Chapter 1

THE EFFECTS OF POWER USERS' DATA PRIVACY MANAGEMENT CONTROLS ON MOBILE DEVICE INVESTIGATIONS

Panagiotis Andriotis and Theo Tryfonas

Abstract There exist different types of mobile device users. Most of them do not seek to expand the functionality limits of their smartphones and prefer to interact with them using predefined user profiles and settings. Others, namely 'power users', always seek different paths to get absolute control of their devices' capabilities. For this reason, these users prefer to get 'super user' privileges (root or jailbreak their devices). In addition, the arising 'Bring Your Own Device' (BYOD) market and the existence of high profile users, that demand enhanced data privacy and protection, creates new trends in mobile computing. In this review paper we discuss the existence of alternative mobile devices that run variations of the Android Open Source Project (AOSP). Also, we highlight the different approaches that deal with the fact that the previous permission model of the Android OS (up to version 5.1) is not very flexible and does not allow the user to restrict access to specific resources. Furthermore, we demonstrate that evidence derived from power users' devices might contain falsified data due to app utilization that employs obfuscation measures to protect users' data and privacy. This fact urges a basic problem for forensic analyses: the level of trust on evidence derived by such devices can be put into question.

Keywords: BYOD, Android, XPrivacy, AOSP, trust, anti-forensics.

1. Introduction

Android is an open source project allowing developers to alter OS characteristics according to their preferences. Data privacy and the lack of user control on installed apps was always a primary concern for security aware developers and users. The previous (but still dominant) permission model of the Android OS (up to version 5.1) has been crit-

icized for limiting users' power on deciding which private data an app should be able to reach. This paper discusses variations of the Android OS that aim to bypass the aforementioned limitations and highlights the fact that forensic analysts might eventually face devices with altered characteristics. Also, we discuss the new permission model that was introduced in the current version of the OS (version M, or 6.0), which will probably change the way users interact with their apps.

We know that contemporary mobile devices are equipped with numerous sensors. The Android documentation lists at least twenty different variables (i.e. 'TYPE_ACCELEROMETER') which can be used by developers in order to get access to these sensors and enrich the functionality of their apps. Sensors are basically divided into two categories according to the documentation: a) hardware and b) software sensors. Apps normally use these sensors to measure orientation, motion and other environmental conditions and finally provide the expected functionality to the user. A portion of the data produced by the apps contains information derived by the devices' sensors. These data are usually stored internally in the device or in the cloud. Some of them may be encrypted, i.e. locations from the Google Maps.

As an example, a call to the camera or the microphone requires the inclusion of the appropriate permissions in the manifest xml file from the developer, in order the user to be informed about the resources this specific app needs to work. Then the users decide if they will accept them and download the app from the Play Store. The previous permission model had a binary 'accept-reject' character. Therefore, if an app needs access to the users' contact lists it will also ask permission to get it (Figure 1). Thus, users are informed that their contact list is going to be shared through content providers to other ecosystems.

2. Data Privacy Concerns

Theoretically, the current model assures that data privacy is not violated without the knowledge of the user. But sometimes this is not the case. We believe that privacy in the smartphone ecosystem is not only related to the stored data, which can be accessed by third party applications via the aforementioned route. Privacy is also associated with the sensors themselves. For example, the system does not require any permission to be declared by an app, when the app requests to access a great portion of sensors existing on a mobile device (e.g. the light sensor) [19]. This fact introduces vulnerabilities in users' privacy because an adversary could utilize sensors to intercept information that might violate their personal lives and habits. For example, via the accelerator

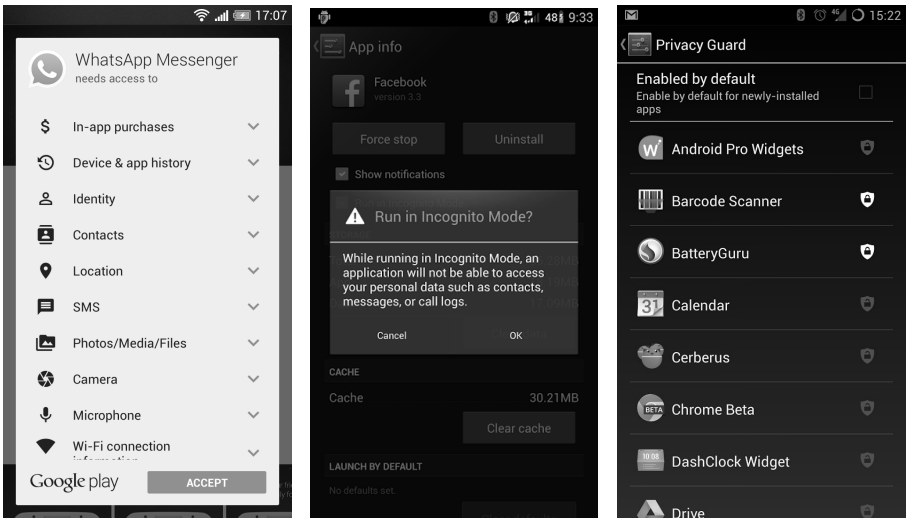

Figure 1. Screenshots Showcasing Android's Privacy Management Control Variations (Permissions, Incognito Mode, Privacy Guard).

one can figure out if the mobile device is currently sitting on a table or not [4]. This information might be useful for someone to understand when the person is moving or even sleeping.

Data sharing in ecosystems that run the Android OS provides flexibility and limitless functionality. It allows developers to implement apps able to communicate with data containers (e.g. contact lists) and get information from sensors (e.g. location services). Recent research work has shown that forensic analyses can benefit from such capabilities because data availability becomes easier via applications that merge similar functionalities. Such an example is the Google Hangouts app, which was primarily a chatting app but it can now act as a text messaging app (SMS) [2]. It is easier now for users to send SMS with embedded information via these apps. As a consequence, these data are stored in the appropriate databases (e.g. babel1.db) that reside internally in mobile devices and are available for analysis. However, users seem to be vulnerable entities in this model because they eventually install applications that request access to the majority of available resources.

Mobile device users must be aware of the resources (data containers, software and hardware sensors) used by an app. Hence, according to their preferences, the operating system should provide solutions that adhere to their privacy concerns. This can be achieved not only by restricting the access to categories of data (e.g. contact list) when the users decide to do that, but also by informing them which parts of the

device are going to be utilized by the particular app. The previous data management model of the Android OS covers the latter. However, the need for a model, which builds a unique relationship of trust between the developer and the user, is apparent and it is now available at version 6.0 of the Android OS. Therefore, the revised security model might force consumers to understand the risks of downloading an app that requires multiple resources from their devices. Hence, consumers in the future might be more cautious by choosing the actions that the apps are allowed to perform in the ecosystems defined by their devices [3].

3. Defence Mechanisms to Enhance Data Privacy

During the recent years we experienced the implementation of different models considering data privacy preservation for Android devices. The community has seen different approaches to the problem of data leaking and permission handling in the Android OS environment. These approaches can be classified in three distinct categories: 1) the app based model, 2) the Android Open Source Project (AOSP) variation and, 3) the 'secure container' model, which is basically used at enterprise environments. Each of these categories handles the Android's architectural weakness of the binary model 'accept-reject' in distinct ways.

3.1 The App Based Model

The first approach includes applications targeted basically to rooted devices. These applications mimic the privacy framework that was first introduced from the Android developers at version 4.3, namely Apps Ops (Figure 2) [15]. Within this environment, users had the choice to restrict access to various sources of data and sensors. For example, if a GPS navigation application required access to the GPS sensor and the contact list of the phone, the user had the choice to restrict access to the contact list and allow access to the GPS. Unfortunately, this feature was removed from version 4.4; Android developers declared that Apps Ops was a framework that was released for internal use and for testing reasons.

However, the control privacy feature was well received in various communities (and power users) like the XDA Developers. Apps Ops allowed the users to have absolute control of the services that the apps were accessing. For this reason, developers from this community used their programming skills in order to bring the Apps Ops functionality back. Among them, an XDA forum member created the Xposed Framework aiming to bring back the service that was removed by the official re-

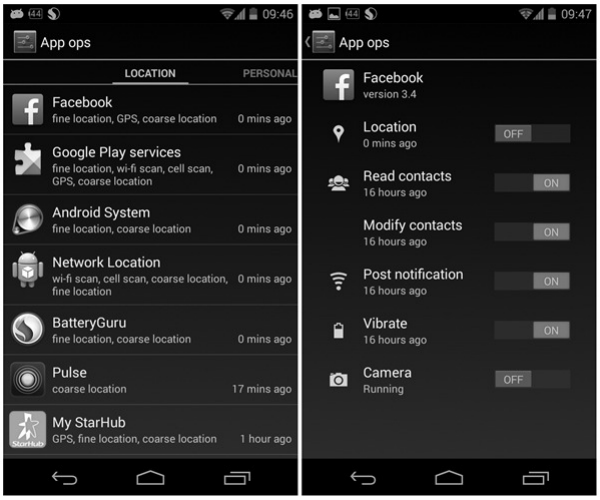

Figure 2. Screenshots Showcasing Android's 'Apps ops' Privacy Management Control. [15]

lease [16]. The disadvantage of this method is that the device must be rooted (super-user privileges) to allow the installation of the Xposed application package file (apk).

App Ops variants are available for installing from several sources (developers) at the Google Play Store. However, numerous users expressed their concerns about the effectiveness of these apps and furthermore, if they keep their privacy safe. These reviews indicate the need for a universal approach that will be safe to use and it will restore the privacy controls that were removed from the successor of version 4.3 of the popular operating system. The new runtime permission model seems to fill this gap.

The 'AppsOpsExposed' framework is an open source project that can be downloaded from Github [9]. It is essential and should be installed in the device in order other applications to be able to restore the Apps Ops functionality. 'XPrivacy' for example is an award winning application, which uses the framework and utilizes obfuscation techniques aiming to prevent sensitive data leaking. It restricts the categories of data an application can access by feeding the application with either fake or no data. It is also an open source project but the device should be rooted in order to provide its functionality.

3.1.1 Experimental Results. We experimented with XPrivacy (version 3.6.19) which was installed on a Samsung Galaxy Pocket

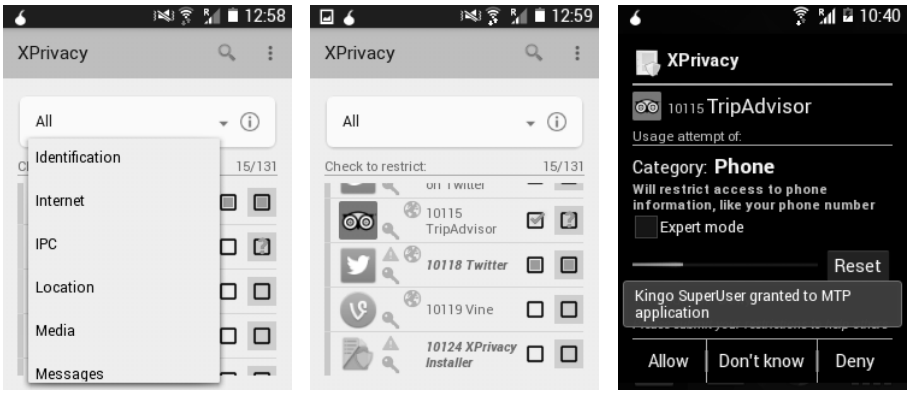

Figure 3. Screenshots Demonstrating XPrivacy Setup.

2 (SM-G110H) running the Android OS (version 4.4.2). First, we had to root the device using a popular exploit (Kingoroot). We should note here that this procedure was chosen just for experimental reasons. Solutions of this kind might introduce additional security vulnerabilities and most vendors discourage users to install them. The XPrivacy installer from the Google Play Store can be helpful in order to install the Xposed Framework and the XPrivacy app. After installation the user can choose which functions would like to restrict on specific apps (Figure 3).

We experimented with location services and the phone's contact list. Our 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, SIM contact list, SMS messages and other information were already registered in the device's internal storage. After XPrivacy was installed, direct access to the location services, the contact list and other accounts was restricted. As a consequence, various apps were not working as expected. For example, a Twitter user had to log in any time the app was invoked or the Facebook Friend Finder was not able to find any new friends by reaching the contact list or Yelp could not function properly (Figure 4).

Further research demonstrated that when we were using location services on Twitter posts, the accurate location was not included in the tweet (Figure 5). Also other apps like Facebook or Swarm were fed with false data provided from XPrivacy according to the relevant settings as Figure 5 demonstrates. Thus, a cautious (or malicious) user could benefit from similar apps and utilize them in order to mislead future investigations. Forensic analysts should be aware of these practices and be very careful when presenting evidence in court from rooted de-

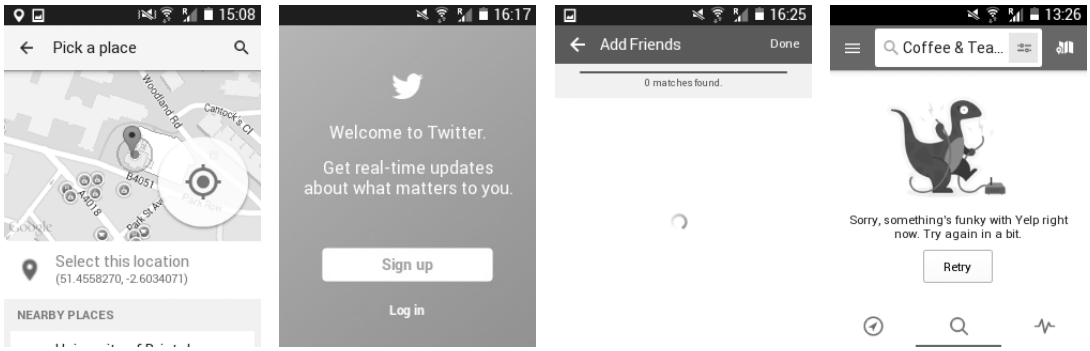

Figure 4. App Screenshots: Using Apps with XPrivacy Restrictions.

vices because such applications might be installed and used at the past from the mobile device owner. Hence the possibility the evidence to be manipulated or falsified is very high.

We also performed forensic analysis on the experimental smartphone using the data acquisition and analysis strategy presented in [2]. Interestingly, the app databases that store locally various app related data, did not contain any information that pointed to our original location (PTL). For example, the database data/com.joelapenna.foursquared/databases/fsq.db at the 'venues' table contained the location (105.667, -10.5), which is the longitude and latitude of 'Christmas Island National Park' provided by XPrivacy (Figure 5). Despite the fact that apps like XPrivacy can mislead the analyst during an investigation, there exist other apps (like Google Maps or the Location Tagger on the Camera app) that were working flawlessly. If the analyst manages to derive data from these apps, there will be a hint that parts of the retrieved data might be manipulated. Thus, trust on the derived evidence can be put in question.

3.2 AOSP Variations

The second category of proposals for data privacy management for the Android OS includes a few noteworthy variations (firmware) of the AOSP. The AOSP offers a common platform, which can be used by developers to modify the orientation of the operating system in various directions. 'CyanogenMod' is among the most popular variations of the AOSP and it implements a different approach to Android's data privacy management. For example, the version with the code name CM11 is based on Android KitKat (version 4.4) and it features the 'Privacy Guard' permission manager app. Privacy Guard (Figure 1) provides the

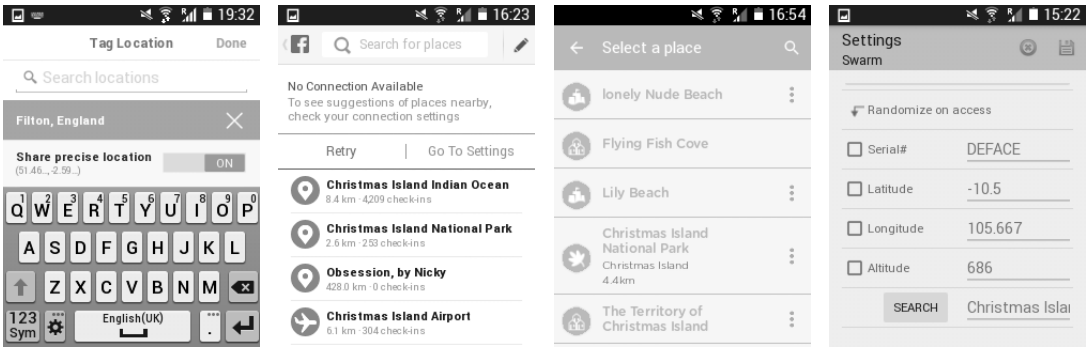

Figure 5. Screenshots: Location Obfuscation Caused by XPrivacy Settings.

same functionality with the XPrivacy app (uses obfuscation, a technique that was proposed in various technical papers [7]) and it is basically an evolution of the previous 'Incognito Mode' (Figure 1). The Incognito Mode is a privacy management feature CyanogenMod used to offer with older versions of the system (starting from the CM7 version). Another popular example of a modified Android OS version is the OxygenOS, which runs on the 'OnePlus 2' phones.

The 'CyanogenMod installer' web page suggests that there is no need the phone to be rooted in order to install and run the latest version. However, users who are not familiar with technology advancements might find the installation process obscure. Privacy Guard offers the capability to turn on and off any feature the users think that is not necessary for an app to function. For example, an individual might decide that a social media app like Twitter should not have access to the location data of the phone. Privacy Guard will restrict the access to the specific information or it will feed the app with limited resources. The main limitation of the proposed scheme is that Privacy Guard does not anonymize users or prevent apps to track their sessions. Another problem is that some apps might become irresponsible and throw exceptions during runtime that will cause them to crash.

AOSP variations like the CyanogenMod demonstrate that there is no need a phone to be rooted in order to be considered as a potential anti-forensic medium. Apps like the aforementioned might create similar environments like those discussed in the previous sub-section. Therefore, smartphone ecosystems that are defined by such devices might contain falsified information too. Forensic analysts must be cautious and able to prove the validity of data that originate from similar devices.

3.3 The Secure Container Model (BYOD)

The release of the Blackphone (and Blackphone 2) introduced a different approach to the problem of data privacy preservation. The phone's operating system, namely SilentOS (previously PrivatOS), is also based on the Android platform. The concept behind this AOSP variation is that data privacy and security should be the most powerful features of the OS. This is why the Blackphone has built-in apps like the Blackphone Security Centre. It also features third party services allowing Blackphone users to remotely wipe and gain control of their data from anywhere in the world. They can also enjoy secure search and browsing, they can safely transfer and store their data and they can speak and chat freely using the offered encryption functionality.

The specific device could be a reasonable solution within a corporate environment and fits the emerging Bring Your Own Device (BYOD) model. However, most of the provided security services come with a considerable cost (they might be free for a period of time but then the user has to pay a subscription to maintain the level of security at high standards). Thus, standard forensic analysis tools and practices cannot be applied to such devices. Analysts should expect that the advanced technology that is bound with this type of hardware and software requires specialized techniques in order to extract useful information.

Finally, the rapid proliferation of mobile devices in our personal and professional lives, the alarming evolution of malware and the latest concerns about data privacy urged companies like Samsung to present various security frameworks targeting corporate environments. Samsung KNOX is a framework that enhances trust by supporting robust, multilayered mobile security. It eventually, initiated a separate data privacy management category by 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 under this scheme made KNOX a pioneer in the Android enterprise mobility space. The users can customize their personal space to allow data to be shared to their (corporate) secure container. These data can be contacts, calendars, browser bookmarks, etc. The current generations of the Android OS are empowered by such enterprise capabilities. This feature adds value to data privacy by separating personal and corporate data utilizing basically different user accounts on the same device. When it comes to forensic analysis these systems will probably need special treatment to reveal evidence, since they are bound with proprietary cryptographic protocols.

Another emerging technology that uses containerization is the 'Android for Work' framework. Within this environment, business apps and personal apps are separated and the mobile device owners can use their Android smartphones or tablets at work and at their personal lives. This can be done by setting up dedicated work profiles for business content which does not interfere with the personal profiles. IT management services cannot reach or manipulate personal data within the specific environment. Thus, the user enjoys a familiar experience when handling equipment at the workplace and gains control over the data to be shared. Security is enhanced via sandboxing, security policies, app verification and encryption. Furthermore, Enterprise Mobility Management (EMM) platforms can be used to manage all engaged mobile devices, (enterprise) apps and business data from a single console. Hence, forensic analysts will probably face various obstacles in order to obtain data related to enterprise environment activities without the assistance of the EMM management vendor.

3.4 Towards the New Era of Mobile Computing

Currently we have seen a trend in the smartphone market to merge enterprise mobility and BYOD concepts under the same environment. EMM applications allow IT administrators to enforce a wide set of policies following possibly the KNOX paradigm. However, these advancements might be overwhelming for the average user. Usability, flexibility and simplicity should be the most critical concepts behind the development of schemes that will protect personal data. The release of the sixth version of the Android OS (version M) brought a radical change to the OS's security model (Figure 6), which takes into account the users' need to control the data they share (Runtime Permissions). This means that forensic analysts in the future will probably handle cases where smartphone users have restricted data sharing among apps making the analysis harder than usual. Also, apps in the near future will probably be more personalized due to the new advancements and restrictions. Thus, generic (traditional) digital forensic models will probably fail to reveal the same amounts of data in the era of the new permission paradigm.

4. Other Open Source Operating Systems

Other open source platforms, such as the Mozilla Firefox OS and Tizen, follow different security and privacy models. They are both using Linux kernels (like Android) but they are also equipped with web runtime layers on top of them. This improvement allows developers to

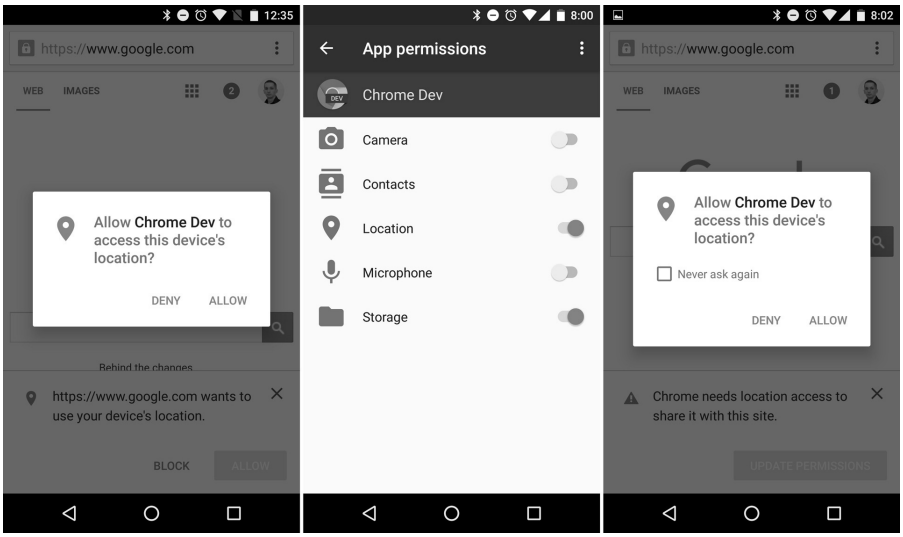

Figure 6. The New Runtime Permissions Model.

create apps (webapps) by using only web technologies (HTML5, CSS and Javascript).

Mozilla developed proprietary APIs for Firefox OS and the way it handles app permissions distinguishes them between hosted and packaged apps. The hosted apps can be downloaded from websites and the packaged apps are already installed on the device. The latter are divided in three sub-categories: web apps, privileged and certified apps: a) Web apps don't use the privileged or certified APIs, b) privileged apps make use of privileged APIs (they are distributed through the Firefox Marketplace) and c) certified apps (which are preinstalled) are able to access privileged and certified APIs. Privileged and certified apps have content security policies but all apps are required to invoke an installation method. This procedure validates the app and asks the user to approve the app installation. In other words, Firefox OS, depending on the app type (e.g. if it is certified or privileged), implicitly grants some of the permissions and then asks the user to approve other permissions (using prompts during run-time just like 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 divided in specific categories. The communication API for example provides functionality for Bluetooth control and messaging, it provides email services and

access to NFC device(s) 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 access control model. Tizen is still in its early days but the developers behind the project aim to build a multi-purpose OS able to serve mobile devices, wearables, in-vehicle infotainment systems and smart TVs. The Android's proliferation currently in the markets makes it an unlike scenario to seize a smartphone that runs such operating systems. However, in the future forensic analysts should be aware that these kinds of technologies might enter the market, because they are open source and can be installed in phones aiming underdeveloped or developing countries. Thus, research in the Digital Forensics field should aim to enhance our knowledge about these systems in order to be able in the future to handle cases where the basic source of evidence is a device running under the aforementioned OS or under other emerging OSs that might acquire a considerate portion of the market in the future (i.e. the Ubuntu Touch OS).

5. The Sixth Android Version

The advent of Android's version M (version 6.0) will probably change the way users interact with their apps considering the Runtime Permissions model, which was revealed at the M Developer Preview. The new permission model ensures that the developers should build their apps requesting permissions from the user only for a limited number of resources. Other permissions should be requested and granted by the user during runtime. The novel permission system will make the smartphone ecosystems unique. These advancements in data sharing among apps will probably change the way we perform forensic analyses because we might face devices, which restrict the access to specific resources. Hence, the data that the analysts will be able to find in the databases might be limited. On the other hand, users who are not privacy and security aware might find it useful to enjoy all features of the provided functionality from the installed apps and therefore allow access to all resources.

The adoption of the users to the new model and their reactions would be an interesting subject for research and analysis in the future. However, we have seen at the past that the adoption rates of the Android's OS newest versions show that their advent does not enforce all users to download and install them on their devices. A great portion of them prefer to use older versions, as we can see in Figure 7. The screenshot 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

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%

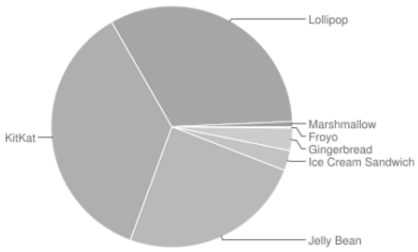

Figure 7. A Screenshot from the Developers Dashboard (January 2016).

Marshmallow edition. Thus, previous versions of the operating system will probably remain in the market for some years.

6. Related Work

Data privacy protection mechanisms are important for the new era of mobile computing, where data sharing can create risks. Small and medium sized enterprises for example seem to be more vulnerable from data leakage because they do not have access to advanced IT resources and capabilities [11], hence the BYOD models they implement might be less efficient. Several approaches have been proposed recently to protect personal mobile computing from limitless data sharing. MyShield [5] for example is a system that supplies anonymized data if requested by user and incorporates another privacy concept, namely Secure Circles which is a control mechanism that allows users to manage app access to sensitive data according to their level of trust.

Other approaches were focused only on location services [6] providing the chance to mobile device users to protect their privacy by adjusting the accuracy of their location in order to be able to use location based apps and at the same time protect their private data by using on-device or service-based obfuscation [12]. Moreover, authors in [8] suggest 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. [18] suggest that when ab-

stract location descriptions are included in privacy protection schemes, then location sharing is more possible to occur.

Henne et al. [13] proposed a crowd-based recommendation system for Android devices which allows users to configure accuracy for location data classifying apps in five precision levels. In addition, they claim that unskilled users will benefit from such approaches. Crowdsourcing for location based privacy settings is also used at [17]. Beresford et al. presented 'MockDroid' [7], a modified version of the Android OS, which works as XPrivacy and its main difference is that it basically feeds with empty resources apps that require access to specific data. This fact reduces functionality at some point but most of the apps work without any other problems. 'AppFence' [14] is another data protection mechanism that uses shadowing and exfiltration blocking on existing applications which aims to reduce side effects. According to its developers, it did not cause problems to 66% of the tested applications. Finally, Fisher et al. demonstrated that iOS users can be classified in three basic 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 some apps they trust [10].

7. Conclusion

To sum up, in this paper we discussed the variety of different ecosystems emerging from the fact that more advanced users tend to change (sometimes dramatically) the expected behavior of their phones. We also highlighted the existing variations in the AOSP's data privacy and security model and stressed that when the phone is rooted and obfuscation apps are installed, the analysis will probably provide false or limited evidence.

Furthermore, we mentioned the applied security models on other open source systems and presented an early estimation of the limitations that the current Android OS version will probably introduce. In this paper we did not refer to iOS devices because the permission system they use is different; the users can restrict data sharing and deny access to specific resources. Thus, user control is in higher standards compared to the aforementioned models. Forensic analysts should expect that in cases where an iOS device is involved, there is a possibility the exchanged resources among apps to be limited.

References

[1] P. Andriotis, G. Oikonomou and T. Tryfonas, Forensic analysis of wireless networking evidence of Android smartphones, *Proceedings*

of the Fourth IEEE International Workshop on Information Forensics and Security (WIFS'12),pp. 109–114, 2012.

[2] P. Andriotis, G. Oikonomou, T. Tryfonas and S. Li, Highlighting relationships of a smartphones social ecosystem in potentially large investigations, *IEEE Transactions on Cybernetics*, vol. PP (99), 2015.

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

[4] Android Developers, SensorEvent, (`developer.android.com/reference/android/hardware/SensorEvent.html#values`), 2016.

[5] R. Beede, D. Warbritton and R. Han, Myshield: Protecting mobile device data via security circles, *Tech. Rep. CU-CS-1091-12*, University of Colorado Boulder, 2012.

[6] 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 (7), pp. 679–694, 2011.

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

[8] A.J. 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.

[9] Caspase, Xposed Module Repository, (`repo.xposed.info/module/at.jclehner.appopsxposed`), 2015.

[10] D. Fisher, L. Dorner and D. Wagner, Short paper: 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.

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

[12] B. Henne, C. Kater and M. Brenner, Selective cloaking: Need-to-know for location-based apps, *Proceedings of the Eleventh IEEE*

Annual International Conference on Privacy, Security and Trust (PST), pp. 19–26, 2013.

[13] B. Henne, C. Kater and M. Smith, On usable location privacy for Android with crowd-recommendations, in *Trust and Trustworthy Computing*, T. Holz and S. Ioannidis (eds.), pp. 74–82, Springer International Publishing, 2014.

[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, *Proceedings of the Eighteenth ACM conference on Computer and communications security*, pp. 639–652, 2011.

[15] T. Kaiser, Google Removes "App Ops" Privacy Control Feature from Android 4.4.2, (`www.dailytech.com/Google+Removes+App+Ops+Privacy+Control+Feature+from+Android+442/article33936.htm`), 2013.

[16] T. Kondrat, Xposed Module Brings Back App Ops on Android 4.4.2 to Give Your Control of Your Application Permissions, (`www.xda-developers.com/xposed-module-brings-back-app-ops-to-android-4-4-2-and\-gives-your-control-of-your-application-permissions/`), 2013.

[17] J. Lin, S. Amini, J.I. Hong, N. Sadeh, J. Lindqvist and J. Zhang, Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing, *Proceedings of the Fourteenth ACM Conference on Ubiquitous Computing (UbiComp2012)*, pp. 501–510, 2012.

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

[19] 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.