human ratings present a bootstrapping problem: Users will likely only use our apps if they contain ratings, but without ratings, the apps would struggle to gain the users necessary to rate apps. Our apps could initially rely only on automated ratings, but they would then suffer from the shortcomings of automated tools.

One option for seeding text reviews would be to mine the existing app reviews in Google Play, searching for permission relevant text. However, Google Play makes it difficult to gather more than a sample of reviews for each app (40 per app, as of June 2016). The Play Store itself, should it ever integrate our apps' features, could leverage the complete database of existing reviews.

To offer human ratings right away, our apps use crowdsourced ratings from Mechanical Turk, which offers a cost-effective platform with a supporting body of academic research [18]. Although the Play store offers millions of apps, many of these apps are not at all widely used, so we have focused our seeding on popular apps by pulling from the Play Store's leaderboards (this is similar to the star ratings in the Play Store, where popular apps generally have numerous ratings while less popular apps may have few, if any). We have seeded our apps with crowdsourced ratings for over 1500 apps, and we are continuing to collect more. (The cost-effectiveness of Mechanical Turk enabled us to do this within the bounds of a limited research budget.)

While crowdsourcing solves the bootstrapping problem, it raises concerns about whether workers take rating tasks seriously. (They might, for example, assign random ratings to finish the task as quickly as possible to maximize their income.) We thus performed a study to evaluate the quality of Mechanical Turk ratings.

We surveyed workers about 14 apps: Facebook, Gmail, Pandora, Angry Birds and ten weather apps, with 20-30 workers per app. For each app, we provided workers with its description and required permissions. We instructed workers to imagine that they were considering installing the given app and asked them, "Which, if any, of the permissions did you find unacceptable, and why?" They had to label each permission as either "acceptable" or "unacceptable," and could explain each rating in an optional text box.

We reviewed the text responses explaining the ratings. First, we found that more than 60% workers did provide explanations for their ratings, despite this being optional. Fur-

[4]The average user is not a security expert, and thus may "mis-rate" a permission because they misunderstand its purpose. However, our apps aim to serve as a communication channel for users and developers, and "incorrect" ratings signal to developers that they are not adequately explaining their apps' permissions, and to the Android team that a permission is confusing or misleading.

thermore, their responses were relevant to the permissions being discussed, indicating that the workers performed the task seriously.

We also evaluated the quality of the binary ratings. This presented a challenge because, as ratings are essentially opinions, there is no ground truth against which to evaluate. We could measure agreement between workers with Fleiss's κ measure of inter-rater reliability, but low agreement would not necessarily mean that workers were negligent, since there could be valid disagreement. However, we *would* expect workers to agree on some of the permissions, particularly non-controversial ones, leading to a range of agreement across permissions. We computed κ scores for each permission and found that the scores ranged from -0.1 (significant disagreement) to 1.0 (total agreement). The scores aligned with our intuition about which permissions would be non-controversial. For example, `coarse-grained location` had $\kappa = 1.0$ for all weather apps, which is unsurprising, as a weather app needs to fetch local conditions.

These findings suggest that Mechanical Turk is a viable method for seeding ratings for an initial corpus of apps. That said, we consider the crowdsourced ratings to be temporary. As we amass ratings from in-the-wild users, we will phase out crowdsourced ratings.

4.3 Merging Human and Automated Ratings

While having both human and automated ratings helps mitigate the shortcomings of each, it could be confusing and overwhelming for users to consider two ratings for every permission and to understand the distinctions between them. Thus, we merge each permission's human and automated ratings together, so that users can see questionable permissions at a glance.

Calculating the combined rating depends on whether the permission is in the DroidRisk corpus. If it is not, and thus does not have an automated rating, we take the average of its human ratings. If the permission *does* have an automated rating, we take a weighted average of the automated rating, denoted by ar, and the average of the human ratings, denoted by hr. The overall rating PR for a permission p is given by:

$$PR_p = (0.25 \times ar_p) + (0.75 \times hr_p) \quad (1)$$

where both ar and hr normalized to be between 0 and 1. Automated ratings are given a lower weight because they are a less nuanced metric than human ratings.

After computing a single rating for each permission, we have to calculate an overall privacy rating for each app. This app-level rating makes it easier for users to compare between multiple apps, and is necessary for ranking apps. An app that requires no permissions is given a privacy score of 1 (the best possible rating), because, from a permission standpoint, it is innocuous. For an app that does request permissions, we need to calculate an overall rating from its permissions' ratings. A naive approach would be to average the permissions' ratings (perhaps with some sort of weighting). However, an average would suffer a significant drawback: the aggregate rating would always be either equal to or *better* than the app's worst rated permission. As a result, an unscrupulous developer could hide a suspicious permission by requesting a large number of innocuous-seeming permissions. To avoid this, our marketplace uses an app's worst permission rating as the overall rating.

5. RANKING APPS

While the privacy ratings can help users choose between apps, a privacy-conscious marketplace should also promote privacy-respecting apps so that users can find them in the first place. In particular, the marketplace should incorporate apps' privacy ratings into its search function so that apps with better privacy scores are ranked higher in results. However, the marketplace cannot simply sort results by privacy rating; users need apps that are functional and relevant to their needs, as well as privacy preserving.

One option would be to replicate the Play store's ranking for a given query and combine those rankings with our privacy ratings to sort apps. However, as discussed in Section 1, the Play store may rank apps in a way that is contrary to users' privacy interests, so integrating their ranking could undercut our goals. Also, the Play store's ranking method is opaque and could rely on privileged information, and so may be irreproducible. Thus, we need another way to incorporate functionality and relevancy.

Our marketplace uses apps' star ratings from the Play store as a proxy for functionality. These ratings are supplied by users, not by Google, and therefore do not present the same concerns as the Play store's ranking function.

To incorporate relevancy, we leverage our database of apps. The scraped app data are stored in a Postgres database. Postgres provides built-in text search that, given a search query, calculates a relevancy score for each record based on how often and where the query appears. Our marketplace searches against apps' title and description to get the relevancy score.

Given privacy, functionality, and relevancy information, we need compute a single ranking number because the marketplace ultimately needs a sort order for apps. Although we are building a privacy-conscious marketplace, relevancy is the most important factor, followed by functionality, since users will not be satisfied with irrelevant or dysfunctional apps, no matter how privacy preserving. We use a weighted sum of all three components, so an app a's rank for a query q is defined by:

$$Rank_{aq} = r_{aq} + (0.25 \times f_a) + (0.2 \times p_a) \quad (2)$$

where r_{aq} is the relevancy score for app a on query q, f_a is its functionality rating, and p_a is its privacy rating, and r_{aq}, f_a, and p_a are normalized to be between 0 and 1.

A close examination of this equation reveals another reason that relevancy is weighted more heavily than functionality and privacy: it is the only component that depends on the search query. If a user does not find an app they want after their initial query, and they try a second query, we want to return different results. If we rely too heavily on functionality and privacy we run the risk of returning identical results for slightly different queries, thus frustrating users.

6. THE USER INTERFACES

The interfaces shown in Figure 1 and Figure 2 were not randomly designed, but rather developed via an iterative process incorporating user feedback. As our apps are intended to communicate privacy information to users, the interfaces are of critical importance; they should help users understand apps' privacy risks so users can make *informed* decisions, without requiring significant effort. During our development process, we discovered that efficiently display-

ing permission ratings without confusing or misleading users is a surprisingly subtle problem. We started with several candidate interfaces, such as those shown in Figure 3, and used feedback from Mechanical Turk to improve on our designs. We discuss this design process in Section 6.1. We also studied our most promising design in greater depth, which we discuss in Section 6.2.

6.1 Iterative Design Process

The iterative design process used feedback from Mechanical Turk workers to evaluate and improve initial designs. We requested workers who were Android users, so that they would be familiar with the permission screen. Although we would expect users of our apps to have some sense of their purpose, we did not give workers any background about the purpose of the interfaces. We only told them that we were testing out a new Android interface, and asked them what they thought the new interface elements meant. (For example, for Figure 3(a), workers were asked, "What do you think the icons next to 'Network communication' means?") While this may seem unrealistic, we chose not to provide background information for two reasons. First, even if a user generally understands that our apps are privacy-conscious, they may not understand which specific elements have to do with privacy, since the marketplace app also displays a lot of non-privacy-related information. Second, we hope that our privacy features will ultimately be incorporated into official marketplaces, such as Google Play. In this case, users may not be aware of the privacy features, and it is important that our interface does not confuse or mislead them.

For each of our candidate interfaces, we surveyed approximately ten workers, and used their responses to improve our designs, or to eliminate designs that were too confusing. We discovered that many designs that we expected to be intuitive failed to convey the desired information, and in some cases *actively misled* workers. To illustrate, we'll discuss some iconographies that proved confounding.

One seemingly-intuitive way to display privacy ratings would be the five-star system (Figure 3(a)), since it is possibly the most common iconography for user ratings, and is already used in Google Play to display apps' functionality ratings. However, workers thought the ratings indicated how well the permissions' services worked. For example, some workers thought the rating next to "Network communication" showed the strength of the network signal.

As this confusion may have been due to the way this interface overloads the star icons (using them for both functionality ratings and permission ratings), we tried replacing stars with other privacy-relevant symbols, such as locks (Figure 3(b)) and Guy Fawkes (or *V for Vendetta*) masks. However, these iconographies caused their own brand of confusion. Workers thought that the masks indicated protection from government data collection, perhaps because they are often associated with the "hacktivist" group Anonymous. Locks led workers to believe that a permission's service was restricted (perhaps because app developers often use locks to mark restricted features). Additionally, workers could not tell whether more locks indicated a more positive or more negative rating. This confusion, which we dubbed the *better-or-worse* phenomenon, arose in multiple interfaces.

We also explored a simple design in which each permission had either a green checkmark indicating users approved of the permission or a red X indicating they did not approve (Figure 3(c)). This interface caused a serious misconception: the red X was meant to indicate a potentially invasive permission, but workers thought it meant that the given permission had been *disabled*. This extreme case of the better-or-worse phenomenon is alarming, as users would think the most disconcerting permissions were completely harmless!

Another interface we explored indicated each permission's rating using a colored rectangular bar. The percentage of the bar that was filled by color indicated approximately the percentage of raters who considered a given permission to be acceptable. Our original design, shown in Figure 3(d), used red, yellow or green on a white background, but some workers perceived the all-green bar as a signal to proceed without caution, which could encourage users to download an app without considering the permissions at all. To rectify this, we used green to indicate the positive percentage of the rating, and a gradient from red to yellow to indicate the negative percentage. Thus, any permission with even some negative rating will have some "warning color," to avoid over-soothing users. The percentage bars proved effective at conveying the rating information without confusing users.

The above discussion offers an overview of design process and a sample of our interfaces. The full collection of iconographies we explored is shown in Figure 5 in the appendix.

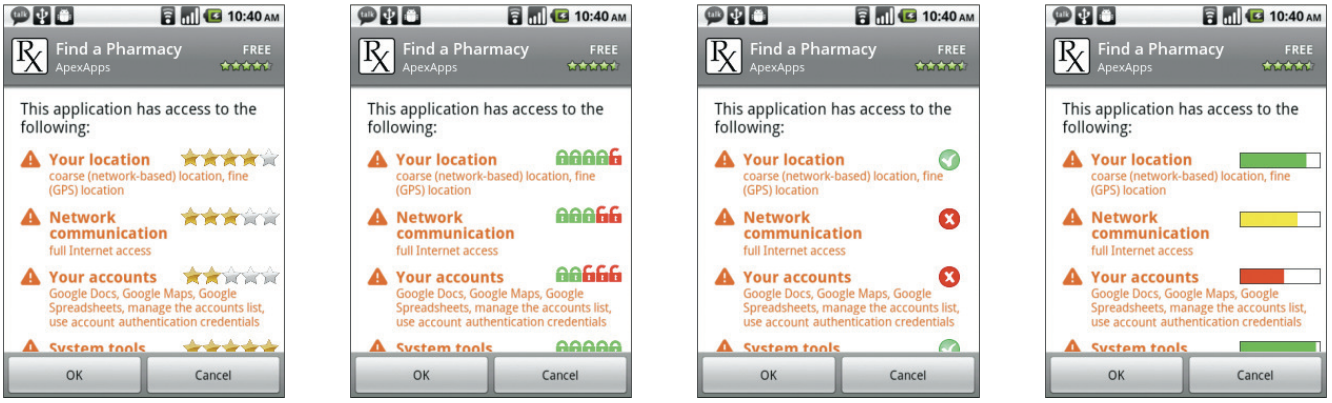
6.2 In-Depth Testing

To validate the percentage bar interface, we performed two larger studies. The first had 83 respondents and, like our exploratory studies, did not offer workers any context for the interface. The second study had 77 respondents, and provided workers with a brief explanation of the interface's purpose. As before, we recruited Android users.

In the first study we started by asking workers for a free-response explanation of the iconography. We manually coded these responses using a rubric that we revised until we obtained an inter-coder reliability score (κ) of 0.835. The rubric had three categories: *Predominantly Correct Interpretation*, *Semi-Correct Interpretation*, and *Incorrect Interpretation*. Over 40% of workers had a predominantly correct interpretation, and over 55% had either predominantly or semi-correct interpretations. As these workers had been given no explanation for the interface, these results represent a worst-case baseline. If our ratings and interface are incorporated into a *general* marketplace where users might lack context for their purpose, users should be given an explanation of the ratings, such as a start-up tutorial.

After the workers completed the free-response questions, we informed workers that the icons represented privacy ratings for each permission. We asked workers to tell us how clear they thought this was, on a four-point Likert-type scale from "completely unclear" to "completely clear." The responses (shown in Figure 4), indicate that workers generally understood that the interface was displaying privacy information, but did not always correctly interpret that information, indicating that some brief on-boarding information, such as tool-tips, may be useful. The ratios of answers is significantly different from chance (χ-squared test, $p < 0.05$).

Our second study measured how well workers understood our interfaces given some context for their purpose. We told workers "we are studying a new app marketplace that includes user ratings of apps' privacy. Please examine the interface shown here and answer the following questions about the interface," but did not tell them how to interpret the



(a) The five-star interface. (b) The five-lock interface. (c) The checkbox interface. (d) The first bar interface.

Figure 3: Four candidate interfaces for permission ratings.

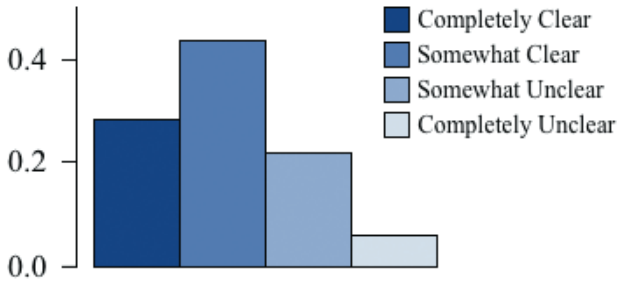


Figure 4: The percentage of responses in each category for the Likert-type question asking workers whether it was clear that the icons represented privacy ratings.

icons. We then asked workers the same free response question as in the previous study. Unsurprisingly, workers who were given this background had a better understanding of the interface. More than 60% of workers had a predominantly correct interpretation, and more than 70% had either a predominantly or semi-correct interpretation. This study reflects a more realistic scenario for our apps (as opposed to privacy features in a general marketplace like Google Play).

To confirm that our permission rating interface was effective in a deployed app, not just static screenshots, we did a small in-person user study of our PerMission Assistant. We chose the Assistant because it is easier for users to jump into, in that it does not force users to have to come up with queries; it is also a more natural entry point, since it helps users audit apps they are already invested in. We recruited eight users (seven undergraduates and one university staff member) to use the app for just 3–5 minutes and then explain what they thought the app did and what the colored bars meant. Seven of them were able to determine that the app provided information about permissions' invasiveness and to assist in adjusting apps' permissions. Six of the eight understood that the colored bar indicated how concerning a given permission was, with more green signifying less concern. This suggests that our interface remains intuitive to users in the context of a fully-functional app. A full-fledged evaluation in lab and in the field is future work.

7. RELATED WORK

There are a number of "permission manager" apps on Google Play, many of which simply reorganize the information provided in the Android settings, and do not offer any additional privacy information. Some highlight "risky" apps, but it is not clear how they are calculating risk [6, 7]. Many appear to use the number of permissions a given app requests, which is an unreliable metric. There are also managers that remove other apps' permissions by altering the apps' APKs [4], or require root access to disable permissions [5], which are significant threat vectors in their own right and do not actually help users make privacy choices (and are of limited use since the release of Android Marshmallow, where permission toggling is a built-in feature). None of these tools provides the structured permission ratings and reviews available in our PerMission Assistant.

Almuhimedi et al. [8] show that a permission manager can be helpful to users in managing their privacy. Liu et al. [17] present a personalized privacy assistant (PPA) that engages users in a dialogue to determine a privacy profile for the user, which the manager then employs to suggest permission settings to the user. Although similar in concept, by focusing on publicly viewable ratings, our system can both let users explore how *other* users understand permissions, and serve as a channel of communication amongst users, developers, and the Android team. Our Assistant could be incorporated with the PPA to provide a more complete tool.

Highlighting the value of privacy information in the marketplace, researchers such as Felt et al. [14] have found that smartphone users take privacy risks seriously, but Chin et al. [11] show that although smartphone users are careful about performing certain tasks, they engage in risky behavior when it comes to installing apps, suggesting that users could benefit from a more privacy-conscious marketplace. Tsai et al. [24] built a search engine annotated with privacy scores for the merchants. They found that users are more likely to purchase products from sellers with higher privacy scores, demonstrating that offering privacy information during the search process can affect user decisions.

Tian et al. [23] use app reviews to give users more privacy information, showing that user reviews can help users make privacy decisions. However, they focus on the consequences of app updates, rather than installing new apps or managing

current apps. Additionally, they draw from existing reviews, rather than gathering privacy-specific reviews.

There are systems that use automated approaches to detect misbehavior or privacy risks in apps (such as Chin et al. [10], Enck et al. [13], Sarma et al. [22], and Wei and Lie [26]), to flag dangerous permissions (such as Wang et al. [25] and Pandita et al. [19]), or to detect malware (like Zhou et al. [29], Zhou et al. [30], and others). All of these systems generate information that could be employed in a privacy-centric marketplace to rank apps and inform users about privacy. Yu et al. [28] and Rosen et al. [21] use API and method calls to generate privacy policies for Android apps, and to highlight privacy-relevant app behavior, respectively, but neither system connects particular behaviors with the permissions that enable them. If developers or Android were to provide this information, our PerMission Assistant could incorporate these tools to help users decide which permissions to enable or disable.

Papamartzivanos et al. [20] analyze smartphone usage patterns across users to find privacy leaks in apps. Lin et al. [16] and Yang et al. [27] use information gathered via crowdsourcing to find unexpected permissions and improve user understanding of Android permissions. These systems aggregate crowd feedback into observations about apps, rather than providing a direct channel of communication for users and developers. Burguera et al. [9] also take a crowd-based approach to app security. Unlike our work, they use the crowd to collect traces of app behavior to detect malware, rather than gathering direct feedback from users on permission use in legitimate apps.

Kelley et al. [15] explore how to present privacy information to users, building "nutrition labels" for privacy policies, and find their display format helped users better understand the policy. Egelman et al. [12] use crowdsourcing to evaluate user comprehension of privacy icons for ubiquitous computing environments. These works demonstrate how an interface can help users better understand privacy, but their icons are intended for different uses.

8. CONCLUSION

We have discussed two apps: the PerMission Store, which allows users to incorporate privacy into the process of searching for apps, and the PerMission Assistant, which helps users manage the permissions of their installed apps. These apps also provide a channel of communication between users and developers. Ultimately, we believe the apps' features should be incorporated directly into platforms, so that they are part of every users' app experience.

Acknowledgments.

We thank Jeff Huang for his advice. This work was partially supported by the US National Science Foundation.

References

[1] Google Play Store: number of apps 2009-2016 | statistic. Retrieved Feb. 2016. statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[2] Google Play Android app store scraper. Retrieved Apr. 2016. github.com/chadrem/market_bot.

[3] Android permissions | help center. Retrieved Apr. 2016. help.pinterest.com/en/articles/android-permissions.

[4] Apk permission remover - Android apps on Google Play. Retrieved Apr. 2016. play.google.com/store/apps/details?id=com.gmail.heagoo.apkpermremover.

[5] Fix permissions - Android apps on Google Play. Retrieved Apr. 2016. play.google.com/store/apps/details?id=com.stericson.permissionfix.

[6] MyPermissions privacy cleaner - Android apps on Google Play. Retrieved Apr. 2016. play.google.com/store/apps/details?id=com.mypermissions.mypermissions.

[7] PermissionDog - Android apps on Google Play. Retrieved Apr. 2016. play.google.com/store/apps/details?id=com.PermissioDog.

[8] 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 *Conference on Human Factors in Computing Systems*, 2015.

[9] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani. Crowdroid: Behavior-based malware detection system for Android. In *Security and Privacy in Smartphones and Mobile Devices*, 2011.

[10] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner. Analyzing inter-application communication in Android. In *Mobile Systems, Applications, and Services*, 2011.

[11] E. Chin, A. P. Felt, V. Sekar, and D. Wagner. Measuring user confidence in smartphone security and privacy. In *Symposium on Usable Privacy and Security*, 2012.

[12] S. Egelman, R. Kannavara, and R. Chow. Is this thing on?: Crowdsourcing privacy indicators for ubiquitous sensing platforms. In *ACM Conference on Human Factors in Computing Systems*, 2015.

[13] 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 *Operating Systems Design and Implementation*, 2010.

[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 *Security and Privacy in Smartphones and Mobile Devices*, 2012.

[15] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder. A "nutrition label" for privacy. In *Symposium on Usable Privacy and Security*, 2009.

[16] 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 *Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2012.

[17] 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 *Symposium on Usable Privacy and Security*, 2016.

[18] W. Mason and S. Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44, 2012.

[19] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In *USENIX Conference on Security*, 2013.

[20] D. Papamartzivanos, D. Damopoulos, and G. Kambourakis. A cloud-based architecture to crowdsource

mobile app privacy leaks. In *Panhellenic Conference on Informatics*, 2014.

[21] S. Rosen, Z. Qian, and Z. M. Mao. AppProfiler: A flexible method of exposing privacy-related behavior in Android applications to end users. In *Conference on Data and Application Security and Privacy*, 2013.

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

[23] Y. Tian, B. Liu, W. Dai, B. Ur, P. Tague, and L. F. Cranor. Supporting privacy-conscious app update decisions with user reviews. In *Security and Privacy in Smartphones and Mobile Devices*, 2015.

[24] J. Y. Tsai, S. Egelman, L. Cranor, and A. Acquisti. The effect of online privacy information on purchasing behavior: An experimental study. *Information Systems Research*, 22(2), 2011.

[25] Y. Wang, J. Zheng, C. Sun, and S. Mukkamala. Quantitative security risk assessment of Android permissions and applications. In *Data and Applications Security and Privacy XXVII*, 2013.

[26] Z. Wei and D. Lie. LazyTainter: Memory-efficient taint tracking in managed runtimes. In *Security and Privacy in Smartphones and Mobile Devices*, 2014.

[27] L. Yang, N. Boushehrinejadmoradi, P. Roy, V. Ganapathy, and L. Iftode. Enhancing users' comprehension of Android permissions. In *Security and Privacy in Smartphones and Mobile Devices*, 2012.

[28] L. Yu, T. Zhang, X. Luo, and L. Xue. AutoPPG: Towards automatic generation of privacy policy for Android applications. In *Security and Privacy in Smartphones and Mobile Devices*, 2015.

[29] W. Zhou, Y. Zhou, X. Jiang, and P. Ning. Detecting repackaged smartphone applications in third-party Android marketplaces. In *Conference on Data and Application Security and Privacy*, 2012.

[30] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In *NDSS*, 2012.

APPENDIX

game	**business**	**medical**
Blossom Blast Saga	Job Search	CareZone
Star Wars: Galaxy of Heroes	ADP Mobile Solutions	MyChart
Clash of Kings	UPS Mobile	FollowMyHealth Mobile
Prize Claw 2	LinkedIn Job Search	Ovia Pregnancy Tracker
Subway Surfers	Job Search - Snagajob	ScriptSave WellRx
entertainment	**health_and_fitness**	**finance**
Netflix	Strava Running and Cycling GPS	Credit Karma
Hulu	Calorie Counter - MyFitnessPal	Chase Mobile
Google Play Games	CVS/pharmacy	Bank of America
Vine - video entertainment	Google Fit - Fitness Tracking	Android Pay
YouTube Kids	Headspace - meditation	PayPal
news_and_magazines	**social**	**music_and_audio**
Yahoo - News, Sports & More	Facebook	Pandora Radio
CNN Breaking US & World News	Instagram	Spotify Music
Viewers to Volunteers	Snapchat	SoundCloud - Music & Audio
AOL: Mail, News & Video	Pinterest	YouTube Music
Fox News	Twitter	Shazam
travel_and_local	**weather**	**white noise***
Waze - GPS, Maps & Traffic	The Weather Channel	White Noise Free
Yelp	1Weather:Widget Forecast Radar	White Noise Pro 2.0
Maps	AccuWeather	White Noise Baby
United Airlines	Transparent clock & weather	Relax Melodies: Sleep & Yoga
Southwest Airlines	WeatherBug	Relax Rain - Nature sounds
brick-and-mortar*		
Stop and Shop		
HSBC		
Wegmans		
Starbucks		
Subway		
Regal Cinemas		

Table 2: Apps considered in classification study (Section 2). Categories marked by an asterisk are not built-in Google Play categories but rather sets of apps with specific qualities of interest to the study: The "white noise" apps have very similar feature sets, and therefore might be likely to be considered by users to be generic, while apps in the "brick-and-mortar" category are closely coupled with real-world products and so might be likely to be single-source. ("Brick-and-mortar" is not mutually exclusive with respect to the other categories, so there are some apps in other categories that are "brick-and-mortar," such as CVS/pharmacy in health_and_fitness and the airline apps in travel_and_local.)

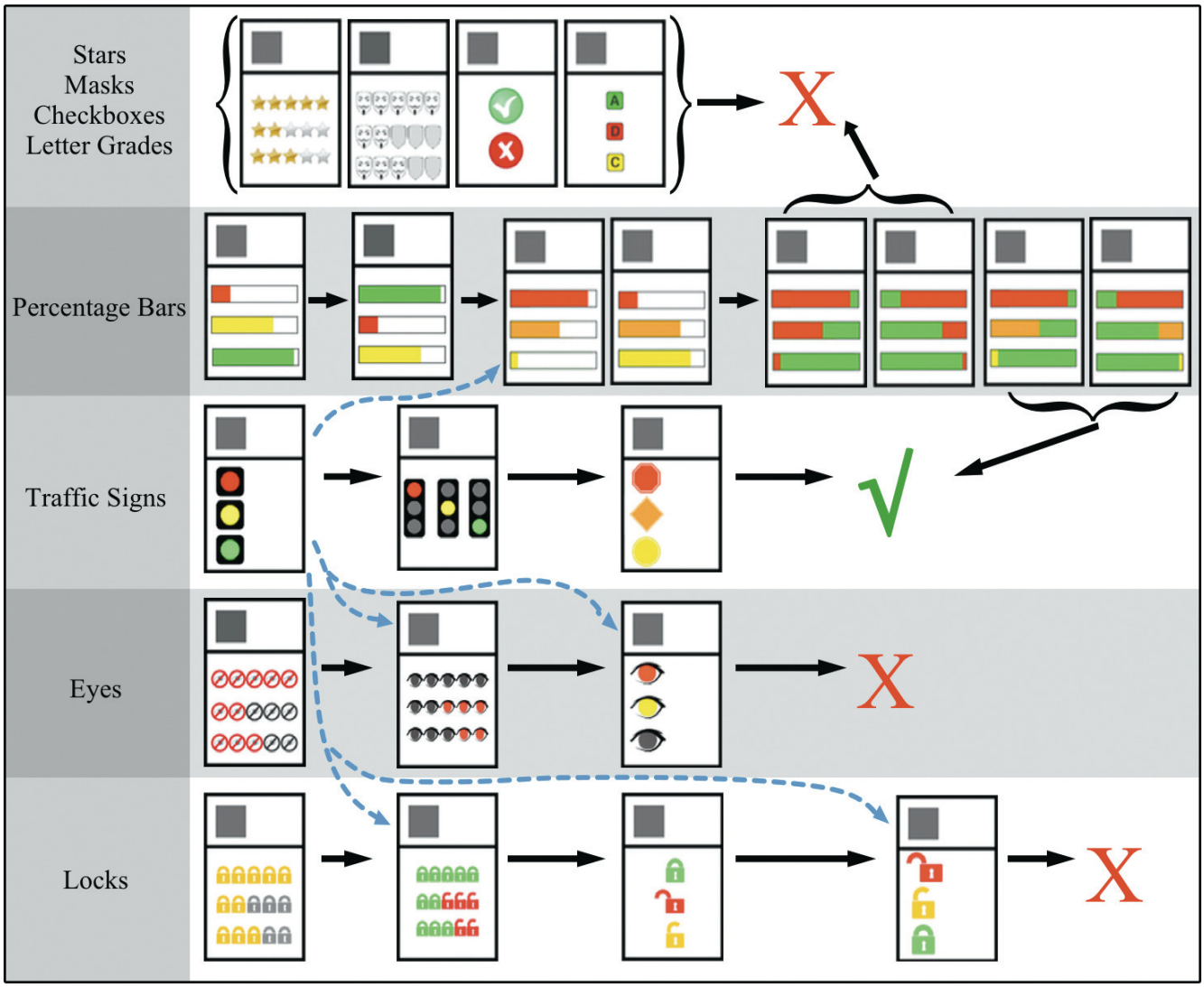


Figure 5: (Note: This figure may be better viewed in color.) An overview of all of the interfaces explored during our iterative design process (Section 6.1). Iconographies not included in the text discussion are eyes, traffic signs, and letter grades. Arrows map the evolution and cross-influences of interfaces; solid (black) arrows show redesigns, and dashed (blue) arrows indicate that feedback on one iconography influenced the design of another. X's (in red) indicate the elimination of an iconography, while the checkmark (in green) signifies the interface was included in our in-depth testing.