Introduction

The permission model currently used for Android smartphones when downloading applications from Google PlayStore or external markets poses several security threats. Up to Android version 5 (Lollipop), users were not given an option to choose which permissions to grant upon application installation. Android v6 (Marshmallow) allows granting and revoking certain permissions upon installation, however, this model is not perfect, and even though it has been almost a year since Marshmallow was launched, it is only running on around 2.3% of Android devices [1].

The current “all or nothing” model forces users to either refrain from installing an application (no permissions granted) or to grant all permissions requested by an application. This can create problems since studies have shown that users tend to ignore the “grant permission” dialog since they have no choice except to avoid installing the app altogether [2]. The issue exists for inbuilt applications as well (most of which are classifiable as “bloatware” [3]), for example the Flashlight application inbuilt on devices running HTC’s Android based Sense UI, requests all permissions that can be granted to an application, and since the app is inbuilt there is no way to uninstall/disable it other than rooting the device.

Least Privilege

Applications on the Apple AppStore are subject to screening before being available for download [4]. After installation, iOS will prompt the user to approve permissions at run-time when they are first accessed. In Android, the permission-based security model requires applications to request permissions up-front before allowing the application to be installed. This process is the same for all permissions, regardless of privacy, data sensitivity or other factors that may differ according to the type of permission. Google uses the principle of “least privilege” for Android permissions, which allows a user to access what is required to complete the task, but not more than that [5].

Android implements two types of security mechanisms; one at system level and another at the Inter Component Communication (ICC) level. ICC security mechanisms build up on the foundation of the Linux kernel, and create a unique user identity for each application [6]. Developers are entrusted to include only permissions that are necessary, and not to misuse the security infrastructure, which generally does not happen, with research revealing that 33% of applications ask for permissions beyond what is required [7].

Data Availability after Uninstallation

Since application permissions once granted are not revoked even upon uninstallation of an application, the data collected through the permissions granted while the application was installed may still be accessible once the app is uninstalled. Upon uninstallation, the user identity belonging to the application is deleted, but the permissions allowed are not revoked, and data still exists as “orphans” without a unique identifier (or “parent”). These “orphans” may later be exploited by malware causing privacy breaches and leaking of sensitive data [8].

Capability Leaks

Applications can sometimes access permissions which are not requested at install time. Such violations of the permission-architecture to access data are referred to as “capability leaks” [9] [10]. A tool named Woodpecker, which analyzes each application to detect readability of permissions from unguarded interfaces, is frequently used in research in this domain [11]. Through woodpecker, two different types of capability leaks are identified; explicit leaks which find loopholes and access data without actually requesting permission and implicit leaks which let applications inherit permissions from another application. Other tools used to detect capability leaks include DroidChecker and IntentFuzzer [12].

Permission Creep

Some Android applications request permission that is not required for the execution of the application because of revenue generation models. Extra permissions may sometimes be requested in cases where developers have difficulty trying to align permission requests with the functionality required for the application, resulting in genuinely having to request extra permissions that seems unnecessary on analysis, but are mandatory for certain functions [13]. For example an update for the popular game Angry Birds caused controversy by requesting permission to send SMS messages, which is not part of the expected functionality of the application. However Rovio (the company behind Angry Birds) later explained that this is due to the payment methodology needed to purchase new levels, where an SMS message is sent to Rovio from the device to be billed later by the carrier [14] [15].

As such, some permission requests that seem unnecessary may be required for the revenue generation methodology of the application, since most “free” applications available on the PlayStore require in-app purchases for extra functionality. Some “free” and low cost applications may sell data to advertisers to generate revenue, without explicit permission from the user. Studies [16] have shown that over 50% of applications that request location access do so with the intent of sharing the information with advertisers for targeted marketing [17]. However, completely disallowing such requests would negatively impact the quality of applications available for Android since the revenue generation model would not survive [18].

Summary

The current Android permission model supports several methods of data misuse caused by capability leaks, permission creep, and data not being deleted after uninstallation of the application. Research has been carried out to propose methods to mitigate the risks caused by data leaks due to permission misuse or to stop applications accessing extra permissions without explicit authorization by the user. Further reading is required to analyze the solutions and alternate security architectures that have been proposed.

References

[1] Android Developers, "Android Developers: Dashboards," 2016. [Online]. Available: http://developer.android.com/about/dashboards/index.html. [Accessed March 2016].

[2] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner and K. Beznosov, "Android Permissions Remystified: A Field Study on Contextual Integrity," August 2015. [Online]. Available: http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/20994-sec15-paper-wijesekera.pdf. [Accessed March 2016].

[3] P. McDaniel, "Bloatware Comes to the Smartphone," June 2012. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.365.5363&rep=rep1&type=pdf. [Accessed March 2016].

[4] P. Gilbert, B. Gon-Chung, L. Cox and J. Jung, "Vision: Automated Security Validation of Mobile Apps at App Markets," 2010. [Online]. Available: http://www.appanalysis.org/jjung/jaeyeon-pub/appvalidation.pdf. [Accessed 2016].

[5] G. Robinson and G. Weir, "Understanding Android Security," 2015. [Online]. Available: https://pure.strath.ac.uk/portal/files/44777319/Robinson_Weir_ICGS2015_understanding_android_security.pdf. [Accessed March 2016].

[6] W. Enck, M. Ongtang and P. McDaniel, "Focus: Understanding Android Security," 2009. [Online]. Available: http://s3.amazonaws.com/academia.edu.documents/30867297/sp09.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1457954969&Signature=bmJDT7ks8xe7t2Lg%2FIUt%2BVXU4uU%3D&response-content-disposition=inline%3B%20filename%3DUnderstanding_Android_Security.pdf. [Accessed March 2016].

[7] A. P. Felt, E. Chin, S. Hanna, D. Song and D. Wagner, "Android Permission Demystified," 2011. [Online]. Available: http://fanfq-android-demo.googlecode.com/svn-history/r168/trunk/doc/android_permissions.pdf. [Accessed 2016].

[8] X. Zhang, K. Ying, Y. Aefer, Z. Qiu and W. Du, "Life after App Uninstallation: Are the Data Still Alive? Data Residue Attacks on Android," October 2015. [Online]. Available: https://xzhang35.expressions.syr.edu/wp-content/uploads/2015/10/android_data_residue.pdf. [Accessed March 2016].

[9] M. Grace, Y. Zhou, Z. Wang and X. Jiang, "Capability Leaks in Stock Android Smartphones," 2012. [Online]. Available: https://dl.packetstormsecurity.net/papers/general/NDSS12_WOODPECKER.pdf. [Accessed March 2016].

[10] M. Grace, Y. Zhou, Z. Wang and X. Jiang, "Detecting Capability Leaks in Android-based Smartphones," 2011. [Online]. Available: http://repository.lib.ncsu.edu/dr/bitstream/1840.4/4289/1/TR-2011-15.pdf. [Accessed March 2016].

[11] Y. Zhou, Z. Wang, W. Zhou and X. Jiang, "Hey, You, Get Off of My Market: Detecting Malicious Apps in Official and Alternate Android Markets," 2012. [Online]. Available: http://www.csd.uoc.gr/~hy558/papers/mal_apps.pdf. [Accessed March 2016].

[12] K. Yang, J. Zhuge and Y. Wang, "IntentFuzzer: Detecting Capability Leaks of Android Applications," December 2014. [Online]. Available: http://netsec.ccert.edu.cn/duanhx/files/2010/12/ASIA-CCS-14-2014-Yang-11.pdf. [Accessed March 2016].

[13] T. Vaidas, L. F. Connor and N. Christin, "Curbing Android Permission Creep," 2011. [Online]. Available: https://www.andrew.cmu.edu/user/nicolasc/publications/VCC-W2SP11.pdf. [Accessed March 2016].

[14] A. Russakovskii, "Rovio Explains Why The SMS Permission Was Introduced In Angry Birds v1.5.1," February 2011. [Online]. Available: http://www.androidpolice.com/2011/02/06/rovio-explains-why-the-sms-permission-was-introduced-in-angry-birds-v1-5-1/. [Accessed March 2016].

[15] P. Nickinson, " Rovio Explains New Permissions in AngryBirds Seasons Update," Android Central, December 2012. [Online]. Available: http://www.androidcentral.com/rovio-explains-new-permissions-angry-birds-seasons-update. [Accessed March 2016].

[16] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," 2010. [Online]. Available: http://www.appanalysis.org/tdroid10.pdf. [Accessed March 2016].

[17] N. Saint, "50% Of Android Apps With Internet Access That Ask For Your Location Send It To Advertisers," Business Insider, October 2010. [Online]. Available: http://www.businessinsider.com/50-of-android-apps-that-ask-for-your-location-send-it-to-advertisers-2010-10. [Accessed March 2016].

[18] T. Moynihan, "Apps Snoop on your Location Way More Than You Think," Wired, March 2015. [Online]. Available: http://www.wired.com/2015/03/apps-snoop-location-way-think/. [Accessed March 2016].