Impact of User Data Privacy Management Controls on Mobile Device Investigations

Panagiotis Andriotis, Theo Tryfonas

Chapter 5

IMPACT OF USER DATA PRIVACY MANAGEMENT CONTROLS ON MOBILE DEVICE INVESTIGATIONS

Panagiotis Andriotis and Theo Tryfonas

Abstract There are many different types of mobile device users, but most of them do not seek to expand the functionality of their smartphones and prefer to interact with them using predefined user profiles and settings. However, "power users" are always seeking opportunities to gain absolute control of their devices and expand their capabilities. For this reason, power users attempt to obtain "super user" privileges (root) or jailbreak their devices. Meanwhile, the "bring your own device" (BYOD) trend in the workplace and increased numbers of high profile users who demand enhanced data privacy and protection are changing the mobile device landscape. This chapter discusses variations of the Android operating system that attempt to bypass the limitations imposed by the previous Android permission model (up to version 5.1) and highlights the fact that forensic analysts will encounter devices with altered characteristics. Also, the chapter discusses the Android permission model introduced in the latest operating system (version M or 6.0) that will likely change the way users interact with apps.

Keywords: Android devices, privacy, trust, power users, anti-forensics

1. Introduction

Android is an open source project that enables developers to alter operating system characteristics according to their preferences. Data privacy and the lack of user controls on installed apps have always been major concerns for security-aware developers and users. The previous – but still dominant – permission model of Android operating systems (up to version 5.1) has been criticized for limiting the ability of users to control the private data that apps may access.

This chapter focuses on mobile devices that run variations of the Android Open Source Project (AOSP). It highlights the various approaches that deal with the fact that the previous – but still dominant – permission model of Android operating systems (up to version 5.1) is not flexible and does not allow users to restrict access to specific resources. Furthermore, it demonstrates that evidence derived from devices may contain falsified data due to app utilization that employs obfuscation measures to protect user data and privacy. This fact raises the specter that the probative value of "evidence" extracted from such devices can be put into question.

2. Data Privacy Concerns

Contemporary mobile devices are equipped with many sensors. The Android documentation lists at least twenty variables (e.g., TYPE_ACCELEROMETER) that can be used by developers to access various sensors and enrich the functionality of their apps. The sensors are essentially divided into hardware- and software-based sensors. Apps normally use sensors to measure orientation, motion and other environmental conditions and provide the expected functionality to users. A portion of the data produced by the apps contains information derived from sensors. This information is stored internally on the device or in the cloud. Some of the information may be encrypted (e.g., locations from Google Maps).

For example, a call to the camera or microphone of an Android device requires the inclusion of the appropriate permissions in the manifest `xml` file from the developer so that a user can be informed about the resources required by the specific app. Next, the user has to decide if he/she will accept the stated policy and download the app from the Play Store. The previous Android permission model has a binary accept-reject character. Therefore, if an app needs access to a user's contact list, it has to ask the user for permission to access it; then, the user is informed that his/her contact list will be shared via content providers to other ecosystems. Figure 1 presents screenshots of Android's privacy management control variations (left to right: permissions, incognito mode and Privacy Guard).

In theory, the Android permission model assures that data privacy is not violated without the knowledge of the user. But this is not always the case. In fact, privacy in the smartphone ecosystem is not only related to the stored data accessible by third-party applications via the aforementioned route, but privacy is also associated with the sensors themselves. For example, an Android device does not require permissions to be declared by an app for access to a number of device sensors

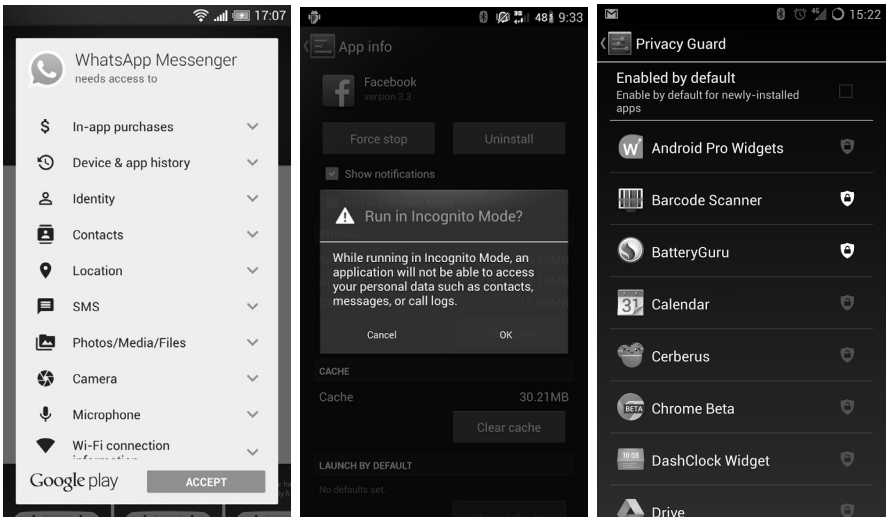

Figure 1. Android's privacy management control variations.

(e.g., light sensor) [16]. This potentially violates user privacy because an adversary could utilize the sensors to intercept information about users' private lives. For example, the information provided by an accelerometer in a mobile device could be used to determine if its user is moving, sitting at a table, or even sleeping.

Data sharing in ecosystems that run the Android operating system provides flexibility and limitless functionality. The sharing enables developers to implement apps that can communicate with data containers (e.g., contact lists) and obtain information from sensors (e.g., location services). Recent research has shown that forensic analyses can benefit from such capabilities because data availability becomes easier via applications that merge similar functionalities. An example is the Google Hangouts app, primarily a chatting app, which now serves as a text messaging app (SMS) [2] because it makes it much easier for users to send SMS messages with embedded information. As a consequence, valuable data is stored in databases (e.g., `babel1.db`) internal to mobile devices that are potentially the targets of forensic analyses. However, users are the vulnerable entities in this model because they install applications that request access to most of the available resources.

Mobile device users must be aware of the resources (data containers, software and hardware sensors) used by apps. Hence, according to their preferences, the operating system should provide solutions that satisfy their privacy concerns. This can be achieved by restricting access to categories of data (e.g., contact lists) if users so desire, as well as by

informing them about the specific portions of their devices that will be utilized by apps. The previous data management model of the Android operating system covers the latter situation. However, the need for a model that creates unique trust relationships between developers and users has been apparent for some time, and it is now available in version 6.0 of the Android operating system. The revised security model may well force consumers to understand the risks of downloading apps that require multiple resources from their devices. In the future, consumers could be even more cautious by consciously controlling the actions that the apps are allowed to perform in the ecosystems defined by their devices [3].

3. Mechanisms for Enhancing Data Privacy

In recent years, several data privacy preservation models have been implemented in Android devices. The approaches for handling the problems of data leakage and permission handling in Android environments fall into three categories: (i) app-based models; (ii) Android Open Source Project variations; and (iii) secure container models, which are essentially used in enterprise environments. Each of the three categories of approaches handles the weaknesses of Android's binary accept-reject model in a distinct manner.

3.1 App-Based Model

The first approach includes applications targeted for rooted devices. These applications mimic the privacy framework introduced in Android version 4.3 (Apps Ops), which is shown in Figure 2 [13]. In this environment, users can restrict access to data sources and sensors. For example, if a GPS navigation application requires access to the GPS sensor and contact list of a phone, the user could allow access to the GPS and restrict access to the contact list. Unfortunately, this feature was removed in version 4.4 – Android developers declared that the Apps Ops framework was created only for internal use and testing purposes.

However, the control privacy feature was well received by XDA Developers as well as by power users. Apps Ops allowed users to have absolute control of the services that apps could access. As a result, developers brought back the Apps Ops functionality. For example, one XDA forum member created the Xposed framework that provided the services that were removed in the official release. However, the disadvantage of this method is that an Android device must be rooted (super-user privileges) to allow the installation of the Xposed application package file (`apk`).

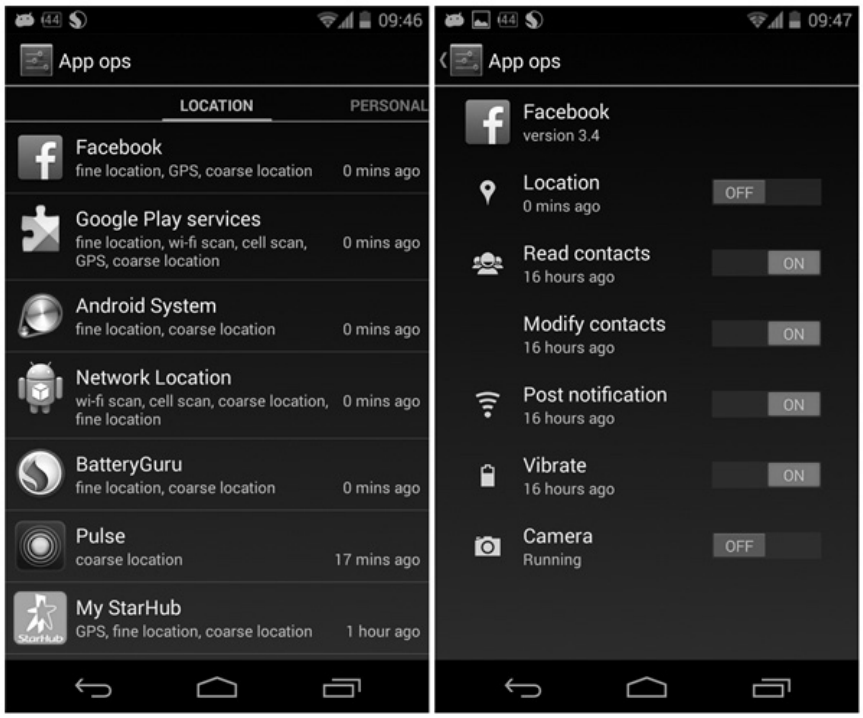

Figure 2. Apps Ops privacy management control [13].

App Ops variants created by several developers are available at the Google Play Store. However, numerous users have expressed their concerns about the effectiveness of these apps and whether or not they protect privacy. Reviews indicate the need for a universal approach that is safe to use and that restores the privacy controls that were removed after Android operating system version 4.3. The new runtime permission model seems to fill this gap.

The AppsOpsExposed framework is an open source project that is downloadable from Github (`repo.xposed.info/module/at.jclehner.appopsxposed`). AppsOpsExposed is essential and should be installed on a device so that other applications can restore the Apps Ops functionality. One example is XPrivacy, an award-winning application that uses the framework and utilizes obfuscation techniques to prevent sensitive data leakage. XPrivacy restricts the categories of data that an application can access by feeding the application with fake data or no data. It is also an open source project, but an Android device has to be rooted to provide its functionality.

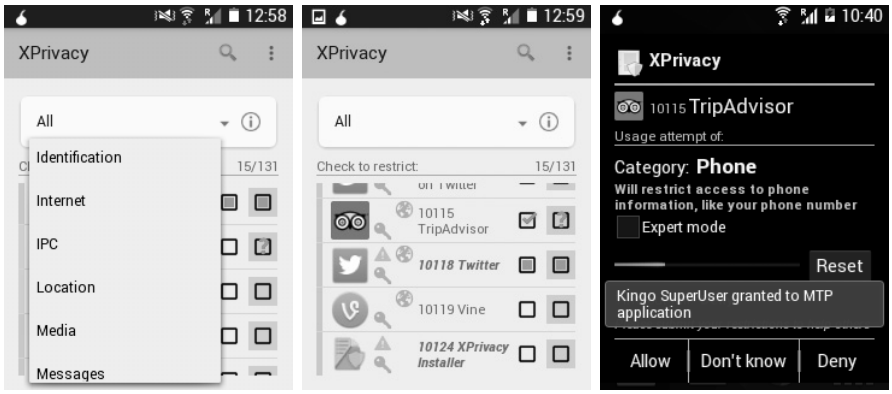

Figure 3. XPrivacy setup.

Experiments were conducted with XPrivacy (version 3.6.19) installed on a Samsung Galaxy Pocket 2 (SM-G110H) running Android operating system version 4.4.2. First, the device had to be rooted using Kingoroot, a popular exploit. Note that this rooting exploit was selected purely for experimental reasons. Such exploits could introduce additional security vulnerabilities and most vendors discourage their installation. The XPrivacy installer from the Google Play Store is useful for installing the Xposed framework and the XPrivacy app. After the installation, the user can choose the functions that should be restricted for specific apps (Figure 3).

Experiments were conducted with the location services and phone contact list. The primary testing location (PTL) was (51.4558270, –2.6034071) (Figure 4). The phone was used for a period of time before XPrivacy was installed. Thus, the SIM contact list, SMS messages and other information were already registered in the internal storage of the device. After XPrivacy was installed, direct access to the location services, contact list and other accounts was restricted. As a consequence, some apps did not work as expected. For example, Twitter required a new log in every time the app was invoked, Facebook Friend Finder was unable to find any new friends by reaching the contact list and Yelp could not function properly (Figure 4).

Further research demonstrated that, when location services were used for Twitter posts, accurate locations were not included in the tweets (Figure 5). Also, other apps such as Facebook and Swarm were fed with false data provided by XPrivacy according to the relevant settings (Figure 5). Thus, a cautious – or malicious – user could benefit from similar apps and utilize them to hinder forensic investigations. Forensic

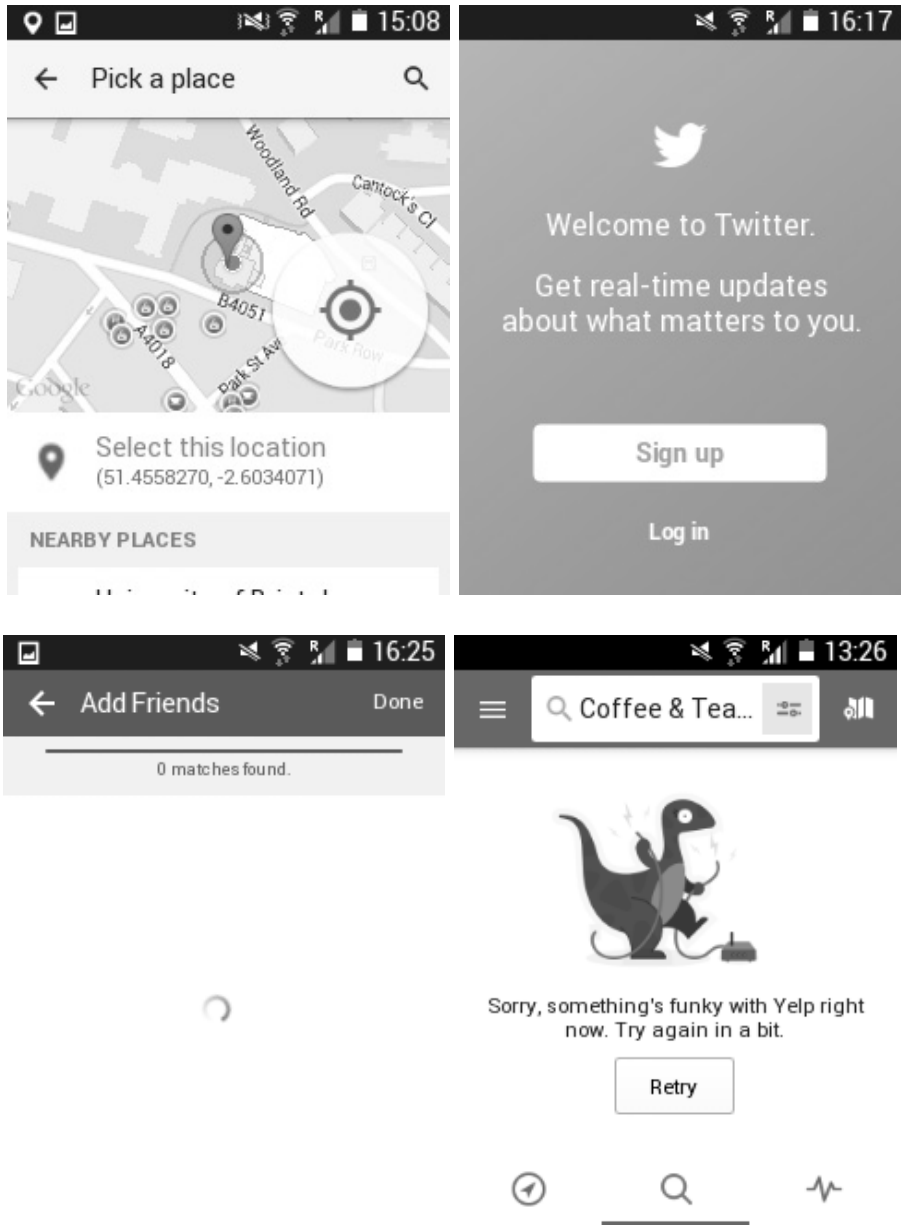

Figure 4. Using apps with XPrivacy restrictions.

analysts should be aware of these practices and should be very careful when presenting evidence from rooted devices in court because such applications could have been installed and used on the devices. These

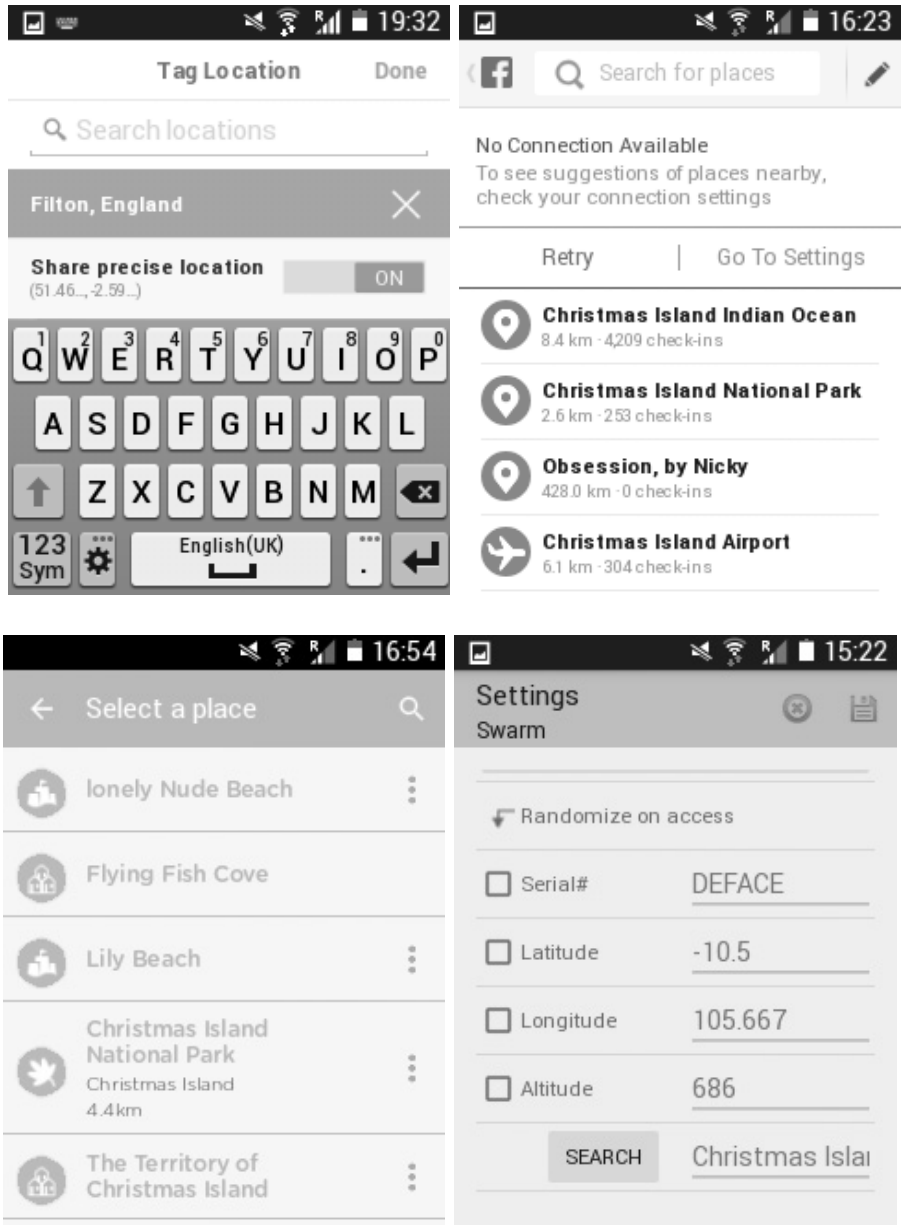

Figure 5. Location obfuscation caused by XPrivacy settings.

applications increase the likelihood that evidence has been manipulated or falsified.

Forensic analysis was also performed on the experimental smartphone using the data acquisition and analysis strategy described in [1]. Interestingly, the app databases that locally store various app related data did not contain any information that pointed to the original primary testing location. For example, the `fsq.db` database in the "venues" table contained the location (105.667, –10.5) corresponding to the longitude and latitude of Christmas Island National Park provided by XPrivacy (Figure 5). Despite the fact that apps such as XPrivacy can mislead forensic analysts, other apps (e.g., Google Maps and the Location Tagger on the Camera app) worked flawlessly. If a forensic analyst is able to extract data from these apps, there is a possibility that portions of the retrieved data might have been manipulated. Thus, the trustworthiness of the extracted evidence can be put into question.

3.2 Android Open Source Project Variations

The second category of proposals for data privacy management for the Android operating system includes a few key (firmware) variations of the Android Open Source Project. The Android Open Source Project offers a common platform that enables developers to modify the orientation of the operating system in various directions. CyanogenMod is among the most popular variations of the Android Open Source Project and it implements a different approach to Android's data privacy management. For example, the CM11 version based on Android KitKat (version 4.4) features the Privacy Guard permission manager app. Privacy Guard (Figure 1) provides the same functionality with the XPrivacy app (i.e., it uses obfuscation, a technique proposed in several technical papers [6]) and is essentially an evolution of the incognito mode (Figure 1). Specifically, CyanogenMod offers the incognito mode privacy management feature for older versions (starting from CM7). Another popular example of a modified Android operating system version is OxygenOS, which runs on OnePlus 2 phones.

The CyanogenMod installer web page suggests that it is not necessary for a phone to be rooted to install and run the latest version. However, users who are not familiar with technology might find the installation process obscure. Privacy Guard enables users to turn on or off any feature that they feel is not necessary for an app to function. For example, a user may decide that a social media app such as Twitter should not have access to location data on the phone. Privacy Guard either restricts access to the information or it supplies the app with limited resources. The main limitations of Privacy Guard are that it does not anonymize users and prevent apps from tracking their sessions. Another problem is

that some apps might throw exceptions during runtime that cause them to crash.

Android Open Source Project variations like CyanogenMod demonstrate that rooted phones are not the only devices that may potentially have anti-forensic capabilities. Indeed, apps such as those discussed above could create similar anti-forensic environments. Therefore, smartphone ecosystems defined by such devices may also contain modified or falsified data. Forensic analysts should be cautious and take strong steps to validate data originating from the devices.

3.3 Secure Container (BYOD) Model

Blackphone (and Blackphone 2) implement a different approach to the problem of data privacy preservation. Their SilentOS operating system (previously known as PrivatOS) is also based on the Android platform. The concept behind this Android Open Source Project variation is that data privacy and security should be the most powerful features of an operating system. This is why the Blackphone has built-in apps such as the Blackphone Security Center. It also features third-party services that enable Blackphone users to remotely wipe and gain control of their data from anywhere in the world. Users can also use encryption for secure search and browsing, data transfer and storage, and voice calls and chats.

These devices offer an adequate solution in corporate environments that have a bring your own device policy. However, most of the provided security services come with some cost – they may be free of charge for a period of time, but users may eventually have to pay subscription fees to maintain high levels of security. Obviously, standard forensic analysis tools and practices cannot be applied to such devices. Forensic analysts should expect to use sophisticated hardware and software to extract useful information from these devices.

Finally, the rapid proliferation of mobile devices across society, the alarming increase and sophistication of malware and grave concerns about data privacy have led companies such as Samsung to offer security frameworks targeted for corporate environments. An example is the Samsung KNOX framework, which enhances trust by implementing robust, multi-layered mobile security. In fact, this framework has created a separate data privacy management category for itself. KNOX offers its own workspace above the Android stack where distinct applications can work safely. It also features hardware components and advanced cryptographic services.

The enhancements presented by KNOX have made it a pioneer in the Android enterprise mobility space. Users can customize their personal space to share data with their (corporate) secure containers. The data could be contacts, calendars, browser bookmarks, etc. The new generations of the Android operating system are empowered by such enterprise capabilities. They add value to data privacy by separating personal and corporate data by essentially creating different user accounts on the same device. In the case of forensic analysis, these systems will likely require special techniques to uncover evidence because they engage proprietary cryptographic protocols.

The Android for Work framework is another emerging technology that uses containerization. The framework separates business apps and personal apps, enabling Android smartphone and tablet users to use the same devices for their professional and personal lives. This is accomplished by setting up dedicated work profiles for business content that do not interfere with their personal profiles. Corporate IT management services cannot reach or manipulate personal data belonging to users. Users enjoy familiar experiences when using their devices in the workplace and gain control over the data to be shared. Security is enhanced via sandboxing, security policies, app verification and encryption. Furthermore, an enterprise mobility management (EMM) system can be used to manage all the engaged mobile devices, (enterprise) apps and business data from a single console. Clearly, forensic analysts will face considerable obstacles when they attempt to obtain data related to enterprise environment activities without the assistance of the enterprise mobility management system vendor.

3.4 Towards a New Era of Mobile Computing

A new trend in the smartphone market is the merging of the enterprise mobility and bring your own device concepts in a single environment. Enterprise mobility management applications enable IT administrators to enforce a wide range of policies by possibly following the KNOX paradigm. However, these advancements may well be overwhelming for the average user. Usability, flexibility and simplicity should be the most critical concepts underlying data protection schemes. The sixth version of the Android operating system (version M) has brought a radical change to the operating system security model (Figure 6) – it allows users to control the data they share using runtime permissions. This means that future forensic analysts will encounter cases where smartphone users have restricted data sharing between apps, significantly complicating forensic investigations of the mobile devices. Also, apps

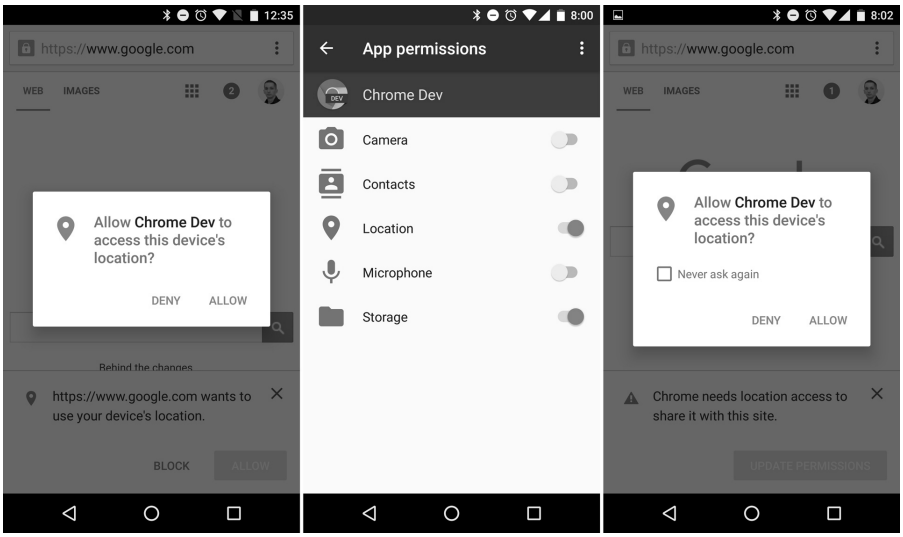

Figure 6. New runtime permissions model.

in the near future will be more personalized due to advancements and restrictions. Thus, generic (traditional) digital forensic models will likely be inadequate in the new era that embraces the permission paradigm.

4. Other Open Source Operating Systems

Other open source platforms, such as the Mozilla Firefox operating system and Tizen, follow different security and privacy models. As in the case of Android, they rely on Linux kernels, but they are also equipped with web runtime layers on top of the kernels. This improvement enables developers to create apps (webapps) using only web technologies (e.g., HTML5, CSS and Javascript).

Mozilla has developed proprietary APIs for the Firefox operating system in which the handling of app permissions is different for hosted apps and packaged apps. Hosted apps are downloaded from websites while packaged apps are already installed on devices. Packaged apps are divided into three categories: (i) web apps, which do not use privileged or certified APIs; (ii) privileged apps, which use privileged APIs (distributed through the Firefox Marketplace); and (iii) certified apps (preinstalled), which can access privileged and certified APIs. Privileged and certified apps have content security policies, but each app is required to invoke an installation method. This procedure validates the app and asks the user to approve its installation. In other words, depending on the app type (e.g., if it is certified or privileged), the Firefox

operating system implicitly grants some of the permissions and then asks the user to approve other permissions (using prompts during runtime as in the case of the upcoming Android version). However, this model does not give the user the power to invoke or deny permissions for certified apps.

Tizen, on the other hand, has a predefined set of APIs that are divided into specific categories. The communications API, for example, provides functionality for Bluetooth control and messaging; also, it provides email services, access to near field communication devices and push notifications. Web apps require authorization to access restricted APIs via a manifest file, which lists the required features from the apps following a (subject, object, permission)-based access control model. Tizen is still in its early days and its developers intend to create a multi-purpose operating system that will serve mobile devices, wearables, vehicle infotainment systems and smart TVs. The proliferation of Android devices makes it unlikely that a forensic analyst would encounter a smartphone that runs a Tizen (or similar) operating system. However, forensic analysts should be aware that these new technologies may well enter the market; because they are open source, there is an increased likelihood that they could be incorporated in smartphones targeted for underdeveloped or developing countries. Thus, digital forensic research should focus on techniques and tools for handling cases where the evidence container is a device that runs one of the aforementioned operating systems or some other emerging operating system that could acquire a significant market share (e.g., Ubuntu Touch operating system).

5. Android Version 6

The advent of Android version M (version 6.0) will likely change the way users interact with their apps, given the runtime permissions model that was revealed at the M Developer Preview. The new permission model ensures that developers will build apps that request user permissions only for a limited number of resources. Other permissions will have to be requested and granted by users at runtime. This novel permission system will make smartphone ecosystems unique. The advancements in data sharing between apps will change the way forensic analysis is performed because the devices will restrict access to resources. Hence, an analyst may only be able to find limited data in a database on a device. On the other hand, users with limited privacy and security concerns or awareness may enjoy all the new functionalities provided by the installed apps while knowingly or unknowingly allowing access to all resources.

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3 - 2.3.7	Gingerbread	10	3.0%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	2.7%
4.1.x	Jelly Bean	16	9.0%
4.2.x		17	12.2%
4.3		18	3.5%
4.4	KitKat	19	36.1%
5.0	Lollipop	21	16.9%
5.1		22	15.7%
6.0	Marshmallow	23	0.7%

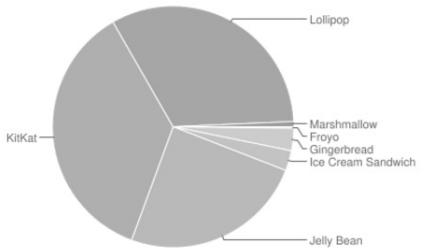

Figure 7. Developer Dashboard (January 2016).

User adoption of the new model and user reactions would be an interesting subject for future research. However, history has shown that all Android users do not immediately download and install the latest operating system versions on their devices; in fact, a large number of users prefer to use older operating systems. Figure 7 shows that four months after the release of the sixth version, only 0.7% of the devices that visited the Google Play Store were running the Marshmallow edition. Thus, earlier versions of the Android operating system will persist in the user community for several years.

6. Related Work

Data privacy protection mechanisms will be vital in the new era of mobile computing where data sharing can create risks. Small- and medium-sized enterprises appear to be more vulnerable to data leakage because they often do not have advanced IT resources and capabilities [9]; thus, they may not implement strong bring your own device models.

Several approaches have been proposed to protect personal mobile computing from unlimited data sharing. The MyShield system [4] supplies anonymized data if requested by users and incorporates Secure Circles, a control mechanism that enables users to manage app access to sensitive data based on the level of trust.

Other approaches have focused on location services [5], providing an opportunity for mobile device users to protect their privacy by adjusting the accuracy of their locations in order to use location-based apps while

simultaneously securing their private data using on-device or service-based obfuscation [11]. Bernheim Brush et al. [7] note that, when individuals agree to share their location data using existing obfuscation methods, their decisions are consistent with their personal privacy concerns. Also, Tang et al. [15] suggest that, when abstract location descriptions are included in privacy protection schemes, location sharing is more likely to occur.

Henne et al. [10] have proposed a crowd-based recommendation system for Android devices that allows users to configure the accuracy of location data provided to apps based on five precision levels; they also claim that unskilled users benefit from such an approach. Crowdsourcing for location-based privacy settings is discussed in [14]. Beresford et al. [6] have developed MockDroid, a modified version of the Android operating system, which works like XPrivacy; however, its principal difference is that it essentially feeds "empty" resources to apps that require access to potentially sensitive data. This reduces app functionality, but the vast majority of apps on the device work without any problems. AppFence is a data protection mechanism that uses shadowing and exfiltration blocking on apps while reducing adverse side effects [12]. According to its developers, the mechanism did not cause problems to 66% of the tested applications. Finally, Fisher et al. [8] have demonstrated that iOS users can be classified into three categories according to their location privacy settings: those who deny access to all apps, those who allow access to all apps and those who selectively permit access to the apps they trust.

7. Conclusions

This chapter has discussed a variety of mobile device ecosystems that emerge from the fact that advanced users tend to change – and sometimes dramatically change – the expected behavior of their smartphones. The chapter has also highlighted variations in the data privacy and security models developed by the Android Open Source Project. The implication for forensic analysts is that when smartphones are rooted and/or obfuscation apps are installed, forensic analyses will likely provide limited or false evidence. Security models used by other open source systems have also been discussed, along with the projected limitations of the current Android operating system version. This chapter has not considered Apple (iOS) devices because they use a different permission system that enables users to restrict data sharing and deny access to specific resources. Clearly, the trend is that future smartphones will provide ever-increasing privacy and security functionality to users. Forensic re-

searchers and analysts should be aware of this trend and attempt to develop sophisticated techniques and tools that maximize evidence recovery while ensuring the probative value of the recovered evidence.

References

[1] P. Andriotis, G. Oikonomou and T. Tryfonas, Forensic analysis of wireless networking evidence of Android smartphones, *Proceedings of the IEEE International Workshop on Information Forensics and Security*, pp. 109–114, 2012.

[2] P. Andriotis, G. Oikonomou, T. Tryfonas and S. Li, Highlighting relationships of a smartphone's social ecosystem in potentially large investigations, to appear in *IEEE Transactions on Cybernetics*, 2016.

[3] P. Andriotis, T. Tryfonas, G. Oikonomou and I. King, A framework for describing multimedia circulation in a smartphone ecosystem, in *Advances in Digital Forensics XI*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 251–267, 2015.

[4] R. Beede, D. Warbritton and R. Han, MyShield: Protecting Mobile Device Data via Security Circles, Technical Report CU-CS-1091-12, Department of Computer Science, University of Colorado Boulder, Boulder, Colorado, 2012.

[5] M. Benisch, P. Kelley, N. Sadeh and L. Cranor, Capturing location-privacy preferences: Quantifying accuracy and user-burden tradeoffs, *Personal and Ubiquitous Computing*, vol. 15(7), pp. 679–694, 2011.

[6] A. Beresford, A. Rice, N. Skehin and R. Sohan, MockDroid: Trading privacy for application functionality on smartphones, *Proceedings of the Twelfth Workshop on Mobile Computing Systems and Applications*, pp. 49–54, 2011.

[7] A. Bernheim Brush, J. Krumm and J. Scott, Exploring end user preferences for location obfuscation, location-based services and the value of location, *Proceedings of the Twelfth ACM International Conference on Ubiquitous Computing*, pp. 95–104, 2010.

[8] D. Fisher, L. Dorner and D. Wagner, Location privacy: User behavior in the field, *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, pp. 51–56, 2012.

[9] M. Harris and K. Patten, Mobile device security considerations for small- and medium-sized enterprise business mobility, *Information Management and Computer Security*, vol. 22(1), pp. 97–114, 2014.

[10] B. Henne, C. Kater and M. Smith, Usable location privacy for Android with crowd recommendations, *Proceedings of the Seventh International Conference on Trust and Trustworthy Computing*, pp. 74–82, 2014.

[11] B. Henne, C. Kater, M. Smith and M. Brenner, Selective cloaking: Need-to-know for location-based apps, *Proceedings of the Eleventh International Conference on Privacy, Security and Trust*, pp. 19–26, 2013.

[12] 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, *Proceedings of the Eighteenth ACM Conference on Computer and Communications Security*, pp. 639–652, 2011.

[13] T. Kaiser, Google removes "App Ops" privacy control feature from Android 4.4.2, *DailyTech*, December 16, 2013.

[14] 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, *Proceedings of the ACM International Conference on Ubiquitous Computing*, pp. 501–510, 2012.

[15] K. Tang, J. Hong and D. Siewiorek, The implications of offering more disclosure choices for social location sharing, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 391–394, 2012.

[16] T. Vidas and N. Christin, Evading Android runtime analysis via sandbox detection, *Proceedings of the Ninth ACM Symposium on Information, Computer and Communications Security*, pp. 447–458, 2014.