Permissions Snapshots: Assessing Users' Adaptation to the Android Runtime Permission Model

Panagiotis Andriotis
Cybersecurity Research Unit
University of the West of England
Email: panagiotis.andriotis@uwe.ac.uk

Martina Angela Sasse
Information Security Group
University College London
Email: a.sasse@cs.ucl.ac.uk

Gianluca Stringhini
Information Security Group
University College London
Email: g.stringhini@ucl.ac.uk

***Abstract*—The Android operating system changed its security- and privacy-related permission model recently, offering its users the ability to control resources that applications are allowed to access on their devices. This major change to the traditional coarse-grained permission system was anticipated for a long time by privacy-aware users. This paper presents the first study that analyzes Android users' adaptation to the fine-grained runtime permission model, regarding their security and privacy controls. We gathered anonymous data from 50 participants who downloaded our application and answered questions related to the new permission model. The results indicate that the majority of users prefer the new model. We also collected data that demonstrate users' security controls at the given time. Our analysis shows that individuals make consistent choices regarding the resources they allow to various applications to access.**

I. Introduction

The release of the sixth Android version, during summer 2015, introduced a major change at the permission system. Prior versions presented to the user a message at installation time, listing the resources that applications were utilizing. The users after reviewing the requested permissions (presented as groups, e.g. Contacts) had the choice to accept or deny the installation. This binary model (accept-reject) has been criticized at the past as being ineffective to provide meaningful information about the way the application to be installed will affect user's privacy [1], [2]. It also limits the ability of users to control applications' access to their private data.

The new permission model is based on the principle of least privilege and assumes that applications will be able to function at a basic level, even if the users do not provide access to resources that might affect their privacy. According to the official documentation for Android Developers[1], there are two basic categories of permissions; *normal* and *dangerous*. The system automatically grants access to resources that applications requested via normal permissions, because these actions are considered to be of low risk. For example, as the documentation notes, setting the time zone requires a normal permission. The risks arising from this action for users' privacy (or for the operation of other applications) are considered minimal. On the other hand, when an application needs to access data related to users' private information, or data stored on the device, the associated permission with this action is considered as dangerous. For example, if an application requests access to the user's calendar, it introduces high risks to user's privacy. Hence, applications designed to function properly under the new permission model, need to request user's permission during runtime, in order to access sensitive information. It lies with the users' discretion if they will accept or deny access to sensitive resources. Additionally, under the runtime permission model, Android users are able to revoke access to resources via the Settings application.

In this paper we present the results of a study about users' adaptation to the Android runtime permission model. We developed and distributed an application at the official Android marketplace (Google Play) to collect anonymous data related to the permissions that were granted (or denied) by users at that time ('Permissions snapshots'). The aim of the study is to examine users' perceptions of the provided security and privacy, and at the same time, to investigate how Marshmallow users adapted to the new permission model. To our knowledge, this is the first work that studies and presents trends associated to the security and privacy preferences of Android Marshmallow users. Our contributions are as follows: a) We collected data from 50 participants[2], which demonstrate that they adopted positively the runtime permission model. b) We present a comparative view of users' privacy settings and other trends associated to the use of popular social media. c) We highlight that our participants presented a consistent behavior related to the resources they allow to be accessed by various applications.

II. Related Work

Previous work showcased privacy- and security-related concerns associated with Android's install-time permission system, such as the proliferation of over-privileged applications [3]. Studies on user-defined security controls on mobile devices showed that users (on average) have to make over a hundred permission decisions per device [4]. In addition, researchers demonstrated than users are unaware of the number (or the context) of permissions they granted to applications at the past [5].

A recent paper by Wijesekera et al. [6] showed that during a study, at least 80% of the participants indicated that they would

[1] http://bit.ly/2d4AdGH

[2] The anonymized dataset is available online at the following address: http://dx.doi.org/10.14324/000.ds.1520825/

prevent at least one permission request of an experimental application if they were aware of its purpose and functionality. The participants also stated that they would block over a third of permission requests if they had this choice. However, other studies highlighted that most of the users do not pay attention to system messages related to permission requests [7]. Additionally, researchers demonstrated that users were often surprised by the ability of applications to collect personal data in the background and share data with third parties [8]. Advertising libraries for example have been consistently examined for data exposure and leakages; a recent study revealed a trend in advertising networks to become more aggressive in collecting reachable user data [9].

In order to protect mobile devices from unlimited data sharing, several systems have been proposed. Some approaches focus on location services [10], offering its users the ability to adjust location accuracy when using location-based applications. At the same time, on-device or service-based obfuscation techniques have been advised as a methodology to maintain users' privacy [11]. Such schemes according to [12], that utilize abstract location descriptions, may cause location sharing to increase. Beresford et al. [13] presented their system, named MockDroid, which essentially feeds 'empty' resources to apps that require access to sensitive data. Another data protection mechanism that relies on obfuscation is AppFence [14]; this system substitutes shadow data in place of data that the user wants to keep private. In addition, there exist methods based on crowdsourcing, which aim to utilize contributors' protection decisions in order to provide application specific privacy recommendations for iOS devices [15]. Note that fine-grained permissions were introduced earlier, when the iOS 6 was launched.

Our study assesses security settings found on devices running the Marshmallow version. In this work we assume that participants do not share their devices with other people and we intent to highlight behavioral patterns related to their perceptions of security.

III. Methodology

We developed an application which was introduced as a survey instrument (*'Permissions Snapshot'*). It was published at the Google Play store, following the example of other recent research works [4], [16]. 'Permissions Snapshot' makes use of the `PackageManager` class (Android SDK) to get the permissions that were granted from the system to each installed application on the device. Participants had to download the application on their devices to complete the survey. In order to proceed with this study (and before publishing the application to Google Play) we obtained approval from the UCL Research Ethics Committee (Project ID Number: 8945/001).

A. Survey Design

The application targets Android Marshmallow users (SDK 23+) and cannot be installed on devices that run an older version of the operating system. Thus, collected data came from participants who were familiar with the sixth Android version (Marshmallow). The application does not collect any personal information apart from the package names of the installed applications and the requested permissions. In order to avoid duplicate entries from participants, the application requests from the system to get the `ANDROID_ID` (which is a number granted by the system, uniquely distinguishing devices). Our application returns a hashed value of this number, which can be used from us to maintain users' anonymity and, at the same time, identify possible duplicate entries.

When the users launch the application they have to read the 'Information Sheet' which describes the aim of the study and the steps that will follow. The participants must click a check box to indicate they give their consent and start the survey. The users are always informed about the current and the next step of the process. When the user's consent is given, the application receives information about the permissions the system granted (or not) for each installed application at the specific (current) time. Our application utilizes the `PackageManager` class receiving information via its `getPackageInfo` public method. This procedure takes usually less than a second on a contemporary device. Then the survey starts, gathering basic demographic data. The next steps require the participants to answer six multiple choice questions (by clicking on radio buttons). Each question is presented on a single `activity` and the user has to click the 'Next' button to view the next question. All questions had predefined answers in order to make analysis easier for us and distinguish participants who were just skipping questions by clicking the 'Next' button. When the six questions are answered, the user sends the responses (along with the demographic and the permissions data) to our server by clicking a button. Finally, the application shows to the participant a short tutorial that discusses the changes at the permission model.

These six questions were aiming to show if our participants had noticed the changes at the Android permission model and if these changes caused security fatigue [17] to them. The questions are summarized as follows: 1) How long have you been using the Android Marshmallow operating system? 2) Have you noticed any changes at the current app permission model? 3) Do you think that now you have more control on the personal data you share? 4) "I can invoke or grant any app permissions anytime from the Settings app of my device." Correct, Wrong, I Don't Know. 5) "I believe that the new runtime permission model is irritating because it asks me too many questions during running time." Agree, Disagree, I Don't Know. 6) Do you prefer the new permission model (runtime permission requests) or the old model?

B. Permission Data Acquisition

Settings for granted permissions for each application on Android devices can be found using the Settings application: Settings → Apps → (application) → Permissions. Users are able to grant, deny or revoke permissions to specific groups. Using the flag `GET_PERMISSIONS` when calling the `getPackageInfo` method of the `PackageManager`, we can get information about the status of each permission for every application (requestedPermissions and requestedPermissionsFlags). Our application does not gather information about system's applications. After experimenting with these settings we concluded that for each requested permission by the application, the system returns public arrays of Strings and ints denoting the current permission settings on the device. In our log files we denoted a granted permission with 3 and a denied permission by 1.

According to the Android Developers documentation[3], the first time an application requests to access sensitive resources (e.g. the SMS list, using for example the permission READ_SMS), it displays a dialogue message to the users to get explicitly their permission to do so. However, if users allowed at the past access to a single permission (e.g. READ_SMS), then anytime the application requests another permission from the same group (e.g. SEND_SMS), the system will immediately grant this permission. Thus, during analysis we took this fact into account to reconstruct the acquired permission settings for each application (and each permission group). For example, if for the package org.telegram.messenger (the 'Telegram' application) the PackageManager returned "android.permission.READ_CONTACTS, 1" (i.e. denied) and "android.permission.WRITE_CONTACTS, 3" (i.e. granted), this means that the application was granted permission to access the 'Contacts' group.

IV. Results and Discussion

We publicized the study at university mailing lists, social media and forums for a period of one month (June - July 2016) and we got responses from 50 participants. No prizes or other raffles were offered at this campaign. We should note that during this period, the Android Developers Dashboards[4] reported that only 10% (approximately) of the Android users that visited the Google Play Store were running the Marshmallow version on their devices.

Among the 50 respondents, 3 Android Marshmallow users provided the predefined answers of the questionnaire, hence their demographic data and their answers were excluded from the presented results. However, their device data were included in the analysis, because they could not somehow be manipulated or falsified from these users; these are device-dependent data. Additionally, although we got permission data from one participant, the file that was sent to us from the specific device did not contain demographic data. Finally, data sent by one participant (No 24) contained only the permission data of our application and the responses to the six questions; these responses were also taken into account during analysis.

A. Demographics and Questionnaire

46 responses were included at the demographic data and questionnaire analysis. 85% of the participants were males and 15% were females. Half of the users were between 18 to 30 years of age (52%), 37% were 31 to 46 years old, 9% were between 47 and 65 years old and 2% were above 66 years old. The majority of the respondents were residents of Europe (74%); the rest were from America-Canada (17%) and 9% of the responses were sent from Asia.

According to the responses at the six questions, 52% of the Marshmallow users in our sample had been using the Marshmallow version for a period of 0 - 6 months. 30% of the participants had been using the operating system for 7 - 12 months and 11% claimed they had been using it for more than a year. The latter might indicate that some of the respondents were software developers who experimented with the Developer Previews; these were released on summer 2015. This is not unreasonable if we take into consideration that we advertized our application to Google+ Communities related to Android. The rest participants (7%) clicked the "I Don't Know" option. Considering the second question, 89% of the participants replied that they had noticed the change at the permission model. 7% said they did not notice any difference and 4% chose the "I Don't Know" answer. This finding shows that our participants were familiar with the change at the permission model.

The next questions were aiming to assess if users were satisfied with the security and usability of the new permission model. At the third question, 65% of the respondents replied positively; they think that now they have more control as users on the personal data they share. However, 17% replied negatively and 17% said "I Don't Know". The responses at question 4 ("I can invoke or grant any app permissions anytime from the Settings app of my device.") showcased that the participants of our study knew how to use the Settings application. 78% chose "I Agree", 7% clicked "I Disagree" and 15% said "I Don't Know". Considering the usability of the model and the security fatigue that might cause, the responses at the fifth question showed that Marshmallow users were not overwhelmed by the security dialogue messages the system produces. The question was presented as "I believe that the new runtime permission model is irritating because it asks me too many questions during running time"; 89% of the participants replied "I Disagree" and only 11% clicked on the "I Agree" choice.

Finally, participants were asked if they prefer the runtime permission model or the old one. 78% of them said they prefer the runtime permission model and 9% said they prefer the previous model. Also, 13% clicked the "I don't have any specific preference" option. The responses at the last question show that users adapted to the changes the permission model introduced in a positive fashion.

To conclude, the majority of Android Marshmallow users in our sample suggested that they think they are able to efficiently control the data they share using the runtime permissions model, they are not frustrated by the dialogue messages the system issues and they prefer the runtime permissions model against the previous.

B. Permission Data Analysis

We analyzed permission data received from 49 devices. As discussed previously in Section IV, one device returned

[3] http://bit.ly/2d4AdGH

[4] http://bit.ly/1kjKifB

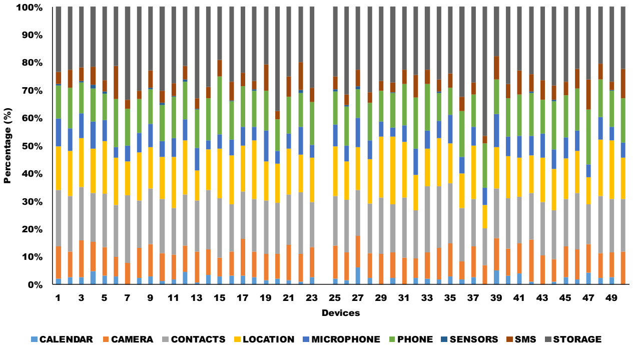

Fig. 1. Requests to Access Dangerous Permission Groups per Device.

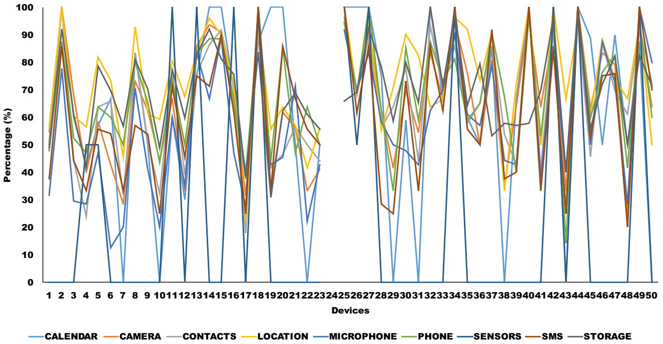

Fig. 2. Accessibility (%) of Dangerous Permission Groups per Device.

TABLE I
AVERAGE USE OF DANGEROUS PERMISSION GROUPS

Permission Groups	Average Use (%)	Standard Deviation
CALENDAR	7.34	5.16
CAMERA	30.46	9.36
CONTACTS	58.02	12.78
LOCATION	49.67	12.20
MICROPHONE	21.63	9.37
PHONE	40.56	10.56
SENSORS	0.93	1.19
SMS	16.51	9.23
STORAGE	76.64	7.18

only permission data derived from our application, hence it was excluded from analysis. The average number of installed applications per device was approximately 71. The maximum number of applications found on a single device was 283 and the minimum was 19. The number of unique applications that were seen in the collected dataset was 1,983 applications.

Developers must declare permissions that will be requested to the system by their applications in the `AndroidManifest.xml` file. This information is made available to the `PackageManager` during runtime. Using the methodology described in Section III-B we gathered the declared permissions for each application. The average number of permissions declared in the `AndroidManifest.xml` file of each application was approximately 12.39 (standard deviation = 10.94). The average number of dangerous permissions per application was approximately 3.85 (standard deviation = 3.21). This means that (statistically) 31% of the declared permissions in the `AndroidManifest.xml` file of each application belonged to dangerous permissions groups. The application `com.baidu.appsearch` contained 118 permissions in its manifest file and 19 of them belonged to dangerous groups. The application 'Signal Private Messenger' (`org.thoughtcrime.securesms`) declared 20 permissions that belonged to dangerous permissions groups. These applications presented the highest number of declared and dangerous permissions in our dataset, respectively.

C. Permission Groups

As mentioned at Section III-B we grouped the declared permissions to simulate what users see when they tap on the Settings application to view the granted permissions per application. According to the Android Developers documentation, there are 9 groups of dangerous permissions: Calendar, Camera, Contacts, Location, Microphone, Phone, Sensors, SMS, Storage. For each device we estimated the average use of dangerous permissions. To do that we counted how many permissions from dangerous groups were included in the manifest files of each application per device. We did not take into account in this calculation if access was granted to the groups at the specific time. The results can be seen at Fig. 1; the majority of installed applications requested access to the Camera, Contacts, Location, Phone and Storage groups. TABLE I shows in average the use of dangerous permissions, derived from their appearance in each device.

Fig. 2 shows the accessibility of permission groups. In other words, it illustrates which permission groups were accessible from the installed applications per device. The figure demonstrates that users tend to preserve a stable behavior when they set permissions for different applications. However, this result seems to be skewed by the fact that there existed numerous applications per device that were not (probably) updated to serve the new permission model. Thus, applications like 'Snapchat' (during the period our experiments took place) appeared to have access to all dangerous resources, because they were not designed (at least at that period) to adhere to the new model. Such applications can be easily identified if the user tries to revoke access to resources via the Settings app; the user will see the message: "This app was designed for an older version of Android. Denying permission may cause it to no longer function as intended". To overcome skewed results caused by the existence of such applications we present a more targeted analysis at the rest of this paper.

D. Fine-grained Permissions on Social Media Applications

In this section we present users' behavioral trends considering the fine-grained settings on popular social media applications. We focused our study on 8 popular applications and instant messengers: (Facebook) Messenger (`orca`), Facebook (`katana`), Whatsapp, Twitter, Skype, Instagram, Linkedin and Telegram. Telegram was added in the analysis because it is known for its 'end-to-end' encryption capabilities. TABLE II shows the popularity of these applications in our sample. Note

TABLE II
INSTALLATIONS OF SOCIAL MEDIA APPLICATIONS IN OUR SAMPLE

Applications - Packages	Installations	(approx.) %
com.facebook.orca	37	76
com.whatsapp	32	65
com.twitter.android	31	63
com.facebook.katana	22	45
com.skype.raider	20	41
com.instagram.android	14	29
com.linkedin.android	13	27
org.telegram.messenger	7	14

TABLE III
USERS THAT SELECTIVELY GRANTED PERMISSIONS TO APPLICATIONS

Applications	All On (%)	All Off (%)	Fine-grained (%)
Messenger	24	22	54
Whatsapp	44	0	56
Twitter	23	19	58
Facebook	23	9	68
Skype	35	15	50
Instagram	7	29	64
LinkedIn	38	15	46
Telegram	0	14	86

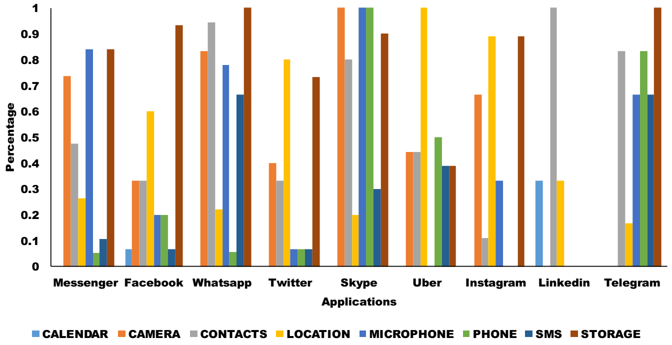

Fig. 3. Percentage of Granted Permissions per Application.

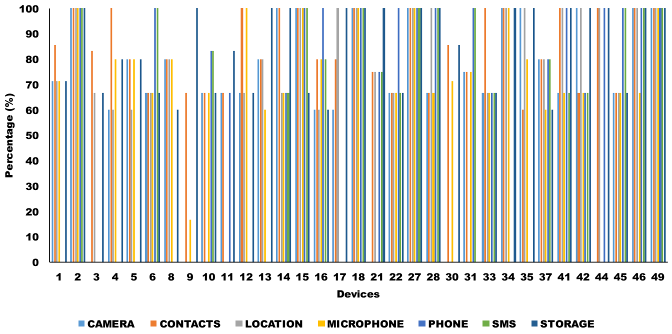

Fig. 4. Common Settings Among Popular Applications per Device.

that Messenger appeared to be more popular in our sample than the Facebook application itself; this indicates that there exists a considerable number of users that prefer to use only the Messenger application.

TABLE III shows the percentage of users that allowed (All On) or denied (All Off) access to all permission groups or used fine-grained settings for each app, respectively. According to this table, on average, approximately 60% of our respondents appeared to selectively grant access to resources when prompted by social media applications. The high accessibility levels seen for Telegram might arise from the fact that it offers end to end encryption, hence users would eventually trust the application more compared to others.

Fig. 3 presents in a concise graph the percentages of users that granted permission to dangerous groups per application. In this graph we only consider participants who selectively granted (or denied) permissions per application. Thus, we only consider participants from the "Fine-grained" category of TABLE III. We can derive that users allow or deny access to permission groups considering the functionality of each application. For example, it seems that users happily allow the LinkedIn application to access their Contact List (100%). However, applications like Messenger and Facebook demonstrate lower rates of accessibility to this permission group (47% and 33%, respectively). Also, users of messaging applications like Messenger, Whatsapp, Skype and Telegram seem willing to grant access to the Microphone permission group. At the same time, Facebook, Twitter, Uber and Instagram present high numbers of accessibility to Location services. We can also extract some additional common characteristics in people's behavior. For instance, if we focus on the Storage group we can see that most of the users allow these applications to access the storage of their devices. Another observation can be made for the SMS group; in general, users do not allow applications to access their SMS list with the exception of Whatsapp and Telegram.

E. Users' Behavioral Consistency

In the final section of our analysis we are interested to see if users present a consistent behavior when they are using different applications on their devices. In other words, we are evaluating if the same participant follows the same patterns when using the fine-grained permission system on a variety of applications on the same device. We identified the users in our sample that had installed at least three applications listed at TABLE II. LinkedIn and Telegram were excluded from the analysis for this section, because the former application declares access to only 3 dangerous groups and the latter was installed only on 7 devices. We did not take into consideration the Calendar permission group, because it is only used on Facebook's manifest file. Hence, we are evaluating similarities to the way users grant or deny access to specific resources considering at least 3 popular social media applications installed on the same device.

Behavioral similarity or *consistency* is defined here as the quality of preserving similar user behavior when various applications (at least 3) request access to a specific permission group. For example, if the user has granted (or denied) access to the Microphone group to at least 60% of social media applications (installed on the same device), then this is considered as a sign of behavioral consistency.

Fig. 4 demonstrates the behavioral consistency (%) for 33 users in our sample. They had installed at least 3 applications

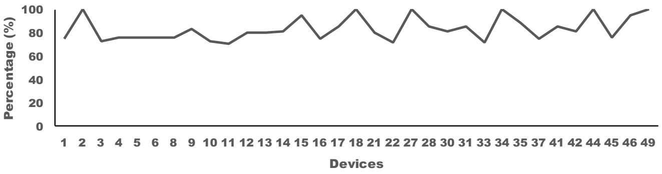

Fig. 5. Average Behavioral Consistency of Users.

from the popular social media applications list, as discussed previously in this section. User No 3 for example presented similarities at the Contacts, Location and Storage groups settings (83.3%, 66.67%, 66.67%, respectively). Finally, we gathered the average behavioral consistency per user and the results can be seen at Fig. 5. The overall average behavior consistency among the 33 users is 83.26% and the standard deviation is 10.05. These numbers indicate that users of popular social media in our sample presented strong similarities, considering their security settings on different applications.

F. Limitations

Due to the chosen methodology to conduct the survey and given that we aimed to offer a user-friendly experience to the participants, we avoided asking extended questions, offering at the same time a limited number of answers. This strategy limited participants' ability to provide very detailed answers. Further research work related to the usability and acceptance of the new permission model can verify or disprove our findings. Also, the anonymous permission data we gathered cannot describe if the user was asked at some point by an application to grant access to a group, and whether access was actually denied to this group.

V. Conclusion and Future Work

To conclude, this paper discussed the results of the first study that focuses on security perceptions related to the use of the new runtime permission model. The responses from our participants indicate that users adapted positively to the new model. Most of them prefer to use the new system compared to the previous and they believe they can control more drastically the sensitive data they share. We also demonstrated that, in general, one third of the requested permissions in our sample belonged to dangerous permission groups. Additionally, we showed that Storage, Contacts, Location, Phone and Camera were the most requested groups. Considering the accessibility of particular applications to resources, we highlighted the persistence of users to allow access to groups that are directly related to their main functionality (e.g. Camera for Instagram). Finally, we showed that (in general) users' behavior is consistent regarding the resources they allow to social media applications to access. More work needs to be done to assess users' trust to specific applications. In the future we also aim to investigate if the users' privacy settings and preferences change by time.

Acknowledgment

This work was supported by: EPSRC grant EP/N008448/1.

References

[1] S. Rosen, Z. Qian, and Z. M. Mao, "Appprofiler: A flexible method of exposing privacy-related behavior in android applications to end users," in *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*, ser. CODASPY '13. New York, NY, USA: ACM, 2013, pp. 221–232.

[2] P. Andriotis and T. Tryfonas, *Impact of User Data Privacy Management Controls on Mobile Device Investigations*. Cham: Springer International Publishing, 2016, pp. 89–105.

[3] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 627–638.

[4] B. Liu, M. S. Andersen, F. Schaub, H. Almuhimedi, S. A. Zhang, N. Sadeh, Y. Agarwal, and A. Acquisti, "Follow my recommendations: A personalized privacy assistant for mobile app permissions," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, Jun. 2016, pp. 27–41.

[5] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal, "Your location has been shared 5,398 times!: A field study on mobile app privacy nudging," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 787–796.

[6] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, and K. Beznosov, "Android permissions remystified: A field study on contextual integrity," in *24th USENIX Security Symposium (USENIX Security 15)*. Washington, D.C.: USENIX Association, Aug. 2015, pp. 499–514.

[7] 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*, ser. SOUPS '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:14.

[8] J. Jung, S. Han, and D. Wetherall, "Short paper: Enhancing mobile application permissions with runtime feedback and constraints," in *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, ser. SPSM '12. New York, NY, USA: ACM, 2012, pp. 45–50.

[9] S. Demetriou, W. Merrill, W. Yang, A. Zhang, and C. A. Gunter, "Free for all! assessing user data exposure to advertising libraries on android," *NDSS 2016*, 2016.

[10] M. Benisch, P. G. Kelley, N. Sadeh, and L. F. Cranor, "Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs," *Personal and Ubiquitous Computing*, vol. 15, no. 7, pp. 679–694, 2011.

[11] B. Henne, C. Kater, M. Smith, and M. Brenner, "Selective cloaking: Need-to-know for location-based apps," in *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, July 2013, pp. 19–26.

[12] K. Tang, J. Hong, and D. Siewiorek, "The implications of offering more disclosure choices for social location sharing," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 391–394.

[13] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "Mockdroid: Trading privacy for application functionality on smartphones," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, ser. HotMobile '11. New York, NY, USA: ACM, 2011, pp. 49–54.

[14] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, ser. CCS '11. New York, NY, USA: ACM, 2011, pp. 639–652.

[15] Y. Agarwal and M. Hall, "Protectmyprivacy: Detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '13. New York, NY, USA: ACM, 2013, pp. 97–110.

[16] P. Andriotis, G. Oikonomou, A. Mylonas, and T. Tryfonas, "A study on usability and security features of the android pattern lock screen," *Information and Computer Security*, vol. 24, no. 1, pp. 53–72, 2016.

[17] S. Parkin, K. Krol, I. Becker, and M. A. Sasse, "Applying cognitive control modes to identify security fatigue hotspots," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, Jun. 2016.