University of Colombo School of Computing

SCS4124 - Final Year Project in Computer Science

Web of Trust for Better Privacy Protection in Android Applications

Researcher:
K.J.S. Fernando
Index Number:
12000434

Supervisor:
Dr. Kasun de Zoysa
Co-supervisor:
Mr. Primal Wijesekara

April, 2016

Preface

This document contains the Project Proposal to be submitted to the University of Colombo School of Computing as a requirement of the subject SCS4124: Final Year Project in Computer Science. It includes an outline of the research question, problem statement in brief, goals and objectives of undertaking the research, methodology to be followed, the scope and delimitations of the project and a tentative schedule for the completion of the project.

The intended purpose of preparing this document is to present it to the academic staff at the University of Colombo School of Computing so that they may determine whether the project should be approved as proposed, approved with modifications or not approved.

Signatures:

K.J.S. Fernando	Dr. Kasun de Zoysa
_____	_____
Signature	*Signature*
_____	_____
Date	*Date*

Contents

1

Research Question

1.1 Introduction

The purpose of this research is to develop a web of trust based implementation to address privacy issues in smartphones, specifically focused on the Android platform.

Android is an open source mobile operating system currently developed and maintained by Google Inc. It was first developed as an advanced operating system for digital cameras, by a team of engineers in Palo Alto, California, and was acquired by Google in 2005. The Android Open Source Project(AOSP) is the collective name for the Linux kernel, middleware components and applications which form the Android Operating System [1]. Analysis of global smartphone market share indicates that as at March 2016, Android with a market share of 60.99% is the leading mobile operating system with a lead of 29.23% over the next most popular operating system, iOS(market share 31.76%).[3]

Users of Android applications are expected to decide on how an application will be allowed to use data without a prior guideline to act as an indicator[4].Users are asked to make privacy decisions before they start using the application, at which time they are not equipped with enough knowledge to do so. Upto Android version 5.0(Lollipop) users were expected to grant permissions at install time. Permissions could not be selected or removed individually and choosing not to grant a permission resulted in the application download being canceled. Since version 6.0(Marshmallow) users are allowed to install applications granting selective privacy options, which can later be toggled depending on individual needs.

Similar situations with privacy on the internet has brought forth standards such as PGP(Pretty Good Privacy), the concepts of which include "Web of Trust". A Web of Trust is a cryptographic term for a security model where participants authenticate the identity of fellow users.The model in its simplest form is used by social networks including Facebook, LinkedIn and Google+ for user validation and networking[10]. In the context of Android security, there are research applying PGP principles to secure message passing, but not for Application permission privacy configuration[5].

1.2 Research Question

Android applications require users to approve a list of permissions that will be accessible by the app before installation. This is a shortcoming in an operating system with such a wide user base as users are not equipped with enough knowledge to make a decision, which leads to making uninformed decisions and compromising their own privacy later on since they have no guideline or benchmark available as an indication of how an application will use data once it is downloaded with the necessary permissions granted. Application ratings can be given by users on most marketplaces including Google PlayStore, but this is based on a host of factors which may or may not have taken privacy and security into consideration.

Due to uninformed privacy decisions taken by users when installing apps, both privacy and security are compromised. However security is primarily threatened through malware apps which access permissions without authority, whereas privacy is compromised through users unknowingly granting permissions for applications which then misuse these privileges. Therefore we will be concentrating on privacy violations that occur through permissions which have been granted by users, as there are many research currently focused on malicious code detection and improved security of Android applications.

Different types of interactions take place in this domain; users interact with applications to use the functionality and applications interact with the Operating System to access data(this is where permissions are called on-through the Linux Kernel where a unique user ID is created for each app upon installation). Further, to confirm the level of privacy provided by an app, users could interact with their peers to share the experience of whether the privacy provided by a particular application is adequate or not. Such interactions between applications and the operating system, applications and users, and users and users can be used to create a Web of Trust, focused on creating an improved permission architecture for Android applications and improving flexibility and control for users of the platform.

2

Problem Statement

2.1 Security and Privacy

Privacy and security, although related are different concepts. Privacy is subjective; the user can decide on how private they want their data to be.However security is objective; it is concerned with 'guarding' something that is universally accepted as confidential, such as password, credit card details, pin number etc.

Users have different needs with regard to privacy, which is why they should be allowed to make their own decisions. Therefore a centralized monitoring system which blocks each privacy infraction would not be ideal since Android does not have information to predict what each user wants beforehand. Privacy infractions are therefore more difficult to predict than security threats, since user preferences also have to be taken into account. Therefore an ideal solution would be to let the users make their own decisions, while providing enough information for them to do so. In the Android platform, users are allowed to make decisions upto a certain extent, but these decisions are usually uninformed since there is no indicator as to what level of privacy will be provided by an application beforehand.

2.2 Comparison of Application Approval Process

The leading mobile operating systems apart from Android are iOS by Apple, Windows Mobile and BlackBerry OS. Each of these have different processes and methodologies for developers to follow before submitting an application to the store. A comparison between application submission processes for these platforms shows that each has a centralized process for validation and/or verifying an application before it is made available for users to download. However these processes are usually in place for checking security or applications, and not privacy related issues.

To submit an application to iTunes, the marketplace for iOS applications, the developer is first required to create an App ID and a Distribution Provisioning Profile and then submit an application through iTunes connect with detailed information on

the app.Three different certificates have to be submitted along with the application; the Distribution Certificate, Push Notification Certificate and Mobile Provisioning Certificate. The app has to be submitted through the Publication Center, where a checklist of complying standards that need to be adhered to has to be filled in.The approval process on average takes six days to one week.[13]

Windows store applications also go through a centralized process before being released. A submission has to be created for each application with a checklist of information. Once the submission is complete and the application has been preprocessed without errors, it is submitted for certification through the Windows Certification Kit. The certification process focuses on three core areas; security tests, technical compliance tests and content compliance tests. The amount of time taken for an application to receive approval fluctuates based factors such as the code and logic complexity, visual content, rating of the developer, other applications in the queue etc. Applications which fail the certification process will be returned with a report indicating where the compliance standards were not met. Developers are allowed to resubmit applications following the same process.[6]

RIM(Research In Motion) BlackBerry requires developers to submit applications for a complex process of reviewing, testing and "readying for publication" before being awarded a Approved/Up For Sale rating. Apps with this rating can be submitted and released on the BlackBerry marketplace; BlackBerry App World.[16]

Android developers add applications to marketplaces including Google PlayStore, Amazon AppStore, GetJar, SlideMe and F-Droid, which can then be downloaded by users. The problem lies in there being no centralized security measurement for applications on such marketplaces. Developers are trusted to prepare an application for release and then release it through a marketplace, email or website.[8] The Android operating system imposes some security and privacy restrictions, including an install-time permission system, where each application declares what permissions it requires upon installation.[2] This can provide users with control over their privacy since the choice to cancel installation lies with the user.

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. [7] 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. [22] The issue exists for inbuilt applications as well (most of which are classifiable as bloatware [19]), for example the Flashlight application inbuilt on devices running HTCs 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.

2.3 Literature Review

2.3.1 Least Privilege

Applications on the Apple AppStore are subject to screening before being available for download [21]. 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 [11].

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 [26]. 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 [4].

2.3.2 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 [27].

This is a key issue that may be caused by users misunderstanding or choosing not to read permissions before installation, since the consequences can be irreversible in such situations.

2.3.3 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' [17] [18]. A tool named Woodpecker, which analyzes each application to detect readability of permissions from unguarded interfaces, is frequently used in research in this domain [28]. 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 [15].

2.3.4 Permission Creep

Some Android applications request permission that is not required for the core functionality 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 [25]. 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 [23] [9].

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 [14] 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 [24]. However, completely disallowing such requests would negatively impact the quality of applications available for Android since the revenue generation model would not survive. [20] In an ideal situation a user should be informed as to why an application is requesting a particular permission; as part of its core functionality, secondary functionality, as a method of revenue generation or any other usage for a permission to be requested. Research has shown that people tend to base their decisions on the reason behind applications data access[12]. However this is not possible with the current model since the level of information made available to users regarding application permission requests is decided on by the developer.

2.4 Problem Definition in Summary

The current permission model used in Android applications does not include a centralized process for certification or testing before an app is released, and instead relies on the developer to act responsibly and request the least number of permissions that are necessary. However in most cases the privileges given to developers are misused causing capability leaks, permission creep and some other issues, the consequences of which cannot be reversed even after app uninstallation. The simplest way to stop the occurrence of privacy breaches caused by the permission model would be to let the user have more control over permissions to be granted. The problem in a nutshell is that at present users are asked to make uninformed decisions, that can have wide reaching consequences, which they are not equipped to answer.

3

Goal and Objectives

3.1 Goal

A preliminary step of letting users have more control over granting permissions would be to provide a way for users to determine whether the application keeps to the expected permissions or accesses more than it is supposed to. This is primarily a security issue since apps access permissions without authorization. However some accesses that violate user privacy occur through permissions that *have* been granted by the user. There are many applications to detect unauthorized access and malicious code, but privacy violations occur through accesses that are not unauthorized and are hence not monitored through these applications.Therefore through this research, the expectation is to determine a method through which users will be informed about the trustworthiness of an application based on peer experience with the application.

3.2 Objectives

- Study applications access of permissions
- Determine a simple scale to rate apps based on privacy breaches that are authorized
- Determine a method to connect a network of users and assign 'trust' ratings to these users
- Define threshold values for the 'Web of Trust' including reach of the web, stopping point etc
- Develop a function to compute the final rating based on the application rating and user's trust rating
- Present the information in a meaningful way for the user about to install the application

4

Methodology

4.1 Study of Related Work

A study of related work in the domain will be conducted. This will focus mainly on determining a method to connect a group of users and rank them with a trust rating, communicating safely between peer devices, determining thresholds for the web of trust in this domain; the reach, how far to check links based on users etc.

4.1.1 The Web of Trust

A PKI(Public Key Infrastructure) would involve a centralized server where the application ratings would have to be stored based on a pre-determined rating which would have to be computed. However applying a single rating for applications would not be practical since most applications tend to have frequent updates, and in some cases get updated in the background without the user being aware. Hence it would be impractical to determine a static rating for how privacy-conscious an application is.

However with a Web of Trust based system, the application rating would be updated in real time based on user experience. Therefore an update which results in more security threats or breaches of privacy would automatically be assigned a lower rating by the system. The final rating would be determined based on the experiences of a users 'trusted' group of peers with the application and hence is more relevant when considering factors such as localization, reasons for use etc. which may differ from user to user and affect security and privacy standards. This will be based on the assumption that those assigned as trusted peers by a user are more likely to have the same purposes for using an application, and same customized settings. Another advantage is that such 'trusted peers' tend to have the same attitude towards privacy and hence the ratings will be more relevant. For example a trusted group of users who use their smartphones for business purposes and a group of students who use their phones mainly for light gaming, social networking and communication will have different expectations about how 'private' data should be.

Threshold values and parameters for the Web of Trust will be determined through experiments and studying current research.

4.1.2 Safe Communication Between Devices

Research in the domain will be studied to determine a method for safe communication between Android smartphones when users need to access a peer rating for an application before installation.

4.1.3 Peer Rating System

Current research on existing methods for assigning a rating for a user based on a trust ranking will be perused to decide on a best practice for use withing this research.

4.2 Evaluation

The evaluation phase will consist of determining how successful and practically applicable the proposed changes to the Android permission system are. Evaluation will be done based on test cases which will then be mathematically analyzed to determine suitability. Further methodology in which the evaluation phase will be conducted will be decided on during the research period.

5

Scope and Delimitation

5.1 Scope

The research will focus on Android smartphones with access to the Google PlayStore. Monitoring will be carried out for a selected number of applications from the PlayStore, and not from apps installed using third party websites, promotional links on social media or email, external markets such as the Amazon Appstore, F-Droid etc. The scale to determine trustworthiness of an application will be assigned simply based on factors such as frequency of uninstallation, whether uninstallation happened after a major privacy breach etc. Depending on resource constraints this rating may be equalized to a user assigned rating for an application as well. The information provided to the user will be limited and focused on the core data needed to determine whether an application can be installed without trust issues.

The research will focus on privacy violations that occur through authorized access. Therefore applications which ask for X permission and then use X to cause some privacy violation will be the focus of this research.

5.2 Delimitation

The period assigned for the research is limited, and hence the amount of work will be limited by time constraints. Since the purpose is to propose a framework through which information flow to users before installing applications can be improved and not to provide a complete practical implementation of the system, some features will not be included during the research period for clarity and ease of understanding.

We concentrate the research on the Google PlayStore since different markets follow different methodologies, and monitoring applications downloaded from various sources can lead to malware and security breaches. For the purpose of this research we will assume that the option to install applications from 'Unknown Sources' is disabled.The reasoning for concentrating on privacy is that there are existing research on security violations which occur through an application requesting only permissions X and Y upon

installation and then using a loophole to access permission Z without authorization, typically malware and spyware. Further, the information provided will be limited by generated data during the course of analysis, and the ultimate decision on whether or not to install an application will be left to the user.

6

Tentative Timeline

Task	Expected Date of Completion
Preliminary meetings with the supervisor to discuss the research	Completed
Confirm supervisor details	Completed
Confirm tentative topic	Completed
Preliminary literature review	Completed
Submit Project Proposal	28/04/2016
Proposal Defense	15/05/2016
Submit introductory chapter of thesis	29/05/2016
Determine scale to assign ratings for monitored apps	07/06/2016
Submit outline of chapter two-literature review	29/06/2016
Determine how to assign user trust ratings based on connectivity	30/06/2016
Submit activity plan for semester two	03/07/2016
Define threshold value for the 'Web of Trust'	31/08/2016
Submit extended proposal	11/09/2016
Interim defense	25/09/2016
Develop function to compute final rating and present information to users	30/09/2016
Detailed outline of thesis	06/11/2016
Evaluation	15/11/2016
Submission of draft thesis	27/11/2016
Submission of final thesis	25/12/2016
Final defense	29/01/2016

Table 6.1: Tentative timeline

7

Personal Statement

As an undergraduate in the final year of a Computer Science degree program, I would like improve my knowledge through contributing to the wide world of research. I am motivated to conduct my research on an area which has interested me for some time; Android phones are objects which are used daily by many people and sometimes they are unaware of the dangers that may be posed by security or privacy breaches on such devices. I believe that the fourth year research project provides an excellent opportunity to use the knowledge accumulated throughout undergraduate life in collaboration with advisers who are more experienced and have more knowledge in this domain.

Bibliography

[1] The android open source project. https://source.android.com/.

[2] Patterns: Permissions. https://www.google.com/design/spec/patterns/permissions.html#.

[3] Market share statistics for internet technologies; mobile/tablet operating system market share. https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1, 2016.

[4] Steve Hanna Dawn Song David Wagner Adrienne Porter Felt, Erika Chin. Android permissions demystified. https://www.eecs.berkeley.edu/~daw/papers/androidperm-ccs11.pdf.

[5] Mario Tejedor-Gonzalez Benjamin Aziz. An android pgp manager: Towards bridging end-user cryptography to smart phones. http://eprints.port.ac.uk/9718/1/IJS_Aziz.pdf, 2012.

[6] Windows Dev Center. Msdn, the app certification process. https://msdn.microsoft.com/en-us/windows/uwp/publish/the-app-certification-process.

[7] Android Developers. Android developers: Dashboards. http://developer.android.com/about/dashboards/index.html.

[8] Android Developers. Publishing overview. http://developer.android.com/tools/publishing/publishing_overview.html.

[9] B.G. Chun L. P. Cox J. Jung P. McDaniel A. N. Sheth Enck, P. Gilbert. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. http://www.appanalysis.org/tdroid10.pdf, 2014.

[10] Patrick Feisthammel. Pgp: Explanation of the web of trust of pgp. https://www.rubin.ch/pgp/weboftrust.en.html, 2004.

[11] G. Weir G. Robinson. Understanding android security. https://pure.strath.ac.uk/portal/files/44777319/Robinson_Weir_ICGS2015_understanding_android_security.pdf, 2015.

[12] Jialiu Lin Bin Liu Norman Sadeh Jason I. Hong. Modeling users mobile app privacy preferences: Restoring usability in a sea of permission settings. https://www.usenix.org/system/files/conference/soups2014/soups14-paper-lin.pdf.

[13] Apple Inc. ios developer library-app distribution guide. https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/SubmittingYourApp/SubmittingYourApp.html.

[14] J. Zhuge K. Yang and Y. Wang. Intentfuzzer: Detecting capability leaks of android applications. http://netsec.ccert.edu.cn/duanhx/files/2010/12/ASIA-CCS-14-2014-Yang-11.pdf, 2010.

[15] Y. Wang K. Yang, J. Zhuge. Intentfuzzer: Detecting capability leaks of android applications. http://netsec.ccert.edu.cn/duanhx/files/2010/12/ASIA-CCS-14-2014-Yang-11.pdf, 2014.

[16] Alex Kinsella. How to submit your apps for blackberry 10. http://devblog.blackberry.com/2012/10/submit-apps-blackberry-10/l.

[17] Z. Wang X. Jiang M. Grace, Y. Zhou. Capability leaks in stock android smartphones. https://dl.packetstormsecurity.net/papers/general/NDSS12_WOODPECKER.pdf, 2012.

[18] Z. Wang X. Jiang M. Grace, Y. Zhou. Detecting capability leaks in android-based smartphones. http://repository.lib.ncsu.edu/dr/bitstream/1840.4/4289/1/TR-2011-15.pdf, 2015.

[19] P. McDaniel. Bloatware comes to the smartphone. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.365.5363&rep=rep1&type=pdf, 2012.

[20] T. Moynihan. Wired: Apps snoop on your location way more than you think. http://www.wired.com/2015/03/apps-snoop-location-way-think/, 2015.

[21] L. Cox J. Jung P. Gilbert, B. GonChung. Vision: Automated security validation of mobile apps at app markets. http://www.appanalysis.org/jjung/jaeyeonpub/appvalidation.pdf, 2010.

[22] A. Hosseini S. Egelman D. Wagner K. Beznosov P. Wijesekera, A. Baokar. Android permissions remystified: A field study on contextual integrity. http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/20994-sec15-paper-wijesekera.pdf, 2015.

[23] A. Russakovskii. Rovio explains why the sms permission was introduced in angry birds v1.5.1. http://www.androidpolice.com/2011/02/06/rovio-explainswhy-the-sms-permission-was-introduced-in-angry-birds-v1-5-1/, 2011.

[24] N. Saint. 50send it to advertisers. http://www.businessinsider.com/50-of-android-apps-that-ask-for-your-location-send-it-toadvertisers-2010-10, 2010.

[25] N. Christin T. Vaidas, L. F. Connor. Curbing android permission creep. https://www.andrew.cmu.edu/user/nicolasc/publications/VCC-W2SP11.pdf, 2011.

[26] P. McDaniel W. Enck, M. Ongtang. Focus: Understanding android security. http://s3.amazonaws.com/academia.edu.documents/30867297/sp09.pdf?AWSAccessKeyId=AKIAJ56TQJRTWSMTNPEA&Expires=1457954969&Signature=bmJDT7ks8xe7t2Lg%2FIUt%2BVXU4uU%3D&response-contentdisposition=inline%3B%20filename%3DUnderstanding_Android_Security.pdf, 2009.

[27] Y. Aefer Z. Qiu W. Du X. Zhang, K. Ying. Life after app uninstallation: Are the data still alive? data residue attacks on android. https://xzhang35.expressions.syr.edu/wp-content/uploads/2015/10/android_data_residue.pdf, 2015.

[28] W. Zhou X. Jiang Y. Zhou, Z. Wang. Hey, you, get off of my market: Detecting malicious apps in official and alternate android markets. http://www.csd.uoc.gr/~hy558/papers/mal_apps.pdf, 2012.