Short Paper: Enhancing Mobile Application Permissions with Runtime Feedback and Constraints

Jaeyeon Jung
Microsoft Research
One Microsoft Way
Redmond, WA, USA
jjung@microsoft.com

Seungyeop Han
University of Washington
Allen Center, Box 352350
Seattle, WA, USA
syhan@cs.washington.edu

David Wetherall
University of Washington
Allen Center, Box 352350
Seattle, WA, USA
djw@uw.edu

ABSTRACT

We report on a field study that uses a combination of OS measurements and qualitative interviews to highlight gaps between user expectations with respect to privacy and the result of using the existing permissions architecture to install mobile apps. Most of our participants expected advertising and analytics behavior, yet they were often surprised by applications' data collection in the background and the level of data sharing with third parties that actually occurred. Given participant feedback, we propose platform support to reduce this "expectation gap" with transparency of data usage and constrained permissions.

Categories and Subject Descriptors

C.2.0 [Computer-Communication Networks]: General – security and protection.

General Terms

Security, Human Factors

Keywords

Smartphone privacy, permission architecture, user study

1. INTRODUCTION

Privacy incidents caused by mobile applications abound. A recent report, for example, reveals that Path and Twitter applications collected the user's contacts without explicit consent. This resulted in public outcry that led the platform provider to revise their permission model [1]. Other prior work has found that many Android and iOS applications share the user's location with third parties and expose the device identifier to trackers [2-4]. This behavior is part of everyday applications rather than malware, and it occurs in spite of the existing permission architecture.

SPSM'12, October 19, 2012, Raleigh, North Carolina, USA.

We report on a user study in which we investigate how the current permission architecture falls short in helping users to make informed privacy decisions when they grant permissions to applications on mobile devices. We use a three-week field study of 20 Android smartphone users. We measure how users' personal data is exposed by the applications that they run and use semi-structured interviews to collect qualitative feedback. Our participants were drawn from moderate to heavy Android application users. Across twenty participants, our logging software detected 129 different applications that transferred at least one of the nine privacy-sensitive data types over the three-week monitoring period.

By comparing the participants' expectations with their actual data exposure, we are able to highlight limitations of the current permission system that permitted "unexpected use" by popular applications.

We find that our participants had a reasonable mental model of privacy and personal data use by the applications with which they were familiar. Some participants were even willing to share more data with applications if that would help to improve the applications' functionality. However, participants pointed out three types of unanticipated data collection when they were shown their measured data exposure: (1) applications discreetly collecting personal data in the background, especially before the first use; (2) applications collecting seemingly unnecessary data with respect to their functionality; and (3) applications collecting an excessive amount of personal data (to use as a tracking ID or to track locations). While our participants might be more technically savvy than an average smartphone user, they are arguably a leading indicator of users as mobile usage and sophistication increase, and more vulnerable to inadequate privacy support on the mobile platform.

To improve the permission architecture, our participants desired better data sharing transparency for making informed privacy decisions, such as whether to uninstall a new application that appears to collect more data than necessary. The current permission architecture shows what data type that an application can access, but not which data has actually left the phone and if so, how frequently. In this paper, we discuss how this limitation can be overcome by

extending the operating system to collect data flows and applications' runtime context.

Our participants also stressed the need to define conditions on which data can be used to exercise meaningful control over their personal data. Currently, users need to decide whether to grant a set of permissions at the time of an application's install, and once granted permissions have a *carte blanche***.** This all-or-nothing design may simplify application development, but it works poorly as an indicator for users to differentiate acceptable level of data collection from excessive, potentially privacy invasive data collection. In this paper, we discuss an idea of "bounding" permissions to limit when and how often an application can access protected data.

To the best of our knowledge, this paper describes the first study to compare people's privacy expectations against the measured data sharing behavior of mobile applications in their own device. We share the vision with [10] that understanding people's expectations of acceptable data collection is important to identify applications' behavior that may violate the users' privacy. Quite a few other studies show the lack of user understanding of the Android's permission architecture [6] [11] and how the install time permission screen can be improved for better privacy risk communication [10], but none of these studies are grounded in the measured data exposure of users' own information.

Next we present the method and details of our user study, followed by the key findings. We then discuss two new extensions to the existing permission architecture—runtime permission constraints and data use feedback. Our proposed designs are still work in progress, but we hope to get feedback from the community on our early designs. Finally we conclude with a discussion of research challenges in building the new permission architecture.

2. STUDY METHOD

We conducted a three-week field study with 20 participants between November and December 2011. While our study was informed by previous user studies [6,7,11], we chose a field study instead of surveys or in-person interviews alone for two reasons. First, we used the field study to measure the types and amount of personal information that was accessed and collected in the day-to-day usage of mobile applications. Second, we used the measured data to solicit users' feedback by contrasting user expectations with what actually happened.

Our user study, therefore, comprises two parts. First, participants were asked to use an instrumented Android phone (Nexus S) that we provided for three weeks. To smooth the transition, we assisted with the initial setup and transferred the applications from the participant's own phone to the instrumented phone. After three weeks, participants were brought back to lab for the exit interview. Upon completion of the study, they were rewarded with a $100 Amazon gift card.

We recruited participants using online and offline flyers. Twenty Android smartphone users participated in our study (9 females, 11 males, and aged 18-41). Although 13 participants were drawn from the school (only 2 from computer science), the rest represented a mix of professions including web designer, artist, cook, and home maker. We screened interested people to select moderate to heavy mobile application users. All participants had a personal Android smartphone that they did not share with others and had at least 200MB data plans with T-Mobile.

The details of the data collection system that we built for the study are available in [12]. In contrast, this paper focuses on qualitative data collected at the exit interviews. We began interviews by asking participants to fill out tables summarizing their expectation as to how often an application accessed a certain data type over the study period. Participants were given a list of several applications that we knew had collected data and were asked to answer only for those applications. We then revealed the observed data and walked through the difference between participants' expectation and what we monitored. We then showed a visual representation of detailed data (an example is given in the next section) to stimulate the discussion of follow-on questions about privacy choices and general perception of personal data privacy.

Each interview lasted about one hour and was semi-structured as we followed up with questions to cover privacy concerns depending on data types, applications, and data sharing with third parties. We audio recorded and transcribed each interview and then analyzed the data using the affinity diagramming technique.

A limitation of our study is that it is based on a relatively small number of participants drawn from the Pacific Northwest area in the USA. As our study involves recruiting participants, handing out an instrumented phone, and conducting interviews, it does not scale well. 10-20 person studies are typical for field studies and considered valuable.

3. USER STUDY RESULTS

We started with twenty one participants (call them P1, P2,…, P21) but P4 dropped out a day after because of difficulties transferring the contacts to the study phone. The remaining twenty participants used the provided study phone as their primary phone for at least three weeks and completed the exit interview. Participants reported a few minor problems with the phone, such as not supporting screen rotation or not being able to run applications provided by the mobile operator. No one reported any major use change during the study.

Overall, our tool logged 223 applications run by participants during the study (avg: 26, min: 12, max: 47)[1]. The top three commonly used applications other than pre-installed Google applications are Facebook (used by 18 participants), One Bus Away (16), and Tmobile (9). Our tool also logged 3.52 GB of data received and 0.65 GB sent by participants during the study (avg: 213 MB, min: 81 MB, max: 413MB).

We found 129 applications transferred at least one of the 11 monitored data types to remote sites. We enhanced the TaintDroid system [2] to track contacts, ICCID (SIM card ID), phone number, SMS, camera, microphone, location, calendar, bookmark, Android ID, and IMEI (device ID). Table 1 shows 9 data types that were transferred to remote sites by at least one application. We manually verified all of the 257 (data type, application) pairs and excluded 30 false positives.

The collected data allowed us to compare participants' expectations[2] and what actually happened to their own data. Most participants had a reasonable understanding of which applications might have collected what types of personal data based on their own application experience. Nonetheless, 14 of 20 participants indicated surprise at discrepancies after a researcher helped them walk through the observed data.

3.1. Unexpected personal data collection

When asked if any of the measured data was unexpected, 5 participants immediately pointed out rarely used applications with collected data: "*GroupMe is a group text-messaging service and I downloaded it to use for a group and never used it. Ever, not once*" said P1. After reconfirming that it was the GroupMe application that collected IMEI 1732 times and location 2 times, P1 responded saying "*That's a lot. For a program that I installed and never used*". P20 showed a similar reaction when finding the Google Books application collected Android ID: "*Like just it being tracked, 'cuz I've never used that before. I might have accidentally clicked on it but I've-, I've never actually used that program. So that's why when I saw your list, I was like, 'What? Books is on here?*'"

This issue results from the Android permission architecture that grants permissions to an application at install time. Since the permission screen is known to be ineffective [6], we find this early permission binding problematic. P21 articulated the issue when asked whether s/he paid attention to the permission screen when installing applications; "*It just feels like it's only for that moment. Like, 'cuz I guess I don't actually think about, 'Oh, it's gonna be accessing the information every single time. Or in the background without me even loading?*'".

Another source of surprise is the applications that collect information that is seemingly unnecessary for their operation. As Table 1 shows, 22 out of 57 applications collected location to share with third parties. Some participants seemed to accept this practice as reasonable for free apps (P2: "*I am assuming* [New York Times and Dictionary applications] *share location with third parties since it is a free app they need to make money out of it in some way. This sounds reasonable*."). However, 13 participants voiced their concern regarding applications that collected data just to share with third parties (e.g., P12: "*Huffington Post is a little surprising because they don't have ads or well, they have ads but it's like a news app. So I didn't expect it to like share my locations*." P8: "*If it was an app like a game or something like that that didn't really need that information and had no kind of use for it other than for advertising purposes then I wouldn't want to share it*.")

Table 1: Personal data collected and transferred by apps: Numbers in parentheses indicates the # of apps that shared the data with known third parties. Due to space constraints, we only list at most the top three application names (sorted by the frequency of data transfers) per each category.

	apps	people	application names
microphone	2 (0)	2	SoundHound, Voice Search
ICCID	2 (0)	5	Facebook Messenger, Antivirus Free
SMS	4 (0)	15	Messages, WhatsApp Messenger, Go SMS
contacts	8 (0)	11	WhatsApp Messenger, Twitter, Facebook
camera	10 (0)	9	Google +, Dropbox, Sugar Sync
phone number	11 (1)	16	Facebook Messenger, Kakao Talk, OneBusAway
IMEI	49 (19)	20	WhatsApp Messenger, Words with Friends for Free, GroupMe
location	57 (22)	20	Weather Gadgets, Google Maps, Facebook
Android ID	78 (51)	20	GameCIH, Dolphin Browser, Yelp

Third, frequent access of personal data by applications was also noted as unexpected by 4 participants. For the applications that use the device ID or Android ID for

[1] We only counted applications that run as a foreground process at least once and therefore excluded applications that were installed on the phone but never used.

[2] At the start of the interview, we asked participants to fill out a table similar to Table 1 using their best guess.

tracking, the access frequency was quite high (P8: "*Hanging Free collecting all of our information was surprising. My IMEI, it looks like* [collected] *72 times. And, Angry Birds, too, collecting the IMEI 59 times*"). Also, as Android applications can run in the background, some applications appeared collecting location on a regular basis unbeknown to participants (P1: "*Well, I guess I expected the Weather Channel,* [but] *not quite 7,000 times. Didn't expect this*").

Overall, our interviews revealed three common cases in which an application's collection of personal data is perceived by the user as unexpected or undesired because it is not associated with the user's direct use of the application. We revisit these cases in Section 4 and discuss how our proposed permission architecture can alleviate them.

3.2. Desire for information flow transparency

To help participants better understand how their own personal data was used by applications, we built an interactive visualization tool. It presents three views of the data collected by a participant's instrumented study phone. Figure 1 shows a modified screenshot of the visualization tool that obscures the participant's location trails[3].

The timeline view plots a colored block if there had been any transmission of the particular data type during one hour period. As IMEI, Android ID, location, contacts had been frequently sent off by applications, we used a different color (IMEI in red, Android ID in orange, location in yellow, and contacts in green) to separate these data types. The rest of the data types (e.g., phone number, photos) are represented together in purple. This particular time view shows that location was frequently transmitted (almost every hour) by the AccuWeather application and contacts were also regularly transmitted by the Twitter application.

After visualizing their own data with our tool, all but P14 answered that they would like to see the history of when and by which applications their personal data is accessed if such information is readily available and easily accessible. P14 said that s/he would be too busy to check on this kind of data.

When probed further, 7 participants stressed the lack of information flow transparency in the current permission architecture (P7: "*it'd be nice if instead of the permissions that you allow if they said that this is where their information goes or like this is the third party*") and expressed their desire "*to be aware of where your data is going and what they are using it for*" (P2).

[3] During an exit interview, a researcher drove the tool to display a participant's data but the participant was free to ask to zoom in, filter out particular data, etc.

Besides improving their general awareness, 5 participants pointed out the information flow log as being potentially useful for auditing a new application (P13: "*if I ever downloaded an application, I'd be interested to see it just-, what it took exactly*.") and for tracking down companies that are responsible for users' data (P2: "*If you think that your data is in of danger or whatever, you should be able to control it in some way and knowing who has your information is part of that.*"). A few participants actually took note of third party companies that collected their location during the interview (P12: "*Like, I really wanna go home and Google that medialytics, right?*").

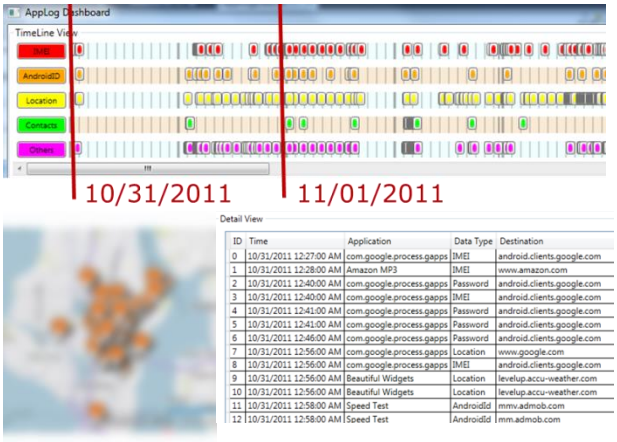

Figure 1: Data presented to participants. The top panel shows the time line view, illustrating how frequently each data type was transmitted off by applications. The bottom right panel shows the detail view that lists what application sent which data to which servers. The bottom left panel shows the location view that displays locations that were exposed. The location view (bottom left) is purposely obscured to respect the participant's privacy.

Based on this feedback, we argue that transparency needs to be integrated into the permission architecture. We discuss new system components needed to collect sufficient logs from mobile devices in Section 4.

3.2. Desire for bounded permissions

After finding that applications collected data in an undesirable way, 6 participants said that they would uninstall the offending applications (P6: "*Well I am definitely uninstalling HomeSmack*,") and 4 participants said that they would look for better alternatives (P13) or access news through the web browser "*instead of downloading the actual app*" (P12). However, some applications were mentioned as too useful to give up (P11: "*I still need to use Skype and SugarSync*") even though participants did not want those applications to share data with third parties.

People often download an application because they "*just wanna try it out*" (P21) and do not become regular users. However, in the current permission architecture, an

application is granted fully with the requested permissions once installed and there are no limits as to under which conditions the application can exercise the granted capabilities. Ideally, privacy-conscious permission architecture should regulate the application's use of the user's data tightly bounded with the user's use of the application as one participant puts it:

> *"So it really depends on how much use I get out of it, so it is like a negotiation thing. So I use uTorrent a lot and they share a significant amount of information to third parties and it seems fair. But I don't use HomeSmack, and they share information, so it is not fair anymore so I would uninstall that."* [P6]

To satisfy the diversity of users' privacy concerns, participants highlighted the need for new types of constrained permissions that provided limited kinds of access to personal data. P6 used Twitter as an example for limiting the location access to only the application's current session saying "*every time I booted up Twitter to like create a tweet, I would want to be asked whether they can use my location in the tweet*". In particular, 8 participants called out their contacts as the most sensitive and therefore would want to review each permission request. (P5: "*For address book, I want it to be asking my permission anytime*").

4. RUNTIME EXTENSIONS TO EXISTING PERMISSION ARCHITECTURE

A number of recent studies proposed various ideas of improving permission architecture on the phone, ranging from enabling users to change access permissions at runtime (e.g., sensor access widgets [9] and user-driven access control [5]) to improving the permission screen [10]. In this section, we briefly discuss two new components that can alleviate privacy concerns raised by our participants.

4.1. Systems support for runtime data use feedback

Existing mobile platforms provide only limited visibility into which data was collected by applications at runtime (e.g., iOS shows applications that have accessed location in the past 24 hours). Both in Android and iOS, a small icon appears in the status bar when current location is accessed by applications. However, types of information that helped participants identify applications' unexpected privacy behavior are beyond just the use of location data.

As shown in Section 3.1, three data types that provide meaningful differentiation between acceptable and unacceptable data collection are (1) whether the data use by the application is reasonable when weighed against the user's actual use of the application; (2) how frequently the data is sent off the phone; and (3) whether the data is shared with third parties. This is not the information that can be statically determined at install time and thus require runtime data monitoring.

First, we need to measure the application context under which the user's data is collected by applications. The application context includes whether the application is actively used by the user, visible to the user, or running in the background. With this contextual information, we can provide meaningful feedback as to when an application's data sharing behavior may violate the user's expectation (e.g., the app continuously collects the data even when the user no longer uses the application).

Second, monitoring applications' data access only is not sufficient as applications can obtain the data once (e.g., device ID) then transfer it many times potentially to many different sites. Dynamic information flow tracking [2] can be useful for properly accounting this kind of accessing-once-using-many-times case.

Third, monitoring which sites that the data is transmitted off the phone and determining whether the given site is a third party is important to generating meaningful feedback to the user. Although a known list of adverting companies can be used for classifying remote sites, as the list may change over time, we need robust mechanisms for updating the list.

However, these components could add overhead to already resource-constrained mobile devices so minimizing any adverse performance impact is important.

4.2. Permissions bounded with runtime constraints

Existing mobile platforms have been loath to interrupt application workflow to ask users to grant permissions to applications. One exception is iOS, which prompts on requests for access to the user's location and, in the recent version, on requests for the user's contacts as well. However, beyond these two data types, users have no options to limit when and how often, data will be accessed once the application is installed. Inspired by the feedback from our study participants, we discuss runtime constraints that need to be imposed on currently unlimited permissions for reducing privacy risks.

Existing proposals such as sensor access widgets [9] and user-driven access control [5] can address some of the issue by inserting explicit user-initiated access control into the operating system. However, applications may have legitimate uses for data at times when users are not available to grant consent. Examples include pre-fetching location-specific data (e.g. weather), data backup, and fitness tracking.

To prevent apps from exploiting unbounded data access permissions, we propose that mobile platforms should support bounded permissions restricted with measurable runtime constraints. For instance, for data types that are static such as phone number and device ID, the repeated collection of these data makes little sense for legitimate applications: A proper setting (e.g., at most once to a first party site when the app is in use) can prevent data

misappropriation such as using these IDs for user tracking. By contrast, for data types that change frequently such as location and microphone input, multiple yet limited number of samplings (e.g., at most one location sample per hour to pull weather update) may suffice to provide desired functionality.

Looking into the actual data use of the 129 applications shown in Table 1, we identified runtime constraints that seem appropriate for these applications. We leave a broader application study as a future work. Table 2 shows the initial list of constraints and one example application whose behavior fits to the proposed model. One open question is to determine an acceptable frequency of data access. One possible approach is to give developers an option to supply information such as "one location per hour" and "two data backups per day" in the application manifest. However, this approach relies on developers voluntarily requesting for the minimum privilege, which has shown to be challenging [8]. Another approach is to analyze existing applications and empirically find out "norms" of data access frequency given data type and data use purpose.

Table 2: Example runtime constraints to bound permissions for some of the 129 applications used by our participants

data type	runtime constraint	example
contacts, calendar	once per foreground run of the application	contacts sync (WhatsApp)
	N per day	periodic data backup (Dropbox)
phone #, device IDs	once per the app's lifetime	phone number registration (Tmobile)
microphone	once per search query	voice search app (Voice Search)
location	once per search query	location-based search (Yelp)
	once per post	location tagging (Twitter)
	N per hour	weather update (AccuWeather)

Operating systems will need to be extended to support bounded permissions. First, we must augment runtime reference monitors, which currently need only guard when a resource is used, to monitor the amount of that resource that is consumed. This requires maintaining state within storage trusted by the reference monitor.

5. CONCLUDING REMARKS

We ran a twenty person, three week field study of Android mobile phone users in which we interviewed users and measured everyday application behavior. We found that users were not well served by the existing permission architecture. While most participants expected tracking as part of applications, they were surprised by the data collection frequency and the application context under which the actual data that was collected. To address privacy concerns raised by participants, we propose two runtime extensions of permission architecture that provide data flow feedback and bound data access with measurable runtime constraints. We discuss technical challenges to developing these runtime components into the existing mobile platform.

6. ACKNOWLEDGMENTS

We thank our study participants, Bongshin Lee at Microsoft Research for providing guidance to design the data visualization tool and many colleagues at Microsoft and UW who provided feedback on an earlier draft.

7. REFERENCES

[1] http://www.wired.com/gadgetlab/2012/02/apple-responds-to-path/

[2] 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 OSDI, 2010

[3] M. Egele, C. Kruegel, E. Kirda, and G. Vigna. PiOS: Detecting privacy leaks in iOS applications. In NDSS, 2011

[4] http:// blogs.wsj.com/wtk-mobile

[5] F. Roesner, T. Kohno, A. Moshchuk, B. Parno, H. J. Wang, and C. Cowen. User-driven access control: Rethinking permission granting in modern operating systems. In IEEE Symposium on S&P, 2012

[6] P. Kelly, S. Consolvo, L. Cranor, J. Jung, N. Sadeh, D. Wetherall. An conundrum of permissions: Installing applications on an Android smartphone. In USEC, 2012

[7] S. Egelman, A. P. Felt, D. Wagner. Choice architecture and smartphone privacy: There's a price for that. In WEIS, 2012

[8] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner. Android permissions demystified. In CCS, 2011

[9] J. Howell and S. Schechter. What You See is What They Get: Protecting users from unwanted use of microphones, cameras, and other sensors. In W2SP, 2010

[10] J. Lin, S. Amini, J. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In Ubicomp, 2012

[11] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. Android permissions: User attention, comprehension, and behavior. In SOUPS, 2012

[12] S. Han, J. Jung, D. Wetherall. A study of third-party tracking by mobile apps in the wild. UW-CSE-12-03-01, 2011