

**Android Smartphone Apps: Privacy
Concerns of Unregulated Permissions on
Social and Psychological Contracts.**

by

Kathryn C. Carstens

University of Portsmouth

AT&T Services Inc.

This thesis is submitted in partial fulfilment of the requirements for the award
of Doctor of Philosophy of the University of Portsmouth

June 2018

Abstract

This research describes how security was being implemented in the smartphone marketplace, specifically on Android smartphones.

The initial work concentrated on security and antivirus app permissions and the APIs that were called. The gap between permissions and functionality was examined.

The first stage involved the antivirus apps that were available in 2011. All 22 free and commercial apps were compared and investigated to determine if there was any relationship between the functions and permissions requested between the two variants. A process tool was developed to extract and analyze the apps.

Stage two, in 2015, consisted of an update of the earlier 2011 investigation and was performed to determine the maturity of antivirus apps over the 4 years. All 67 apps in 2015 were compared to the apps from 2011 and the changes between the apps were evaluated. There were some tools available that could assist in this investigation and the extraction and an automated analysis method was developed called Permission Extraction and Method Process (P.E.M.P.). This reduced the extraction and evaluation processing times from 10 hours for 20 apps to less than 30 minutes. Subsequent development has reduced the time further.

In Stage 3, the research moved from analysis of security apps to analyzing 60 free Children's apps. As the market place had evolved to supplying apps with adware or in-app purchases rather than offering paid apps, 20 of the top free game apps for each age group; 0-5 years, 6-9 years and over 9 years. The research concentrated initially on the evaluation of privacy and security of children with the apps installed and if there were differences between the permissions requested in the different age groups.

Stage 4 of the research developed and created a model of the impact of social and psychological contracts through the installation and use of the apps. In addition, this thesis makes contribution of a model for the comparison of an app to evaluate the user's expectation of privacy and if the app is fulfilling the social contract between the user, developer and marketplace owner.

Keywords: Security, Android, Smartphone, Privacy, Social Contracts, Antivirus, Children, Psychological Contracts.

Declaration

Whilst registered as a candidate for the above degree, I have not been registered for any other research award. The results and conclusions embodied in this thesis are the work of the named candidate and have not been submitted for any other academic award.

This thesis has a word count of 43,015 (excluding ancillary data).

Copyright

Copyright © 2017 Kathryn Casey Carstens. All rights reserved.

The copyright of this thesis rests with the Author. Copies (by any means) either in full, or of extracts, may not be made without prior written consent from the Author.

Abbreviations

These are the abbreviations used in the thesis.

Apps/apps	Android applications
AV	Antivirus
AV_Perm(s)	Antivirus permission(s)
Base_PI	Base-line Privacy Impact
Base_PI_Perm	Baseline Privacy Impact Permissions
GDPR	General Data Protection Regulation
OS	Operating system
P.E.M.P.	Permission Extraction Method and Process
PI	Privacy Impact
PI_gauge	Privacy Impact Framework Gauge
PI_Perm(s)	Privacy Impact Permission(s)
PI_perms	Privacy Impact permissions

Table of Contents

Abstract	ii
Declaration	iv
Copyright	v
Abbreviations	vi
Table of Contents	vii
List of Tables	xi
List of Figures	xiii
Publications	xvi
Chapter 1. Introduction.....	1
1.1 Mobile Devices.....	10
1.2 Online Crime.....	11
1.3 Malware Growth.....	13
1.3.1 Malware Examples	15
1.4 Anti-Malware Protection.....	17
1.4.1 Antivirus Functions	18
1.4.2 Antivirus on Mobile Devices	19
Chapter 2. Literature Review	21
Chapter 3. Evolution of Research Design.....	27
Chapter 4. Antivirus and Anti-Privacy (2011-2012)	30
4.1 Antivirus Functions and Permissions	30
4.1.1 Real-time monitoring	32
4.1.2 Malware Removal	32
4.1.3 Scanning	33
4.1.4 Update Signature Definitions	33
4.2 Antivirus Verification Method	34
4.3 Anti-privacy Functions and Permissions.....	39
4.3.1 Anti-privacy Permissions	39
4.4 Antivirus Apps Permission Analysis	43
4.5 Selection Criteria and Sample Selection.....	44
4.6 Antivirus Functions	45
4.7 Anti-privacy Permissions Requested	47
Chapter 5. Preparing the Test Environment	50
5.1 Preparing the Test PC	50
5.1.1 Software Environment	51
5.1.2 Installing the app onto a Clean Device	52
5.2 Rooting an Android Smartphone.....	53

5.2.1	Root the Device	54
5.2.2	Method 1 - Using the Setup Utility	54
5.2.3	Method 2 - Using a PC	55
5.2.4	Custom Recovery Image	55
5.3	Upgrading to Android Version 2.2 (Froyo).....	56
5.3.1	Upgrading the Operating System	57
5.3.2	Security Implications	58
Chapter 6.	Analysis of Antivirus Apps.....	60
6.1	app Status in 2011.....	60
6.2	Android Antivirus Apps in 2011.....	61
6.2.1	Investigation Method in 2011	63
6.2.2	Obtaining a Commercial app for Testing Without Incurring a Cost.	65
6.2.3	Selected Security and Antivirus Developers	66
6.2.4	Comparative Analysis Results	73
6.2.5	Review Program Source	82
6.2.6	Efficacy of Free Antivirus Apps	85
6.2.7	Results	88
6.3	Android Apps in 2015.....	89
6.4	Android Antivirus Apps in 2015.....	93
6.4.1	Investigation Method - 2015	95
6.4.2	2015 Security and Antivirus Apps	95
6.4.3	Results	104
6.5	Commercial Testers Results.....	105
6.6	Comparison of 2011 and 2015 Antivirus Apps.....	106
6.7	Conclusion.....	108
Chapter 7.	Permission Extraction Method and Process (P.E.M.P.)	110
7.1	Generic PEMP	114
7.2	Generic PEMP Process and Guidance	115
7.3	Initial 2011 Method	118
7.3.1	Tools	118
7.3.2	The Phases of PEMP	119
7.4	Generic Method (2015).....	126
7.5	Tools	126
7.6	APK Downloader	126
7.7	Extraction and Processing Tools	128
7.7.1	Process	129
7.7.2	Additional Actions	138
7.8	Conclusion.....	140
Chapter 8.	Analysis of 2015 Children's Apps	142

8.1	Introduction.....	142
8.2	Motivation	143
8.3	Method	144
8.4	Permissions that Affect User Privacy.	146
8.4.1	Apps for Children aged 0-5 years	146
8.4.2	Apps for Children aged 6-8 years	149
8.4.3	Apps for Children aged over 9 years	152
8.5	Results	155
8.6	Conclusion.....	158
Chapter 9. Privacy and Social and Psychological Contracts		161
9.1.1	Apple’s Privacy Terms and Conditions	164
9.1.2	Google’s Privacy Terms and Conditions	166
9.1.3	Protection Normal	167
9.2	Social Contract Obligations.....	169
9.3	Psychological and Implied Contracts.....	170
9.4	GDPR – EU Privacy Regulation	175
9.5	Privacy Impact of Location Trackers	178
9.6	Spyware	182
9.6.1	Mobistealth	182
9.6.2	FlexiSPY’s Android Spy App	182
9.6.3	Android Monitoring App	183
9.7	Big Data.....	183
Chapter 10. Research Review.....		185
10.1	Overview and discussion.....	185
10.2	Differences between 2011 and 2015 Research	188
10.2.1	Initial Research in 2011	188
10.2.2	Subsequent Research in 2015	191
10.3	Guidance for Regulators.....	194
Chapter 11. Contribution.....		196
11.1	Privacy Impact Framework Model Evolution.....	197
11.1.1	The Antivirus Efficacy Gauge	197
11.1.2	Privacy Impact Analysis	205
11.1.3	Antivirus Privacy Impact	207
11.1.4	Protection_Normal Privacy Impact	209
11.2	Privacy Impact Framework Model	212
11.2.1	Privacy Impact of Children’s apps	216
11.2.2	Antivirus app Privacy Impact 2011 vs 2015	220
11.3	Summary.....	223
Chapter 12. Addendum.....		229

12.1.1	Permission Control	230
12.1.2	RA Remover	234
References		236
Appendix A Detailed Permission tables		246
A.1.	Antivirus Apps Analysis Input Tables	246
A.2.	Analysis of Children’s Apps Input Tables	275
A.3.	Protection Normal Input Table	281
A.4.	Detailed Permissions of Antivirus apps in the study	282
Appendix B Android Operating System		330
Appendix C Antivirus Function testing		336
Appendix D Evolution of Android Malware		344
Appendix E UPR16		350

List of Tables

Table 1-1 Worldwide Smartphone market, in 2010	13
Table 1-2 Android Malware release 2010 to 2011	16
Table 4-1 Basic Antivirus Features	31
Table 4-2 Primary anti-privacy permissions and their activity and privacy rating	42
Table 4-3 Products and stated Antivirus Features	46
Table 6-1 The list of companies that provide the security apps, grouped by version type.	65
Table 6-2 User ratings for the six suppliers	77
Table 6-3 Rating of app by app type (free or commercial)	78
Table 6-4 Correlations of features and user rating	82
Table 6-5 Comparison of program sizes of the packages as downloaded 27/05/2011	84
Table 6-6 Antivirus apps and their described features	86
Table 6-7 Antivirus function testing summary	87
Table 6-8 Smartphone OS market share growth	89
Table 6-9 Antivirus apps that requested the most Antivirus permissions	103
Table 6-10 Antivirus apps that requested the most Anti Privacy permissions	103
Table 7-1: Worldwide Smartphone market	120
Table 8-1 Age group 0-5 apps requesting anti-privacy permissions.	149
Table 8-2 Apps requesting anti-privacy permissions for 6-8 year group.	151
Table 8-3 Apps that have requested anti privacy permissions	154
Table 9-1 Summary of contract types	174
Table 11-1 Antivirus Products Efficacy in 2011 and 2015	203
Table 11-2 Permissions required for Antivirus function	208
Table 11-3 Base-line permissions and their privacy rating	210
Table 11-4 Resultant list of permissions to perform basic Antivirus function	213
Table 11-5 The main 11 Privacy Impact permissions	216
Table 11-6 Privacy Impact changes for antivirus apps 2011 vs 2015	221
Table A-1 Security Applications on Androlib Marketplace as at 28/02/2011	247
Table A-2 Companies with free and commercial versions of Antivirus apps	249
Table A-3 Comparison of features of Antivirus products in the study	251
Table A-4 Android permissions requested by each app.	252
Table A-5 Details of non-android permissions requested	257
Table A-6 Android Antivirus apps in 2011 and 2015	259
Table A-7 List of Security and Antivirus apps in 2015	261
Table A-8 Antivirus apps in 2015 and their properties	268
Table A-9 Apps that requested “old” permissions.	272

Table A-10 Developer and their Antivirus apps available in 2011 and 2015	273
Table A-11 Ranking order and details of top 20 apps in the 0-5 age group by app name	275
Table A-12 Age group 6-8 top 20 apps selected for the study.	277
Table A-13 Apps selected for this study with number of permissions requested.	279
Table A-14 Permissions classified as Protection_Normal in Android V6.0	281
Table A-15 Set one consisting of 12 Apps	284
Table A-16 Set two consisting of 14 Apps	293
Table A-17 Set three consisting of 15 Apps	303
Table A-18 Set four consisting of 13 Apps	312
Table A-19 Set five consisting of 13 Apps	321

List of Figures

Figure 3-1 App Download and Extraction process	28
Figure 4-1 Flowchart illustrating the initial app installation and scan function	35
Figure 4-2 Flowchart illustrating the malware detection.	37
Figure 4-3 Flowchart of update of signature file.	38
Figure 4-4 Permission types: No privacy issues and two types of Antiprivacy concerns	40
Figure 4-5 Investigation method flowchart	44
Figure 4-6 Permissions by security type for each app (2012)	47
Figure 4-7 Number of Antivirus designated permissions requested by each App	49
Figure 6-1 Android permissions requested by each app.	74
Figure 6-2 The figure shows the total permissions requested by each Antivirus app.	74
Figure 6-3 Application Features and Android Permissions	75
Figure 6-4 Features, permissions and ratings for each product	75
Figure 6-5 The rating of the app by type.	79
Figure 6-6 Cluster analysis of relationship between Features and Total Requested Permissions	80
Figure 6-7 Relationship between features and user ratings.	81
Figure 6-8 Distribution of Android versions as at 5 th September 2016	91
Figure 6-9 Top Ten Mobile Apps in the U.S. for 2017	92
Figure 6-10 Most essential apps according to millenials.	93
Figure 6-11 The total permissions requested by the 2015 Antivirus apps.	97
Figure 6-12 Permissions requested by Type (Antivirus, Anti Privacy or Neither)	99
Figure 6-13 Antivirus permissions requested by apps	100
Figure 6-14 Anti Privacy permissions requested by apps	101
Figure 6-15 Antivirus and Anti Privacy permissions requested by the Antivirus apps.	102
Figure 6-16 Comparison of permissions of 2011 apps still available in 2015	107
Figure 6-17 The features and requested permissions of Free and Commercial apps in 2011	107
Figure 7-1: Flowchart illustrating the overall method for extraction of permissions	116
Figure 7-2: Permissions requested by Free and Commercial Antivirus apps in 2011	124
Figure 7-3: Cluster analysis of the relationship between features and total permissions requested	125
Figure 7-4: Chrome APKdownloader plugin installation notes	128
Figure 7-5 Analysis process flow - simplified.	129

Figure 7-6: Example python script to create the batch file entries	133
Figure 7-7 Sample Log from the APKTool call	134
Figure 7-8: Sample Manifest file extract	135
Figure 7-9 Sample contents from a manifest.csv file	136
Figure 7-10: Python code to compare app permissions to a master permission file	137
Figure 7-11: Snapshot of Permission Database entries	138
Figure 7-12: Python code to compare permissions of different versions of Android OS and mark the origin.	139
Figure 8-1 Frequency of the requested permissions in the 0-5 age group apps	147
Figure 8-2 Permissions requested by each of the studied apps in the 0-5 years age group	148
Figure 8-3 Frequency of the requested permissions for the 20 apps in the 6-8 age group.	150
Figure 8-4 Permissions requested in apps for children in the 6-8 years age group	151
Figure 8-5 Ages 9+ app permission frequency	152
Figure 8-6 Permissions requested in apps for children in the 9+ age group.	153
Figure 8-7 Number of permissions by age category	155
Figure 8-8 Frequency of permissions requested in each age group.	156
Figure 8-9 Number of permission requested by age group	158
Figure 8-10 Requested antiprivacy permissions by app	159
Figure 9-1 Development of a psychological contract	172
Figure 9-2 Development of an implied contract	173
Figure 9-3 Graphical representation of Über permission requests.	180
Figure 10-1 A summary of regulatory control	195
Figure 11-1 Initial Design using the Goldilocks Method	198
Figure 11-2 Framework using antivirus as a base.	198
Figure 11-3 Initial antivirus framework as a pie chart with exploding slice	199
Figure 11-4 Antivirus framework guage for Aegislab apps	200
Figure 11-5 Results of Bluepoint antivirus apps comparison	201
Figure 11-6 Lookout antivirus apps comparison	202
Figure 11-7 Base-line Privacy Impact Status using the Privacy Impact Framework Model	211
Figure 11-8 Privacy Impact Framework Model for Antivirus Function	214
Figure 11-9 Privacy Impact analysis of a free app with it's commercial version	215
Figure 11-10 Childrens apps 0-5 age group privacy impact	217
Figure 11-11 Childrens apps 6-8 age group privacy impact	218
Figure 11-12 Childrens apps 9+ age group privacy impact	219
Figure B-0-1 Android Architecture (2011)	331
Figure B-0-2 Android Internals for API Level 9	332

Publications

The research contains research suitable as input for four papers:

A method paper based on the P.E.M.P process. This will be submitted to the ACM Journal TOMPECS - ACM Transactions on Modelling and Performance Evaluation of Computing Systems (TOMPECS) for peer review.

An analysis paper on Children's apps, how children are tracked and what protection is in place to protect the child. The IEEE publication ICSSP is most suitable for this research.

A discussion paper on the management of the apps on the Google Play store and adherence to GDPR and how the apps will be policed.

The fourth paper describes the social and psychological implications of the "accept all" that the user is obliged to agree to, to have access to the app and who the contract is with, Google or the developer? This paper would be suitable for the "Psychology Now" publication.

Chapter 1. Introduction

This thesis investigates security and privacy on mobile devices. From initial investigation into device protection the thesis analyses people protection and recommends guidelines for Regulators, Marketplaces and developers to ensure that user information is not abused. The thesis concludes with framework, initially created for measuring the efficacy of antivirus apps and subsequently updated to measure and display the privacy status of the app and its impact on the privacy of the user as related to Social and Physiological contracts.

The initial null hypothesis of the research was that antivirus apps for Android mobile devices was not effective. This was in response to the hypothesis that Security apps protected the user.

As a security professional, the author was concerned that security products were being marketed to mobile users that were not fulfilling the function of protecting the user from malware.

The purpose of the study was to analyse existing security and antivirus products to determine their effectiveness in protecting the user. What vulnerabilities or gaps exist in the protection and who should be responsible for the protection of the user and what the user expects from the product.

The questions to be answered were; are security and antivirus products protecting the user, what are the shortfalls in the protection and how can it be improved? With an additional question related to user privacy, does the security app introduce vulnerabilities onto the device and make the user or their data insecure?

Introduction

The lack of available protection meant that the average user was open to attack, vulnerabilities on their device were not being fixed, updates to their phones operating system were not occurring on a regular basis to resolve and close those vulnerabilities. As the computing power of these devices increased owners of these devices started to use them in place of PCs (laptops and desktops). The protection of these devices was in a similar state to the PC market place in the early 2000's.

The aim of the research was to provide users with the knowledge to protect themselves, primarily by providing the user with a simple snapshot of their security status.

As vulnerabilities to the user's data was detected the research scope increased to incorporate privacy analysis and therefore provide the user with a snapshot of the impact to the user's privacy whilst using an app.

The growth of mobile devices and the increase in their capabilities has meant that smartphones shipments in 2012 were almost 3 times higher than shipments of Notebook PCs, 694.8 million units as compared to 215.7 million. ("Mobile device market to reach 2.6 billion units by 2016 | Canalys," 2013).

Previous research concentrated on the effects and workings of the apps in general, mainly consisting of the API calls made by the apps, geo-tracking of the user, identifying and testing malicious apps and the use of permissions to identify these malicious apps.

The geo-tracking research performed by Balakrishnan et al on real-time privacy monitoring (Balakrishnan, Nayak, Dhar, & Kaul, 2009), concentrated on permissions, whilst Gibler et al reviewed potential privacy leaks (Gibler, Crussell, Erickson, & Chen, 2012).

Introduction

Identification of malicious intent focussed on the basic mobile device functions; accessing private information, calling or texting premium numbers without the user's knowledge, but did not concentrate on specific types or genres of apps. Results of the research was presented by summarising the number of apps that could perform these functions but again did not detail this by genre. The malicious apps were also immature, the analysis of malware, for example Trojans, installed through phishing showed how data was captured and with the antivirus market being so immature, there was no adequate protection for the user.

Spyware was also prolific, although it did require physical access to the device for installation. The spyware was also able to hide itself on the mobile to avoid detection by the user. Existing antivirus products had difficulty in detecting these hidden programs. Additional security vulnerabilities were introduced by the user through "rooting"¹ the device, which facilitated the installation of malware.

Security protection was limited and many of the main security companies had not started releasing security products for mobiles. Android devices were particularly vulnerable to malware due to the open source nature of the operating System. Eventually with the growth of mobile forensics and the alignment to digital forensics researchers were able to evaluate the forensic data to review security holes in the android operating systems.

Academic research continued concentrating on the devices and the API calls, but privacy research lagged physical and software research. The main privacy issues investigated related to wireless security. Brunk (Brunk, 2002) conducted

¹ Rooting the device permitted the user to install apps from sites other than Google.

Introduction

a detailed examination on 133 privacy tools and services and using analytical techniques created a framework which describes a “privacy space”.

The main source of understanding privacy was the article by Brunk in 2002. He switched the perspective of privacy from the viewpoint of threats and intrusions to the persons perspective of how their privacy was being invaded. He analysed a variety of packages freeware, shareware, and other solutions he was investigating this on the PC or desktop arena.

Tsavli (Tsavli, Efraimidis, Katos, & Mitrou, 2015) used Brunk’s work as a base to explore privacy concerns of mobile devices, which apps were storing data about the user, and if these could be used to detect trends in the user usage. The paper explored dataflow of users’ personal information and a data taxonomy was proposed.

A major hole in research was how security changes due to multiple external influences and how corporations should be more responsible was not investigated and research concentrated on the detection test and notification of the malware for creating virus signatures. However, many of the initial antivirus apps tested did not have access to or use a virus signature database, which made malware detection very limited.

The objective was to test all security apps with an antivirus component and determine their effectiveness. The approach was to evaluate the antivirus apps first by using the android permission model to confirm functionality and then to test each app with viruses and virus signatures. The privacy impact of the apps was also tested to determine if there was a negative impact to the user installing and running the app. (Part of the test process was to compare the apps functions with its advertised functions.) And to provide the user with a simple snapshot of the efficacy of an antivirus app and the privacy impact to the user to apps that the user has downloaded and installed.

Key terms used in the research are;

Introduction

Security and antivirus apps are software programs that run on mobile devices to protect the device from software attack.

Privacy and its impact on users was described by Brunk on his paper on Understanding the Privacy Space (Brunk, 2002). The privacy impact to the user is hard to quantify as the monetary value of a users' information is not defined.

Social contracts are the voluntary agreements reached between individuals within a society. In this research, a social contract is an agreement between a user and the app developer. In this case the user purchases an app and expects to be able to use/play the app.

A psychological contract differs from a social contract as it is predominately used to define a contract between employer and employee. In this research this type of contract is applied to the relationship between a user of the app and the app developer and marketplace (app provider). A new definition of the contracts as related to app users and providers is proposed.

There are several contributions to research that have been made, these are:

1. A robust automated process to extract and analyses android permissions
2. A unique historical database that contains the source code and package code of all antivirus apps on the Google Play Store in 2011 and 2015. In the research all security packages which contained an antivirus component in 2011 and in 2015 were extracted and their source code maintained in a database for future analysis. Testing of all antivirus products in 2011 was unique, the major antivirus test organisation, AV_test.org only started testing android antivirus apps in 2014.
3. The research crosses the boundary between technology and psychology, mainly by assessing mobile apps as they relate to social and psychological contracts and defining Android permissions in psychological terms.

4. The creation of a framework which is used to illustrate the efficacy of an antivirus app. The creation of a privacy impact framework model. This display model takes the output from the initial analysis of an antivirus app and provides the user with an immediate display of the privacy impact of the app on the user.

There are several limitations of the research;

- The antivirus testing is performed using snapshots in 2011 and 2015 and the antivirus apps in the study are products available on the Google marketplace only and not on other app provider sites.
- Access to testing equipment limited the number of tests that could be performed on the antivirus apps, primarily the number of malware available for testing and their criticality, specifically zero-day malware.
- There was a limitation on the number of antivirus apps available at the time of testing, although there was a greater number in 2015, the marketplace was still immature with many of the specialist security companies not providing apps.
- The research was performed on a part-time basis and would've benefited by testing all the antivirus apps each year over a five-year time span to record the evolution of the antivirus apps every year, rather than a before and after snapshot.
- The research assumed that the apps would be similar across the different operating platforms. The research only concentrated on Android antivirus apps and could have benefitted with a comparison of iOS antivirus apps and Android apps to determine if the apps provided by the same company were more effective on the different platforms.

The format of the research and layout of the thesis follows. The research starts with a review of mobile devices, operating systems and how users are being protected whilst using the devices. The initial objective was to evaluate security apps to determine their efficacy and if they really protected the user. This initial

premise evolved into creating a permission extraction method to automate the manually download, transfer, extraction and analysis of the apps. The process was tested on other app genres to verify the robustness of the method. Analysis of other genres necessitated changing the hypothesis to test for privacy issues instead of security ones.

The research progressed from assessing and analysing the physical device to the available third-party security software. As the Android operating systems is permission based, a modicum of security was built in to prevent apps from using unauthorised system resources or accessing data or coding in other apps. Analysis of the permissions being requested showed that the requesting of permissions was arbitrary and the responsibility of deciding which permissions to use were left entirely to the developer. Analysis of the permissions requested to perform the antivirus function was tested. None of the controls in place were protecting the user and analysis of the interaction between the operating system and these apps demonstrated that the user had no benefit by using free or commercial security products.

Further analysis of the app permissions showed that the user's information was being obtained with uninformed consent. The user was unaware of what information was being gathered and how it was being used and by whom. This prompted the research to progress into the privacy ramifications of the data collected and use and if this was beneficial to the user by applying social and psychological concepts to the agreements. This led to the analysis of the collection of user information related to the protection of user privacy, especially with the future GDPR regulations being mandated in the EU in 2018.

The research also reviewed later versions of the antivirus apps to determine if the app had evolved and improved in protecting the user. To test the efficacy a framework was developed that provided a snapshot of the antivirus app efficacy. As privacy had been reviewed in testing the automation process the

Introduction

efficacy framework was adapted to test the privacy impact that apps had on the user.

The research concludes with a model defining privacy levels of Android permissions and how to evaluate their effect on the user of the app. Advice is provided on how to evaluate the privacy status of an app using a simple “fuel gauge” diagram and how the user can view and request the data held on the user.

To perform the research, security apps available in the Google Store from 2011 to 2015 were analysed for efficacy and privacy and subsequently Children’s apps from each genre were compared to determine if there were changes to the privacy controls for different age groups.

In 2011, free Security apps that contained an Antivirus component were investigated for their efficacy. Most of the apps did not perform the Antivirus adequately to protect the user’s device. Some of the app developers also provided a commercial variant. The research then concentrated on comparing the free and commercial variants to determine if the commercial variant provided any additional functionality and if it was effective. A comparison of the source data was performed to determine if there was any difference between the variants.

In 2015, using the same keyword criteria as in 2011, 67 apps were downloaded and analysed for efficacy. This was an increase of 30 from the number of apps available in 2011. The apps that were available in 2011 and 2015 were selected for comparative testing. An automated testing method had been developed between 2011 and 2015, called PEMP. The originally extracted 2011 apps were prepared for analysis using this final method.

To confirm the robustness of the P.E.M.P method Children’s apps were selected due to the sensitive nature of children’s protection. The top twenty apps from each of the three age groups were analysed. Initially the expectation was that

Introduction

children were unprotected and were being tracked and monitored through these apps.

The results from the analysis prompted a review of apps in relation to the new data protection laws, GDPR. The privacy requirements described in the GDPR articles raised questions around the ownership and accountability of the marketplace owner and the developer. This in turn raised questions on the social and psychological contracts between the user and the app owner/distributor.

A solution to address the responsibility of app owners and distributors is provided and discussed and is adaptable to any genre of apps.

The document starts by describing mobile devices, crime related to the devices and what is available to protect the device. This leads to the chapter which describes the Android operating system and how the various components interconnect.

The next chapter introduces the software available to protect the device and an analysis of the software for efficacy.

Chapter 5 describes the PC test environment, the software requirements and how to ready the mobile device to test the apps.

The following chapter describes the analysis of the apps from 2011 and 2015 and the comparison of the apps available in 2011 and 2015.

Chapter 7 describes the P.E.M.P. developed during the research.

This is followed by an analysis of children's apps to determine the privacy implications of children's apps.

Social and psychological contracts are reviewed in Chapter 9 and includes the GDPR articles and their required adherence by May 2018.

Chapter 10 provides the results of the testing and tools that are available to mitigate the issues uncovered in the research.

Chapter 11 contains proposals to control or eliminate issues and has guidance for regulators and a method to evaluate apps in relationship to the user's privacy.

Chapter 12 and the appendices contain information about additional tools to remove permissions from apps, input tables used in the research and information on the Android operating system to provide background for the reader.

All figures and tables in the thesis apply to global data and information unless otherwise specified.

1.1 Mobile Devices

There is a great deal of material available to assist consumers and enterprises in choosing security and Antivirus software to secure standard computing equipment; laptops, netbooks, desktops, etc. This comparative information is not yet widely available in the mobile sector (Smartphones, e-readers, iPads etc.) where the increase in acquisition of these devices has far outstripped the growth of legacy platforms. Additional issues are also introduced as the users of the devices either do not know or do not care about the potential security vulnerabilities of the devices, and the increase in criminal activity targeting the devices. There are many documents and advice, in the format of blogs and white papers, and company promotional material available to aid consumers and enterprises in securing standard computing equipment; laptops, netbooks, desktops, etc. There are also a variety of tools which are freely available to perform vulnerability assessments of these devices and networks that they use

for connectivity, e.g. Nessus (<http://www.tenable.com/products/nessus/>), Nmap (<http://www.nmap.org>) and Wireshark (<http://www.wireshark.org>), to name but a few. However, this availability of tools and knowledge had not transferred into the mobile sector (Smartphones, e-readers, tablets etc.). In this sector the increase in acquisition of these device types continued to exceed the growth of legacy platforms (laptops, netbooks), PCs and shipments increased to 92.1 Million in the last quarter of 2010 ("Tablet Computers Hold Back PC Sales Growth," 2011) whilst Smartphones grew by over 100 Million in the same period (Canalys, 2011).

This thesis identifies the Antivirus applications that are available as both free and commercial products for Android Smartphones, analyses them to discover any differences between the free and commercial apps and the privacy issues associated with the apps. These apps are then reviewed 4 years later to investigate the maturity of the apps, if they still exist, what new apps are available and how the existing apps have matured. Children's apps are then investigated to determine if the privacy issues detected in the antivirus apps exist in the children's games apps across different age ranges. This theme was continued in the light of the impending GDPR regulation required by any company trading in the EU. The thesis then concludes with the proposal for a privacy monitor which can be used by consumers and developers to determine if the apps contravene privacy requirements, especially with respect to the new GDPR regulations in Europe.

1.2 Online Crime

There is a great deal of material Online crime took off as a serious crime in 2004 (Moore, Clayton, & Anderson, 2009) after actors had realized the potential

opportunities once amateur hackers had shown the ease that websites could be defaced and malicious software circulated. Criminals have moved from cloning ATM cards and stealing pin numbers, to insider call centre employees collecting password data to establish entire networks, where wrongdoers have specialized roles and trade skills and resources with each other (Thomas & Martin, 2006). A new specialized role has emerged, that of a “botnet herder” (a person who manages a large collection of compromised computers and rents them out to spammers, phishermen and other actors to enable their criminal activities).

One of the ways to steal data (banking info, passwords, etc.) is to introduce malware onto the device. As most spyware requires physical access to the device, the goal of the attacker is to trick or persuade the user to install the malware themselves, thereby removing the obstacle of physical access.

There are a variety of methods in use to place malware on portable devices. Android vulnerabilities permit actors to install malware without the user’s knowledge. One example was the unsuspecting user downloads an application from the manufacturer’s store which is fake but contains malware injected into the application and placed on the app store. One example of this was the case of the Fake Angry Bird update application, that downloaded additional apps which accessed the phones contact list, location and SMS functionality, and transmitted it to a remote server (Goodin, 2010).

Other methods are to infect the device whilst the user is browsing the web, a strategy commonly called a Drive by Exploit, (Lu, Yegneswaran, Porras, & Lee, 2010) whereby the user’s device is infected merely by visiting the website, or where the device’s off the shelf OS security has been breached by “Jailbreaking” which leaves the device vulnerable to malicious software, as in the case of the Dutch phones with default SSH credentials (Lu et al., 2010) and in further exploitation of the vulnerability with the iKee.A as described by Porras et al in

their analysis of the ikee worm in Australia (Porras, Saïdi, & Yegneswaran, 2010).

1.3 Malware Growth

Mobile phones are growing at an unprecedented rate, overall the Smartphone sector grew by 64% in the year from 2Q2009 to 2Q2010 (“Google Android phone shipments increase by 886%,” 2010). With the sale of Android phones growing by 886% and Apple’s iPhone growth was around 61% during the same period. Although the Android growth slowed to 148.1% between 4Q 2010 and 4Q2011 its share of the market grew to over 51%, thus becoming the most popular mobile operating system (Canalys, 2011) in Table 1-1.

Table 1-1 Worldwide Smartphone market, in 2010

<i>Operating system</i>	<i>Q2 2010 shipments</i>	<i>% share</i>	<i>Q2 2009 shipments</i>	<i>% share</i>	<i>% Growth</i>
Symbian	27,129,340	43.5	19,178,910	50.3	41.5
RIM	11,248,830	18.0	7,975,950	20.9	41
Android	10,689,290	17.1	1,084,240	2.8	885.9
Apple	8,411,910	13.5	5,211,560	13.7	61.4
Microsoft	3,083,060	4.9	3,431,380	9.0	-10.2
Others	1,851,830	3.0	1,244,620	3.3	48.8
Total	62,414,260	100	38,126,660	100	63.3

Introduction

The table shows the worldwide market share and growth of the share of the smartphone operating systems from 2009 to 2010. Although in 2010 Symbian (by Nokia) had the major market share, the introduction of multiple cheap Android handsets from a small number of manufacturers has fuelled the growth of the Android market to take the biggest market share in 2011. This indicates that the sector is probably growing faster than controls can be developed to secure the products, this is like the growth of PCs within the general population in 1999 and the subsequent development of security controls, Firewalls, Antivirus, anti-Spyware applications, etc.

It was therefore natural to believe that there would be an increase in criminal activity in proportion to the growth of the Android operating system market share.

As an operating system becomes more prominent, actors are adapting the malware to target it. Initially actors adapted PC viruses and Trojans to the mobile market as in the case of the Zeus Trojan, which once installed uses the mobile to forward SMS messages, bypassing the 2FA (two factor authentication) systems used by a variety of UK banks to confirm identification by forwarding the Banks SMS containing a one-time-password (Raywood, 2010) to the actor. The installation of this sort of malware would normally be prevented by the Antivirus software on the device, but this is not a standard installation for Smartphones during 2010. Android handsets are very susceptible to these types of threats due to the availability of the open source of their operating system. Their applications are also available outside of the control of the Google Marketplace (<https://market.android.com/>) on a variety of online sites. Research has been conducted in the placement of malware masquerading as a legitimate application on the Marketplace.

However, in 2011 applications already containing malware are were infiltrating the Google Operating system faster than the increase in malware attacks against personal computers at a similar stage of development (Browning, 2011).

Vincent Wafer, senior vice president of McAfee Labs, said the year so far (referring to 2011) has seen "record-breaking numbers of malware, especially on mobile devices," and directly in proportion to the devices' increase in popularity. One of the most favoured techniques is infecting apps so users download and spread the malware themselves. Other trends, he said, include attacks that are stealthier and more sophisticated, which could mean some attacks go unnoticed for a substantial period. Stealth attacks have increased more than 38 percent over the last year (Ally Zwahlen, Heather Edekk, 2011).

Subsequent attacks and the availability of re-packaged applications pre-infected with malware (Taylor, 2010) forced the release of an application by Google for the removal of malware from infected devices (Kellex, 2011). Google notified affected users by email and supplied the removal tool, called Android Market Security, on the marketplace. Once the application has run and removed the malware, the application then removes itself from the device.

1.3.1 Malware Examples

Mobile malware has evolved since the initial Symbian viruses that spread via Bluetooth in 2007. In 2008, malware stole data and directed text messages to premium-rate numbers. 2010 saw the introduction of malware on iOS and the first ever trojan on Android. A detailed table of the Android malware evolution is included in Appendix D.

The following table (Table 1-2) provides a summary of the Android Malware released over a one-year period between August 2010 and August 2011.

Table 1-2 Android Malware release 2010 to 2011

Date	Malware Name
Aug 9 2010	SMS.AndroidOS.FakePlayer.a
Aug 17 2010	AndroidOS_Droisnake.A
Sep 14 2010	SMS.AndroidOS.FakePlayer.b
Oct 13 2010	SMS.AndroidOS.FakePlayer.c
Dec 29 2010	Android.Geinimi
Feb 14 2011	Android.Adrd AKA Android.HongTouTou
Feb 22, 2011	Android.Pjapps
Mar 1, 2011	Android.DroidDream AKA Android.Rootcager AKA AndroidOS_Lootoor.A
Mar 9, 2011	Android.BgServ AKA Troj/Bgserv-A AKA AndroidOS_BGSESV.A
Mar 20, 2011	Android.Zeahache
Mar 30, 2011	Android.Walkinwat
May 9, 2011	Android.Adsms AKA AndroidOS_Adsms.A
May 11, 2011	Android.Zsone AKA Android.Smstibook
May 22, 2011	Android.Spacem
May 31, 2011	Android.LightDD
Jun 6, 2011	Android/DroidKungFu.A AKA Android.Gunfu
Jun 9, 2011	Android.Basebridge
Jun 9, 2011	Android.Uxipp AKA Android/YZHCSMS.A
Jun 10, 2011	Andr/Plankton-A AKA Android.Tonclank
Jun 15, 2011	Android.Jsmshider
Jun 20, 2011	Android.GGTracker
Jul 1, 2011	Android.KungFu Variants
Jul 3, 2011	AndroidOS_Crusewin.A AKA Android.Crusewind
Jul 6, 2011	AndroidOS_SpyGold.A AKA Android.GoldDream
Jul 8, 2011	DroidDream Light Variant
Jul 11, 2011	Android.Smssniffer AKA Andr/SMSRep-B/C AKA Android.Trojan.SmsSpy.B/C AKA Trojan-Spy.AndroidOS.Smser.a
Jul 12, 2011	Android.HippoSMS AKA Android.Hippo
Jul 15, 2011	Android.Fokonge
Jul 15, 2011	Android/Sndapps.A AKA Android.Snadapps
Jul 27, 2011	Android.Nickispy
Jul 28, 2011	Android.Lovetrap
Aug2 2011	Android.Premiumtext
Aug 9, 2011	Android.NickiBot

One example of a major infection in 2011 was DroidDream. This malware had the capability to root the mobile and install infected applications without direct

intervention by the user. The malware operated between 11pm and 8am which the developers determined were the quietest time, termed “Dream time” and the phone would be rarely used. This resulted in Google supplying an app on the Play Store (Kellex, 2011) to scan apps and remove the malware infected app on the user’s device, it was not able to repair the app. Since Android V6.0 (Marshmallow) Google has provided a facility to manage any app’s permissions (Hoffman, 2017). Again, there was no guidance to which permission to deactivate.

Currently there are a variety of malware targeting the Android operating system, a few of these are; SMS Trojans (examples are RuFraud, Fancy), Trojans/bots (examples are DroidDream, Basebridge, PJapps, DroidKungFu), SMS/Spyware (examples are NickySpy, Mobi stealth, ZiTMO, SpiTMO) and Dataleak.

1.4 Anti-Malware Protection

The Antivirus products for legacy PC environment has matured greatly since its introduction in the late 1980s to early ‘1990s. At that time malware was introduced into the device by sharing files between users, either via email or by sharing floppy disks. The introduction of the World Wide Web made file sharing easier and also made many devices open to attack from the web, this lead to a proliferation of online crime in 2004 (Moore et al., 2009) once hackers had demonstrated the ease that websites could be defaced and malicious software circulated.

Antivirus software was first designed to detect and remove computer viruses. The software has developed to detect and remove a variety of malware, including worms (a self-replicating virus), Trojan Horses (a malicious program that appears harmless), rootkits (a collection of program tools that enable an

attacker to have administrative access and remain hidden from the user), spyware (spies on the user and violates the user's privacy), keyloggers (monitors and records each keystroke that the user types), ransomware (hijacks the users data by encryption it and demanding a ransom to provide the user with the decryption key), and adware (banners or pop ups that advertise products. The malevolent ones point to a website so that malware can be installed on the device).

There are now a variety of free and commercial antivirus products available for the user to choose from that will detect and remove malware. To assist in the choice research has been conducted into the effectiveness of these products ("AV Test Reports," 2011) and the comparison results are published regularly and even graded by the reviewers ("Top ten reviews," 2011).

1.4.1 Antivirus Functions

The objective of any Antivirus product is to prevent a device being infected with malware. This is achieved by either preventing the malware installing onto the device and removing any existing malware detected on the device. To do this the product must be able to detect incoming malware and prevent it installing and detect any pre-installed malware and remove it.

Detection of incoming malware is known as Real Time Monitoring and consists of scanning downloads (programs and documents), emails and messages (SMS and MMS) and preventing the installation of the malware onto the device by either deleting it or quarantining it into a secure non-executable environment.

Detecting pre-installed malware is performed by scanning the device for malware also either removing or quarantining it.

Antivirus products use a two-pronged approach to detect malware; basic and advanced. Basic detection consists of hashing the suspected file and comparing it to a known virus “signature”. The advanced detection uses a heuristic approach, which is behaviour based and assigns a score to the suspected file depending on a combination of factors, e.g. malicious links, code behaviour, etc. The score is then used to indicate if the file is infected or not. If both approaches are used, then the product would require a “signature” database which would need to be updated as new signatures are created. If only the advanced approach is used then although the signature database is not required, the product may detect more false positives than by using a combined approach.

1.4.2 Antivirus on Mobile Devices

Smartphones by their nature are continually connected to a network and are ready to receive calls and messages. Wireless networks are less secure, and the device communication is open to interception (sniffing) and Man-in-the-Middle attacks. Being always connected also increases the time available for an attacker to monitor and access the device to obtain banking security codes or to install malware on the device. Data encryption is not installed and configured as standard (unlike iOS smartphones) which means that data is readable and useable if physically accessed. Pins and Password technologies which are available on most smartphones are predominately not activated, therefore facilitating access to the data once the device is physically acquired. Malware is also introduced to the device on receipt of an infected message (SMS, MMS, email attachment) or downloaded from a website, as was the case with the Fake Angry Bird app (Goodin, 2010) or even by accessing a website as is the case with Drive-by-exploits (Lu et al., 2010).

Introduction

Attackers have now progressed to offering ranges of free apps, normally with advertising, that can track the user's activity, read their contacts or SMS texts, or take control of the phone's functions. A recent article in *The Sunday Times* described how 70% of users rarely or never read the Terms and Conditions when they download an app (Henry & Flynn, 2012), which in most cases requested permissions from the user to access their private data and the handset's functions, including well known apps like Facebook and Twitter. The article continued with a description of the information that could be gleaned from the device and the types of functions that the app could control. The number of these intrusive apps are increasing as developers realize the income from advertising exceeds the income from selling an app, with one company producing one free app a week and expecting to have created more than 1,000 by the end of 2013 (Henry & Flynn, 2012).

Antivirus software will not prevent all exploits but will aid in protecting the device when accessing an infected site and preventing the downloading and installation of infected files and messages.

In 2012, Gibler et al (Gibler et al., 2012) used a static analysis tool *AndroidLeaks* to evaluate privacy leaks of Android apps. Of the 24,350 apps tested, 7,414 showed potential privacy leaks, of which 2,342 were manually verified leaking privacy data. They also concluded that the requested permissions are not informing the user of how they are being used. They also had concerns about the install adware and what data was being collected on the user. They used a program analysis tool *WALA* to process the Java source and bytecode but had to perform the mapping manually. In my research I created an extraction and analysis tool in Python to perform this called *PEMP* (see Chapter 7). However, their research also tested adware libraries, where my research concentrated on the apps, the genre types and the privacy issues in the genres rather than the apps.

Chapter 2. Literature Review

The previous chapter introduced the mobile devices available at the time of the research and how online crime and malware grew to match the popularity and growth of these devices.

This chapter describes the research active at the time of this study and how that research scope has grown to match the increase in handset availability and use.

Initially mobile devices were large, and the battery pack was carried separately. The devices contained proprietary software, e.g. Symbian by Nokia and RIM by Blackberry. Android was developed by Android Inc. and was bought out by Google in 2005. The Android operating system is based on a Linux kernel and has gradually taken over as the most common OS for mobile devices.

As the most common OS and as an open source product research started with reviewing the Android operating system concentrating on the API calls of the OS and then to security of the device and privacy issues such as Geo Tracking.

S mobile systems (Vennon & Stroop, 2010) performed a threat analysis of the Android market. Here the author describes the openness of the Android environment, the flexibility of any-one has access to develop apps and publish apps. He identified the market security model, where it is the community's responsibility to identify and test if an app is malicious. He described the Bank

phishing malware Droid09. However, there is no process for the detection, test and notification of malware. He explained the difficulty of detecting virus signatures.

Developers are required to declare their permissions for the app. The author described a method and technology to determine potential malicious apps depending on the permissions.

They performed a market analysis of 48,694 apps (68% of the 2010 apps available for download) and noted that 20% requested permissions that could access private information and 5% that could call any number without user intervention and 3% could send a premium SMS message.

This analysis was used create a behaviour-based detection model. (patent pending).

Sandminer, a context aware sound Trojan was used as an example of a trojan developed to steal user's credit card data and have access to the microphone and dialler (Schlegel, Zhang, & Zhou, 2011). The authors explored the increase in data-stealing malware on mobile phones and how antivirus companies are moving their products from the PC arena to mobiles. They showed how supposedly secure apps could be attacked and sensitive information, like credit card data, could be detected. This is achieved by the trojan recording the digits from a user's conversation (either spoken or typed). The research did not provide a solution but suggested a defence to sensory malware.

Xu et al (Xu et al., 2009) described using the video function to capture data and developed a video spyware called Stealthy Video Capture, to record the video.

New security services were described by Enck et al (Enck, Ongtang, & McDaniel, 2009) and methods for retrofitting security requirements in Android, a method for certifying apps at install time. They provide a product Kirin as a security add-in to supplement the then existing security framework. A set of rules were defined, and security requirements were identified. Individual permissions were designated as “dangerous”. Although their malware mitigation rules provided rules for single and multiple permissions, there was no investigation into how some combination of permissions could be identified as potentially dangerous if used concurrently. They tested the top 20 applications in each of the 16 categories (a total of 311 apps). Only 12 failed their 9 security rules. Some of these were false positives where the app required the permission described to function. They also discovered flaws in the operating system that permitted malicious apps to make API calls without the required permissions.

More recently researchers created or reviewed mobile forensics to analyse mobile device security. In Digital forensics, investigators use similar techniques to obtain evidence from Mobiles and PCs. Mobiles have a limited amount of storage, so the main function of the forensic tools are to extract personal data. Most of the mobile companies have proprietary Sockets to access the device, even if they are using open source operating systems and more models are increasingly available. Vinit Shah (Shah, 2012) described a model that forensic scientists could use as part of their forensic extraction.

Dehghantanha et al (Dehghantanha, Udzir, & Mahmud, 2011) discussed mobile device functions and possible vulnerabilities and proposed a security model to protect the data on these devices. Their research described the vulnerabilities and how the loss of data affects the user and their company if the user has a COU (company owned unit). A financial figure for this loss has not been stated. The top ten cyber security risks were described by the SANS

group (a leading source for information security and training) in 2009. One point raised in the paper was that as these devices were connected to the computer at some stages, to perform synchronisation or backups/restores, the malware could move to the computer. Additionally, was the discussion on how the isolation of apps (sandboxing in the Android environment) made it difficult for Antivirus apps to detect malware.

Thus, was the case in this research, where it showed in 2011 that Antivirus apps were not effective.

The research concluded that mobile devices needed security protection, especially as the devices were being used more and more in the commercial world.

Felt et al (A. P. Felt, Chin, Hanna, Song, & Wagner, 2011) provided an in-depth analysis of Android permissions. They created and built a tool called Stowaway that detected privileges in API calls. The tool was used on 940 applications and detected that a third of these apps were over-privileged. Their research then moved to analysing the API's permissions and the tool calculates the maximum number of permissions that an app needed. 40 applications were used to verify the tool efficacy and compared to the manual calculation. Their conclusion was that the extra permissions was caused by developer confusion. Stowaway has since been superseded by PScout in 2012. The PScout tool was developed by Wain et al (Wain, Au, Zhou, Huang, & Lie, 2012). Again, the research was performed across the plethora of Android apps and not at a specific genre.

The main research was into API calls and if the app was obtaining more access than was required to perform the function of the app.

In my research I decided to concentrate into specific genres and if the permissions requested were adequate or excessive to perform their function. The initial genre was in the utility genre, specifically Security apps as they were the first line of defence to protect the user.

An earlier paper into wireless security and privacy (Katos & Adams, 2005) explored the relationship between wireless security and privacy. They introduced the concepts of security and privacy and how the concept of security changes over time due to multiple external influences and especially in response to the increase in malware. They equated the “focus on privacy because of increased awareness of human rights”. The paper mainly concentrated on the responsibilities of corporations to “adopt appropriate policies to conform to privacy rules”, here the responsibility was placed on the user (corporation) to protect themselves.

A paper focusing on privacy tools (Brunk, 2002), performed a detailed examination of 133 privacy-related tools and services. The examination discovered 1,241 features relating to privacy. Their work formulated a framework to describe "privacy space" and provided a statistical analysis of the raw data. The paper concentrated on the software tools from a user perspective and reviewed a sample of web sites. Due to time constraints he was only able to evaluate 50 sites.

The solutions investigated were in many formats, freeware, shareware, adware, spyware and demonstration packages (a.k.a. crippleware) etc, but did not review a group of specific formats.

Privacy concerns for mobile devices was explored by Tsavli et al (Tsavli et al., 2015). Then the number of smartphones and apps had increased enormously. There were apps that provided business application access e.g. email, file and document management, as well as educational apps and games etc. Many of

these apps stored data about the user and were used to detect trends in user usage of the device and apps as well as enriching the user experience of the app. The paper explored the data flow of the user's personal information. A "data taxonomy" was proposed to determine which data was being requested and by whom. The data was defined into one of seven categories and this study classified the apps into five different genres. The results of their research were like my research into Antivirus apps showing the lack of control that a user has on agreeing to permissions especially related to the fine control of the permissions.

Chapter 3. Evolution of Research Design

The introduction described the available mobile devices, the vulnerability of the devices and how they need to be protected. The previous chapter illustrated the available research and how this research had concentrated initially on the device security and the security of the running software.

This chapter describes the research design of the study and how it evolved to meet the changing environment of the mobile device and Android app market.

The original research questions were to answer how secure mobile devices were, if there was security software available, did it provide more protection to the user and was there a difference in protection between free and commercial products.

The Android mobile operating system was selected due to its open source nature, which meant that there was more opportunity for coding malware to attach the system. All products on the Google Marketplace that contained a security keyword were selected. Specifically, security apps that contained an Antivirus component. There were a variety of free and commercial apps and the research initially concentrated on the differences between the variants of these apps.

To perform the analysis the app needed to be in a PC readable format.

In 2011 the process to download and extract the app involved performing the upload and installation steps in reverse.

The app was downloaded to an Android device, in this case a T-mobile G1 mobile. The app was in Davlik executable format.

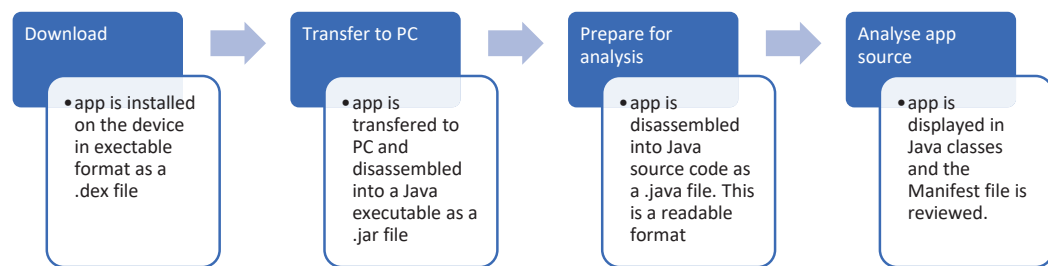


Figure 3-1 App Download and Extraction process

The app is downloaded from the Google Marketplace using the Android downloader tool. The software required to perform the transfer and analysis are Android Developer tools.

The time taken to download the app was dependant on the mobile device connection and the size of the app, normally took about a couple of minutes. The transfer and preparation of the app for analysis was also a manual process but took much longer due to the many steps involved. This part of the process took approximately 25 minutes.

This made the preparation of the app for analysis and review a long process and limited the selection of the apps to analyse. Details of the process is described fully in Chapter 5.

A process was also required to obtain the commercial apps at a minimum cost. This is described in section 6.2.2

The design was finalised in 2015. The main aim in 2015 was to be able to obtain the app, transfer the app to the PC and perform the analysis automatically.

The initial refining method involved automating each step and manually providing the data for input to each automated process. This involved too much manual intervention and an automated process was required to transfer the data for input to each step.

The automation is described fully in the created method, Permission Extraction Method and Process (P.E.M.P.) in Chapter 7.

This method enabled the download, transfer and analysis of each app to under 5 minutes. The reduction in the download and transfer of the app was greatly reduced due to the availability of a new tool in 2012 which provided the ability to download the app directly to the PC. The app still needed to be in an input format suitable for the disassemble and analysis. The method finalised in 2015 provided the method to download and process over 60 apps within a 30-minute window. This was a reduction from 30 minutes to 30 seconds for each app.

Chapter 4. Antivirus and Anti-Privacy (2011-2012)

The introduction described the available mobile devices, the vulnerability of the devices and how they need to be protected.

This chapter describes the relationship between antivirus and privacy of the devices and software. How the device and user should be protected and if the software aimed primarily at protecting the user is effective and if there are privacy issues with the software.

Having researched the Android operating system on mobile devices and the function of permissions on the apps on the device, the security of the device was tested. First the Antivirus functions needed to be defined and understood.

The Antivirus functions are defined below and linked to the permissions needed to perform that function.

4.1 Antivirus Functions and Permissions

To perform Real Time Monitoring, scanning, removal of malware and updating of a signature database the app would require access to system resources to read incoming messages, downloads and storage and to either prevent installation or storage and to delete any pre-existing infections.

The table describes the minimum functions that an Antivirus program should be able to perform to be effective, the reason that the function is required and the threat that the function mitigates.

Table 4-1 Basic Antivirus Features

<i>Basic Security Requirements</i>	<i>Threat mitigation or Reason</i>
Real time monitoring: Scan downloads Email scanning SMS scanning	Scanning of apps, files, email, SMS, etc. during download or transfer to prevent malware being downloaded and installed on the device
Passive Monitoring: Device scanning	The ability to perform a scan of the device, either manually or on an automatic schedule is needed to detect if malware has been introduced to the device via physical access (e.g. SD card, 3 rd party installing spyware, etc.), or it has slipped though the Real-time monitoring.
Maintenance: Virus signature update	Scans should always be performed with the latest virus signatures to reduce the incidence of zero-day attacks.

To perform these functions a basic set of permissions are needed, Antivirus (AV) Permissions (AV_Perm), for the antivirus app to be effective.

The following AV_Perm for each of the Antivirus functions are defined and a verification method is provided.

4.1.1 Real-time monitoring

Real time monitoring consists of reviewing incoming apps (during downloads) and messages (SMS) to detect any malware inside the app code or text message. The permissions that would permit this are;

RECEIVE_MMS - Allows an application to monitor incoming MMS messages, to record or perform processing on them. Monitors incoming MMS messages, to detect malware and to remove it or to perform other processing on them.

RECEIVE_SMS - Allows an application to monitor incoming SMS messages, to record or perform processing on them. Monitors incoming SMS messages, to detect malware and to remove it or to perform other processing on them.

4.1.2 Malware Removal

To remove the malware from the device the antivirus product needs to have access to the storage areas on the device (RAM, Memory, device storage and SD card storage) and to prevent or disable the app if it is running. To access these areas the following permissions are needed;

CLEAR_APP_CACHE - Allows an application to clear the caches of all installed applications on the device. Clear the device cache of detected running malware.

DELETE_PACKAGES - Allows an application to delete packages. Deletes malware app from the device

KILL_BACKGROUND_PROCESSES - Allows the application to call the process to force the process to end. Stops process if it is running in the background

WRITE_EXTERNAL_STORAGE - Allows an application to write to external storage. Clears or deletes data on external storage (SD card).

4.1.3 Scanning

The antivirus app needs access to scan the installed device for malware and to remove any infection if it is running. The permissions are;

GET_TASKS - Allows an application to obtain information about the currently or recently running tasks. Obtain information about running tasks or recently run tasks.

READ_EXTERNAL_STORAGE - Allows an app to read from the external storage to determine if malware is already installed or if there are infected files on the card.

4.1.4 Update Signature Definitions

An antivirus app needs to be able to recognise malware and to do this it must have access to a database of malware signatures. Signatures are used to detect malware that has small variations from the original malware. To do this the app must either be able to download the latest signatures or have access to a

signature database, or to heuristically predict the malware signature from existing available signatures.

ACCESS_NETWORK_STATE and CHANGE_NETWORK_STATE - These allows applications to access information about networks; used to determine if the device is connected to the network and if not to active the network connection to either access a cloud signature file or download signature updates.

INTERNET - Allows applications to open network sockets and connect to the Internet.

4.2 Antivirus Verification Method

This method can be used by a user to verify that their installed antivirus is working and detecting malware. To be effective in securing the device an Antivirus product should be able to:

- Scan the device, detect and remove malware
- Detect malware at download or installation
- Update a signature file or have access to the latest virus signatures

Other options which are advantageous but are not necessarily essential are automatic or scheduled scanning and automatic updating of the signature file. This ensures that the product requires no intervention from the user and is protecting the device against the latest attacks.

Software products containing antivirus should be able to scan and detect malware as standard.

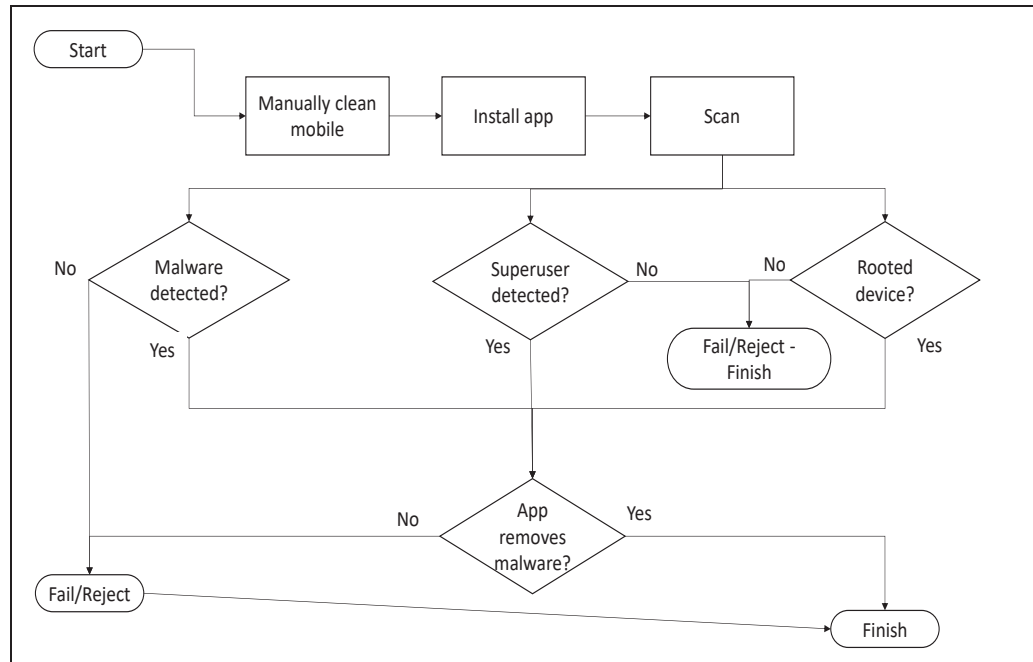


Figure 4-1 Flowchart illustrating the initial app installation and scan function

The initial procedure is to download and install the security product that contains the Antivirus component. Once the product is installed, activate the app by opening it. In the open app determine if the app needs manual intervention to perform a scan? Perform a scan of the handset to provide a base for comparison. If the app detects any malware, follow the instructions to quarantine or delete the affected file or application. If the mobile has been Jailbroken or rooted, then the security app should detect that there is Superuser access on the device.

If this is the case then ignore the message that occurs during scanning that this access is suspicious or malware, if the product doesn't detect the root access then the product may not detect rootkits or spyware installed on the device. If the app detects Superuser access and you the device has not been Jailbroken or

rooted, then follow the instructions to remove the access². Once the device is clean (no malware detected, except for the intended Superuser access), deactivate or stop the security product. Once the app is no longer running, download a test virus. Test viruses are not malware but contain a malware signature and will be detected as malware by the Antivirus software. Two such test viruses are P.Defender's Antivirus TESTVIRUS available from the Google marketplace (<https://market.android.com/>) or the Eicar Test Virus from Extorian (http://eicar.org/anti_virus_test_file.htm). Once the test viruses have been downloaded, activate the security product and scan the device. The app should detect the test virus on the device.

If you have not done so before, quarantine or remove the test virus as per your Antivirus instructions. Then rescan to ensure it has been removed. If the product does not detect this test virus, then the product is not performing the scanning adequately and is not fully protecting the device.

Antivirus products should detect malware at time of download to protect the device whilst on the Internet or to prevent malware being downloaded over a Bluetooth or Wi-Fi connection; this is known as real time monitoring.

² From 2014 the App, SuperSU, available on the Google Play Store removes Root access.

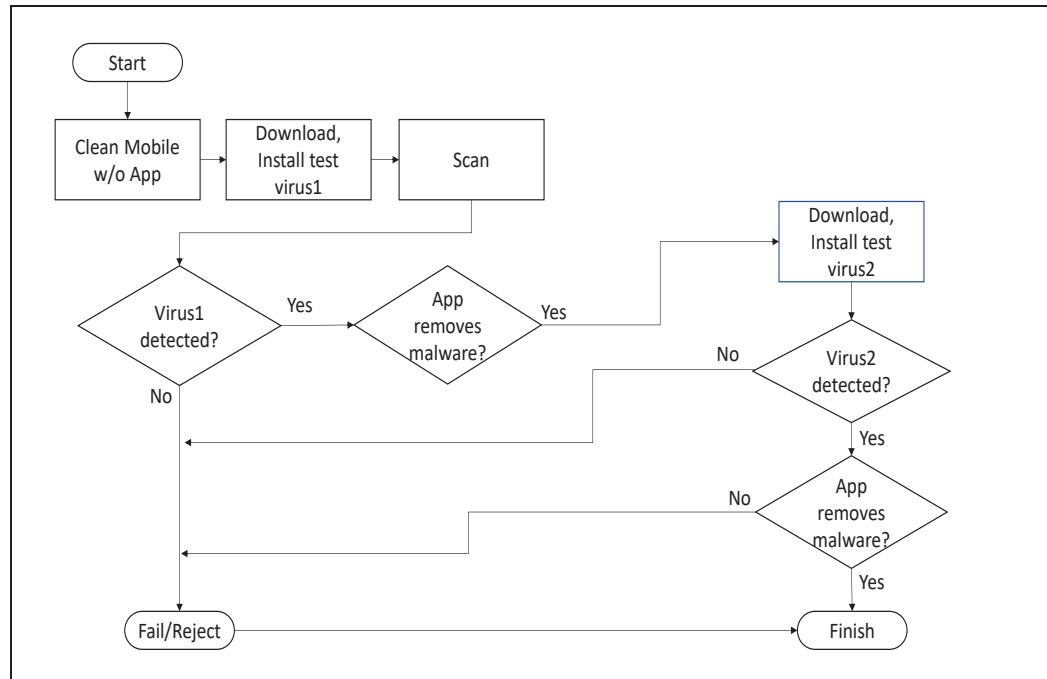


Figure 4-2 Flowchart illustrating the malware detection.

The flowchart displays the steps required to perform Real Time Monitoring verification. Monitoring is a similar function to scanning but contains additional steps due to the proactive nature of the function. The monitor analyses the app during download (installation process) for known malware or known virus signatures.

With a clean device (follow previous instructions to scan and remove malware, installing the app onto a clean device as in chapter 5.1). Ensure that the security product is active and download a test virus onto the device. The product should detect this at download and either prevent the download or provide a notification that the app contains suspicious content. If the product does not detect this test virus and permits installation, perform a scan to verify that it is detected by the product, if the malware is not detected then the product is not intercepting the download and is therefore not performing real time monitoring.

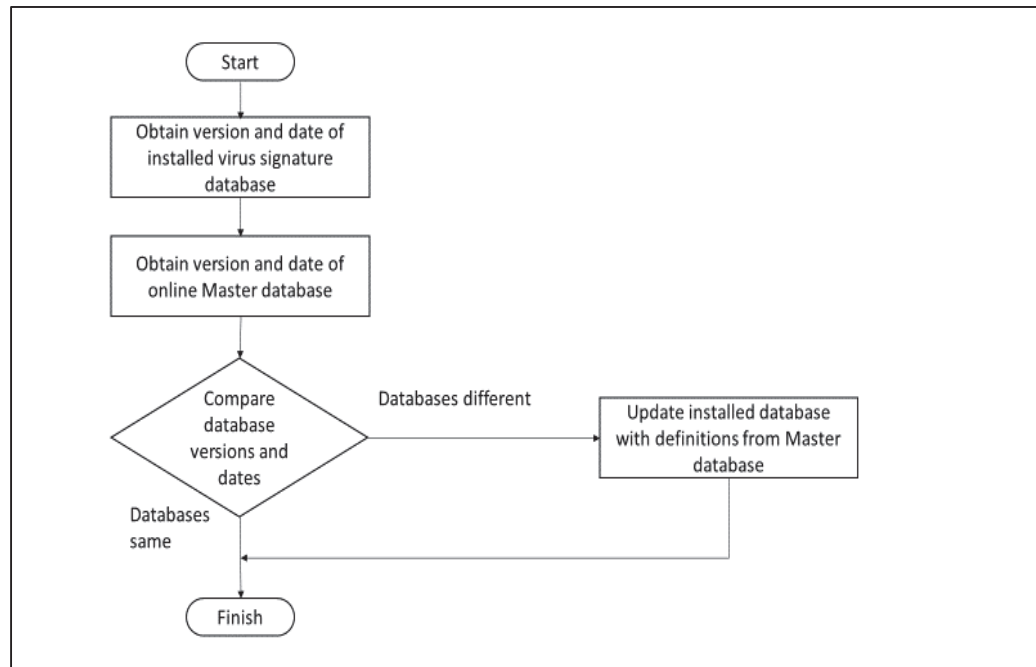


Figure 4-3 Flowchart of update of signature file.

A signature is an algorithm or hash that identifies a specific virus. A signature may be consistent amongst various viruses; in this case an Antivirus scanner can use this signature to detect known and new viruses. The signature file contains the signatures and is updated when new signatures are detected. To maintain effectiveness all Antivirus software should be able to access updates to the file or perform heuristic analysis for suspicious content.

The signature file updates are normally performed automatically but some products prefer the user to initiate the file update or only check for updates when activated. Check the product settings to verify that updating is performed on a schedule or if it is performed manually. If manual update is proscribed, then initiate an update request. The expected response is either confirmation of the update (and normally the version number of the update) or that the database is up-to-date. If the product does not use heuristic analysis

of files, then the lack of updating of the signature file means that the device is not protected against newer threats.

4.3 Anti-privacy Functions and Permissions

There are a variety of permissions that permit an application to access the user's private details on the device. The following permissions requested by any of the analysed security app manifest files are deemed to contravene the user's personal privacy. These six permissions are; `CALL_PHONE` (provides the ability to make phone calls without the user's knowledge), `GET_ACCOUNTS` and `MANAGE_ACCOUNTS` (obtains a list of the user's service accounts and permits the app to add or delete accounts from this list or to read account details, e.g. GMAIL or Facebook or Twitter account ids and Pins/passwords), `READ_CONTACTS` and `WRITE_CONTACTS` (read and write to the user's phonebook) and `WRITE_CALENDAR` (allows an app to write but not read the user's calendar - perversely none of the apps in this analysis asked permission to read the calendar).

4.3.1 Anti-privacy Permissions

In 2010, Android version 2.2 had a total of 105 permissions that could be selected by a developer. Each permission was evaluated to determine if it contravened the user's privacy.

The permissions which were considered to cross the concept of privacy were recorded in Table 4-2 . The Oxford English Dictionary defines privacy as:

1. a state in which one is not observed or disturbed by other people
- or
2. the state of being free from public attention:

Brunk’s research on privacy examined the privacy tools and services on the internet and created a framework to describe a privacy space (Brunk, 2002). His research was based from the user perspective, other works had focused on technology. He defined Role categories; awareness, detection, prevention, response and recovery. This research concentrates on the awareness and detection categories and are further divided into four sub categories. Therefore, any permission which can be used to monitor activity; track location, overhear or spy on the user can be considered as a contra-indication of privacy.

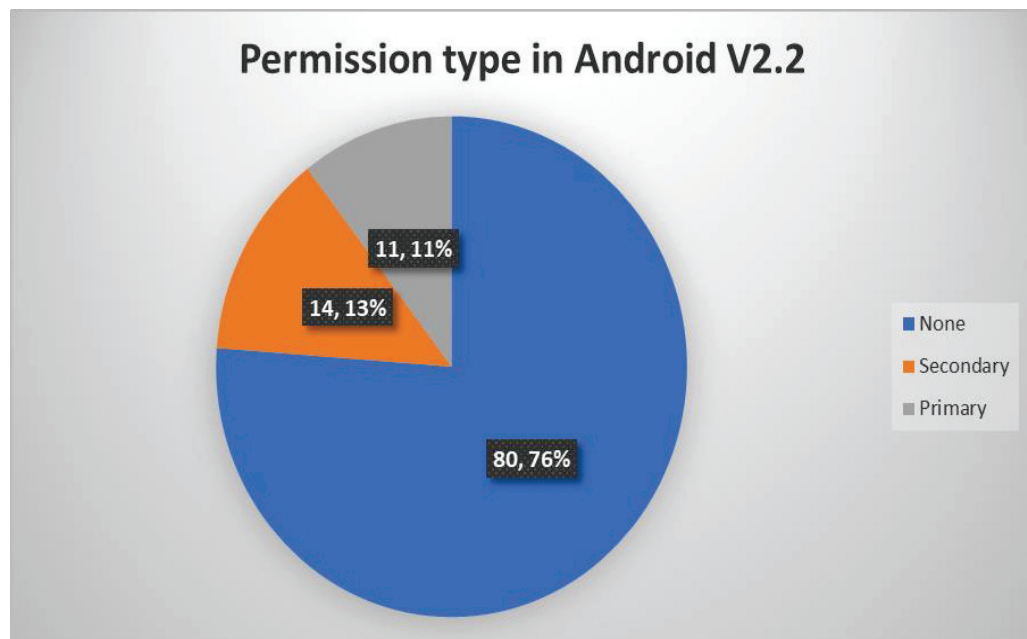


Figure 4-4 Permission types: No privacy issues and two types of Antiprivacy concerns

There are approximately 25 permissions which either contravene the concept of privacy (primary role) or perform secondary roles to enable the devices that permit the eavesdropping/monitoring.

The chart in Figure 4-4 shows the available Android permissions evaluated and placed into one of three groups, no privacy concerns, secondary antiprivacy concerns and primary antiprivacy infringements.

However, only the eleven (11) permissions that perform the primary roles are marked and described (Table 4-2). To determine the impact on the user's privacy, each of these permissions are given a rating. The ratings are:

- High - control permits all anti-privacy activities
- Medium - control permits most but not all anti-privacy activities
- Low - control permits few but not most anti-privacy activities
- None - control does not affect user's privacy

The permission that is of most concern is the one marked High, which permits an app to capture secure video output. This enables the app to track, spy and overhear the user. The remaining 80 permissions not affecting Privacy are marked None and are not listed here.

Table 4-2 Primary anti-privacy permissions and their activity and privacy rating

Permission	Description	Activity	Rating
ACCESS_COARSE_LOCATION	Allows an app to access approximate location.	Track	Low
ACCESS_FINE_LOCATION	Allows an app to access precise location	Track	Low
CAMERA	Required to be able to access the camera device	Spy, Overhear	Medium
CAPTURE_AUDIO_OUTPUT	Allows an application to capture audio output	Spy, Overhear	Medium
CAPTURE_SECURE_VIDEO_OUTPUT	Allows an application to capture secure video output	Spy, Overhear, track	High
CAPTURE_VIDEO_OUTPUT	Allows an application to capture video output	Spy, Overhear	Medium
READ_SMS	Allows an application to read SMS messages	Spy	Low
READ_VOICEMAIL	Allows an application to read voicemails in the system	Spy, Overhear	Medium
RECEIVE_MMS	Allows an application to monitor incoming MMS messages.	Spy	Low
RECEIVE_SMS	Allows an application to receive SMS messages	Spy	Low
RECORD_AUDIO	Allows an application to record audio	Overhear	Low

The majority of designated antiprivacy permissions are classified as low, and only CAPTURE_SECURE_VIDEO_OUTPUT is viewed as a major abuse of privacy, as this permission permits an app to track the user and to record sound and vision of the user's location/user.

4.4 Antivirus Apps Permission Analysis

The Google Marketplace (<https://market.android.com/>) contained 37 security products that had either security or antivirus in their names or contained them as keywords in their descriptions. These products constituted the base for the investigation. The permissions requested by these apps were recorded and reviewed against the API list to determine the requested access to system resources. Of the 130 permissions, available at the time of the study (in 2011 the most common version of Android was Froyo) 103 were requested by the security apps analysed.

Firstly, the permissions that were determined to provide the Antivirus functions and those which were detrimental to the user's privacy were noted for each security product.

The flowchart, Figure 4-5, illustrates the method used during the investigation. The initial step is to define the parameters for the product type for the investigation. Apply the sample criteria and select the samples. A method was not available to examine the permissions and one was written to fulfil this function. The method was applied and updated to create a robust method. Comparisons were performed, and the results documented. Analysis of the results indicated the next steps of the research.

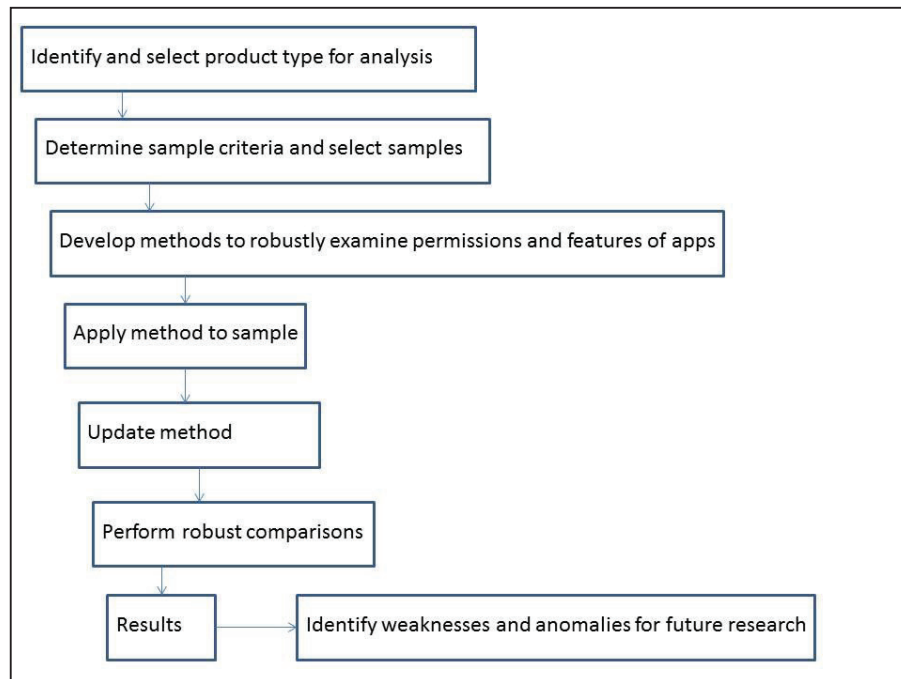


Figure 4-5 Investigation method flowchart

4.5 Selection Criteria and Sample Selection

A search was performed in the Google Marketplace (<https://market.android.com/>) and the keywords used for the selection criteria was; *antivirus* and/or *security*. The security apps were required to have an Antivirus component, or they were dropped from the study. The objective was to test all the available apps, and these were selected.

4.6 Antivirus Functions

The next step was to provide a common base for the comparison, the author considered the following features to be the basic functions that should be in any security product containing an anti-virus component (Table 4-1), the selected products were then compared to the basic features (Table 4-3) for each of the selected products.

Table 4-3 Products and stated Antivirus Features

<i>Product</i>	<i>Real time monitoring</i>			<i>Device scan</i>	<i>Virus signature update</i>
	<i>At Download</i>	<i>Emails</i>	<i>SMS</i>		
Lookout mobile security (free)	√			√	√
Lookout mobile security (premium)	√			√	√
AVG Antivirus Free	√	√		√	√
AVG Antivirus Pro	√	√	√	√	√
Dr. Web Anti-Virus	√		√	√	√
Dr. Web Anti-virus lite	√			√	√
Aegislab Antivirus Free	√			√	√
Aegislab Elite	√			√	√
Bluepoint Antivirus Free	√	√	√	√	√
Bluepoint Antivirus Pro	√	√	√	√	√
Android Defender Virus Protect (free)	√			√	
Defender Pro Virus	√			√	

Each product describes its features and functions that it performs. The functions that relate to antivirus processing are marked in the table.

4.7 Anti-privacy Permissions Requested

Using the defined grouping described in Figure 4-4 each app was analysed, and the number of permissions requested in each group was recorded. The number of permissions requested by each app in each group is shown in Figure 4-6.

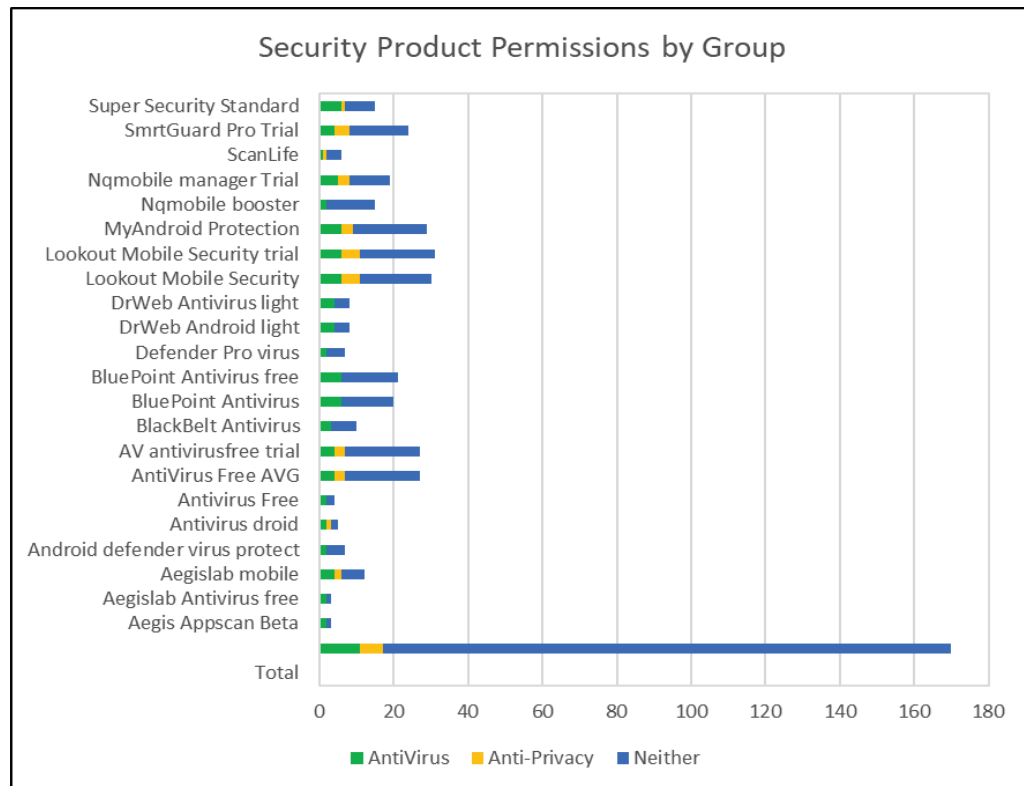


Figure 4-6 Permissions by security type for each app (2012)

The apps did not request the full set of permissions to perform antivirus monitoring and removal. There were 11 permissions that were required to fulfil the antivirus function and 6 permissions that contravened the user's privacy.

The permissions for antivirus processing were determined to be the following;

- ACCESS_NETWORK_STATE
- CHANGE_NETWORK_STATE
- CLEAR_APP_CACHE
- DELETE_PACKAGES
- GET_TASKS
- INTERNET
- KILL_BACKGROUND_PROCESSES
- READ_EXTERNAL_STORAGE
- RECEIVE_MMS
- RECEIVE_SMS
- WRITE_EXTERNAL_STORAGE

The highest number of antivirus designated permissions that was requested by an app was 6. Figure 4-7 displays the apps and the number of antivirus permissions requested.

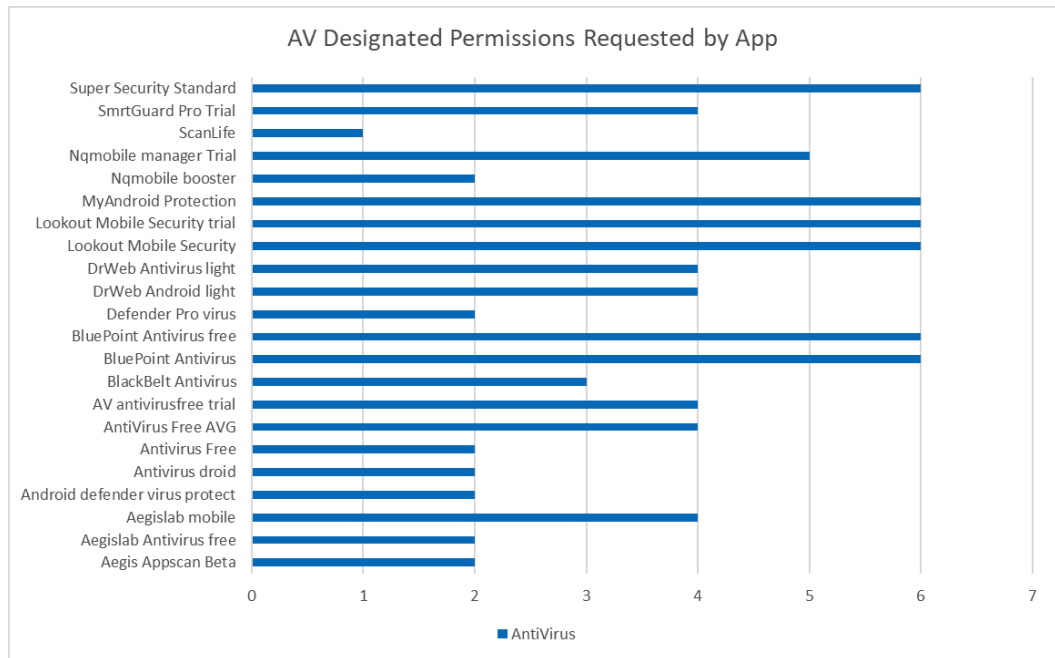


Figure 4-7 Number of Antivirus designated permissions requested by each App

The figure shows that none of the Antivirus apps requested all the antivirus permissions required to be effective. The maximum number requested was 6 which indicated that Antivirus functions were not being performed adequately.

This chapter has described the Antivirus functions and their related permissions and defined the categories of anti-privacy permissions and graded them for severity of privacy infringement. The next chapter defines the test environment and how to prepare the device and software for testing the Antivirus apps. This test environment is used during the analysis of the Antivirus apps.

Chapter 5. Preparing the Test Environment

Part of research is the ability to perform a repeatable and robust process to obtain and analyse data. This research obtains data from apps freely available on the Google Marketplace (<https://market.android.com/>) and uses commonly available tools to process the data. In that aspect this chapter is devoted to the creation of a test environment.

A tool to download the App, in this case a T-Mobile G1 and another to perform the analysis, in this case a Windows PC. The software used in the analysis is also discussed.

5.1 Preparing the Test PC

A PC is required to perform the analysis of the packages downloaded from the Google Marketplace. The software on the PC enables the package to be transferred to the PC and de-compiled and dis-assembled into its source code for analysis.

5.1.1 Software Environment

The software is freely available and can be installed on either a Windows or Linux PC. This environment used a PC running Windows XP.

Tools were required to de-encrypt and dis-assemble the compiled code into a readable format, so that the source code was in a readable format and the Manifest (permission request description file) could be accessed.

Tools are required for performing the transfer of the app and to extract the permissions from the binary files.

The software used was:

- Android Development Kit (ADK) –
 - Android Virtual Devices (AVB),
 - Android Debug (ADB),
- Java Development Kit (JDK) and a Java graphical interface – JDGUI (JDGUI Download, 2011),
- Eclipse (Eclipse Download , 2011),
- Software Development Kit (SDK),
- a .dex decompiler – DEX2JAR (Dex2jar Download page),
- a reverse engineering tool – APKTOOL (Apktool downloads , 2011)
- a script programming - PYTHON (Python Downloads, 2015)

Linux only tools

- SANTUKO performs package analysis
- DROZER –analyses the interaction between apps.

Install the tools from above. (use the recommended links). ADK creates a virtual machine with the same characteristics as the device (AVD). The package

is downloaded onto the PC via the debug function (ADB). The package has a suffix of .apk. The preparation of the package for analysis is performed. The downloaded package apk file is decompiled using dex2jar, which creates a java compiled file. This .JAR file is dis-assembled using the JDK into Java code (or the eclipse product can be used to perform the dis-assembly and provide the code in Java source available for editing). Once de-compiled and dis-assembly the package is available for analysis.

APKTOOL is used to extract and decode the Manifest file and placed in a readable format. The file is now ready for analysis. A python script is used to extract the permissions from the manifest file for the comparison analysis.

5.1.2 Installing the app onto a Clean Device

Ensure that you have a clean operating system; reset the mobile to the factory defaults and clear the storage by re-formatting the storage card. The instructions to perform this are available in the user manual. To factory reset a T-mobile G1 perform the following steps;

1. Power off the G1
2. Hold the Home key and the End key simultaneously for at least 20 seconds or until the G1 displays a triangle, an exclamation point and a picture of the G1.
3. Open the QWERTY keyboard and press ALT and W

The device is now restored

To perform a soft reboot (general reset) replace step 3 by pressing the HOME and Back buttons simultaneously.

(This can be done on a PC or Laptop if there is no option on the mobile).

5.2 Rooting an Android Smartphone

Some Android users that wanted more control of their device and the ability to obtain apps from locations other than the Play Store, “rooted” their devices. Some of these locations were uncontrolled and some apps contained malware masquerading as a genuine App. Gordon Kelly extracted from a report by F-Secure in 2013 the results tested the Google Play store and found that only 0.1% of apps were infected with malware. This contrasts with other 3rd party sites tested; Mumayi – 6%, AnZhi – 5%, Baidu – 8%, oeoMarket – 7%, liqucn – 8% and in Android159 33% of apps were infected (Kelly, 2014).

Rooting an Android smartphone consists of removing the original Android OS and “skin” provided by either the smartphone manufacturer or the mobile network provider and replacing it with an “open” OS provided by a 3rd party, for example CyanogenMod. The open OS removes the sandboxing security feature of the OS by permitting the user to have “superuser” access to the device. This enables the user to upgrade the OS to a newer version, install or customise skins and to install apps from multiple providers.

To perform the rooting the user must uplift their access to be a privilege user (“Superuser”) of the device as the smartphone ROM must be accessed. During the process the original OS on the ROM is removed and replaced with the open

OS. The following example of upgrading to a higher version OS and rooting an Android smartphone is using the CyanogenMod ROM, although other developers' ROMs are available.

5.2.1 Root the Device

The rooting and upgrading the device requires the device to have a specific firmware. The European firmware is RC7. The firmware file is DREAIMG.nbh and is stored in the root of the device's SD card. To load this firmware the device is powered off and then rebooted by pressing the Power and Camera button simultaneously until the device enters bootloader mode. The on-screen instructions guide you through the process to flash the new firmware image. When this is complete you are requested to press the "trackball" on the keyboard. You then need to reboot the device with the new image, this is done by pressing the Call, Menu and Power buttons simultaneously.

Rooting the device is performed via a Telnet session and there are two methods available to install and start telnet.

5.2.2 Method 1 – Using the Setup Utility

1. Finish booting up the G1 & sign-in to a Google account.
2. Once at the Home screen, open the keyboard and press the **Enter** key twice.
3. Type `telnetd` & press **Enter**. The Contacts screen will come up, just ignore it. **There will be no indication that you did it right.**
4. Open the Android Market and install [Telnet](#) by ClockworkMod.

Alternatively, you can install Telnet from the device's browser. First, go to Settings » Applications » and check Unknown Sources. Then, from the device's browser, go to <http://koushikdutta.blurryfox.com/G1/Telnet.apk>. Wait for the file to download, then tap on icon to install it.

5. Open Android Telnet Client; type `localhost` in the large text box and `23` in the smaller text box on the right. Press Enter.

5.2.3 Method 2 - Using a PC

This method is used if there are any connectivity issues signing into the Google account.

1. Enable a WiFi connection and connect to your local home network
2. On the Android screen, type `<enter>telnetd<enter>`
3. On the PC, open a new Command Line
4. On the device, press on your connection to know your local IP
5. Back on PC, type this to the fresh command line: `telnet [your_local_ip]`. This should connect you to the device, and you should see this: `# #`
6. If it's right, copy these lines and press enter on the end of them:

```
mount -o rw,remount -t yaffs2 /dev/block/mtdblock3 /system
dd if=/system/bin/sh of=/system/bin/su
chmod 4755 /system/bin/su
```

7. Type `su` in the console. The correct response is a new line (`# #`). If the response is "permission denied", repeat step 6.

5.2.4 Custom Recovery Image

To install the recovery image, the image must be flashed, and this is done via the Android Telnet client (see earlier step) and entering the flash command.

```
flash_image recovery /sdcard/recovery.img
```

Once the recovery file has finished installing, `# #` is displayed on the screen below the command and Amon_Ra's Recovery image is now installed.

The recovery image needs to be installed and this is performed by activating the Radio update.

Boot the device into recovery mode (press Home and Power buttons) The device prompts you for the boot type, scroll down and select Flash zip from SD card. Select “radio update.zip”. The G1 will reboot to install the update. Once the update is finished, select “Reboot system”.

The device is now “rooted” and has recovery image and radio update installed in preparation of the OS version upgrade.

5.3 Upgrading to Android Version 2.2 (Froyo)

The T-Mobile G1 released in the UK in the 2008 was a re-badged HTC Dream G1 and was sold to consumers with the HTC skin with the original Android version 1.0. This example will describe the process of upgrading the OS from version 1.5 (Cupcake) to version 2.2 (Froyo) onto the G1. This upgrade was unsupported and was not available from the mobile suppliers or manufacturers of this device.

To perform the upgrade (install the Froyo ROM) the device will need to be at a specific firmware level and have custom recovery images installed to recover the original OS version.

The files are downloaded to a PC. The G1 is connected to a USB port on the PC in debug mode and the SD card is mounted. The SD card must be in FAT32 format. The files are then copied to the SD card’s root directory. The files to download are:

- CyanogenMod 6 Stable for the G1 (which contains the Google apps file for the version of Android)
- DREAMIMG.nbh (firmware file)
- recovery-RA-dream-v1.7.0-cyan.img (recovery image from Amon_Ra) renamed to recovery.img before copying to the G1.
- Recovery Radio file 2.22.19.26i (used with the recovery image to boot the device)

Once the files have been copied the G1 can be disconnected.

5.3.1 Upgrading the Operating System

Now that the user has root access and there is a recovery image, the OS can now be upgraded to a more recent or previous OS version.

- Reboot the phone in recovery mode and on the Backup/Recovery screen follow the instructions to do a Nandroid backup.
- The device's existing OS is deleted to enable the installation of the new OS, to do this;
 - From recovery, scroll down using trackball to Wipe or press ALT + W on the keyboard.
 - Select Wipe Data/Factory Reset and press home to confirm the WIPE.
- Once all data and cache has been wiped, return to the main recovery menu and navigate to Flash Zip from Sdcard option. Press trackball and the installation will commence. (Note: switching off the phone

at this stage of the installation will cause the phone to be “bricked”³ and therefore be unusable).

- After the installation is complete, install Google apps by repeating the above procedure.
- Once the Google apps have been installed, navigate to Reboot Your System Now and press the trackball.
- This first reboot will take some time. The device will then start the normal setup for the Google instructions to complete your account setup (This step can be skipped for later).

To verify that the OS has been upgraded to Froyo, select *Menu > Settings > About Phone* and the Android version should display as 2.2 Froyo with the build number FRF91.

5.3.2 Security Implications

Research by Luyi, X. et al (Xing, Pan, Wang, Yuan, & Wang, 2014) described the new challenges in updating the mobile’s OS. The length of time between updates being available and being installed provides the actor with a large window of opportunity to develop an exploit of the update installation process. Their study focussed initially on the Android package manager but can be applied to other internal updaters. The study highlighted a how unprivileged malicious apps can acquire system capabilities after the OS has been upgraded and to be unnoticed by the user. These vulnerabilities, which

³ Bricked is a term used to denote that the device is permanently unusable. The device is unable to boot and it has the same value and usability as a brick.

Test environment

they called *Pileup* (privilege escalation through updating) exploits the OS updating system not an app and can therefore create new permissions for malware to exploit. Manually updating the OS as described in section 5.3 makes the OS more vulnerable to attack as the time that the update is available is far greater than for normal updates.

The limitation of this test environment is the manual steps to move the apps from the download device to the PC and the steps required on the PC to prepare the app in a format for analysis. The manual interaction is time consuming and an automated process is required.

The tools available in 2011 to perform this download and preparedness were very limited and the process used was like performing a reverse engineering of the App.

Although this process was used in the initial extraction and analysis an automated method was developed and used in future analysis (see PEMP in Chapter 7). The results from the manual process was used as a comparable base when the automated method was tested on the original set of apps.

This section described the test environment and the following section details the steps to obtain and analyse Antivirus apps.

Chapter 6. Analysis of Antivirus Apps

The previous section detailed the test environment required to test the Antivirus apps and this section concentrates on the test process of the apps. The process is described for obtaining and testing the apps available in 2011 and 2015 and their selection criteria. The results obtained in each year's group and makes a comparison between the two sets to show the evolution of Antivirus apps from 2011 to 2015.

6.1 app Status in 2011

Articles and white papers are available to assist consumers and enterprises in choosing Antivirus software to secure standard computing equipment; laptops, netbooks, desktops, etc. This comparative information was not available in the mobile sector (Smartphones, e-readers, iPads etc.) The increase in acquisition of these devices has far outstripped the growth of legacy platforms. Additional issues were introduced as the users of the devices either do not know or do not care about the potential security vulnerabilities of the devices and the increase in criminal activity targeting these devices.

This chapter explores the variety of security and anti-virus tools that were available for installation on Android mobile devices. There were many

products which were available as either Free or Commercial applications, but this research focuses on the products available for Android devices available as both Free and Subscription (commercial) variants.

These variants were then compared to find the differences that could have been used as the criteria to determine the difference that were used to determine the product availability as free or require a charge.

6.2 Android Antivirus Apps in 2011

There are a variety of antivirus and security protection products for Android mobiles. The difficulty occurs in deciding which product to use and whether it is effective in protecting the device and owner data. There are a variety of sites where Android applications can be obtained. The best known is the Google Play Store (<https://play.google.com/store/>), some other known app providers are; Amazon (<http://www.amazon.co.uk/appstore>), Phandroid (<https://www.phandroid.com/>), the Android Freeware store (<http://www.androidfreeware.net/>) Android Software Download store (<http://androidsoftwaredownload.com>), Androlib marketplace (<http://www.androlib.com>) or Best Android downloads (Best Android Downloads, 2011) which uses the iliVid Download Manager.

Tripwire magazine also compiled a list of the top 15 best websites for Android app downloads (Angus, 2011).

A thorough analysis of the Android marketplace antivirus and security protection applications was performed. The criteria for the included security applications were that it had to have an Antivirus component. The Google Marketplace (<https://market.android.com/>) contained 37 security products

that had either security or antivirus in their names or contained them as keywords in their descriptions.

The initial analysis was to determine the number of free applications available, how popular the apps were by their download count and the rating submitted by users of the tools' performance or ease of use and to compare free and commercial Antivirus apps to determine if there were differences in their efficacy.

The highest downloaded free security apps, according to the Androlib Market site (<http://www.androlib.com>) on February 28th, 2011, that contain an Antivirus component and the developer are contained in Table A-1. Details of the user rating and the number of reviews and downloads are recorded. The number of reviews as a percentage of the download were calculated to determine if the rating value was a true representative of the users downloading the product. The lower the figure indicated that more users that download the product provided a rating. This was then used to rank the apps.

The list of applications was used to determine if the supplier also provided a similar commercial version, for which the user either paid a one-off or a regular subscription charge like Antivirus products in the PC/Laptop world. This incorporated small changes to the product list. Some suppliers only concentrated in providing free applications and there were also additional suppliers who did not provide a free version of their application but did offer trial periods or paid versions only. Six of the suppliers provided free and commercial versions.

6.2.1 Investigation Method in 2011

Initially the permissions and features were compared between the original free apps and their commercial variants. The null hypothesis was there were no major differences between the free Security products that contain an Antivirus component and their commercial versions.

The materials used in the study were the free and commercial versions of the Antivirus programs available for download to Android mobile devices, primarily Smartphones, from the Google Marketplace (<https://market.android.com/>)⁴. The data about features and permissions have been obtained from either the supplier web site or from various online Marketplace libraries and search engines such as; Androlib (<http://www.androlib.com>) Android Market (<https://market.android.com/>), Cyrket (<http://www.cyrket.com/m/android/>) or Android Zoom (<http://www.androidzoom.com>).

6.2.1.1 Procedure

The initial task was to remove the applications that are presumed to be unique, these were the applications which were only available as a single version, either free or fee paying. For the investigation a trial version is a fee-paying version if once the trial period has expired users are required to pay for a monthly or yearly subscription to continue using the product and users do not have to perform any additional downloads to the trial product.

⁴ Applications are also available from other locations, but were not used in this case

The applications that are included in the comparison study are those security programs that are available as both free and paid versions.

Table 6-1 The list of companies that provide the security apps, grouped by version type.

Both Free and Commercial	Free Only	Commercial only
Lookout Inc	Creative Apps	McAfee
AVG Mobilation	NetQin Mobile Inc.	MyMobile Security
Doctor Web Ltd	SuperDroid.net	UMU Ltd
AegisLab	Hauri Inc.	DMA
BluePoint Security Inc.	TrustMobi	Livezen
MoonBeam Development	CPU Media Sarl	P Defender
	ShipWreckTech	
	Qianjun	

The table shows that there were 6 suppliers that provided free variants of their commercial applications, either as basic or Lite versions and it was these products that formed the base of the comparison testing.

6.2.2 Obtaining a Commercial app for Testing Without Incurring a Cost.

To compare the free and commercial variants of the app to determine if the paid version provided additional facilities, required the purchase of the paid version of the App, which ranged from £0.85 to £19.95. As the app was needed as input to a comparison and not for use as an Antivirus product, therefore an alternative method was needed to acquire the app for no cost.

After downloading the app and transferring it to the PC, the author inadvertently selected the option to reject the payment on the mobile device. The app was then removed from the device but was still available on the test PC for analysis. Further investigation revealed that Google provided a 15-minute window in which a user could refuse/reject the App. The app was then removed from the device as part of the reject process. This did not affect the app stored on the secondary device.

The author then used this process to obtain commercial apps for the investigation. The method to obtain the app is as follows.

1. Download the app to the mobile device (I used a T-Mobile G1 device), Agree to the payment as part of the download process.

Using the debug function in eclipse transfer the Davlik module to the PC. Save this module for input to the dis-assembly and comparisons.

On the device reject the app after download. The rejection kicks off the automatic process to remove the app from the phone and the user's account is not charged.

Note: The user only has 15 minutes to transfer the app to the PC and reject the app on the phone, otherwise they will be charged.

6.2.3 Selected Security and Antivirus Developers

Security product suppliers are dependent on their Google Store ranking to encourage users to download and install their product. The companies use a variety of ways to do this. Some provide free apps and then offer in app purchases, a method where the user is required to make an additional payment for increased functionality, or the free app contains ads to entice the user to buy other apps or other products or services, the developer then receives a

recommendation fee if the user selects the ad. Other companies use a list of functions/features that their app contains hoping that this will differentiate them from other providers.

An overview of each of the six suppliers from the study and a brief description of the two security products, the free and the commercial app, as provided by the suppliers are described below.



Lookout Inc <https://www.mylookout.com/> has two products that are included in the comparison, both are called Lookout Mobile Security, but the premium or commercial application is only available as an upgrade. The premium application includes a Privacy Advisor, additional backup/restore capabilities and the ability to remotely wipe and lock the device.



Lookout Free

Security

- Block malware, spyware, and phishing apps
- Scan every app you download to ensure it's safe
- Schedule daily or weekly security scans
- Automatic protection against the latest threats
- Prevent a virus from transferring from your phone to your PC
- Doesn't drain your battery

Find My Phone

- Phone Locator: Locate your lost or stolen phone on a Google map
- Activate a loud alarm, even if it is on silent
- When possible, Lookout will remotely enable GPS to help you find your phone even if GPS is turned off
- Log in to myLookout.com from any web browser to locate your phone

Backup and Restore

- Securely backup your contacts
- Restore your backup data to an existing phone
- Access all backed up data securely at myLookout.com
- Lookout is certified by TRUSTe (privacy and data are protected)



Lookout Premium contains all the features in Lookout free and the following additional features:

- Remote Lock, a security lock for your phone to lock others out. Set a secret passcode to unlock your phone.
- Remote Wipe to delete your data from logged-in accounts like Facebook, Twitter, Gmail, and YouTube. Delete contacts, SMS text messages, photos, call log, web browser history, calendar, sync settings, and full SD card data. With enhanced protection, you can do a full factory reset.
- Privacy Advisor to identify which apps can access your personal data such as contacts, location, SMS text messages & identity information
- Additional Backup of photos & call history; restore your backup data to a new phone

- Premium Support for priority response



AVG Mobilation <http://www.droidsecurity.com/> is a joint venture between AVG and Droid Security and has three products on the market, Antivirus Free, Antivirus Pro and Security Pro. All products are available from the Marketplace, and all have the same file size.



Security

- Scan whole device and identify and remove viruses with a simple click
- Automatic scans can be run weekly, daily, or on demand
- Check apps for malware before downloading from app stores
- Check website content, emails, and SMS for malware before downloading to device

Theft protection

- Locate lost or stolen device using GPS
- Create and display message on screen remotely
- Lock device and wipe content
- Manage applications remotely
- SMS Spam Protection provides basic protection from SMS Spammers



Anti-Virus PRO has all the features of Antivirus FREE, plus: Premium SMS security, whose feature set includes

- All SMS checked in real time for malicious content and spam
- SMS spam blocked at source

Anti-Virus PRO customers receive premium level support whenever they need it

Anti-Virus PRO is free of advertising and other disruptions



Doctor Web

<http://products.drweb.com/mobile/?lng=en> has two Antivirus products for Android devices. The light version has a smaller file size than the commercial version and does not contain SMS filtering.



Dr. Web Anti-virus Light scans the file system of your Android device, including the "hidden" area and user applications. Detected malicious objects are moved to quarantine. A real-time file monitor automatically scans applications being installed and all files written to the SD card.

The feature set consists of:

- Non-stop anti-virus protection. Non-stop, real-time file system scanning.
- On-demand scanning. Scan options are either fast or full file-system scans as well as scan individual files and folders.
- Filtering mode selection.
- Black list editing. Block incoming calls and messages from certain numbers.

- Filter creation. Dr. Web Anti-virus lets you configure custom filtering modes for calls and messages.
- Viewing of blocked calls and messages.



DR Web Antivirus has all the features of Anti-virus light with an additional anti- SPAM feature to filter and block SMS messages.



Aegislab <http://www.aegislab.com/> have two products, Aegislab Antivirus Elite and Aegislab Antivirus Free, which were previously known as Appscan beta. The commercial variant has a larger file size than the free version and requests more permissions. The cost of the commercial variant is the yearly subscription for the application.



Aegislab Antivirus Free identifies Spyware/Malware. Supports advertisement detection (especially from Admob). Provides network/traffic statistics for both mobile and WiFi interfaces to assist finding suspicious background usage.



Aegislab Antivirus Elite has the following feature set;

- Real Time and Manual cans
- Remote lock
- Remote Data wipe
- Search/Query signature database prior to downloading
- SMS check for Phishing



BluePoint Security Inc.

<http://www.bluepointsecurity.com/presentationlayer/pages/home.aspx>

has two products Antivirus Free and Antivirus Pro. The commercial version has a much larger file size and incorporates additional settings and scheduled scanning. The company utilises a cloud-based Antivirus database to detect all viruses not just phone viruses.

Bluepoint Antivirus Free Features

- Realtime protection
- Battery efficient
- Automatic scans of email, SMS and other downloads
- Scan memory cards
- Uses a cloud-based Antivirus engine

Bluepoint Antivirus Pro contains the same features of Antivirus Free with additional settings and the ability to schedule scans.



MoonBeam Development <http://moonbeamdevelopment.com/> has two products Android Defender Virus Protect (free) and Defender Pro Virus (commercial). The commercial version has a larger file size than the free version.

Key features are:

- Block viruses, spyware and malware
- Scan all apps installed on device
- Scan new apps when first installed

6.2.4 Comparative Analysis Results

Comparative analysis was performed of the twelve security products from the six companies to determine if there were any differences between the commercial and free versions. Each product was investigated for the following; feature set, Android permissions, other permissions, ratings (popularity) and file sizes. The results of the comparisons are summarized in Figure 6-4, which displays the features of the tested Antivirus products, their requested Android permissions, other permissions and user ratings.

Each product (free and fee paying) was investigated for the following:

- Android File permissions requested
- Other Permissions requested
- Features
- User rating
- Antivirus function efficacy

The detailed results of the comparisons are provided in the tables and figures in the appendix. These are comparisons of the product features (Table A-3), the Android permissions requested (Table A-4), additional permissions (Table A-5) and user rating of the product (Table 6-2).

Additionally, a comparison of the package sizes was performed to determine any variations between the free and commercial versions of the apps (Table 6-5).

Finally, the apps were tested to verify that they performed the Antivirus function through detection and removal of malware.

Figure 6-1 Android permissions requested by each app.

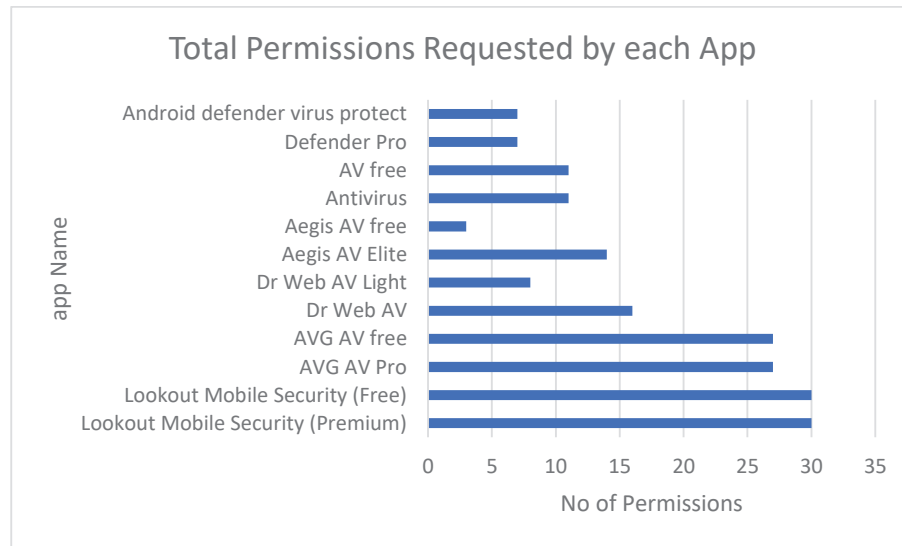


Figure 6-2 The figure shows the total permissions requested by each Antivirus app.

A summarization of the total permissions and the number of features is shown in. Figure 6-3 shows the Android permissions requested and the user rating.

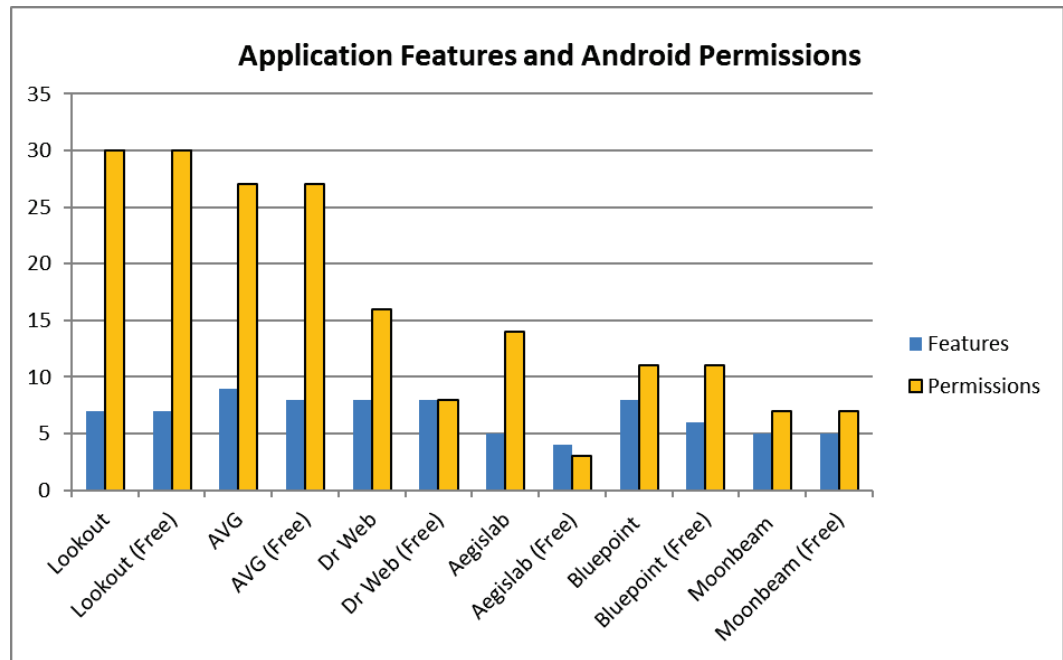


Figure 6-3 Application Features and Android Permissions

The figure indicated that there was no correlation between the permissions requested and the number of features, which needed to be researched.

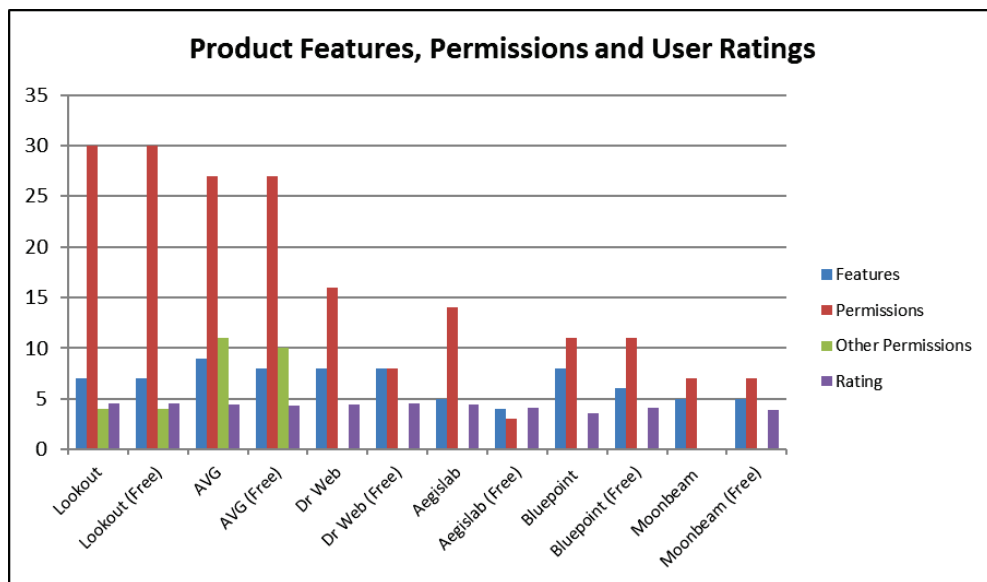


Figure 6-4 Features, permissions and ratings for each product

Figure 6-4 shows the total features and permissions for each of the products with any additional non-Android permissions and the user rating. The expectation was that the more features that an app has, then this should be reflected in the increased number of permissions requested and that the more features defined then the higher the user rating.

Of the six suppliers in the analysis, three (Lookout Inc, AVG Mobilation and BluePoint Security Inc) used the same Android permissions on both the commercial and free applications. Two suppliers (Lookout Inc. and AVG) requested non-Android permissions, whilst the other suppliers only requested Android permissions. Of the non-Android permissions, Lookout Inc. used the same permissions on both products, whilst AVG performed License checking and used different C2D_MESSAGE permissions between its PRO and Free versions. The user's rating of the product was obtained from the Androlib market site. Bluepoint used only Android permissions.

Two of the developers requested additional permissions, Lookout Mobile and AVG. The permissions requested by each of these developer's apps are recorded.

The analysis of the ratings was similar irrespective of the number of features of the app. The reviews as a percentage of the downloads was calculated to determine if there was any correlation between the number of reviewers rating the app and the number of downloads (Table 6-2). The range of the result demonstrated that there was no correlation.

Table 6-2 User ratings for the six suppliers

<i>Company</i>	<i>Application</i>	<i>Rating (out of 5)</i>	<i># of reviews</i>	<i># of downloads</i>	<i>Reviews as a % of downloads</i>
Lookout Inc	Lookout Mobile Security	4.59	169987	20,587,202	0.83
Lookout Inc *	Lookout Mobile Security Premium	-	-	-	-
AVG Mobilation	AntiVirus Free AVG	4.36	98907	13,082,544	0.76
AVG Mobilation	AntiVirus Pro	4.44	2221	50,000	4.44
AVG Mobilation	Security Pro	4.31	356	7,739	4.60
Doctor Web Ltd	Dr Web Anti-virus	4.43	69	515	13.40
Doctor Web Ltd	Dr Web Antivirus light	4.57	19267	1,177,978	1.64
Aegislab	Aegislab Antivirus free	4.43	126	10,000	1.26
Aegislab	AntiVirus Elite	4.09	11	157	7.01
Bluepoint security Inc	BluePoint Antivirus	4.12	321	14,793	2.17
Bluepoint security Inc	BluePoint Antivirus	3.56	36	720	5.00
MoonBeam Development	Android defender virus protect	3.89	66	10,312	0.64
Moonbeam Development	Defender Pro virus^	0	0	49	0.00

The user rating for each product (Table 6-2) was obtained from the Androlib market site. Data was not available for the Lookout mobile premium App; however, it was possible to obtain the premium version by upgrading from the

free version. Defender Pro virus was removed from the Marketplace 23/01/2012.

A statistical analysis was performed to determine if the rating was related to purchase price of the product. The resultant means, and standard deviation is shown in Table 6-3 and a box graph showing the overlap is in Figure 6-5.

Table 6-3 Rating of app by app type (free or commercial)

Status	Mean	Std. Deviation
Free	4.33	0.27
Commercial	3.51	1.78

The analysis showed that overall the free products received a slightly higher mean user rating than the commercial products and the standard deviation shows that the user ratings of the commercial products had a greater range than the ratings for the free products.

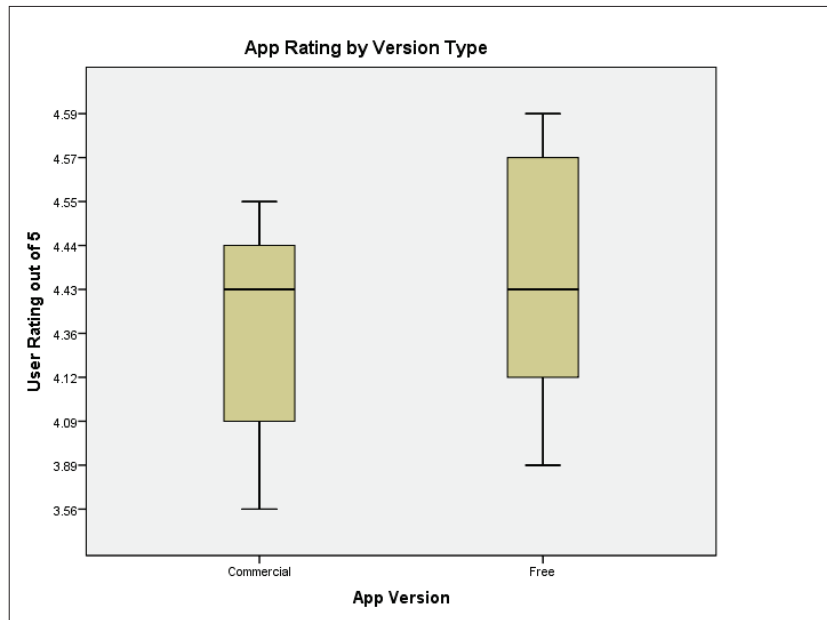


Figure 6-5 The rating of the app by type.

The figure displays the user rating of the app by type, commercial or free. Free apps tended to have a slightly higher rating than the commercial apps.

Next analysis was to determine if there was any relationship between the number of features and the number of requested permissions. A simple bivariate plot of the two variables by version is in Figure 6-6. The cluster analysis produced an unexpected visual analysis. The plot shows a positive relationship between the number of features and number of permissions although grouped into clusters and there appeared to be no relationship to the version of the app (free or commercial).

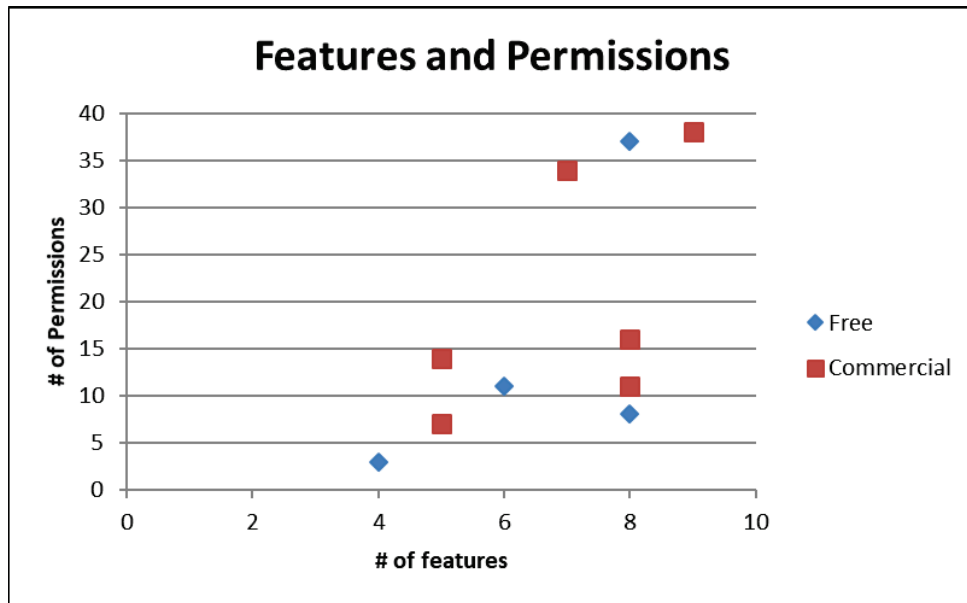


Figure 6-6 Cluster analysis of relationship between Features and Total Requested Permissions

The correlation value for the relationship was calculated. The result for the 12 cases was 0.61, which is a strong relationship. A significance test was then performed to determine the probability that this relationship had occurred by chance. Using an alpha level of 0.05, the critical value for $df=10$ is 0.576, therefore as the correlation coefficient is 0.61 the relationship is not a chance occurrence and is statistically significant.

The final analysis was to determine if there was any relationship between the number of features and the user rating (Figure 6-7).

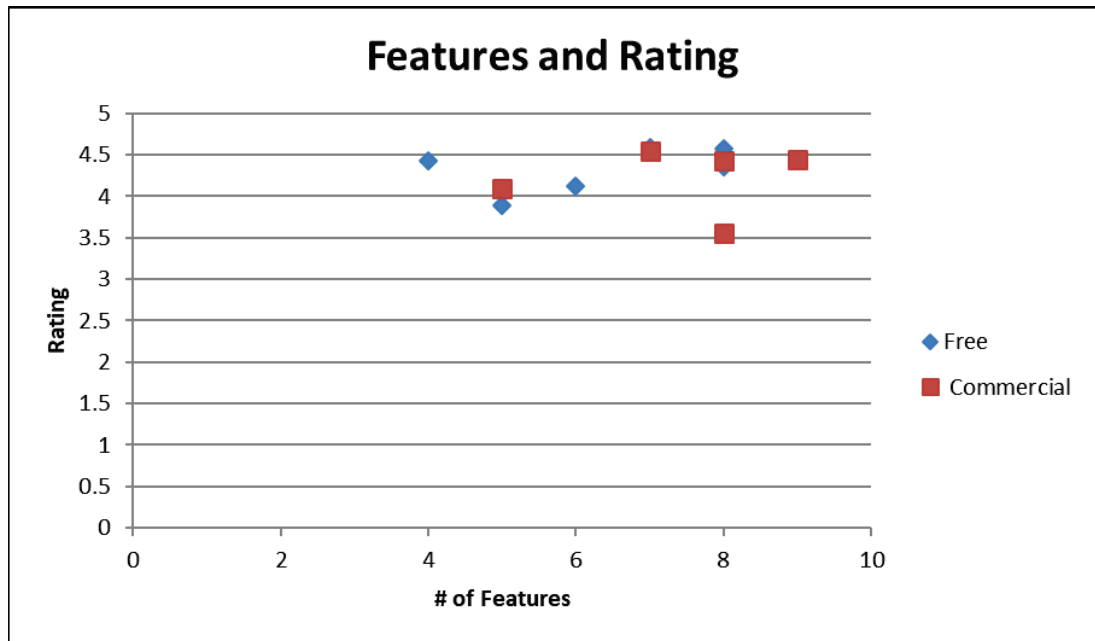


Figure 6-7 Relationship between features and user ratings.

The graph did not display any relationship between the features and user ratings, so a Spearman's rho correlation was performed to determine if there is any relationship between them (Table 6-4).

Table 6-4 Correlations of features and user rating

Spearman's Rho		Feature	Permissions
Feature	Correlation Coefficient	1	.376
	Sig. (2-tailed)		.228
	N	12	12
Rating	Correlation Coefficient	.376	1
	Sig. (2-tailed)	.228	
	N	12	12

The resulting coefficient was 0.376 which is lower than the critical value of 0.576 for the requisite degrees of freedom and therefore there is no correlation between the rating and features as shown in Figure 6-7.

6.2.5 Review Program Source

The next step was to review the program source of the Antivirus App. This involved downloading the app to a smartphone, in this case a T-mobile G1, and then transferring this package in Davlik format to a PC for analysis.

The transfer of the package required the smartphone to be connected via USB to the PC and Android app developer tools installed on the PC.

The tools required for the transfer of the app from the mobile device to the PC and the software required for the dis-assembly to the source code are described in section 5.1.1

Initially the package was converted from Davlik into compiled Java and then de-compiled to Java source code. Table 6-5 provides a comparison table of the program sizes of the decompiled packages. Packages that are signed and are therefore protected from disassembly were supplied in a non Davlik format (zipped XML files), their file sizes are shown for information only.

**Table 6-5 Comparison of program sizes of the packages as downloaded
27/05/2011**

<i>Application</i>		<i>Cost</i> <i>(GBP)</i>	<i>Package name</i>	<i>Davlik</i> <i>Size</i> <i>(KB)</i>	<i>Java</i> <i>Size</i> <i>(KB)</i>	<i>Zip</i> <i>Size</i> <i>(KB)</i>
Lookout Security	Mobile	0.00	Com.lookout-1.apk	1335	862	
Lookout Security Premium	Mobile	18.51	Com.lookout-1.apk	1335	862	
AntiVirus Free AVG		0.00	Com.antivirus-1.apk	1349	677	
AntiVirus Pro		6.09	La.droid.gr-1.apk	1169	622	
Dr Web Anti-virus		3.68	Server error prevented purchase			
Dr Web Antivirus light	Antivirus	0.00	Com.drweb-1.zip			677
Aegislab free	Antivirus	0.00	Com.aegislab.sd3prj.antivirus.free-1.apk	670	233	
AntiVirus Elite		4.88	Com.aegislab.sd3prj.eigismobile-1.zip			966
BluePoint Free	Antivirus	0.00	bluepointfree.ad-2.apk	3476	211	
BluePoint Antivirus		3.09	Bluepoint.ad-1.zip	2813		
Android defender virus protect		0.00	Com.moonbeamdevelopment.riskdetec tor.android-1.apk	261	296	
Defender Pro virus		4.99	Com.moonbeamdevelopment.riskdetec torPRO.android-1.zip	56		

The table displays the sizes for each package (executable app name) as the Davlik executable component and then the size of the decoded Java source.

Both Lookout Inc. products had the same file sizes and the resultant md5 Hash showed that there was no difference between the files. This is possibly due to the free version acting as a trial version of the premium product and those features are in an inactive state.

Once the free version of Lookout Mobile Security was downloaded and activated the company offered the option of a 14-day trial of the premium version. There was no additional downloads or updates once the 14-day trial was opted for. An MD5 hash was performed to detect any differences between the free and commercial source codes.

- Free variant MD5hash
 - 41593367DF5FDBC8005F71048FC61E95
- Commercial variant MD5 hash
 - 41593367DF5FDBC8005F71048FC61E95

The two hashes were identical, and this indicated that the premium functions are not included in the package but were instead available as host (web) based functionality and are available as part of the user registration.

Note: I was unable to purchase the Dr. Web Anti-virus due to a server error on the 27th May 2011 during the purchase of the product. This also occurred on multiple occasions during that week. This prevented the comparison of the free and commercial versions.

6.2.6 Efficacy of Free Antivirus Apps

The free apps were tested to determine their efficacy and if the user would obtain more benefit (security) from buying the app rather than use the free version.

Each product defined in its feature list which Antivirus functions it could perform (Table 6-6).

Table 6-6 Antivirus apps and their described features

<i>Product</i>	<i>Real time monitoring</i>			<i>Device scan</i>	<i>Virus signature update</i>
	<i>At Download</i>	<i>Emails</i>	<i>SMS</i>		
Lookout mobile security (free)	√			√	√
Lookout mobile security (premium)	√			√	√
AVG Antivirus Free	√	√		√	√
AVG Antivirus Pro	√	√	√	√	√
Dr Web Anti-Virus	√		√	√	√
Dr Web Anti-virus lite	√			√	√
Aegislab Antivirus Free	√			√	√
Aegislab Elite	√			√	√
Bluepoint Antivirus Free	√	√	√	√	√
Bluepoint Antivirus Pro	√	√	√	√	√
Android Defender Virus Protect (free)	√			√	
Defender Pro Virus	√			√	

All stated that they would detect malware at download of an app and during a device scan.

Android defender virus protect, and Defender Pro virus did not use a virus signature database. This meant that they relied on using a heuristic method to detect malware which indicates that they need frequent updates to ensure that their detection method could detect the newer types of attack.

Four products monitored emails for malware, with two of them also monitoring SMS texts. Non-monitoring of SMS texts exposes the user to Man in The Middle (MITM) attacks. MITM attacks are used to intercept SMS messages before passing them on, thus obtaining one-time-passcodes (used by Financial institutes for mobile authentication) to access a user’s account or email password change links. This vulnerability exposes the user to identity theft and theft of assets and money.

The testing of the app was performed on the T-Mobile G1 device running Froyo. The testing was performed using the Antivirus verification method as described in section 4.2.

The results of the Antivirus function testing are summarised in Table 6-7.

Table 6-7 Antivirus funtion testing summary

<i>Task</i>	<i>Lookout</i>	<i>AVG</i>	<i>Dr Web</i>	<i>Aegislab</i>	<i>Bluepoint</i>	<i>Moonbeam</i>
	<i>Mobile</i>					
Update Virus database	No	Yes, optional	Yes, optional	Yes, optional	No, database in cloud	No
Scan options	On demand	On demand	3 options	On demand	On demand	Automatically
Scan scheduling	manual	manual	manual	manual	manual	manual
Virus detected (number out of 2)	2	1	1	2	2	1
Adware detected	No	No	No	Yes	No	No
Root/Superuser app detected	No	Yes	No	No	No	No
Malware detected during download	Yes	Yes	Yes	Yes	Yes	No
Malware detected during install	Yes	Yes	Yes	Yes	Yes	No
Malware removal or quarantine	Yes	Yes	Yes	Yes	Yes	Yes

Full details of the testing results are in the Appendices.

6.2.7 Results

The products Antivirus functions were very similar, but there was a difference in the quality of the applications in comparison to classical Antivirus products available on PCs and Laptops. There were differences in the permission requests and file sizes of the selected products and most suppliers provided additional functions or features to their suite of security products to differentiate them from competitors. These functions were mainly backup/restore utilities, location and data removal utilities and these were primarily included in the commercial variant of the product. These functions required user registration.

Testing the Antivirus function of the 6 free Antivirus apps that were analysed with their commercial version resulted in none of the free versions fulfilling the full requirements of an Antivirus product.

Three products detected both viruses and the majority detected malware during download. Only 1 detected adware and another detected that the device had been rooted. Quantifying the results against the required function showed at best a 75% match to the required functions, with one app detecting an installed virus and nothing else.

Essentially an Antivirus app should detect and remove malware, but sadly this was not the case. As the free and commercial versions had no variation for the

on-device Antivirus functions, there is no benefit to the user to purchase the product unless they desire the remote or cloud facilities.

6.3 Android Apps in 2015

Since 2011 the Android operating systems has increased its share of the smartphone market, so by 2014 it had over 80% of the market-share. 2011 data is provided by Canalys (Canalys, 2011). IDC investigated the growth of the smartphone market (“IDC,” 2011). By 2015 the three main operating systems were Android, iOS and Windows phone (Table 6-8). The market share of the Android operating system has grown by 160% from 2011 (51.6%) to 2015 (82.8%).

Table 6-8 Smartphone OS market share growth

Operating system	2011Q4(1)	2012Q2	2013Q2	2014Q2	2015Q2
Android	51.6%	69.3%	79.8%	84.8%	82.8%
iOS	23.4%	16.6%	12.9%	11.6%	13.9%
Windows Phone	1.6%	3.1%	3.4%	2.5%	2.6%
Blackberry OS	8.3%	4.9%	2.8%	0.5%	0.3%
Others	15.1%	6.1%	1.2%	0.7%	0.4%

The table shows the market-share of each of the main smartphone operating systems from 2011 to 2015. Between 2011 to 2014, the Android OS grew in market-share at the expense of iOS, Blackberry OS and other proprietary operating systems like Symbian. The only operating system to recover was iOS and between them Android and iOS had over 96% of the market.

As the most ubiquitous Smartphone OS (operating system), Android had become the main target for attacks (Table 6-8). As an open source OS, the availability of the source code was one of the drivers of the take up of the OS. Most smartphone manufacturers installed Android on their hardware and installed their own front end, called a skin, on top of the OS. This gave a different “feel” to each manufacturer’s device. This feel also created hardware manufacture loyalty and the Android smartphone market become delineated by the manufacturer of the device rather than by the level of the operating system. The only “native” Android device was manufactured by Google (the owner of the operating system) and was the Nexus series of smartphones, phablets and tablets.

One of the problems with the diversity of hardware manufacturers and the range of devices was the delay in updating the software. Software updates became dependent upon the manufacturer’s schedule rather than on the new releases of the operating system. This left the operating system increasingly vulnerable to more malware as actors had more time to create and or adapt malware.

Some manufacturers “pushed” the updates out to the smart phones within a short time of the new release or version, whilst some either did not publish an update or if they did, they left it to the carrier to “push” the update out. This meant that the marketplace had a great variety of levels in circulation as can be seen in Figure 6-8. Smartphone Android version distribution figures and release dates were provided by IDC (“IDC,” 2011).

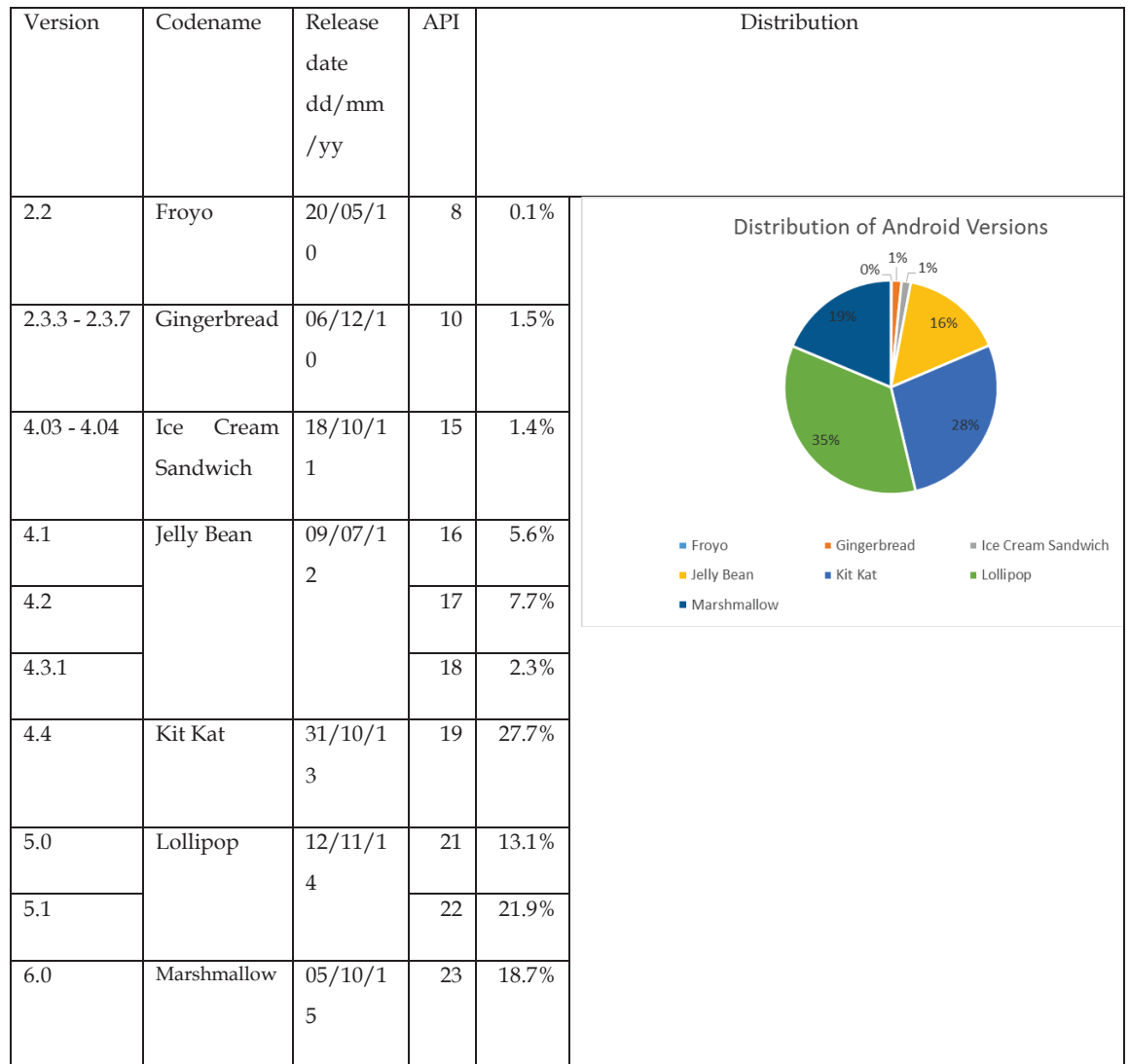


Figure 6-8 Distribution of Android versions as at 5th September 2016

The most common version (Kit Kat) is two levels behind the latest release (Marshmallow). The exception to this was Google’s Nexus devices which were updated when (or shortly after) the new release was published. The figures for Honeycomb (version 3.0 – 3.2.6) are not included as this was a tablet only operating system, released 22nd February 2011. The figures for Nougat (version 7.0) which was released 23rd August 2016 are not yet available.

At each new release permissions were added or deleted and updates to apps running on the devices were also subject to performing updates to incorporate the changes to permissions. Apps also required updates to resolve bugs or to make the app more attractive, for example; more levels in gaming apps, additional functions in business or lifestyle apps.

As app updates became more prolific and the reticence of Android users to purchase apps the app developers turned to adware to earn an income for the apps. Initially the user was offered a one-off charge to remove the adware but as the income from adware grew many developers moved away from this option. The exception to this were the major app developers who continue to provide their apps free. The Top Ten Mobile apps in 2017 as provided by comScore (Dan Frommer, 2017). Show that the most popular app was Facebook, closely followed by YouTube, two major social media sites.

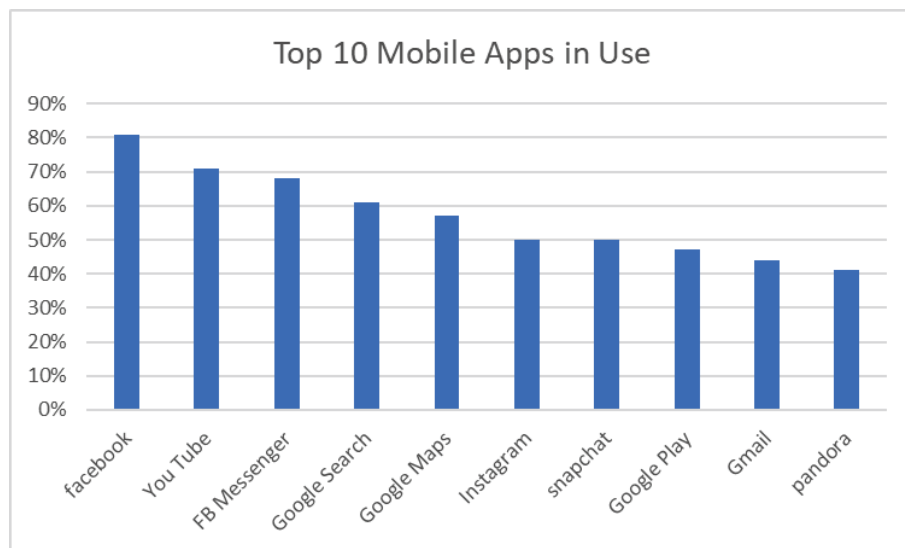


Figure 6-9 Top Ten Mobile Apps in the U.S. for 2017

The major use of social media sites is reflected in the Essential Apps that Millennials “said they couldn’t do without” according to comScore Whitepaper report on mobile apps (Lella & Lipsman, 2017).

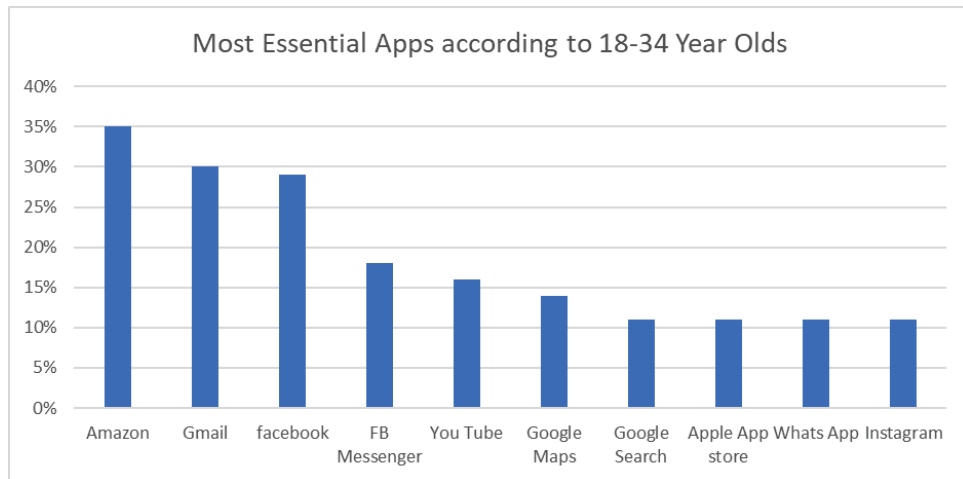


Figure 6-10 Most essential apps according to millenials.

Although Facebook and YouTube were the top two apps being used, they were only the third and fifth essential apps according to the 18-34-year olds.

The main interests and usage of millennials in social media sites and sharing data is in contrary to securing data and privacy concerns.

6.4 Android Antivirus Apps in 2015

As the Android OS grew in popularity so did the malware aimed at it. By 2013 when the Android OS held a market-share of 87%, it also accounted for 97% of all mobile malware (Kelly, 2014). The Antivirus and security apps developed for the Android OS to protect the user and remove malware from the device had also matured. The apps were available in two variants, free and commercial (which included both one-off or monthly payments). The commercial apps offered additional functionality (in some cases) see initial research into the comparison of free and commercial antivirus apps features and permissions in 2011/2012 (Chapter 5).

As the popularity of the Android OS grew many antivirus and security developers were bought by the mainstream Security software companies. The Antivirus arena on mobiles which was in its infancy in 2011 matured over the four years. The major providers of Antivirus programs from the PC/Laptop arena consolidated their position by purchasing or by merging with other companies, as in the case with AVG entering the mobile Antivirus market by purchasing DroidSecurity (Horn, 2010). This meant that multiple Antivirus products were available from one company, whilst the products were consolidated, incorporated into an existing product or dropped from the marketplace altogether.

The growth of apps with Antivirus components from 2011 to 2015 is shown in Table A-6.

In 2011 there were 22 apps with Antivirus components. In 2015 the number of apps with Security or Antivirus functions was 240, of which 67 were Antivirus apps. Developers use multiple tags or keywords to provide greater visibility of their apps during searches. The 240 apps contained the keywords “security” or “antivirus” or both. These apps were reviewed to confirm that they did possess an Antivirus component. In total 67 of the 240 apps performed Antivirus functions. (Table A-7)

This research added to the initial 2011 research and concentrated on analysing the permissions of the 67 Antivirus apps in 2015. The permissions and features from the initial 2011 Antivirus apps were available to perform comparison testing between the apps that were available in both 2011 and 2015, albeit at a newer release.

Of the 67 Antivirus apps the 64 free apps were downloaded and prepared for analysis. The app name, package name, developer, rating, number of downloads and size were recorded (Table A-8).

A summary of the number of permissions requested by each of the apps included in this study are shown in Figure 6-11 and the detailed tables of permissions requested are provided in the Appendix section A.4 Detailed Permissions of Antivirus apps in the study

6.4.1 Investigation Method - 2015

The materials used in the study were all security apps that contained an Antivirus component. There were 67 Security apps that contained an Antivirus component of which there were three commercial variants. Only the free apps were used in this study.

In 2011 there were 82 permissions specified for the Froyo version of Android. In Kitkat the number of specified permissions had grown to 154. At the time of testing six permissions flagged as no longer available were requested by eleven of the apps. The permission “android.permission.ACCESS_COARSE_UPDATES” was the most requested old permission (six times) and had not been superseded by another permission in the newer versions of Android.

6.4.2 2015 Security and Antivirus Apps

The permissions for the sixty-four free apps were extracted for analysis (Figure 6-11). The permission figures were then analysed. Four of the apps did not request any permissions and were ignored for the analysis as outliers. The maximum number of permissions requested was forty-nine and the least requested was four. Eighty-six percent of the apps requested between four and

forty permissions. Only five apps requested more than forty-one permissions and four apps didn't request any permissions.

The permissions requested were reviewed to determine if any old permissions were being requested. Old permissions were those designated as no longer valid in this version of Android. There were six old permissions that were being requested (Table A-9).

The requesting of these non-valid permissions could be due to a variety of causes, these include (but are not limited to); backward compatibility, incomplete code review or no code review or updates. The lack of code review indicates that the Antivirus is not being updated and is not protecting the device against new malware.

As previously the apps were checked to see that they were requesting Antivirus permissions and if any Anti Privacy permissions were also being requested.

Antivirus permissions;

```
android.permission.ACCESS_NETWORK_STATE
android.permission.CHANGE_NETWORK_STATE
android.permission.CLEAR_APP_CACHE
android.permission.DELETE_PACKAGES
android.permission.GET_TASKS
android.permission.INTERNET
android.permission.KILL_BACKGROUND_PROCESSES
android.permission.READ_EXTERNAL_STORAGE
android.permission.RECEIVE_MMS
android.permission.RECEIVE_SMS
android.permission.WRITE_EXTERNAL_STORAGE
```

Anti-Privacy permissions;

```
android.permission.CALL_PHONE
android.permission.GET_ACCOUNTS
android.permission.MANAGE_ACCOUNTS
android.permission.READ_CONTACTS
android.permission.WRITE_CALENDAR
android.permission.WRITE_CONTACTS
```

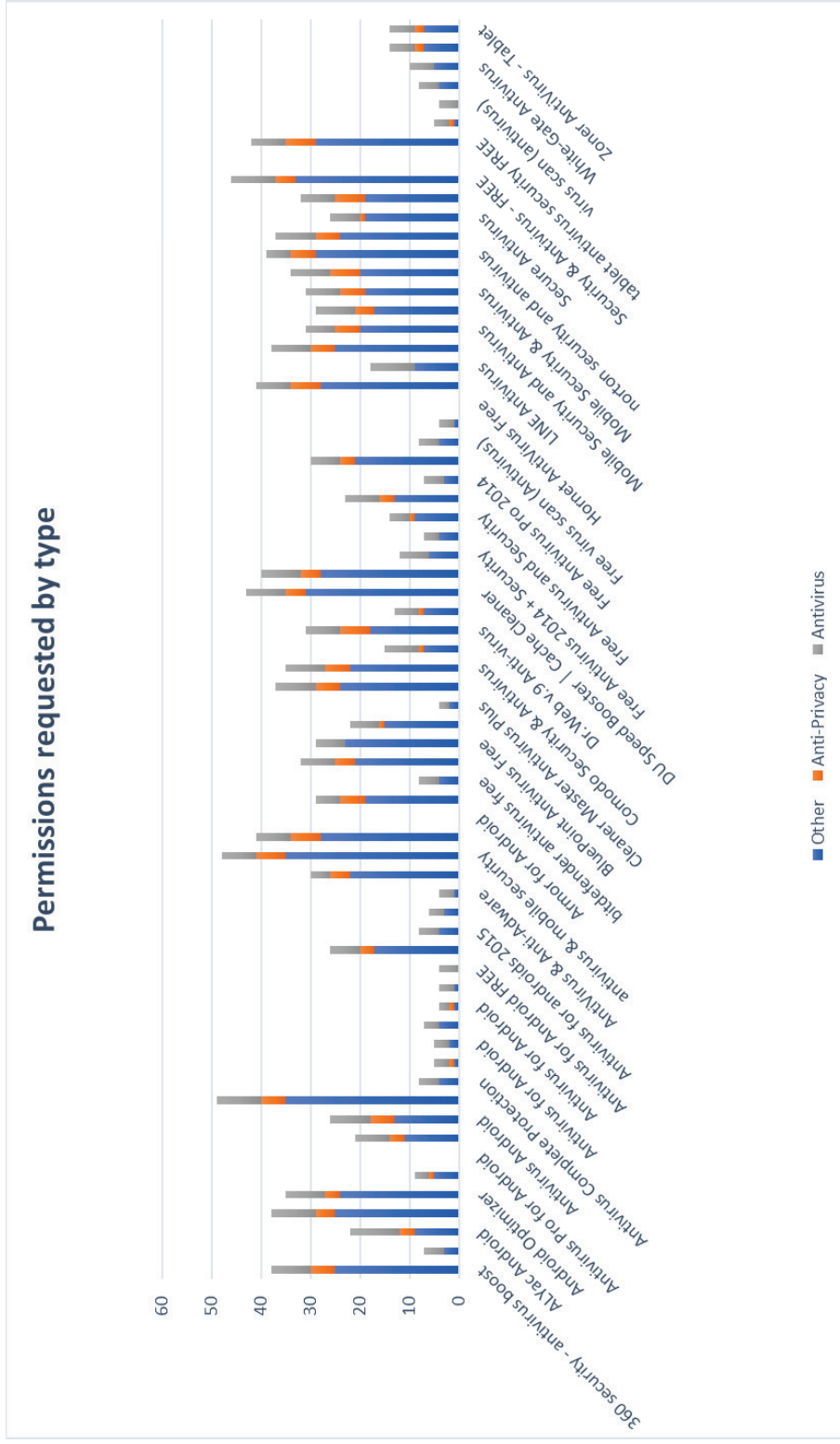


Figure 6-12 Permissions requested by Type (Antivirus, Anti Privacy or Neither)

Analysis of Antivirus Apps

None of the apps in the study requested the eleven permissions required to perform Antivirus functions. ALYac Android requested the most (ten) permissions.

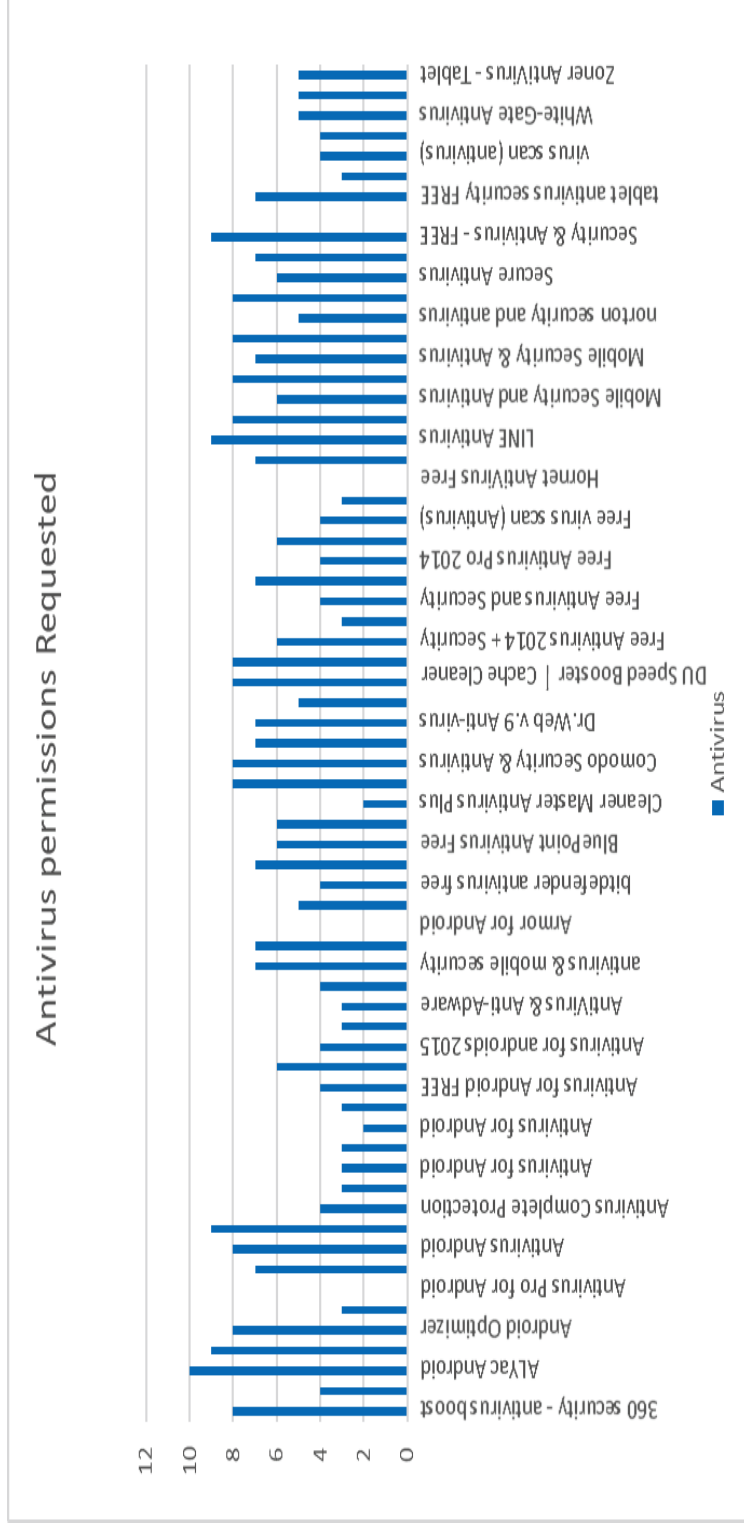


Figure 6-13 Antivirus permissions requested by apps

Requesting so few Antivirus permissions will affect the efficacy of the Antivirus function of the security app (Figure 6-13).

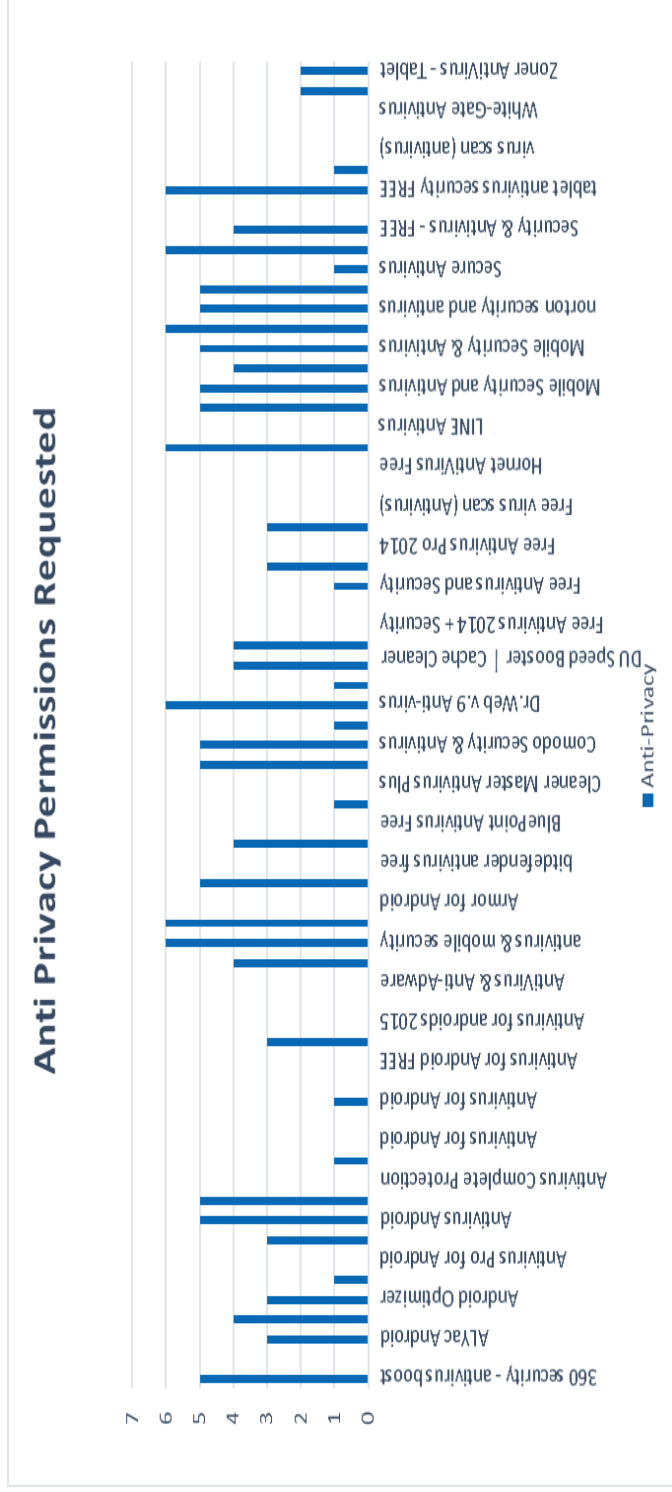


Figure 6-14 Anti Privacy permissions requested by apps

The apps were then analysed to compare Antivirus and Anti Privacy permission (Figure 6-15).

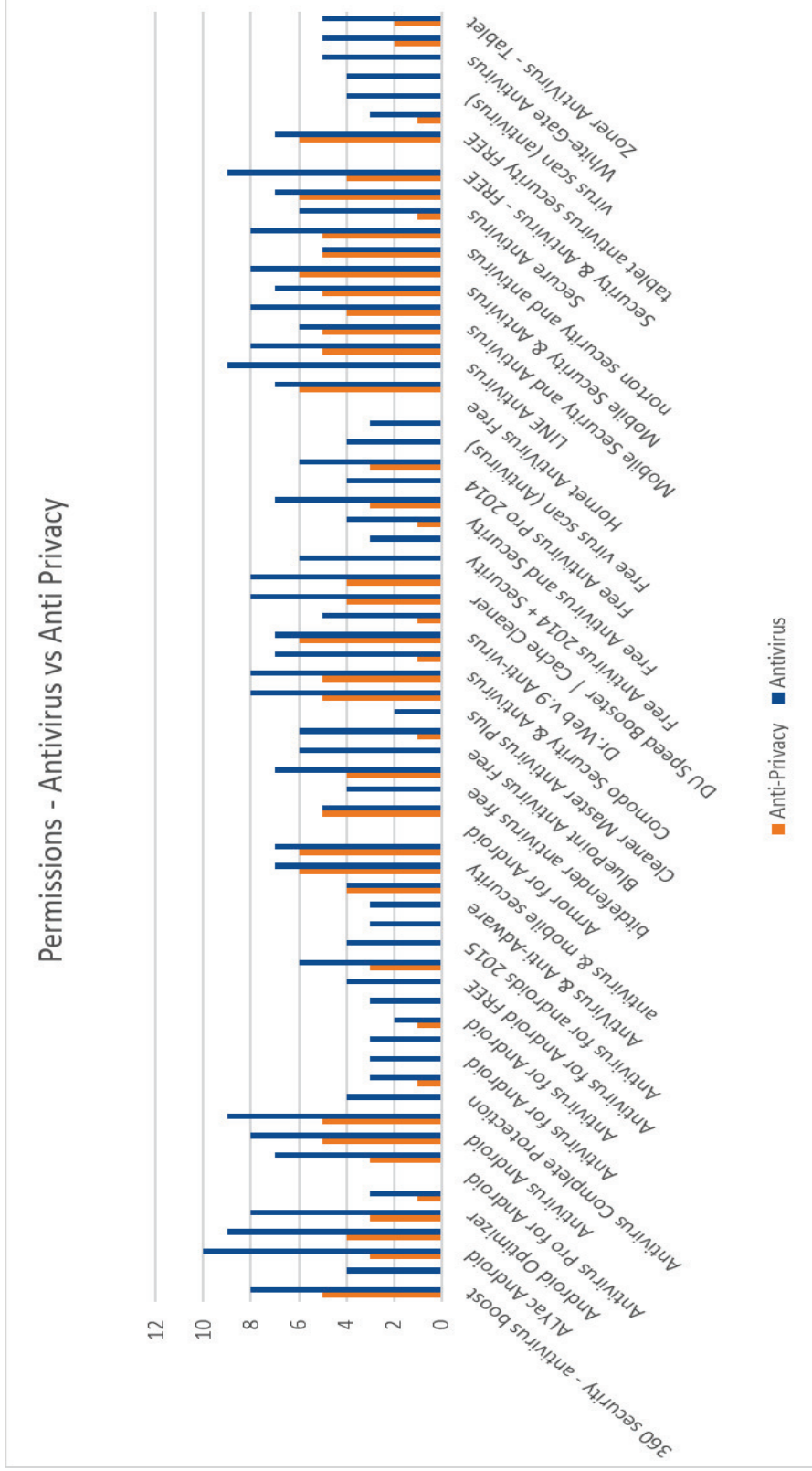


Figure 6-15 Antivirus and Anti Privacy permissions requested by the Anrivirus apps.

The apps were then analysed to determine if there was any correlation between the number of Antivirus and Anti Privacy permissions requested.

The apps that requested the most Antivirus permissions requested approximately half the Anti Privacy permissions. Except for “Line Antivirus” that requested 9 of the eleven Antivirus permissions and none of the Anti-Privacy permissions.

Table 6-9 Antivirus apps that requested the most Antivirus permissions

<i>app Name</i>	<i>Anti-Privacy</i>	<i>Antivirus</i>
ALYac Android	3	10
AMC Security - Clean & Booster	4	9
Antivirus Booster & Cleaner	5	9
LINE Antivirus	0	9
Security & Antivirus - FREE	4	9

Seven of the apps requested all six permissions that were designated as anti-privacy (Figure 6-14). These apps also requested a high number of Antivirus permissions.

Table 6-10 Antivirus apps that requested the most Anti Privacy permissions

<i>app Name</i>	<i>Anti-Privacy</i>	<i>Antivirus</i>
antivirus & mobile security	6	7
antivirus Security - FREE	6	7
Dr.Web v.9 Anti-virus	6	7
Kaspersky internet security	6	7
Mobile Security & Antivirus	6	8
Security - Free	6	7
tablet antivirus security FREE	6	7

The correlation between Antivirus and Anti-Privacy was tested. The result was 0.71 which indicates that there is a strong positive correlation between the requested number of Antivirus permissions and Anti-Privacy permissions.

Meaning that many of the permissions requested to perform Antivirus functions were contrary to a user's privacy. Therefore, Antivirus apps require more controls to protect the user's information from abuse.

6.4.3 Results

In 2011 there were 15 developers with 22 apps on the Google Store, of which 5 developers in their original state were still in existence in 2015. The 22 apps available in 2011 had reduced to 7 which had been updated during the 4 years to 2015.

In 2015 the number of developers had increased to 57 and the number of products available to 67.

The number of permissions had also changed but had not increased across the board as expected with the increase in permissions available. In 2011 the median number of permissions requested was 15, the maximum requested was 82 and the minimum requested was 3. In 2015 the median had increased to 21, but the maximum requested had dropped to 49. Three of the apps did not request any permissions at all, which does question the efficacy of the App. Removing these outliers showed the minimum that was requested was 4.

During the 4 years of the study the number of developers had increased four-fold, but the number of apps had only increased by a factor of three. This showed that the market was maturing, and developers were concentrating on a main app rather than providing multiple variations and names. The main commercial Antivirus providers were now providing Antivirus and security products to the mobile environment in addition to their PC portfolio.

6.5 Commercial Testers Results.

The main antivirus testing organisation (AV-test.org) had started testing and publishing these results in 2010. They tested four apps. The feature sets of the four apps were proscribed for the following OS; Windows Mobile, Symbian, Android, Android 7 and iPhone. All 4 apps were available on Windows Mobile and Symbian, 2 apps were available on Android and only 1 for the iPhone. In 2010 there were 35 security apps available on the Android platform, but these were not tested. The test results in the report showed that the apps were tested on the HTC Touch Pro 2, which is a Windows Mobile device. Details of the device type; Android, Symbian or iPhone used were not available. The Antivirus testing consisted of loading two viruses onto the phone and then testing the detection and quarantine functions of the apps. Browser detection and Firewall protection of the Security function of the app was also tested and their results published online ("Product Review: Mobile Security - August 2010," 2010).

Their subsequent testing occurred in 2011 and in this and future tests the company concentrated on the Android OS, with the first report available in August 2011 containing the results of the testing of six (6) security products on an LG P500 running Android 2.2. This testing was of the feature set of the products. Their first test of the products to the Android Permission set was performed in 2014 with the report published in September 2014.

6.6 Comparison of 2011 and 2015

Antivirus Apps

As the popularity of the Android OS grew many antivirus and security developers were bought by the mainstream Security software companies. In 2011 there were 22 apps with Antivirus components, this had grown to 63 apps in 2015.

Of the 22 security apps in the marketplace in 2011, 7 had been updated and were available in 2015. Five of the developers from 2011 were still active as developers in 2015, the rest had either gone out of business or had been subsumed by other companies. Table A-10 shows the apps available in 2011 to 2015, the developer name and the number of permissions requested in that year's variant. Some of the app's names changed between 2011 and 2015, but their package name (installable component) remained consistent with version variations.

The 2015 analysis consisted of comparing the differences between features and permissions of the 2011 apps that were still in existence in 2015. The extraction and comparison of the apps permissions and feature used the latest methodology as described in the PEMP chapter (Chapter 7).

The comparison of the permission changes during the 4 years of the antivirus apps are shown in Figure 6-16.

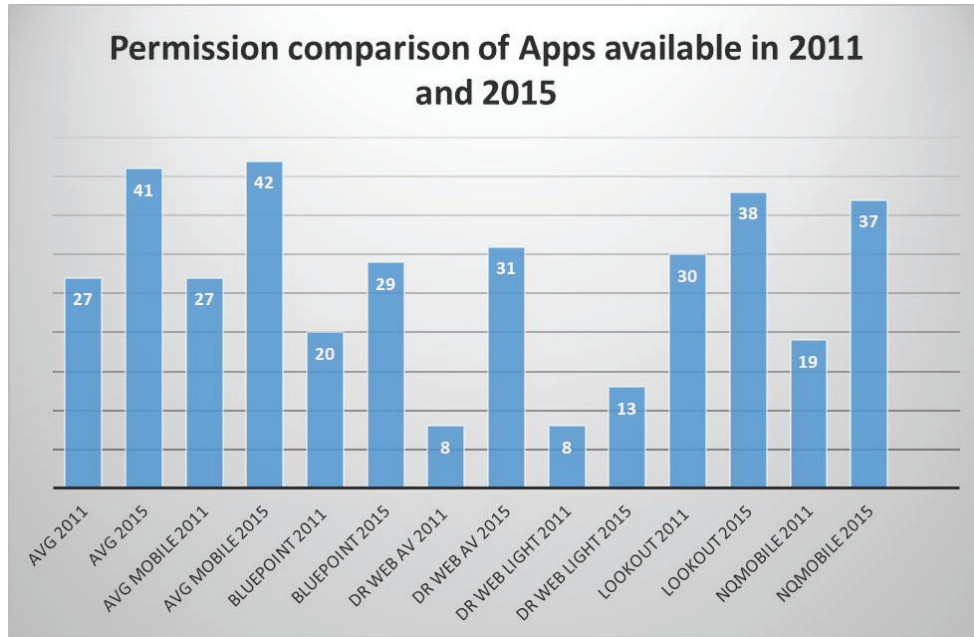


Figure 6-16 Comparison of permissions of 2011 apps still available in 2015

A comparison of the defined features was also made.

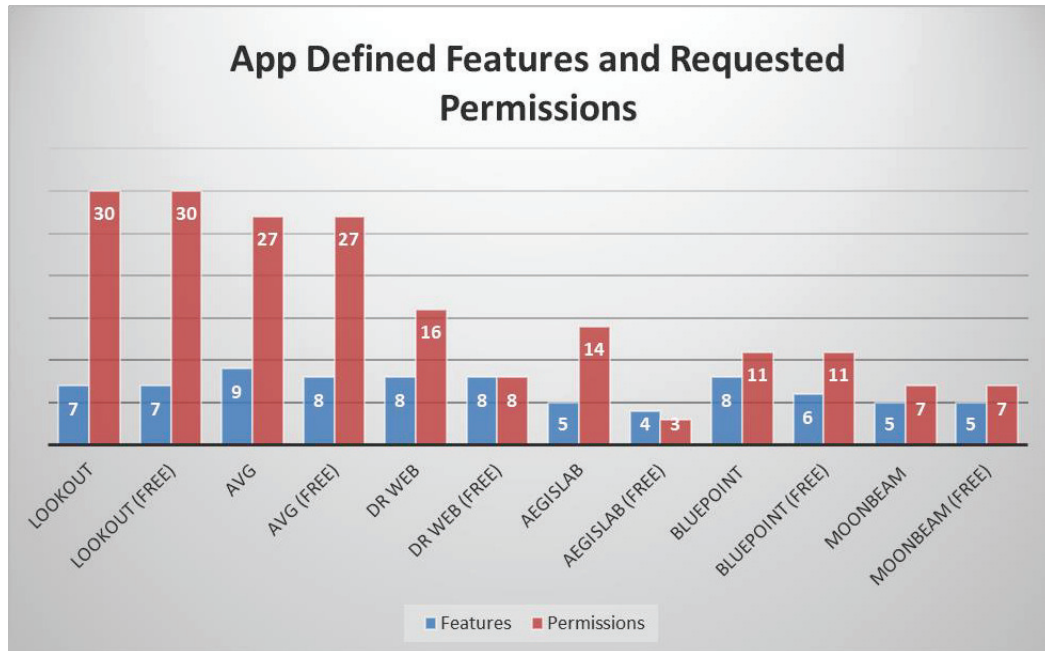


Figure 6-17 The features and requested permissions of Free and Commercial apps in 2011

6.7 Conclusion

Five Antivirus developers from 2011 were still in existence in 2015. The 22 apps available in 2011 had been reduced to 7 which had been updated during the intervening 4 years to 2015.

In 2015 the number of developers had increased to 57 with the number of products available to 67.

The number of permissions had also changed but had not increased across the board as expected with the increase in permissions available. In 2011 the median number of permissions requested was 15, the maximum requested was 82 and the minimum requested was 3. In 2015 the median had increased to 21, but the maximum requested had dropped to 49. This indicated that developers were either being more selective about the permissions to perform the function or were using the higher-level permission, which would cover multiple permissions, rather than select individual permissions (see the section in 9.1.3 which describes “Protection Normal”). Three of the apps did not request any permissions at all, which does question the efficacy of the app. Removing these outliers showed the minimum that was requested was 4.

Testing the correlation between Antivirus and Anti Privacy permissions showed that there was a strong positive correlation.

During the 4 years of the study the number of developers had increased four-fold, but the number of apps had only increased by a factor of three. This showed that the market was maturing, and developers were concentrating on a main app rather than providing multiple variations and names. To be able to provide the security for the user, the user’s privacy was severely impacted. This was not communicated to the user as many of the apps used the high-level

permissions provided by Google, which did not ask for user approval. The main commercial Antivirus providers were now providing antivirus and security products to the mobile environment in addition to their PC portfolio.

The next section improves on the testing process by introducing an automated method created to reduce the preparation of the app for analysis. The method is tested on various genres to ensure that it is repeatable and robust.

Chapter 7. Permission Extraction Method and Process (P.E.M.P.)

The previous chapter described the analysis of apps using a manual process. This was time consuming and an automated method was required to extract the app and perform some initial processing before the final analysis. This chapter describes the improved automated extraction process and its use.

Mobile app permissions are increasingly attracting interest from the mobile industry, researchers, standards bodies and protection agencies. Previous studies have concentrated on the technical aspect of the permissions and related API calls (Wain et al., 2012) introducing methods for the static (Bartel, Klein, Monperrus, & Le Traon, 2014) and dynamic (Barrera, Kayacik, van Oorschot, & Somayaji, 2010) analysis of the extracted permissions.

The extraction of the permissions is a laborious process and repeatable methods are needed to automate the extraction itself.

This chapter provides a repeatable and robust method, which is subsequently referred to as the Permission Extraction Method and Process (P.E.M.P). The method extracts the permissions from the app and provides the permissions in a suitable format for processing.

The development of the PEMP method described has been tested and refined over four years of research. The method has been used primarily to evaluate Android apps' permissions, although the method is easily adapted to other permission-based systems.

The initial method and use and the evolution to the current version is described. A discussion on the observations on the success of the method and additional functionality which could be incorporated to fully automate the process are explored.

The initial method that is described first was used to extract antivirus and security apps in the Google play store⁵. The purpose of the extraction was to compare the coded permissions and features with those described on the Play Store. Previous research had reviewed the efficacy of free antivirus scanners but had not analysed the permissions requested by the scanner apps (Ramachandran, Oh, Stackpole, & Smartphone, 2012). Before the app could be processed it had to be downloaded to a device capable of running the app and then transferred to a PC for the evaluation.

However, the initial method was very labour intensive, initially taking 1 hour to extract and process each app, but with repetition the author managed to reduce it to 30 minutes per app. The thirty minutes processing for each app consisted of; the download took 5 minutes and to transfer, decrypt and extract the permissions took an additional twenty-five minutes. Therefore, preparing the 20 apps for comparison analysis took 10 hours. The final product contains

⁵ Google play store is also known as the Play store and Google Market place (<https://play.google.com/store>).

automation and has reduced the time taken to prepare the app for processing to slightly more than the download time of approximately 3 minutes. The disassembling, decryption and preparation of the app for further processing is then done in bulk and takes less than 5 minutes for 20 apps. The updating of the permission database is still manual, but the format of the extraction output has reduced the time taken to populate the database. This is an area for future automation (Chapter 10).

The chapter provides guidance on PEMP for Android, the initial overall model and extraction, code segments and guidance on selection of apps for a robust and repeatable evaluation. And provides insights into how app permissions have changed over the last four years.

The chapter concludes with a discussion on observations on the success of the method and the additional functionality which is required to fully automate the process, and then indicates additional areas for further research. The research will concentrate on analysis rather than the extraction and data collection as in this research.

Previous researchers concentrated on the permissions specified API calls (Wain et al., 2012) and permission mapping analysis (Bartel et al., 2014). In these cases, research concentrated on the permission framework and analysis of the Android framework and those permissions requested by categories of apps rather than the permissions of individual apps in one category.

A survey by (Mylonas, Kastania, & Gritzalis, 2013) has found that there is a suggestion of complacency by users to security on personal devices, initial research investigated the efficacy of security products, especially Antivirus, available in the market place in 2010 (Pilz S, 2012). At that time, the Android

operating system was selected due to its rapid growth on devices and the abundance of free apps (Enck et al., 2010). Whilst investigating the selection of the apps, the analysis detected a distinct variation between the numbers of permissions requested, the lowest being 3 and the highest was 31. Using security knowledge and experience the permissions which would be required by an app to be able to perform basic antivirus functions was defined. The apps were then tested to evaluate their ability to perform the antivirus functions⁶ and then the results compared across the apps. One section of the research reviewed the permissions of the apps to determine if the previously defined required minimum had been requested and if the permissions requested had any impact on the efficacy of the app's functioning as an Antivirus app. Further comparisons were made between the free app and its commercial version (if available) including the source code.

It was during the extraction of the source code in 2011, that a generic process was required to enable mass extraction of the apps, irrespective of the genre or category of the app, or the OS version that it was written and compiled for. This would enable the research to be concentrated on the analysis rather than the extraction tasks. The generic Permissions, Extraction Method and Process (PEMP) evolved from this need and verified using the earlier tested process models. The method has also been tested by another researcher to extract and analyse First Person Shooter (FPS) games.

⁶ Full research results are available on request.

7.1 Generic PEMP

This section describes the generic permission extraction process for apps. It also captures a discussion of a fuller extraction process, PEMP, towards a repeatable and mass extraction capability. It also provides guidance to implement PEMPs through the extraction process of Android apps.

The overall research method was first developed in 2011 to download Antivirus apps from the Google Store (now called the Play Store), verify that the app performed Antivirus functions; scanning, detection and removal of malware from the device. The app was then transferred and decompiled into readable format to check the permissions coded into the app were as described on the Play Store.

The Google Play Store in 2011 (<https://market.android.com/>) contained 37 security products and these constituted the base for the investigation. The permissions requested by these apps were recorded and reviewed against the API list to determine the access to system resources.

The permissions that were determined to provide the Antivirus functions and those which were detrimental to the user's privacy were described and documented for each security product.

Other methods have been described to perform analysis of the apps, API calls, Permissions etc. Two examples are static analysis to map API calls and Permissions (Bartel et al., 2014) and empirical analysis (Barrera et al., 2010) using Self-Organising Maps. The methods describe the analysis of the permissions but have assumed that the author already uses an undisclosed method for the extraction of the app, it's source and database repository. The method described in this paper is one example to fill this gap and provides an extraction method to enable researchers with little or no Android development skills, Java or Eclipse knowledge to prepare the Android framework and code for analysis.

7.2 Generic PEMP Process and Guidance

A generic model of the process is shown (Figure 7-1). The model has been extended with further contribution to existing models. A detailed description of the phases, with examples of the implementation of the phase and the results are discussed in the Initial 2011 Method section (Section 7.3) and the subsequent evolved method in the 2015 Method chapter (7.4).

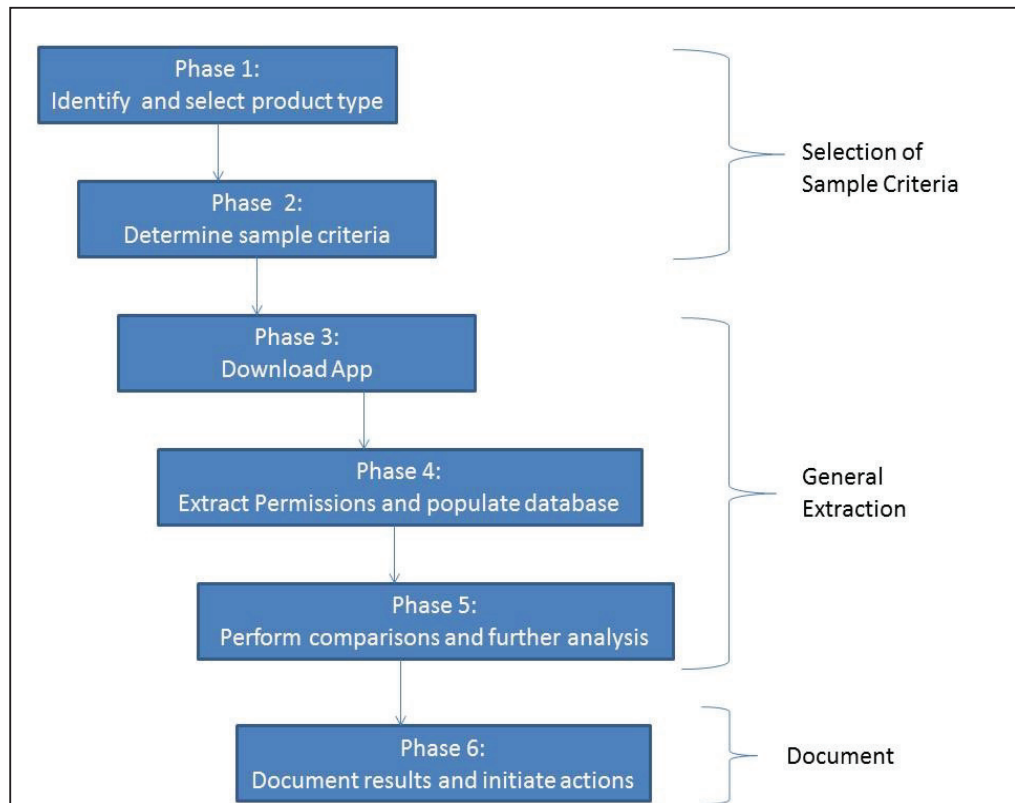


Figure 7-1: Flowchart illustrating the overall method for extraction of permissions

The generic PEMP process model consists of 6 phases. A summary of each phase is described and more detailed description of the phase in action is in

Phase 1:

The identification of the product type is important as it will indicate which types of permissions will be in the selection. Antivirus apps will be concentrating on permissions related to scanning either the device itself or during downloads, so the ability to read and write to storage is necessary. If an app is performing photographic tweaks, then access to the camera and the photo album is required.

P.E.M.P.

Phase 2:

Sample criteria needs to be robust and repeatable. One option is to select apps that have a minimum number of downloads or a certain rating. Google also displays apps from a search by popularity, but this is hard to repeat by researchers performing subsequent searches.

One method that the author used was to create a co-efficient related to the rating and number of downloads. apps that had too few downloads were discarded to prevent the data being skewed. Subsequent selection was to create a co-efficient on the rating and number of people rating the app.

Phase 3:

Downloading the app can be performed in a variety of ways. The main options are to download to a valid device and transfer to the processing PC or to download either directly to the PC or via a download server.

Phase 4:

Extraction of permissions can be performed manually for each app, or automation can be used to simulate the manual extraction. The output from the extraction entered into a database to facilitate later analysis.

Phase5:

Comparison of the permissions requested will depend on the product selection and the researcher's area of interest.

Phase 6:

Document the results. As the selection criteria is repeatable, evolution of the app permissions can be compared over time, as well as changes to the rating

co-efficient. As bulk extraction is relatively simple, other analysis can be initiated by the category of app extracted.

7.3 Initial 2011 Method

In 2010, mobile phones were growing at an incredible rate, overall the smartphone sector grew by 64% in the year 2Q2009 to 2Q2010 (“Google Android phone shipments increase by 886%,” 2010). The author’s research concentrated on the analysis of security and privacy of Antivirus apps.

The question that the research intended to answer was, “Is there a correlation between the permissions requested and the features specified, and do they effect the efficacy of the Antivirus function?”.

7.3.1 Tools

Prior to 2012, the download of the app from the Google Play Store (<https://market.android.com/>) was performed using the Google Installer and installed directly onto the device that it would be run on. See Chapter 4

Additional tools are then required to transfer the app and its code to the processing device (a laptop or PC). Tools were also needed to de-encrypt and disassemble the compiled code into a readable format, so that the permission file could be accessed.

7.3.1.1 Tool Installation.

The base tools had to be installed prior to the testing.

7.3.2 The Phases of PEMP

The description of each phase of the initial method and the results follow.

7.3.2.1 Phase 1: Identify and select product type

At the time of selection in 2010 and 2011, there were a variety of documents and advice, in the form of blogs and white papers, and company promotional material available to aid consumers and enterprises in securing standard computing equipment; laptops, netbooks, desktops, etc. There was also a variety of free/shareware tools available to perform vulnerability assessments of these devices and the networks that they use for connectivity, e.g. Nessus (<http://www.tenable.com/products/nessus/nessus-product-overview>), Nmap (<http://www.nmap.org/>) and Wireshark (<http://www.wireshark.org>). However, this availability of tools and knowledge had not been transferred into the mobile sector (Smartphones, e-readers, tablets etc). In this sector the increase in acquisition of these device types exceeded the growth of legacy platforms (laptops, netbooks), PC shipments increased to 92.1 Million in the last quarter of 2010 ("Tablet Computers Hold Back PC Sales Growth," 2011) whilst Smartphones grew by over 100 Million in the same period (Canalys, 2011).

A study by Nielsen shows that the choice of Smartphone software is also age related with Android being the main choice in the 18-34 age group (Study: Ages of social network users., 2010). Therefore, there was an increase in criminal activity in proportion to the growth of the Android operating system market share. Android phones growing by 886% between Q2 2009 and Q2 2010 whilst Apple's Smartphone growth was around 61% during the same period (Mobile Snapshot: Smartphones Now 28% of U.S. Cellphone Market. , 2010). Although the Android growth slowed to 148.1% between 4Q 2010 and 4Q2011

the market share grew to over 51%, thus becoming the most popular mobile operating system as per research by Canalsys (“Smart phones overtake client PCs in 2011.” 2012). The mobile operating systems and their market share from Q4 2010 to 4Q 2011 are in Table 7-1.

OS	Q4 2011 Shipments (millions)	% share	% Growth Q4'11-Q4'10
Symbian	18.3	11.6	-40.9
RIM	13.2	8.3	-9.7
Android	81.9	51.6	148.7
Apple	37	23.4	128.1
Windows	2.5	1.6	-14.0
bada	3.8	2.4	39.1
Others	1.8	1.1	117.91
Total	158.5	100	

Table 7-1: Worldwide Smartphone market

As occurred on the Windows OS for PCs as an operating system becomes more prominent, actors are adapting existing malware, PC viruses and Trojans, to target it.

Therefore, this research concentrated on the most popular mobile OS, which was Android smartphones and how they were being protected from not only malware but the security products themselves.

7.3.2.2 Phase2: Determine sample criteria

A search of the google store produced a result of 37 apps that had keywords or tags of security or antivirus, 23 contained tags for Antivirus, and were selected for the research. The selection method was to sort the free apps by number of downloads and select the top 10. These apps were sorted by user satisfaction co-efficient. This was calculated using the user rating and number of downloads.

If a company provided a commercial version of the app, this was also selected for download so that comparisons could be performed.

7.3.2.3 Phase 3: Download

At the time of the testing the apps had to be downloaded to an Android smartphone and then transferred to a PC to perform any extraction of the source code.

The app downloads were performed on a T-Mobile G1 smartphone. The device was running the original installed Cupcake version of Android (V1.6). This proved to be inadequate to run the Antivirus apps and was not supported by some of them. Therefore, the decision was made to update the software to the latest operating system, which at the time was Froyo (V2.2). Once the device had been rooted and updated to the latest version of the OS, the apps installed with no problems and each app was tested to determine that it performed the basic Antivirus functions; that is detecting and removing malware. Once the app passed the verification checks it was a suitable candidate for further processing.

The author did not test against all the malware available but used a small subset of test malware to verify functionality. Antivirus test companies, AV-Test.Org had started testing mobile security apps and had large databases of malware to use as part of their test process (Pilz S, 2012).

7.3.2.4 Phase 4: Extract permissions and populate database

The Android operating system (OS) is a privilege-separated OS and by default applications (apps) or packages are not permitted to perform any operation that would impact another app, the operating system or the user, this is known as Sandboxing. The sandbox creates an area for applications to run in and the access that the installed app must a system resource is controlled. Android uses a system of permissions. These permissions form part of the application sandbox and provide a modicum of basic security to the operating system. These permissions are declared in an application's manifest file.

By default, an application does not have any associated permissions and must declare in the manifest file which permissions it needs. At installation time the user is notified by the installer the permissions that the app is requesting, and the user then has the option to deny (don't install) or accept (continue install) the request.

The user is not able to select which permissions the app can receive during the installation process.

To extract the permissions, the app code had to be transferred to the PC. To do this several software tools was required. The software required was; Android Development Kit (ADT), Java Development kit (JDK), Android Virtual Devices

(AVB), Android Debug (ADB), a Java graphical interface (e.g. JDGUI), Eclipse, Android Software Development Kit (SDK), a .dex decompiler (dex2jar), and an app package extraction tool.

The smartphone was connected using a USB cable. The app was located on the phone main storage and had a .dex suffix. This suffix describes the package as a Davlik EXecutable. Only apps in this compiled format can run in the Android operating system environment.

The transfer and conversion commands are run using the command line on the PC. To transfer the executable to the PC the ADB Pull command was used.

To be able to read the Android manifest file (Manifest.xml) the transferred executable must be converted from dex to a readable format. This was performed in two steps; first de-compiling from dex to a compiled Java code (jar) using the dex2jar tool and then from the compiled Java (jar) to Java source code. This was done via the JDGUI interface which displays the Java classes of the app in a GUI format. The Manifest file was selected, and the permissions were manually extracted from the source code and saved in a database for later analysis.

7.3.2.5 Phase 5: Perform comparisons and further analysis

Excel was used as the database platform, due to its ease of use, inbuilt programmability and the various file formats that the data can be converted to and saved. A spreadsheet was created which contained the following fields; app name, developer name, package name, rating, number of downloads, coefficient, package size, and the permissions selected.

7.3.2.6 Phase 6: Documented results

Comparisons between the Free and Commercial (paid for) Antivirus apps were documented. The results⁷ of the comparison of product features, permissions and user ratings are in Figure 7-2 and a cluster analysis of features and permissions illustrated that there was no relationship between the number of features of the app and the permissions (Figure 7-3).

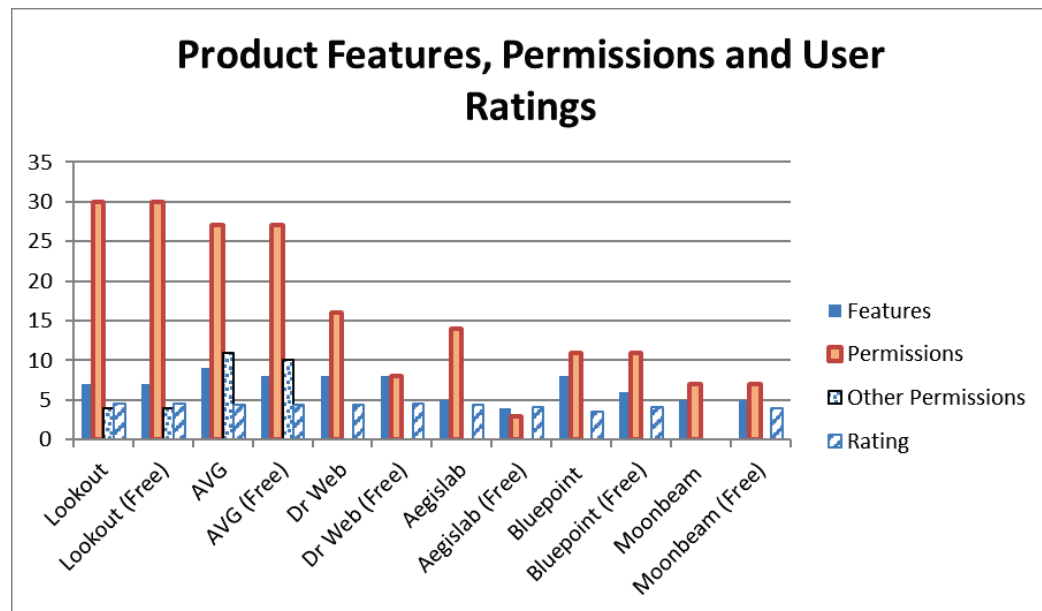


Figure 7-2: Permissions requested by Free and Commercial Antivirus apps in 2011

⁷ These results are available on request.



Figure 7-3: Cluster analysis of the relationship between features and total permissions requested

The analysis demonstrated that there was no correlation between features and permissions. Therefore, any additional features in the commercial versions, either did not require any additional permissions or were external to app. External features consisted of remote lock/wipe, find my phone, and other online/cloud-based functions.

7.4 Generic Method (2015)

7.5 Tools

One of the objectives in the evolution of the method was to automate the various functions of the method. There are tools available that perform many of the previous manual steps and these are incorporated into the generic method. The download of the app was performed by using a PC browser tool, APK Downloader (“APK Downloader V2,” 2014). The extraction was performed with APLtool (“APKtool,” 2015) and the processing was performed by running a Python script.

7.6 APK Downloader

Google Chrome (“Chrome Browser Download,” 2012) was selected as the browser for the Play Store access and for downloading the apps, the APK Downloader was used. APK Downloader, which became available in 2012 and is obtained directly from the developer (“APK Downloader V2,” 2014).

Apk downloader is a browser extension which downloads the app directly to your PC. The extension version used in this method was version V2 and was available for both Chrome and Firefox.

Other, non-official versions of this Chrome extension are available (“APK Downloader,” 2014). In that instance of the extension, the app is downloaded to a server and a link is made available for the user to then download to the

PC. Commonly accessed apps are stored on this server to reduce download times. This method provides the ability to bypass the normal device constraints. (The browser plugin requires that you define the device that the downloader is emulating. The downloader then downloads the related code for that device. This way you can download apps for tablets or mobiles.) Users are also able to download apps from a variety of app stores and transfer them to their smart device. When using this extension, the user needs to be aware that there is an extra step in the download process where malware could be introduced to the app. It is also possible to obtain apps from non-legitimate sites, which could also contain malware. Using this option means that any app can be downloaded, irrespective of the device requirements.

APK Downloader is also available as a Windows executable. This program downloads the app but requires the package name or full URL as input to the program to download the app, whereas the browser extension can download the app directly from the Play Store.

The method used in this stage of the study was to use the Chrome extension to download the app directly to the PC. This method requires that the Chrome extension has access to the user's email and password, (as this access is a security risk, I recommend creating a dummy Userid to perform the downloading) and the Device Id that the app needs to work on as described in the installation notes (Figure 7-4).

To be able to get the Android Market cookie, it needs a valid email and password to login. Once the initial login has occurred, download(s) can commence. The password is not stored after this initial login, the email, Device Id and Cookie are stored for later requests.

1. **Enter email and device ID on Options page.** There are two ways to get Email and Device ID
 - a. **With the Device ID app** which is obtained from the [<https://market.android.com/details?id=com.redphx.deviceid>] , it will show you your emails and Device ID
 - b. **On the smart device:** Open dial pad, call *##*8255##* (8255 = TALK). If it opens "GTalk Service Monitor", find lines that begin with **JID** and **Device ID**. Your email is **JID**, and your device id is a string that after **android-** prefix

Figure 7-4: Chrome APKdownloader plugin installation notes

7.7 Extraction and Processing Tools

To extract the manifest file from the packages the Android reverse engineering tool APLtool ("APKtool," 2015) is used. This tool is a small debugger and can decode Android apps from binary to their nearly original form. This is required to extract the Manifest.xml file which contains the app permissions. The Java v1.7 or higher development kit ("Java Download," 2015) is needed to run the tool. A knowledge of the Android SDK is useful but not essential.

To perform the processing, automation code was written in Python and needs Python version 2.7.9 or higher ("Python Downloads," 2015).

7.7.1 Process

In 2015 the method was simplified (Figure 7-5) to incorporate the APK-Downloader plugin on Chrome. Once installed and enabled it permits the user to download the app package directly to the PC. This removes the laborious steps of downloading the app to a smartphone, transferring the executable to the PC and de-compiling to a readable format. The file is saved with a suffix of .apk.

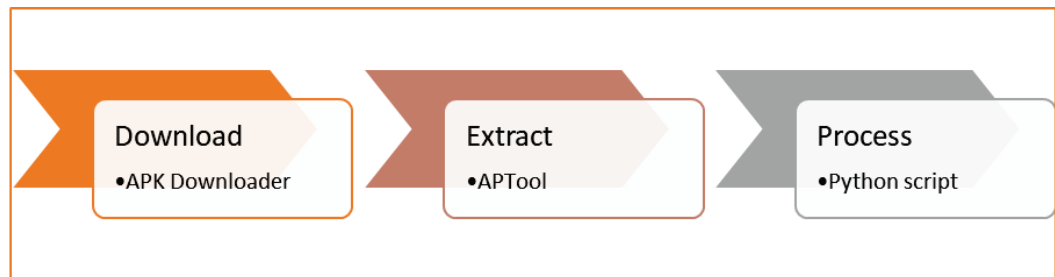


Figure 7-5 Analysis process flow - simplified.

The Extract code (Figure 7-6) provided, requires that the app descriptions are inputted into to a flat file for pre-processing. The pre-processing prepares the app's information for the decode. The decode is performed in a batch file (cmdlist2.bat). This extracts the app's code and decrypts and dis-assembles it into a source readable file so that the AndroidManifes.xml file, containing the permissions can be accessed.

This automation requires that basic details are in an input file, which is used to create the output folders for each app. A comma separated values file that contains the app name, package name and company has been used in this case. These values are also used to create the pre-processing list for the tools.

7.7.1.1 Phase 1: Identify and select product type

In 2015 the selection criteria from 2011 was replicated; Android apps on the Play store with keyword tags of Security and/or Antivirus. Primarily of interest were apps that had been in existence in 2011 and had evolved with Android and smartphones.

7.7.1.2 Phase 2: Determine sample criteria

The search of the Play store produced a result of 65 apps that had keywords or tags of security or antivirus. Seventeen of the apps were updated versions of 2011 apps. These apps, in both versions were used to test the validity of the process.

7.7.1.3 Phase 3: Download

Two additional options are available to download apps. Both provide the functionality to download directly to a PC thereby by-passing the requirement to download and install the app to a suitable device and then transfer the executable to the PC for decompiling.

This reduces the time to prepare the app for extraction. Although the download time, from Play Store to device is constant, irrespective of the device, Smartphone or PC, the transfer process has been eliminated.

The APK Downloader that downloads directly to a PC was used. This requires that the device type and its operating System is configured in the tool at first use.

During the download process the tool verifies that the app is suitable for the device type and OS configured prior to the download. The tool downloads both the .apk file and the .obb file, which contains additional data that is used when the app runs.

To verify that the tool was not injecting any additional code into the app, a sample app was downloaded using the traditional method to a device and the executable transferred to the PC, where the extraction was performed to obtain the package and the two MD5 hashes compared. Subsequent downloads were performed only using the APK-downloader.

The Play store was accessed from the PC using Chrome with the APK-Downloader plugin and the search keywords used were security and/or antivirus. Snapshots of the app pages were taken so that the app presentation order was recorded.

Each app was downloaded, and the package stored in an input folder (apkin). The name of the developer, app, package name, rating and number of downloads and package size was recorded in an excel spreadsheet. Once the spreadsheet has been populated with the selected apps it is saved in .csv format so that it will be readable by the batch extraction process.

7.7.1.4 Phase 4: Extract permissions and populate database

The format of the input file is app names, the company/developer name, the package name, rating, number of downloads and package size. To differentiate between duplicate app names, the name is updated to include the company name in parenthesis and any names that contain an and symbol (&) will have it replaced with a 'n'. All spaces in the app name are replaced with an underscore '_' which prevents processing errors.

Prior to executing the script, the following tasks are required. Create an output folder (apkout)

- Create a folder to contain the python output and the call command batch file.
- Create a batch file to contain the APKTool commands to decode the pkg and output the code and androidmanifest.xml file
- Create an output folder, apkout

The sample python script (Figure 7-6) reads the .csv input file and for each entry creates a corresponding entry in an output file, containing the APKTool command. The code is provided as guidance, as complementary code to help guide other researchers to provide a robust and repeatable extraction.

The format of the command is

```
"call apktool d -f -s /apin/package_name.apk -o /apkout/app_name/ \n"
```

This creates a folder in apkout for each app_name. Each of these folders contain the decrypted and dis-assembled package including the AndroidManifest.xml file in readable format.

The script is run in a python shell (from the IDLE editor GUI).

```
''' APKextract
This code reads the downloaded apk file.
For each line open the package and run APKtool to decode the xml file
and output to the app name folder.

list of packages is in c:/apkin/apklist/

cmd string is c:apktool d -f -s /indir/app_name -o /outdir/app_name

'''
import csv

fi = file('/apkin/apklistn.csv','r')
#fo = file('/apkout/appout.csv', 'wb')
fo = file('/apkout/cmdlist2.csv', 'wb')

ci = csv.reader(fi)
co = csv.writer(fo)

#for each input row

for master_row in ci:
    app_name = master_row[0]
    pkg_name = master_row[1]
    company = master_row[2]
    print pkg_name
    cmd1 = "call apktool d -f -s /apkin/"
    cmd2 = "-o /apkout/"
    cmd3 = "/ \n"
    cmd = cmd1 + pkg_name + cmd2 + app_name + cmd3
    fo.write(cmd)
    #fo.write('\n')

fi.close()
fo.close()
```

Figure 7-6: Example python script to create the batch file entries

P.E.M.P.

The output file cmdlist2.csv is renamed to a batch file (cmdlist2.bat)

A cmd GUI is opened and the batch file is run.

Example log of one batch call which corresponds to the APKTOOL cmd;

```
call apktool d -f -s /apkin/org.sample.av-53.apk -o /apkout/Sample_Antivirus/
```

and the output log is:

```
I: Using Apktool 2.0.0-RC3 on org.samplem.av-53.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: C:\Users\KC\apktool\framework\1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Copying raw classes.dex file...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Figure 7-7 Sample Log from the APKTool call

This creates a folder in APKout called Sample_Antivirus

The folder contains the AndroidManifest.xml which has been decoded so that it can be viewed by any text editor (e.g., Notepad or Wordpad).

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:installLocation="internalOnly" package="org.whitegate.av">
    <uses-feature android:name="android.hardware.telephony" android:required="false"/>
    <uses-permission android:name="GET_TASKS"/>
    <uses-permission android:name="RESTART_PACKAGES"/>
    <uses-permission android:name="INTERNET"/>
    <uses-permission .....
    .....
</application>
</manifest>
```

Figure 7-8: Sample Manifest file extract

7.7.1.5 Phase 5: Perform comparisons and further analysis.

This is still a manual process and will be automated in future work.

Excel is used to open the AndroidManifest.xml. When Excel opens the file, it asks for the format. Select the XML Table format. Once open, search for (use the find all option) the permissions will all be in one group. Permissions that have been created by the developer may show up but are not part of this comparison of Save the permissions in a file called manifest.csv (Figure 7-9) in the packages folder. This will be used as input to the permission checker (Figure 7-10).

	A	B	C	D	E	F	G	H
1	android.permission.CAMERA							
2	android.permission.FLASHLIGHT							
3	android.permission.INTERNET							
4	android.permission.READ_CONTACTS							
5	android.permission.WRITE_EXTERNAL_STORAGE							
6								
7								
8								
9								
10								
11								
12								

Figure 7-9 Sample contents from a manifest.csv file

A file containing all the Android permissions for the OS version is used as the master permissions file (f1) to compare the app permissions with. The script has 2 input files, the master permission list and the app's package list. The output file from apklist contains a list of all the app names and this used by the permission checker to open each app folder and open the manifest.csv file (this contains the android.permissions specified in AndroidManifest.xml). The app's permissions are compared to the master permission list and the result is the full list of permissions with the requested permissions marked with a 'y', which is stored in pkg_perm in the apps folder.


```
#!/c:\python27\scripts
# f1 is the file containing the full list of permissions for Android Lollipop
# f2 is the manifest file of the package
# f3 is the resultant file with the permissions confirmed (y) or not (n)
import csv
fp = file('/apkout/uni_pkg.csv', 'r')
cp = csv.reader(fp)
pkglist = [row for row in cp]
for unip in pkglist:
    uname = unip[0]
    print 'unique name is ', uname
    #uname = 'testfold'
    f1 = file('/apkout/masterperm.csv', 'r')
    f2 = file('/apkout/' + uname + '/manifest.csv', 'r')
    f3 = file('/apkout/' + uname + '/pkgperm.csv', 'wb')

    c1 = csv.reader(f1)
    c2 = csv.reader(f2)
    c3 = csv.writer(f3)
    permlist = [row for row in c1]
    manflist = [row for row in c2]
    row = 0
    man = 0
    for perm in permlist:
        result = perm
        for manf in manflist:
            if manf[0] == perm[0]:
                result.append('y')
                break
        c3.writerow(result)
    f2.close()
    f3.close()
    f1.close()
fp.close()
```

Figure 7-10: Python code to compare app permissions to a master permission file

7.7.1.6 Phase 6: Document results and initiate actions

The pkg_perm file for each app is stored in the excel database. A snapshot of the database (Figure 7-11) is shown.

	app name	Aegis AppsScan Beta	Aegislab Antivirus free	Aegislab mobile	Android defender virus p
	Unique_app	Aegis AppsScan Beta	Aegislab Antivirus free	Aegislab mobile	Android defender virus p
	pkg_name	com.aegislab.atrprj.appsca	com.aegislab.sd3prj.antiv	com.aegislab.sd3prj.egisr	com.moonbeamdeveloppr
	company	Aegislab	Aegislab	Aegislab	MoonBeam Development
	rating				
	downloads				
	Size (KB)				
Permission	Status	2011	2011	2011	2011
android.permission.ACCESS_CELL_ID	old				
android.permission.ACCESS_CHECKIN_PROPERTIES	new				
android.permission.ACCESS_CHECKING_PROPERTIES	old				
android.permission.ACCESS_COARSE_LOCATION	both				y
android.permission.ACCESS_COARSE_UPDATES	old				
android.permission.ACCESS_FINE_LOCATION	both				y
android.permission.ACCESS_GPS	old				
android.permission.ACCESS_LOCATION	old				
android.permission.ACCESS_LOCATION_EXTRA_COMMANDS	both				
android.permission.ACCESS_MOCK_LOCATION	both				
android.permission.ACCESS_NETWORK_STATE	both	y	y	y	y
android.permission.ACCESS_SURFACE_FLINGER	new				
android.permission.ACCESS_WIFI_STATE	both			y	
android.permission.ACCOUNT_MANAGER	new				
android.permission.ADD_SYSTEM_SERVICE	old				
android.permission.ADD_VOICEMAIL	new				
android.permission.AUTHENTICATE_ACCOUNTS	both				
android.permission.BATTERY_STATS	both				
android.permission.BIND_ACCESSIBILITY_SERVICE	new				
android.permission.BIND_APPWIDGET	new				

Figure 7-11: Snapshot of Permission Database entries

7.7.2 Additional Actions

Permissions can be checked for any version of the OS and even across the versions using a simple script. A sample script (permver_chk.py) to compare permissions for different Android versions is provided (Figure 7-12).

In this case each version has a master file created and the sample code will compare each file and output a file containing all the permissions in both input

files marked with either 'new' (only in this release), 'both' (present in both versions) and 'old' (only in older version and now discontinued).

```
#! c:\python27\scripts
'''
Permission Version Checker
This script compares two Android versions permissions file and creates an output file with the permissions marked
as valid in both versions, only in previous version or new for this version.
'''
# f1 is the file containing the full list of permissions for Android Lollipop
# f2 is the manifest file of the package
# f3 is the resultant file with the permissions confirmed (y) or not (n)
import csv

f1 = file('/apkout2011/masterperm.csv', 'r')
f2 = file('/apkout2011/permv2.2.csv', 'r')
f3 = file('/apkout2011/permdiff.csv', 'wb')
c1 = csv.reader(f1)
c2 = csv.reader(f2)
c3 = csv.writer(f3)
permlist = [row for row in c1]
manflist = [row for row in c2]
row = 0
man = 0
for perm in permlist:
    result = perm
    for manf in manflist:
        if manf[0] == perm[0]:
            result.append('y')
            break
    c3.writerow(result)

f2.close()
f3.close()
f1.close()
```

Figure 7-12: Python code to compare permissions of different versions of Android OS and mark the origin.

The final output is an Excel Spreadsheet with the permissions selected by each app. This data can then be used as input for analysis of trends, most commonly selected permissions by genre/category etc. When used as the base for checking apps requested permissions, it is useful to review if discontinued permissions are still being requested.

7.8 Conclusion

The method arose to fill a need to extract and process Android apps to perform permission analysis in another stage of the research. The time taken to extract and prepare the permission list for analysis was too time consuming. It meant that the research was concentrated on the extraction and decoding of the manifest file instead of the analysis of the file between different categories of apps and within the categories. Removal of the manual intervention at each stage of the extraction and decoding through the automation of the basic tasks has enabled the research to process 20 apps within a few hours rather days.

The method has been tested against different categories of apps and in each case the mass extraction of a minimum of 60 apps at a time. This proved successful, especially when comparing apps across multiple genres and updated versions of the apps.

Although the method⁸ has been tested and refined to be as automated as possible, it still requires further automation. Work will be concentrated on the

⁸ The python code used in the method is provided by CC licence ©AT&T and University of Portsmouth. The author would appreciate feedback on the code as well as any suggested improvements.

P.E.M.P.

next two labour intensive areas. The first area for automation will be with the app download and the next area will be concentrated on the permission extraction and population of a database for each app's permissions. This would facilitate access to the data for processing and analysis.

The process uses open source software and the code is easily updated to incorporate changes to the permission databases or for the author to concentrate on another part of the app code. Use by other researchers has shown that the process is robust and is easily used to extract and analyse multiple apps/genres.

The next chapter utilises this method to review children's apps.

Chapter 8. Analysis of 2015 Children's Apps

8.1 Introduction

The analysis of the Antivirus apps concluded with the development of a method to automate the analysis process of the apps, resulting in the PEMP method. The method had been tested in the utility genre apps with apps across a four-year time span but needed to be tested in another genre to show robustness. Children's apps were selected for the next stage of testing.

In 4Q15 Android had over 80% of the Worldwide Smartphone Market share (Puneet Sikka, 2016). Research in the US in 2013 by Vicky Rideout (Rideout, 2013) on the usage of mobile media of children under 8 revealed that 38% of children under 2 years of age had used a smartphone or a tablet, this is up from 10% two years ago. By the age of 8, 72% have used one of the devices. This has increased from 52% of 8-year olds using these devices in 2011.

The increase in usage of these age groups increased the concern that had been raised by other researchers and business groups, Mumsnet etc, that children are vulnerable to being tracked (geo-location) or monitored (camera, voice recording) inadvertently by the apps that they were using (either games or educational apps).

The increased usage of these devices by young children has exposed them to being tracked through location sharing technologies (de Souza e Silva, 2013) as well as affecting their privacy (Duncan, 2011).

This part of the research concentrated on the privacy aspect of apps aimed at children. The objective was to determine if the privacy of children was being abused either by monitoring or spying on children without the parents or guardian's permission.

8.2 Motivation

Earlier this research into security apps showed that the apps that were supposed to protect the user, also abused the user's privacy. Where adults have a reasonable awareness of privacy and are concerned at the erosion of their privacy online, the research of Palfrey, Gasser and Boyd (Palfrey, Gasser, & Boyd, 2010) showed that youngsters are also concerned by this erosion but have a different perspective to what can or should be disclosed. Often the skills and knowledge to protect themselves are missing. Children are especially vulnerable and through peer pressure will disclose private information. Therefore, it is important to provide some modicum of protection until the requisite skills are learnt. This part of the study was to determine if there were differences or similarities between the permissions requested for the different age groups or whether similar apps used the same permission requests across all the age groups. The apps were also reviewed to determine how many of the apps requested permissions perceived to be "privacy" related permissions.

8.3 Method

Google uses multiple categories to group apps. Children apps are in the Family category and are also divided into 10 sub-categories. The twenty most popular free children's apps were selected from the following sub-categories; Ages 5 & Under, Ages 6-8 and Ages 9 & Over.

The Google Play Store (and other app stores) contain thousands of apps and increasing daily. When a new app is released it has a temporary visibility as a new app and then drops to become part of the general apps unless positively promoted. All app stores use a ranking system which is kept confidential. Google's ranking was used when selecting the apps and the top 20 were downloaded from the Popular apps & Games group in each age range. The rankings are not constant and to ensure consistency in the selection of the apps over time the initial ranking of the app was recorded.

Recent research into app ranking performed by Stuart McIlroy et al (McIlroy, Ali, & Hassan, 2016), used Distimo, an app analytical tool. These tools are commonly used by developers including crashing and bug tracking analytical tools to obtain a higher ranking of their app.

These 60 free apps were analysed as follows:

- Define the permissions considered as conflicting with the user's privacy
- Number of permissions requested for each app
- Redundant permissions
- Similar apps across the different age ranges
- The variety of developers
- The similarity or not between the app's functions

- The similarity or not of the permissions requested.
- The permissions were analysed and marked for privacy or not markers. The number of anti-privacy permissions for each app.

8.4 Permissions that Affect User Privacy.

Permissions are requested to permit the app to access core or system facilities in accordance with the sandbox design of the Android operating system. There are apps available on the Google Play store that will list the number of permissions of each installed app and will list the permissions for that version of Android, but will leave the decision to block the permission (if possible) or to uninstall the app to the user, three examples are; PrivacyBlocker, the free version is Privacy Inspector ("Privacy Blocker," 2017), Permissions – Privacy ("Senior Lab DE Apps," 2017) and Snoopwall ("Snoopwall App," 2017).

8.4.1 Apps for Children aged 0-5 years

The apps were selected using the default Google Ranking system, this is the order that the app is displayed to the user on the Play Store. The top 20 free apps were selected. The ranking order of the app and the number of downloads, user rating and permissions are described in Table A-11.

The 20 apps in this category were supplied by 7 app providers. The most popular apps were the ones supplied by the Lego Group with 4 apps. The Disney group was second with 3 apps.

The permission frequency of the apps in this age group of the study is shown in Figure 8-1. The median permissions requested was 5 and the median rating for these apps was 3.8 (out of 5).

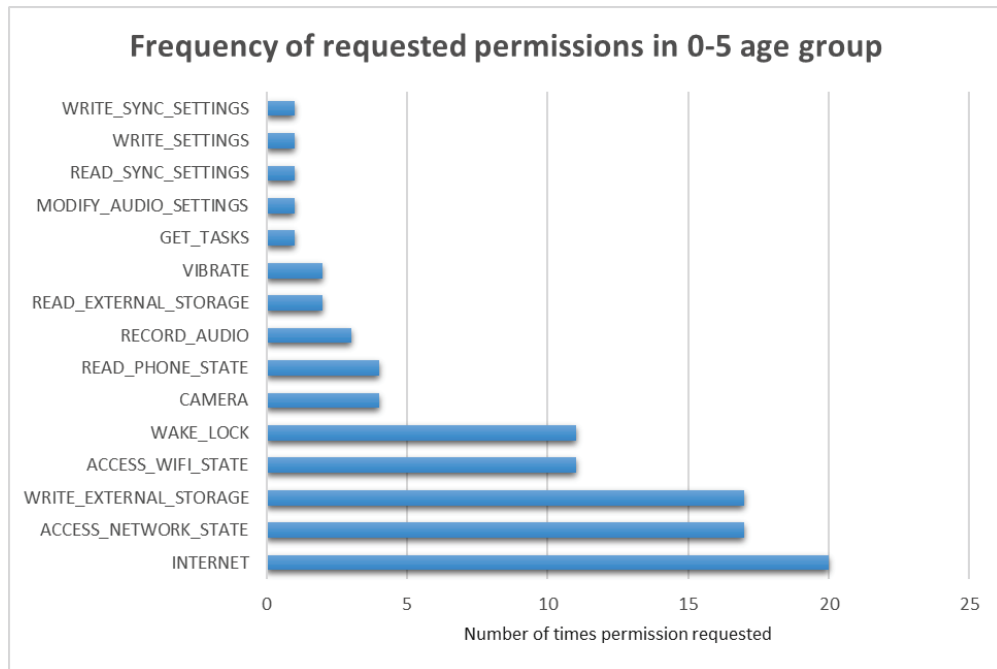


Figure 8-1 Frequency of the requested permissions in the 0-5 age group apps

In this age group the most frequently requested permissions were for the Internet, Access Network State and Write to external Storage. These permissions allowed an app to determine the network access and to connect to the Internet. The requesting apps were also able to write to an SD card if installed or to mobile device memory that has been configured as external storage. The request for WRITE automatically assumes READ access.

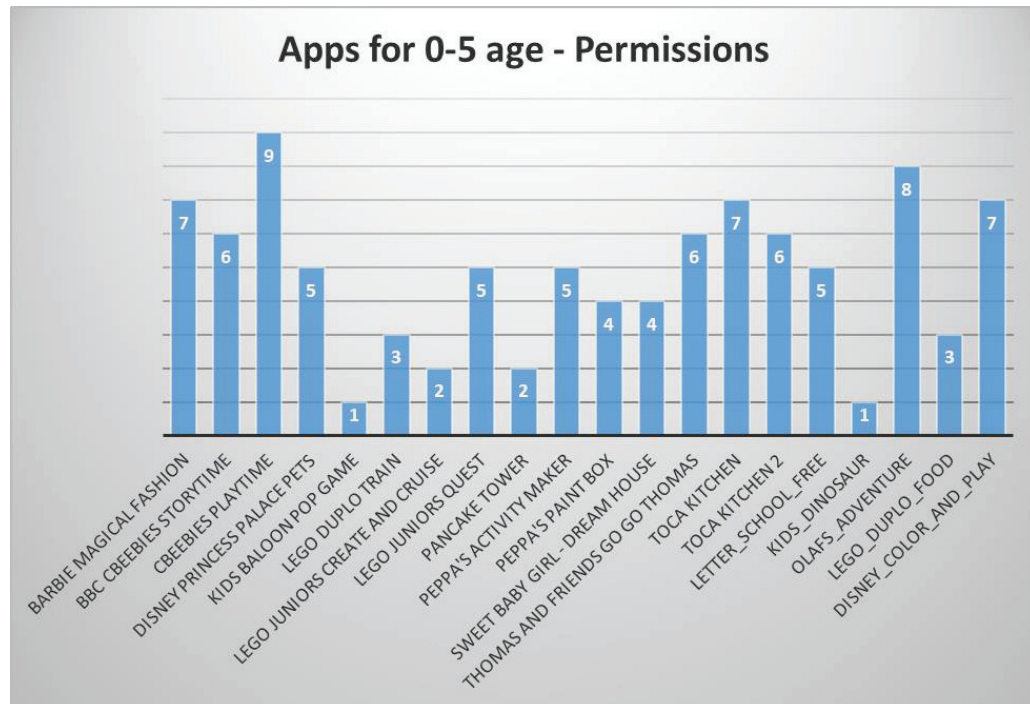


Figure 8-2 Permissions requested by each of the studied apps in the 0-5 years age group

The privacy marked permissions for each app were analysed and 4 of the apps requested permissions that were marked as anti-privacy (Table 8-1). Of these two apps requested multiple anti-privacy permissions. These apps were Cbeebies Playtime and Disney color and play, who both requested access to the camera and to audio.

Table 8-1 Age group 0-5 apps requesting anti-privacy permissions.

<i>app Name</i>	<i>CAMERA</i>	<i>record_AUDIO</i>
Barbie magical fashion	Y	
BBC Cbeebies story time		Y
Cbeebies Playtime	Y	Y
Peppa's activity maker	Y	
disney_color_and_play	Y	Y

8.4.2 Apps for Children aged 6-8 years

The 20 apps in the study for the 6 to 8 age group is shown in Table A-12. The table shows the package name, developer, user rating, number of downloads and number of permissions requested.

The 20 apps in this category were supplied by 11 app providers. The most popular apps were the ones supplied by the Disney, Lego and Budge, who supplied 4 each.

The permission frequency of the apps in this age group of the study is shown in Figure 8-4. The median permissions requested was 6 and the median rating for these apps was 3.8 (out of 5). With the King of Math Junior – Free not requesting any permissions.

The median permissions requested was higher than the younger age group although the median rating was the same.

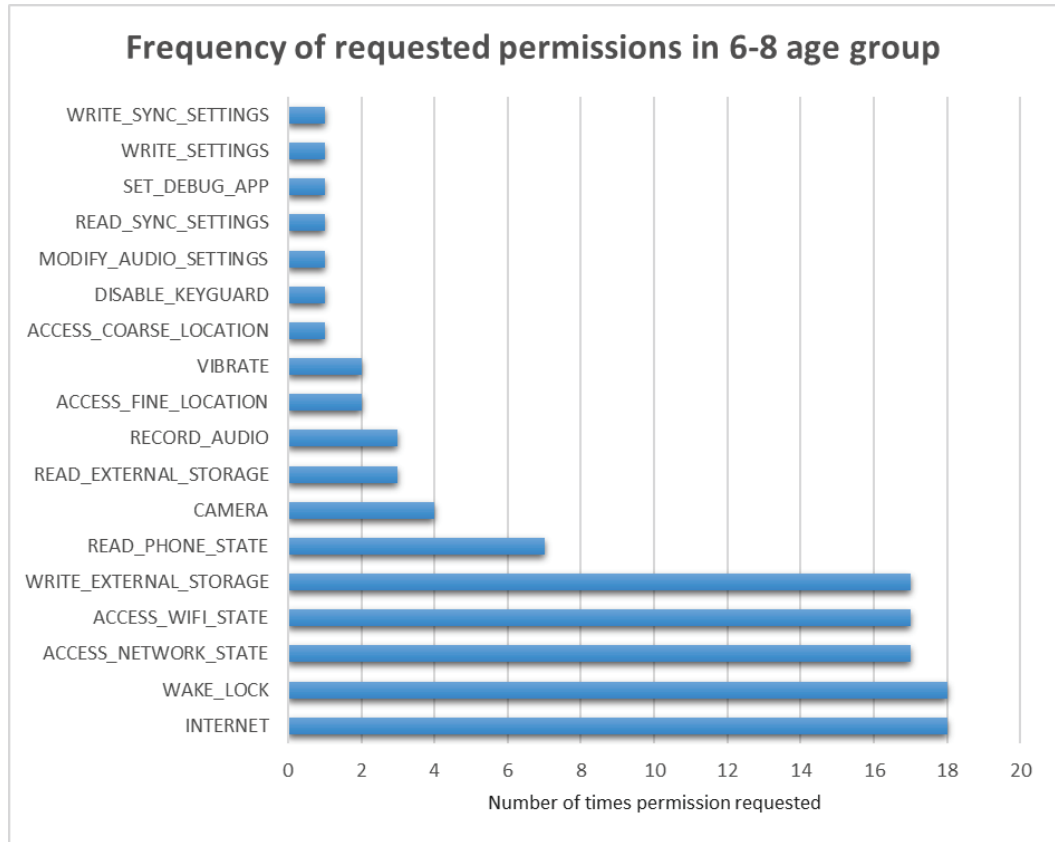


Figure 8-3 Frequency of the requested permissions for the 20 apps in the 6-8 age group.

In this age group 5 permissions were requested most frequently. Again, Access Network State, Write External Storage and Internet were requested, with the addition of Wake lock and Access WiFi state. These last two permissions permitted apps to activate the phone without the user's knowledge, for example at night, and to be able to determine the WiFi access and to activate it to logon to a WiFi network also without the user's knowledge.

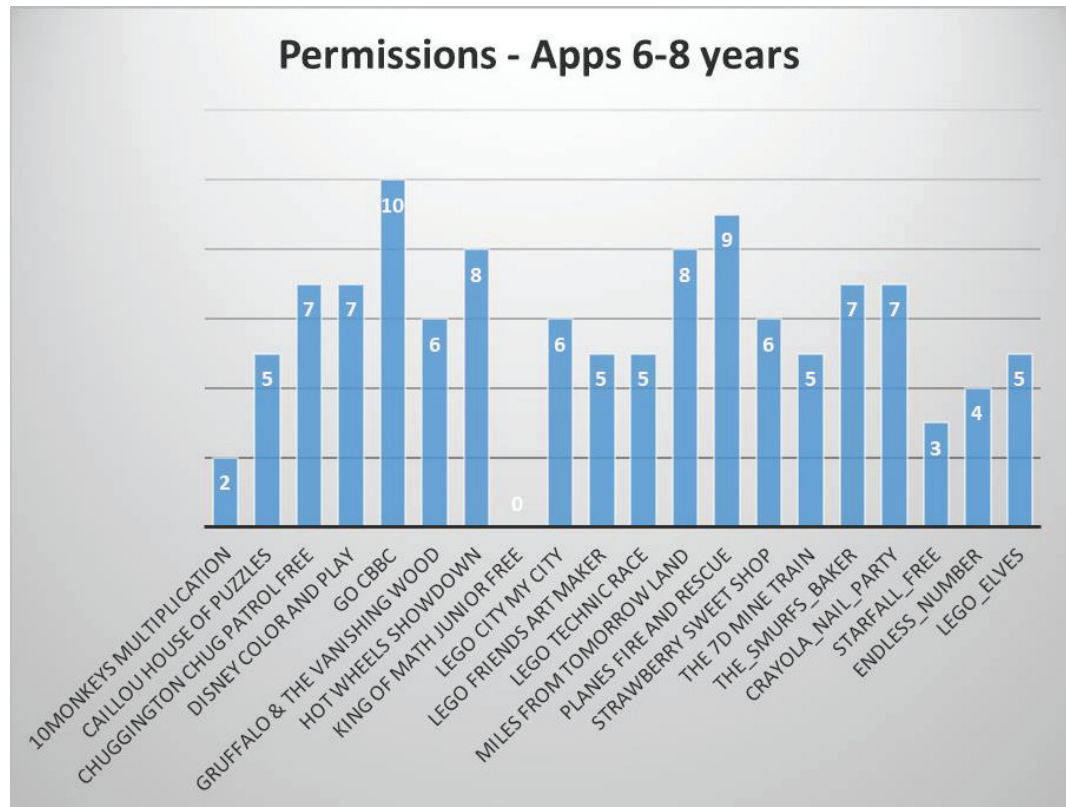


Figure 8-4 Permissions requested in apps for children in the 6-8 years age group

The privacy marked permissions for each app were analysed and only one of these permissions was requested and this was requested by three of the apps.

Table 8-2 Apps requesting anti-privacy permissions for 6-8 year group.

<i>app Name</i>	<i>record_AUDIO</i>
Disney color and play	Y
Go CBBC	Y
The_Smurfs_baker	Y

8.4.3 Apps for Children aged over 9 years

The 20 apps in the study for the over 9's age group is shown in Table A-13. The table shows the package name, developer, user rating, number of downloads and number of permissions requested.

The 20 apps in this category were supplied by 10 app providers. The most popular apps being the ones supplied by Disney (5) and Gameloft (4).

The permission frequency of the apps in this age group of the study is shown in Figure 8-6. The median permissions requested was 7.5 and the median rating for these apps was 4.2 (out of 5). The median permissions and rating were the highest in this age group.

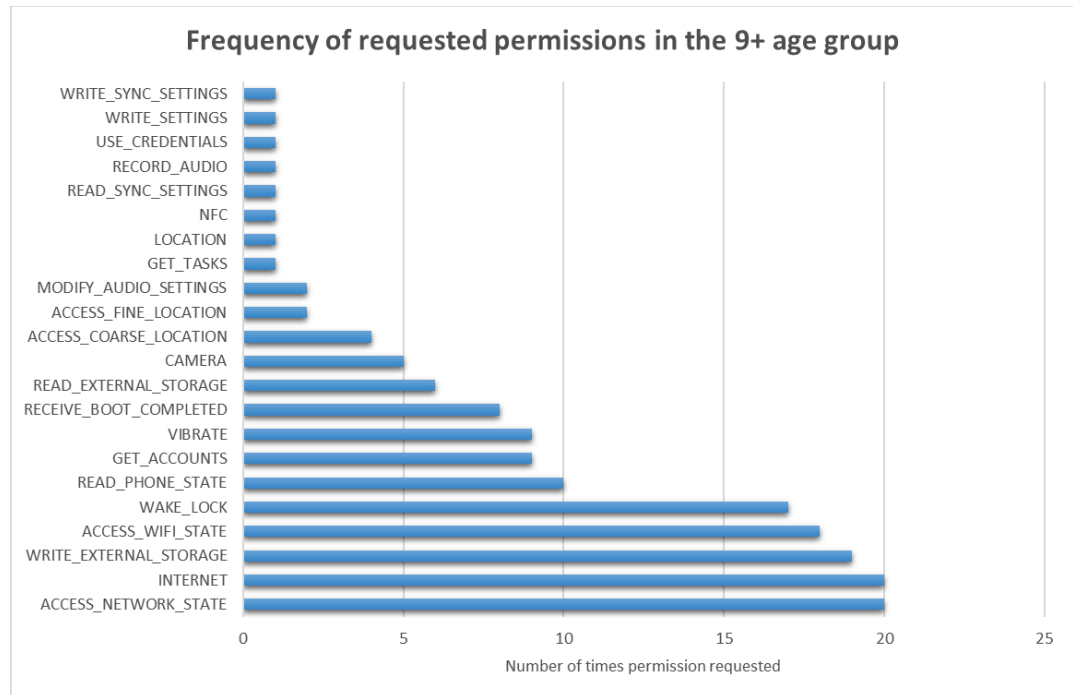


Figure 8-5 Ages 9+ app permission frequency

In this age group, the same five permissions were requested, Wake lock, Access WiFi state, Write external storage, Internet and Access network state. All the apps in the study requested the Internet and Access network state permissions.

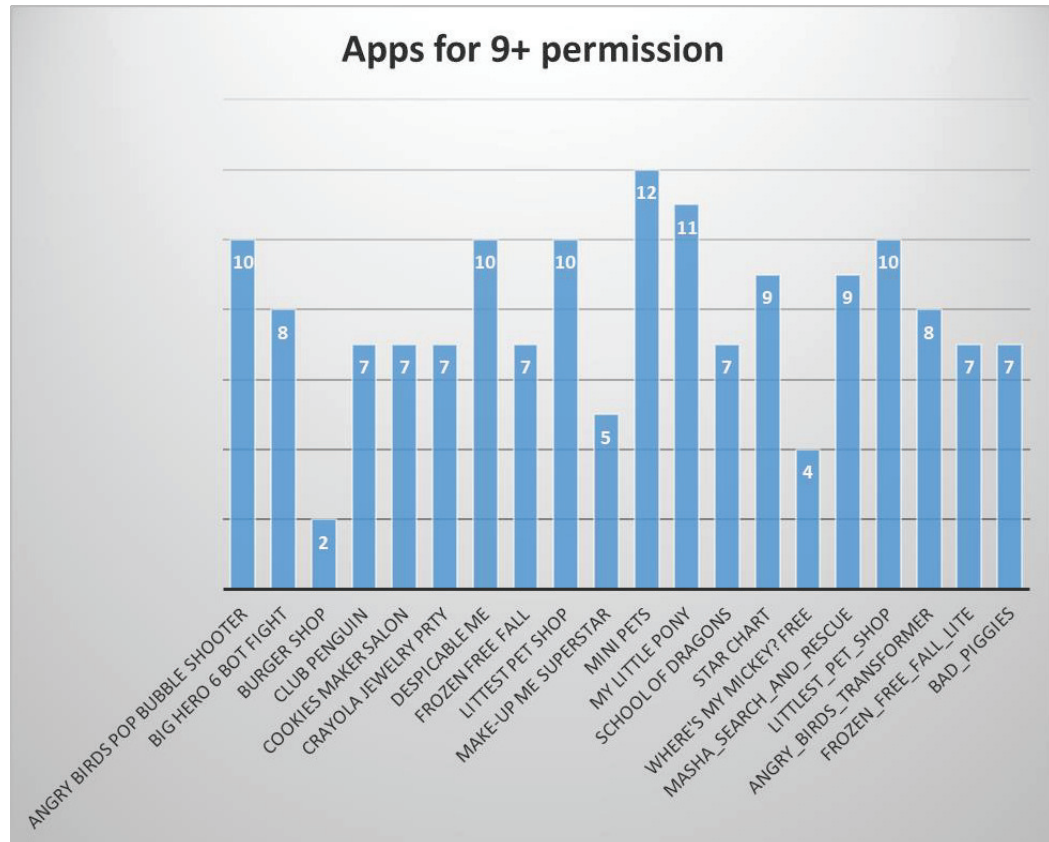


Figure 8-6 Permissions requested in apps for children in the 9+ age group.

The privacy marked permissions for each app were analysed. 10 of the apps requested permissions marked as anti-privacy. Two of the apps requested more than one anti-privacy permission (Table 8-3).

Table 8-3 Apps that have requested anti privacy permissions

<i>app Name</i>	<i>ACCESS_COARSE_LOCATION</i>	<i>ACCESS_FINE_LOCATION</i>	<i>camera</i>	<i>record_AUDIO</i>
	N	N	a	O
Cookies maker salon			y	
Crayola jewellery party			y	
Despicable me	y			
Littlest pet shop			y	
Mini pets	y	y		
My little pony	y			
Star chart		y		Y
Littlest_pet_shop			y	
Angry_birds_transformer				
Bad_piggies	y			

8.5 Results

Comparing the three age category's permissions illustrated that the mean number of permissions had increased in relation to the age of the user (Figure 8-7).

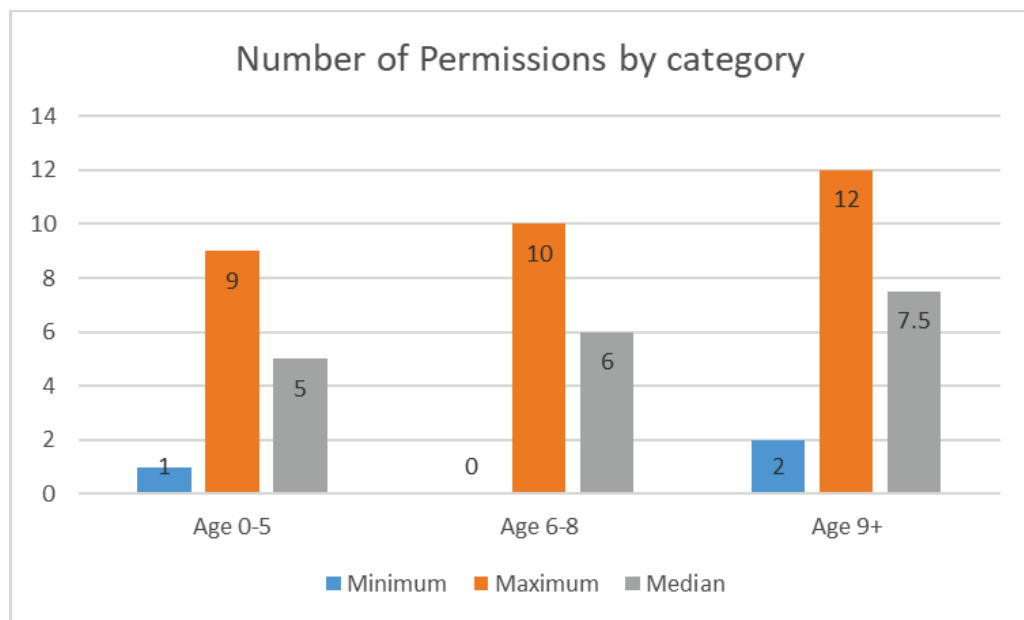


Figure 8-7 Number of permissions by age category

The frequency of the permissions requested across the age groups was then evaluated and Figure 8-8 shows the frequency that a permission was requested of the apps in that age group.

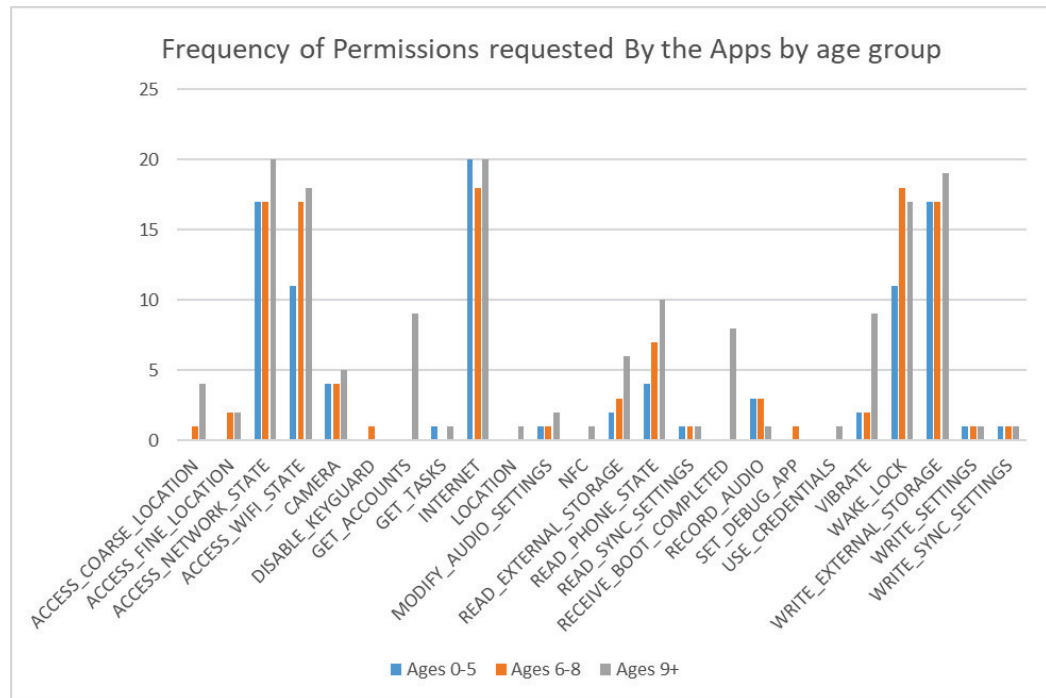


Figure 8-8 Frequency of permissions requested in each age group.

These frequencies illustrate that predominately the number of permissions requested increased in the older age bracket, the main exception to this was the requests for DISABLE_KEYGUARD and SET_DEBUG_APP, which were only requested in the 6-8 age bracket and RECORD_AUDIO that was only requested in one app in the ages 9+ bracket.

Reviewing the frequencies of the apps across all three age groups showed that the apps use a standard set of permissions. These permissions are; Access_Network_State, Access_WiFi_State, Wake_Lock, Write_External_storage and Internet.

The main concern is that the apps were requesting permissions deemed contrary to the user's privacy also had Internet access. Over the 3 age brackets 97% of the apps had requested Internet access.

Although these apps are not necessarily collecting camera, location data, account data etc, the Internet access could be used by colluding with data collection apps and using Covert Channels to provide this data to a third party (Marforio, Francillon, & Capkun, 2011). Marforio et al, describe how applications can collude on smartphones by bypassing the restrictions of their own permissions and using covert channels. This technique is useful to a data collector as once installed an app's permissions in the Manifest file are normally immutable.

8.6 Conclusion

Reviewing the range of permissions requested by the apps in each age group, the maximum number of permissions increased by age.

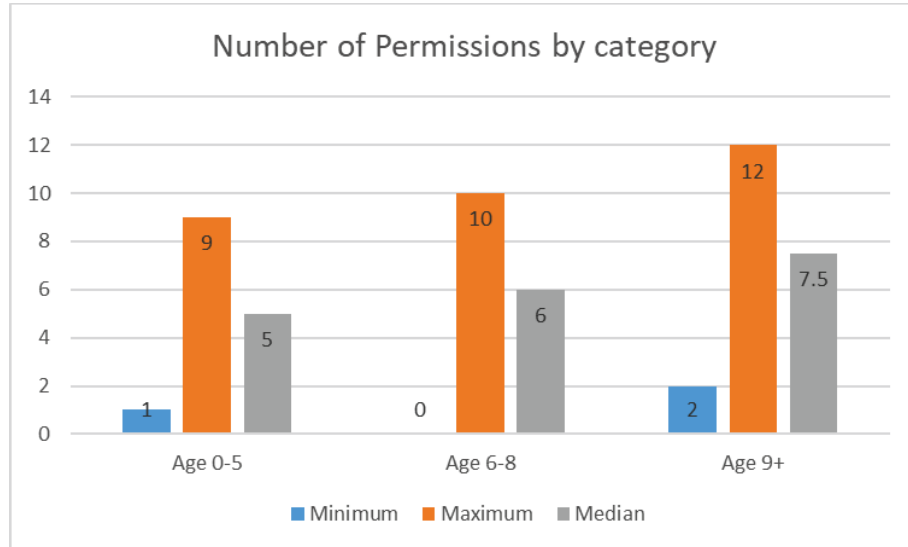


Figure 8-9 Number of permission requested by age group

A variation to the increase in the number of permissions by age group was in the 6-8 age group where one app did not request any permissions. This prompted the question “Are there any permissions that are added as default and therefore not proffered to the user to accept or reject?” The app requesting no permissions was the “King of Math Junior – Free”. This app is aimed at parent schooling of mathematics in this age group. The interaction is only on the device and therefore does not need any permissions.

Reviewing the privacy permissions requested by age (Figure 8-10) indicated that more anti privacy permissions were requested of the older age group.

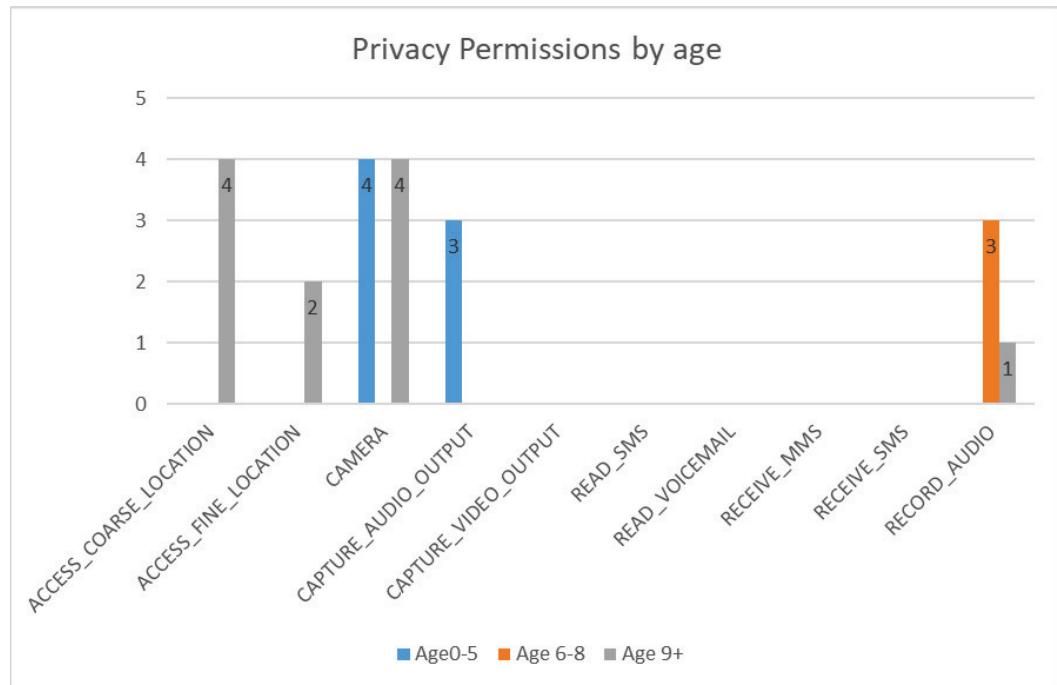


Figure 8-10 Requested antiprivacy permissions by app

The permissions in this age group related to location, camera and audio access. This permitted the child to be location tracked and overheard as well as visual surroundings being recorded.

In summary, the initial hypothesis that children were not fully protected was incorrect. However, Anti-privacy permissions requested increased in the older age groups. The main concern was that the privacy permissions requested in this age group was for tracking these were the locations, both coarse and fine and the camera and audio. Fortunately, the apps did not request these at the same time, but it indicates that it is not single permissions which could be a problem but how multiple permissions are being used in conjunction with each other.

The next chapters build on this initial research by describing the expected privacy needs of a user in relation to social and psychological contracts and

what the app marketplace owners have incorporated into their sites to “*protect*” the user.

Chapter 9. Privacy and Social and Psychological Contracts

The previous chapter introduced the concept of apps privacy and how the marketplace owners are “protecting” the user. This chapter builds on this by associating the privacy of the apps with the presumed privacy requirements of the user and the user’s perception of their privacy protection.

Previous research into privacy reviewed it from the perspective of the user (Brunk, 2002), the technical view (Enck et al., 2014) and from a forensic perspective (Tsavli et al., 2015), this research reviews privacy from a social and psychological perspective.

Social contracts were first coined and described by Jean-Jacques Rousseau in 1762 in his work *The Social Contract*. Where Thomas Hobbes in *Leviathan* (1651), proposed that individuals relinquish their individuality to obtain security through a holder of absolute power, Rousseau advocated that individuals surrender their rights under a social contract to form one moral will. Both theorists are advocating an individual’s right to use free will. A social contract is defined in the 2017 Oxford English Dictionary as:

An implicit agreement among the members of a society to cooperate for social benefits, for example by sacrificing some individual freedom for state protection.

Therefore, the concept of Social contracts which apply to users of mobile apps can be described as;

Users are permitted to utilise an app and to do so agree to relinquish personal information to the app developer or provider as defined in the requested permissions.

Users are unable to use the app, even in free mode or as a trial, unless they click the “I agree” button. Interviews of Android users (Kelley et al., 2012)(Kelley et al., 2012) found that users paid little or no attention to the permission request screens and that they did not understand the implications of the permissions requested. This study was corroborated (A. Felt, Ha, Egelman, & Haney, 2012) by similar research based on Internet surveys and lab studies.

Users are inadvertently, unknowingly or unconcernedly relinquishing ownership of their privacy, so they can install and run an app. This brings into question the social contracts between users and the app providers, permissions vs data protection and who benefits from this partnership. In GDPR, the main principle is the protection and control of the user’s privacy data. It is not known yet what the impending implementation of GDPR will have on these agreement forms. What can the developer/marketplace keep and in what format. How is this big data summarized by the likes of Google?

Users do not normally think that their private information is a commodity that can be sold or shared. Information sold or shared to external groups is summarized, but the level of granularity is not shared, and the initial collectors do not state the level of detail that is being kept and for how long and who has access at this level of detail.

Big Data has become an industry and many companies use this data to target users. Telecoms mobile companies collect data of customer usage as part of their vision to enable it to build a better customer experience. This ranges from

making more bandwidth available for customers during the times of heavy usage, e.g. downloads of videos, live streaming of TV programs, sports events and movies. Adhoc data usage, e.g. the upload/download of personal files, facetime or skype or Whatsapp calls are more difficult to predict but many of these calls will be performed during time ranges that the provider can plan for by viewing historical data. If the provider detects that more usage of a specific type of traffic, SMS, voice calls, data usage occurs in specific areas then it can install more cell towers or even convert the existing towers to a newer technology (LTE/5G).

All this data is being kept about the user, totally unknown to the user.

Some mobile network providers sell their mobiles with a skin (as do mobile manufacturers) the data collected by these widgets is not disclosed. For example, if a user has a weather app that is active on their device and they use it to track more than one location's weather the provider can extrapolate that the user (based in a location acquired from geo tagging) is monitoring their own weather and that of another location that they are interested in. Once the user goes to that other location it is detected (geo tagging or cell tower tracking) the provider can then extrapolate that this user will in future travel or connect with somebody in that location or future locations that the user will go to, based on the usage of the weather app and the locations checked and visited. The same is true of social media sites that track the user's location and their friends using geo tagging. The benefit to each party is slightly skewed to the provider as the provider can target ads for the additional location (hotels, restaurants, travel options) whilst the user has access to a single piece of information (weather in that location).

Therefore, the user has given up their personal information, location, expected travel plans, an area of interest or where contacts are located to be able to view

the weather at that location. The permission request is not made as the app is pre-installed on the device. The user is unaware that additional data of the other locations could be obtained and used by the provider as a saleable product.

Marketplace app marketplaces provide terms of contract for users to agree to. These contracts are long and complex and are available online. The main areas relating to Privacy in Apple's and Google's Terms and Conditions are summarized below.

9.1.1 Apple's Privacy Terms and Conditions

Apple's Media Service Terms and Conditions contains a section that refers to a separate privacy terms.

"PRIVACY

Your use of our Services is subject to Apple's Privacy Policy, which is available at <http://www.apple.com/legal/privacy/>.

Apple has country specific privacy policies. In the UK version Apple states that it "may collect a variety of information, including your name, mailing address, phone number, email address, contact preferences, and credit card information" it may also "collect the information you provide about those people such as name, mailing address, email address, and phone number. Apple will use such information to fulfil your requests, provide the relevant product or service, or for anti-fraud purposes."

Apple will use it for contacting for promoting services as well as auditing, data analysis and other research."

Fundamentally Apple will collect the data, perform data analytics and then either use it to promote its own services and/or provide this data to strategic partners, law enforcement and other service partners.

Apple has a separate clause re Children and Education.

“We understand the importance of taking extra precautions to protect the privacy and safety of children using Apple products and services. Children under the age of 13, or equivalent minimum age in the relevant jurisdiction, are not permitted to create their own Apple IDs, unless their parent provided verifiable consent or as part of the child account creation process in Family Sharing or they have obtained a Managed Apple ID account (where available) through their school. For example, a parent must review the Apple ID and Family Sharing Disclosure and agree to the Consent to Apple’s Collection, Use and Disclosure of Your Child’s Information; and the iTunes Store Terms and Conditions, before they can begin the Apple ID account creation process for their child. In addition, schools that participate in Apple School Manager and have reviewed and consented to the Managed Apple IDs for Students Disclosure may create Managed Apple IDs for students. The Managed Apple IDs for Students Disclosure describes how Apple handles student information and supplements Apple’s Privacy Policy. Learn more about Family Sharing, the Managed Apple IDs and Restrictions for children’s accounts.

If we learn that we have collected the personal information of a child under 13, or equivalent minimum age depending on jurisdiction, outside the above circumstances we will take steps to delete the information as soon as possible.”

If at any time a parent needs to access, correct, or delete data associated with their Family Sharing account or child’s Apple ID, they may contact us through our Privacy Contact Form.

9.1.2 Google's Privacy Terms and Conditions

The Google T&Cs also refer to a specific Privacy Policy.

As with Apple Media, Google states the type of information that it collects. These include but are not limited to:

When Google services or view content is used, Google automatically collects and stores certain types of information. These include;

- Details of how the service is used, for example search queries
- Telephony log information, such as telephone number, calling number, time and date of calls, SMS routing
- IP address
- Event informant, such as crashes
- Cookies, that identify your browser or Google account

As well as location information, local storage (web storage) and cookies

Google also performs data analytics of this data and profiles the Google Account to target services and products.

Google also has the right to use the name specified in your Google Profile across all its services that require a Google Account, including replacing all past names associated with the account across all services. Other users that your email or other identifying information may be shown your publicly visible Google Profile information, such as your name and photo.

Fundamentally you have loosened any control over who has access to your profile. Your data is shared across all the google services irrespective if the service is used or not.

Personal information is also provided to Google affiliates, law enforcement and partners (like publishers, advertisers or connected sites).

Google Age Restrictions

“Age Restrictions. In order to use Google Play you must have a valid Google account, subject to the following age restrictions. In order to serve as the family manager of a family group on Google Play, you must be at least 18 years old. You must not access Google Play if you are a person who is either barred or otherwise legally prohibited from receiving or using the Service or any Content under the laws of the country in which you are resident or from which you access or use Google Play. You must comply with any additional age restrictions that might apply for the use of specific Content or features on Google Play.”

The passage on the rights/protection of Children places the responsibility of the download/installation of the app with an adult. However, there are no controls in place to verify that the consentor is an adult.

Fundamentally, Google will track the user and their online behaviours, irrespective of age as by selecting “I agree” you have confirmed that you are over 18.

9.1.3 Protection Normal

apps on Google Play must also follow Google Play's policies. Google removes apps that are found to violate these policies. Google also has systems that analyze new and existing apps, along with developer accounts to help protect users against potentially harmful software

Google has designated a base set of permissions as *protection normal*, to indicate that there's no great risk to the user's privacy or security in letting apps have those permissions. For example, users would reasonably want to know

whether an app can read their contact information, so users must grant this permission explicitly. By contrast, there's no great risk in allowing an app to vibrate the device, so that permission is designated as *normal*.

If an app declares in its manifest that it needs a normal permission, the system automatically grants the app that permission at install time. The system does not prompt the user to grant normal permissions, and users cannot revoke these permissions.

To minimise the number of permissions that the user consents to at app download and install, Google introduced the designation “*protection_normal*” (“Protection Normal,” 2017). This designation applies to permissions which Google has determined that there's “no great risk to the user's privacy or security in letting apps have those permissions”. If the app declares in the manifest file that it needs a normal permission, then the system automatically provides the app with that permission at install time. The user is not prompted at install time to agree to these permissions and is not able to revoke any of them. These designated permissions are listed in in the appendices in Table A-14.

Unlike the explicit permissions request made by apps, these permissions are implicitly accepted as part of using an Android handset, the app permissions are requested for acceptance as normal.

This acceptance permits the provider to track the user, change how the mobile is connected (Bluetooth, network and/or WiFi), change the look and feel of the device (manufacturers or providers skin). Reboot the device and override physical security (fingerprint). Therefore, the user's privacy and security are already compromised.

Individually the permissions are not greatly impinging on the user's privacy, however when used in conjunction with other permissions the privacy infringement increases.

9.2 Social Contract Obligations

In social contracts the user expects the device manufacturers and app developers to treat them fairly. Access to a device for them to control their environment or provide other services, for example voice activated or visual commands. And not to spy on them in the confines of their own home.

Although the user is prepared to pay for the app or in-app purchases, the user is unaware of who the contract is with. Currently the purchase of the app is made via the Marketplace provider so that the actual contract is with the app provider and not developer. Therefore, it should be the provider's responsibility to protect the user during the purchase and use of the app. However, the providers limit their accountability by requesting that the user agrees to their Terms and Conditions which are complicated and long. Copies of the T&C's are provided in the appendices.

Rarely do the providers admonish the app developer, as occurred between Apple and Über, which was poorly reported at the time. There was no mention of a similar occurrence between Google and Über, even though the apps performed the same on both operating systems (iOS and Android).

9.3 Psychological and Implied Contracts

Two other unwritten agreements are psychological and implied contracts. The terms psychological and implied contracts was originally developed by Denise Rousseau. She described the subjectivity and nature of the contracts and how it is applied to organisations (Rousseau, 1989). Her definition is

“The term psychological contract refers to an individual’s beliefs regarding the terms and conditions of a reciprocal exchange agreement between that focal person and another party”

In a psychological contract the individual expects the company to reciprocate or be obligated to the individual due to the contribution that the individual makes. This belief is held only by that individual that a contract exists.

Denise Rousseau used a biblical parable about the vineyard owner employing workers to work for him as an example, their expectations and the owner’s contract with them. His contract with the workers was that he would pay a fair day’s wage. At payment time the workers that had worked the full day expected a higher payment than those employed later during the day. These workers were aggrieved as they felt that they had been unfairly treated. Whilst the owner felt he was keeping to the agreed contract to pay a fair day’s wage. The workers psychological contract was with the term “fair”, where they felt that the wage would be in proportion to the hours worked, an implied contract.

Implied contracts are a mutual obligation between the two parties and the relationship evolves over time. The evolvment binds the two parties together and makes exit a possible expensive option.

There are also differences in the level and point of view of these contracts. Employees that work for a company for many years and in their view, go above

and beyond (working late and over weekends) expect the employer to recognise and reward this loyalty. However, the employer only acknowledges the content of the contract that an employee works for a fixed number of hours and is paid for those hours. The employee's psychological contract is that "the more I work and perform, the better my standing is with the company and my employment is more secure". The employee's extra hours are not expected and are not part of the agreed contract and thus do not have any bearing with the employer.

Applying the concept of psychological contracts to app purchase and use, the user's expectation is to be able to play the app with no hindrance. Most of the apps that are free on the Store contain adware or in app purchases. This disrupts the continuous flow of the app and contravenes the psychological contract that the user perceives to have with the developer. From the developer's perspective, the app is developed to provide income. Managing this perception is key to encourage the user to continue using the app and continue to update the app if available and to purchase powerups or another add-ins. This "loyalty" to the app ensures that there is more opportunity for the user to click on the adware or the in-app purchases, thereby providing more income to the developer.

Breaches of the perceived contract can severely damage the relationship between the user and the developer. These range from deleting the app and possibly providing a negative review on the Store, thereby putting other users off and reducing income.

An implied contract is often considered to be legally binding. In this case the contract is that the user may download and play the app, but may not plagiarise the app. This would be to copy the coding of the app, making minimal changes and selling it on the store as a unique app, thereby reducing

the real developer's income. Currently this does not seem to be policed proactively and there are many apps on the store with very app similar names. The onus is on the developer or user to report this to the Store monitors.

Rousseau (Rousseau, 1989), illustrates the differences between these two types of contracts, and I have used her diagrams as a foundation to reflect the usage of these contracts in the mobile app environment.

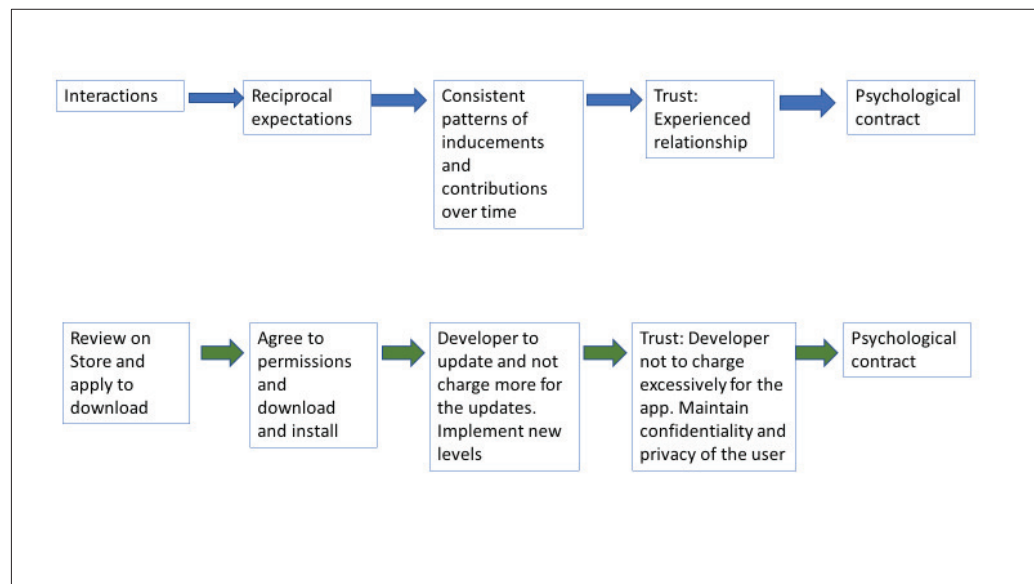


Figure 9-1 Development of a psychological contract

The psychological contract is the individual's (user's) perception of the contract. In this contract there is an expectation of trust between the user and the developer. This can include confidentiality of the user's details (name, age, etc.) and their privacy expectation, e.g. location. The developer trusts that the user will not cheat him and will pay for using the app in one form or another.

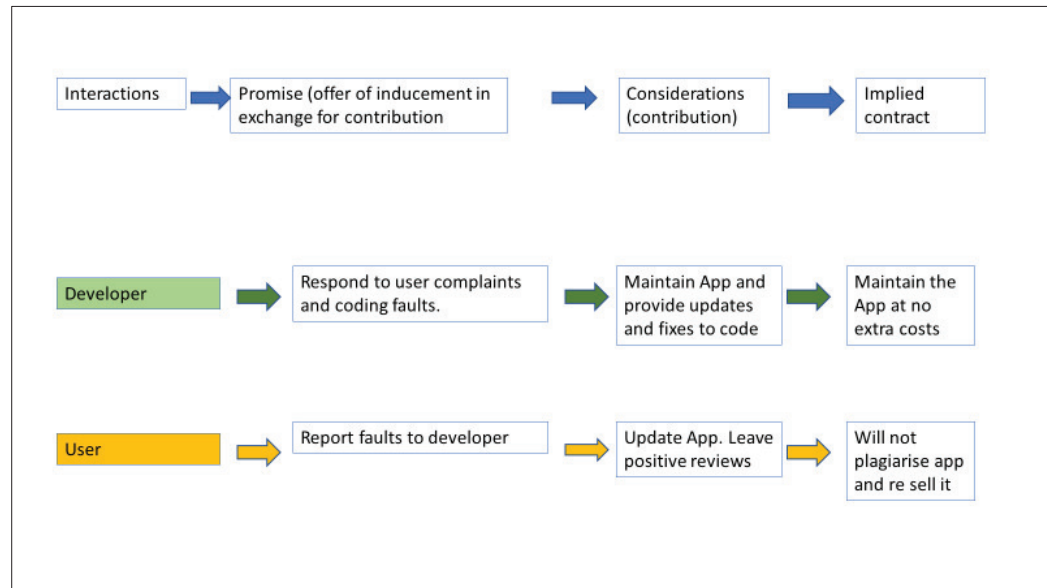


Figure 9-2 Development of an implied contract

The implied contract is the relationship between the user and the developer, and both perspectives are displayed in Figure 9-2. For the contract to be reciprocal, both parties have similar goals. The developer wants their app to be popular (and provide income) and the user wants the app to fulfil their requirements, for which they will pay an amount, either for the app itself or for in-app purchases.

The comparison of the different types of contracts as applied to app purchase and use are summarised in Table 9-1.

The user plays the app, only by agreeing to a permission list at download. The app may contain adware or in-app purchases. The app may or may not be maintained or updated with new levels. The play is interrupted to offer in-app purchases or be delayed whilst a timer runs down. User privacy is impacted, and the user has no control over their data and no recourse to control its use. The user may incur more costs to continue to play the app, even after initial payment to purchase the app has occurred.

Table 9-1 Summary of contract types

<i>Contract type</i>	<i>Activity</i>	<i>In actuality</i>
Social	Developer protects user data and ensures that the app is maintained.	Developer may update app or fix coding issues. Developer may also use the user's data to provide a target group for new apps or provide additional functionality.
Psychological	The user expects the developer to maintain the app without interference	Developer often uses adware and in-app purchase to increase income. Usage is interrupted and may affect the user's enjoyment of the app.
Implied	The developer sells the use of the app and the user agrees to pay and not steal the app code.	If either party breaks the agreement there could be legal consequences

The user is at a disadvantage as none of the three contract types are active. Equity Theory (J. S. Adams, 1965) deals with exchange and fairness. The psychological belief of the employee is expecting an exchange of fairness. Adams suggests that the employee comparing him/herself to a neighbour and believing that they should earn more is an expectation not a psychological contract. Reciprocal expectations in a contract believe their actions are bound to another, employee and employer. An employee expects more income but understands that there is no obligation for the employer to give them a raise.

However, the experience of inequity differs from an actual or implicit contract as it is not enforceable by law. As discussed above, the employee who is loyal and works hard, expects to be rewarded. When the reward is not forthcoming the employee becomes dissatisfied and their performance will be affected, with withdrawal of the employee being the last resort. This relationship is far easier to repair than contracts.

Violating a psychological contract has similar results. The employee begins to distrust the company and the relationship is badly damaged and is difficult to

repair. Sometimes the trust must be rebuilt as happened at the beginning of the relationship of the employee and employer part company.

In the app development arena, the user has paid or accepted free use of an app and agrees to some in-app purchases. Some developers force the user into paying large sums of money to un-lock game levels or remove ads. This creates an atmosphere of inequity, *“I’ve paid for the app, why should I pay more to play it?”* attitude. This resentment can spill over into the reviews for the app, advising other users not to buy it. A similar resentment occurs when a developer does not repair errors in the app. This creates a deep psychological distress, leading to frustration and disappointment.

Once a user experiences one problem it will heighten the psychological distress of any subsequent problems. This could lead to negativity in the user’s life experience. This is a concern where minors or teenagers are concerned as their life perception could be tainted.

9.4 GDPR – EU Privacy Regulation

The EU has decreed that by May 2018 all companies that process data about individuals in the context of selling goods or services inside the EU must comply to their General Data Protection Regulation (GDPR).

The GDPR not only applies to organisations located within the EU but it will also apply to organisations located outside of the EU if they offer goods or services to, or monitor the behaviour of, EU data subjects. It applies to all companies processing and holding the personal data of data subjects residing in the European Union, regardless of the company’s location. (“GDPR FAQs,” 2017).

Any company breaching the regulation can be fined up to 4% of annual global turnover or €20 Million.

The main article that will affect app providers is article 7, which deals with consent.

The conditions for consent have been strengthened, and companies will no longer be able to use long illegible terms and conditions full of legalese, as the request for consent must be given in an intelligible and easily accessible form, with the purpose for data processing attached to that consent. Consent must be clear and distinguishable from other matters and provided in an intelligible and easily accessible form, using clear and plain language. It must be as easy to withdraw consent as it is to give it (“GDPR Summaries of Articles,” 2017).

The introduction of this regulation will provide some measure of protection to the users, but only if they understand how their data is being harvested and processed.

The introduction of the GDPR requires that a provider must obtain consent from the customer to collect personal data. Marketing consent is required, and the user must be informed of how their data is to be used. This is one of the ways that the regulation empowers the individuals to control their own personal data.

Comparison to the DPA (Data Protection Act) indicates that many of the user’s rights are strengthened; article 22 (decision making), articles 12/13/14 (transparent information) and article 15 (right of access to the data). Articles 20/18/17 (processing, portability and right to be forgiven) are all new. The data must be provided to the individual upon request and must provide an

overview of how it is to be used and the information must be clearly legible (in a child friendly manner in the case of minors).

Article 5 relates to the processing of the data specifies that the individual must be treated with lawfulness, fairness and transparency, like the concept of a “social contract”.

Compliance to the regulation is yet undefined. The controls for guidance to obtain compliance certification is not in existence and are only expected to be available after the implementation deadline in May 2018.

Obtaining personal data from the use of mobile apps has yet to be defined by the app stores and the GDPR regulators. Android apps usage and data collection of the individual’s data is not specified. Is the data collected for the developer’s benefit or for the Store, for example, Google Store or Apple’s app store? If the developer is the recipient, then the developer must provide the consent form and describe how the user’s data will be used. If Google or Apple are the recipient, then the responsibility for the consent form and description lies with Store owner.

There are multiple problems with either acting as the data collector.

If the developer collects and stores the data for their use to market their products or products that they will receive a fee for, then it means that for every app, the developer must provide a consent form. The developer will also be required to provide a revoke form so that the user can revoke their consent to collect or use the data. At this point, will the user be able continue to use the app and what happens to the historical data that has been collected and possibly sold prior to the revocation request. The administration becomes very costly for a developer to provide and maintain the consent/revokes of users. It

also means that a consent agreement is required for every app and will be more complicated than a permission agreement.

If the Store acts as the collector, then they provide the consent and revoke forms and maintain them. This very advantageous for many Store owners as they already collect and process customer data. The companies already have “privacy” sections to their terms and conditions, and it would be easy for them to add a consent to collect and consent to market from the user as part of the user’s agreement to use the Store rather than at the app download stage.

Again, there is no benefit to the user, especially if the usage of the app is revoked the same time that the user’s revoke request is processed. This then contravenes the concept of *fairness*, one of the principles of the GDPR. Or would the revoke request be treated the same as *the right to erasure*?

There is no definition of what level of granularity the anonymised data is maintained in either storage, transference or sale of the data.

9.5 Privacy Impact of Location Trackers

Research into geo tracking of mobiles using Cell towers, WiFi, RFID and GPS is very popular and there are a variety of papers describing the tracking and how to simplify and improve it from a basic paper in 2009 describing current geo tracking and how to improve the tracking of mobiles (Balakrishnan et al., 2009) to using third party services, such as apps, social media as well as the normal physical tracking (WiFi, Networks, etc) (Razaghpanah et al., 2018). The trend towards inbuilt location awareness was described by Adams and Katos (C. Adams & Katos, 2005). Geo tracking is a useful source of data to companies that collect and use user data. Google publishes the estimated location of

millions of iPhones, laptops, and other devices with Wi-Fi connections. Without the knowledge of the user. Android phones with location services enabled regularly beam the unique hardware IDs of nearby Wi-Fi devices back to Google.

Google make their location databases linking hardware IDs to street addresses publicly available on the Internet. If the hardware ID is known it is possible to determine the physical address of the device, a major privacy concern.

This is how it works: Wi-Fi-enabled devices, including PCs, iPhones, iPads, and Android phones, transmit a unique hardware identifier, called a MAC address, to anyone within a radius of approximately 100 to 200 feet. If someone captures or already knows that unique address, Google services can reveal a previous location where that device was located, a practice that can reveal personal information including home or work addresses or even the addresses of restaurants frequented.

This tracking was highlighted in the case between Über and Apple, where Über defended the tracking by saying “that the tracking is a common industry practice used to prevent fraud and account compromise.” (Conger, 2017). Über used this method for fraud prevention (especially in China) where drivers would register multiple accounts (and thereby rides) to receive additional bonuses.

Über continued to track iPhones even when the app had been deleted on the device. In 2015 they were forced to comply with Apple policy and the fingerprinting was removed.⁹ However, in 2016 an app update re-introduced

⁹ There is no data that Google held a similar intervention with Über.

the fingerprinting and Uber defended the tracking by saying that they only track users five minutes before and after a ride to obtain accurate pick-up points and safe exit afterwards.

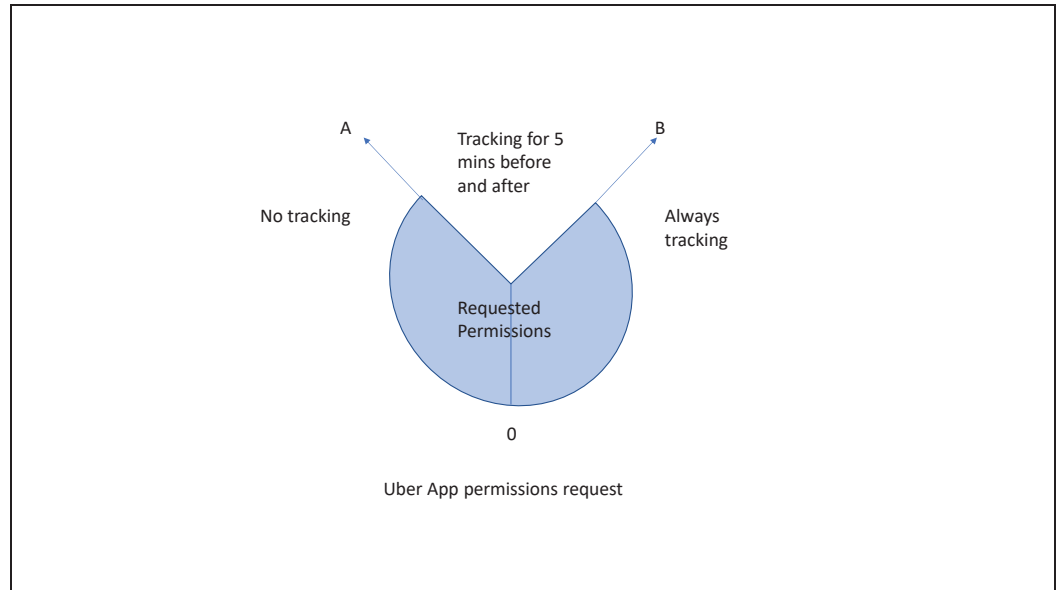


Figure 9-3 Graphical representation of Uber permission requests.

This statement needs to be validated as Uber needs to know the location of the user to provide them with a graphic showing the nearest Uber drivers. Who may be more than 5 minutes away.

A graphical representation of the privacy infringement can be shown in the Privacy Impact Framework Model in Figure 9-3.

The number of permissions required by the app to start and validate user are represented by 0 to A permissions. Once the user requests a ride then the app needs the additional permissions A to B, so total number of permissions required to operate are 0 to B. Additional permissions, B onwards are superfluous and should not be requested. This is where the main privacy infringement occurs. An additional infringement is that the app still records the user location even when not active and in use by the user.

With the ease to track users, advertisers plus parents that want to be able to track their kids, asks the question about how protected is that data from prying eyes. And at what granularity.

What about the expectations of privacy, in 1888 when the first portable Kodak camera was introduced there were many privacy panics and the camera was banned in a variety of places, e.g. businesses, beaches, monuments, etc. However, twenty-two years later this panic had ended with the increasing adoption of the camera. This showed that as technology is accepted and utilised, previous concerns about its misuse diminishes. As more data is collected, is this data copyrighted? Can the owner of the data have it removed from the collector's databases? This is one of the principles of the EU GDPR regulation as described in Section 9.4.

Facebook terms and conditions state that any image uploaded to their servers becomes their property. This ownership has not yet been tested in the courts.

With education apps the data collection has another aspect, where parents or educators view the education app with an implied trust which is not contractual. Certain companies are viewed as trustworthy especially as relating to children, for example, the Disney Corporation, although there is no legal contractual obligation of the company to protect a child using their app. This negates the assumption that the app provider is acting in "loco parentis" of the child as would normally happen with children in schools or colleges.

Some parents use the apps as digital nannies so is the trust implicit or explicit?

9.6 Spyware

Another method to detect usage and location is by installing spyware on the physical device.

9.6.1 Mobistealth

Mobistealth (“Mobistealth,” 2017) is an Android tracking app

There are a million reasons why parents would want to monitor their children’s mobiles as well as companies wanting to monitor company owned devices. Mobistealth is marketed as an All-In-One Android monitoring Software solution. Once installed, the app remains hidden from the mobile user and begins instantly sending information directly to your online user account. The information that is collected and sent are; Real-Time Location of User Even When GPS is not Working (in buildings, etc.), Monitor Skype, Whatsapp and Viber Chat communications, Call Details and Complete SMS Data, Browsing History and Pictures or Videos Available on the Target Phone

As the app remains hidden the device user is unable to access the app to either remove or disable it. Physical access to the device is required for the download and configure.

9.6.2 FlexiSPY’s Android Spy App

FlexiSPY’s app (“FlexiSPY™ Unique Android Spy app – Reveals Secrets Others Cannot,” 2017) monitors messaging, application usage, GPS location

and can perform live listening and recording of phone calls and device surroundings. The app is also able to spy on instant messaging services. Again, physical access to the device is required to install the app. The full functionality of the app is only achieved if the mobile has been rooted.

9.6.3 Android Monitoring App

The Mobile Spy app, ("MobileSpy," 2017) also requires physical access to the device to install the app. Once installed the app monitors and records SMS, Social networking usage (Facebook, Whatsapp etc), YouTube videos that have been watched, what apps have been installed, URLs accessed, GPS locations visited, phone calls (incoming and outgoing) numbers, messages, emails from the primary account, contacts, calendar date and time logged. apps can be blocked, all photos taken are saved and viewable. The update interval is customisable.

All the logged data is available and accessed remotely. There is an optional LIVE Screen Viewer feature, which permits the user to see what is on the mobile phone in real time.

9.7 Big Data

Now that the mobile interacts with so many other devices (IOT) what about the permissions for these apps? A future tool is needed to analyse IOT controls and accesses

Originally smart home devices were controlled in the home using short range devices, these were either using a form of WiFi or Bluetooth. This evolved to each device having a mobile app which could access the device through the consumer's network. Initially within the user's network and then via the router into the network.

Access via the router opens the user network to the outside.

Eventually users will require a single app to be able to control all the smart devices in the home. The devices need to be interconnected and using a common protocol to communicate. This increases the possibility of an external actor accessing the range of devices to "spy" on the user.

Currently there are many smart devices available to watch users, for example, smart TV's, internal CCTVs, web cams, etc. Devices can overhear user's, for example, Alexa, Google Home, Siri, or Cortana, etc.

Once the actor is inside the network the app will have full control of these devices and can monitor the user and their environment.

There is currently no constraint on what the smart devices can do, and most users are unaware of the access or control that they have.

Chapter 10. Research Review

10.1 Overview and discussion

The research initially concentrated on user security, the available tools and how they protected the user. The questions asked were;

- Do anti-virus products work as intended?
- Do anti-virus products protect the user?
- How have anti-virus apps matured?
- Is there additional security for minors?

The Antivirus products were tested for efficacy and the results demonstrated the limitations of the free products. The maturity of the Antivirus apps was inconclusive. Some of the apps had matured and were providing a basic security service to the user. Other apps had disappeared altogether, and other apps had not been updated or improved since their initial addition to the Marketplace.

The security for minor's investigation showed that minors were being protected if the apps were aimed at their age group, but there was no protection if the minor accessed an app aimed at a higher age group and there was no real protection to stop the minor accessing these apps. Most companies circumvented this requirement by adding into their Terms and Conditions that

it was the responsibility of the parent to stop the child. This can be interpreted as “we are in Loco Parentis, but we’re not!”

The app download and installation permission request require an all or nothing approach, i.e. accept the app permissions so that you can download the app and install it or refuse the permissions and therefore are unable to install the app. At the time of the initial research there were few apps available that could review the permissions of the app and there was no basic guidance for the general user to be able to decide which app permissions to turn off, even if it was possible at the time.

How does this relate to the social peer pressure to run apps and play multi-user apps? Both Millennials and Generation Z have grown up with mobile technology and social media and use it as just another tool in daily use. Generation Z have not known anything different and are used to sharing every moment of their life online. This always on-line approach to the minutia of their lives makes them vulnerable to peer pressure to install the current popular app with little or no regard to the permissions being requested or if there is any impact on their privacy. When multi player apps are being used in these online social groups how much data does the developer collect that not only is from each user but their interaction and their relationship to each other. It also means that the developer can consolidate usage/privacy of the player not only from this app but any other apps that the users uses from that developer. The developer has a viewpoint of the user’s activities, apps that they like (from usage stats) as well as do they play them with their friends (multi-play) if available or alone (either multi-play or not) do they only play multi play with their group of friends or do they interact with other groups. Is there even a group dynamic which can be obtained from the usage data of the multi-play groups? Then there is the issue of the ads being targeted to these users. Are they different when the user is playing alone or in a multi group? Are the ads

targeted at specific ages and is this representative across single player apps and multiplayer apps? All these questions need to be investigated and answered in future research.

What about apps that permit you to change the permissions of the app. Does the app work properly after you have switched some of the permissions off? What is the number of permissions that the app requires to work? What about adding permissions, is that feasible?

At the time of the 2015 research, there were apps available to aid the user in the review of the permissions and permit the user to switch permissions off, but again how does the user know which permissions should be deactivated. First the user had to determine what permissions that should be revoked and had to install apps that could review the permissions for each app and then permit the user to revoke the permission or to switch a permission on. Invariably these apps required “root” access, which then made the device vulnerable to external attack. The decision to which permission to switch off or on was left to the user with no guidance of permissions, only a brief tag, e.g. camera or read contacts. Some of these apps had the disclaimer that switching off permissions could prevent the app from working. There was not any indication of whether the app would work without this permission. Some of these apps at least provided an indicator of last used to aid the user.

In October of 2015, Google provided the ability in Android 6.0 (Marshmallow) to control the permissions. See section 12.1.1 on Permission control.

10.2 Differences between 2011 and 2015 Research

10.2.1 Initial Research in 2011

The security app database shows the maturing of an app or its removal and any new apps and developers that are now available in the marketplace. Comparisons have been made of apps that were in existence in 2011 and 2015 albeit with or without updates or improvements and show how the permissions changed over time, good and bad. Thereby, providing a reflection of the maturation of the market and how the market has moved to free apps with in-app payments and how some developers moved to providing online services rather than incorporate them into the app. This enabled the developer to reduce the number of updates to the app as the services were being controlled in the cloud. This introduced additional problems with data security as the cloud services were more detached from the user and the user did not have the control as was the case with requesting permission updates. Often the developer did not have control of the Cloud as this was a purchased service. Many of these suppliers than used rolling payments rather than requesting if the user wanted to renew, an opt-out rather than an opt-in.

Research moved away from practicalities of the extraction to the analysis of the permission use and practice. The extraction and preparation for analysis was automated and documented as the P.E.M.P process. The initial requirement for the automation was the download of the app. In 2011 this was a laborious process as it required the author to download the app onto a mobile device and then transfer the app to a PC for analysis. Once on a PC the app had to be

decoded and dis-assembled so that it was in a readable format. At the time, the process from executable app to readable format had to be performed serially one app at a time. The longest part of the process was the download and transfer. Download speeds were slow as the cellular network was immature, and the network was 2G (2nd Generation also known as Edge). The mobile device had to be disconnected from the PC to perform the download and then re-connected to the PC for the transfer process. This involved connecting the device in debug mode to perform the transfer. Decoding and disassembly were quicker but still had to be performed one at a time. Initially this manual process took 1 hour per app, but by repetition the process (still manual) was performed in 30 minutes. The permission list was then manually added to an Excel database. This also involved manual intervention to sort the permission list ready for analysis.

The research initially was to investigate the security apps available to protect users and their devices and the efficacy of these apps. All security apps that had an antivirus component or keyword tag in 2011 were downloaded. At the time there were no freely available tools to perform this download and de-compile, so the initial apps were downloaded to a T-mobile G1 smartphone, transferred to a PC for the analysis. The download, transference and extraction of the app's manifest file and the multiple de-compiles (Dex [package on mobile] to Jar [compiled java code] to source code [java]) and the extraction of the manifest file was taking approximately 30 minutes for each app. The length of time taken to perform this meant that it was prohibitive to download and analyse large numbers of apps. An automated process was required to extract the manifest file from multiple apps and the download and extraction was reduced to less than 5 minutes per app. This method has been further augmented and comparison steps have been added and automated.

The antivirus component of the app was tested on the smartphone using available test viruses. The results of the free and commercial versions were compared, and the efficacy of the apps recorded. The conclusion being that there was no difference between the free and commercial versions in detecting and quarantining the virus.

There was a large difference between the apps functioning, with some apps able to detect both viruses, during scanning and downloading, whilst one app didn't detect either virus during download, installation or scanning.

Once the app was extracted and decompiled, a rudimentary analysis was made of the *Android. Manifest* file that contained the actual permissions defined by the app. These permissions required acceptance by the user before download and installation was permissible. In some cases, the permissions requested were not the same as those that were described on the marketplace site.

The free security apps were compared to their commercial variants to determine if there was any benefit to the user to purchase the product. The differences in their permission requests and features were recorded and analysed to determine if the commercial versions provided the user with more features or better protection. Part of the analysis was to record the sizes of each of the packages to determine if additional code was used in the commercial version to differentiate it from the free version. Hashes were performed on the app source codes versions that had the same size to determine if there were any actual differences between the free and commercial versions. The hashes were identical, indicating that there were no source code differences, therefore, the main differences were probably related to online services.

Reviewing the feature sets of the apps, indicated that the additional features of the app (used as a differentiator on the marketplace) were in fact online and not included in the app itself. One of the apps had the same size and hash of

the free and commercial variants. Of the six suppliers in the analysis, three (Lookout Inc, AVG Mobilation and BluePoint Security Inc) used the same Android permissions on both the commercial and free applications. Two suppliers (Lookout Inc. and AVG) requested non-Android permissions, whilst the other suppliers only requested Android permissions. Of the non-Android permissions, Lookout Inc. used the same permissions on both products, whilst AVG performed License checking and used different C2D_MESSAGE permissions between its PRO and Free versions.

10.2.2 Subsequent Research in 2015

In 2015 the same analysis was performed using the method from 2011. New tools were available to perform the download directly to a PC, namely the APK Downloader tool ("APK Downloader," 2014). The author developed analysis code using Python software ("Python Downloads," 2015). This script extracted, dis-assembled and decoded the app and recorded the permission list into the database in the correct format without any user intervention. Once all the requisite apps had been downloaded to the PC, they could be processed in one batch rather than serially. The automation reduced the download, extraction and decode process by over 80% (from 30 minutes to 5 minutes). The permission list database was then available for analysis. The refining and automation of the method was named P.E.M.P. (see Chapter 7).

The PEMP process was tested against the initial (2011) set of apps as well as the 2015 set. The results from the manual and the PEMP process was identical, confirming the robustness of the process. The process was also used to extract

and analyse children apps¹⁰. The code has been used to prepare various set sizes with very little increase in processing time.

The process was also used by another researcher for preparing First Person Shooter (FPS) games for analysis. This showed that the process could be used across genres.

As in 2011 the available Antivirus products were tested. During the intervening 4 years the antivirus apps had matured and many of the industry security market leaders had entered the mobile marketplace. Some of the original developers and/or their products had been bought and integrated into the market leader's portfolio of products, this meant that these companies had a security presence across all platforms.

As in 2011 the apps were available in two variants, free and commercial. With the commercial variants either charging a one-off or monthly payment. In some cases, the commercial apps offered additional functionality.

In 2011 there were 22 apps with Antivirus components, this had grown to 67 apps available in 2015. The main PC Antivirus testing company (AV-Test.org) had also matured its testing of antivirus products on mobiles. Testing had increased from 4 apps in 2010 to 16 in 2015. By which time this research had already tested all 67 antivirus apps. Of the 15 developers (22 apps) on the Google Store in 2011, 5 developers were still in existence in 2015. The 22 apps that they had available in 2011 had reduced to 7 which had been updated during the 4 years to 2015. In 2015 the number of developers had increased to

¹⁰ The process has also been used by another researcher for preparing First Person Shooter (FPS) games for analysis.

57 and the number of products available to 67. The market was maturing as the number of developers had increased fourfold, but the number of apps had only increased by a factor of three. The author's conclusion was that developers were concentrating on a main app rather than providing multiple variations and names.

The same tests and analysis were performed as in 2011. First the antivirus component and then the permissions and features.

An additional analysis was to compare the permissions and features of antivirus apps in 2015 with their predecessors in 2011. There were 7 apps from 2011 that were still in existence in 2015.

The number of permissions had also changed but had not increased across the board as expected with the increase in permissions available. In 2011 the median number of permissions requested was 15, the maximum requested was 82 and the minimum requested was 3. In 2015 the median had increased to 21, but the maximum requested had dropped to 49. This indicated that developers were either being more selective about the permissions to perform the function or were using the higher-level permission, which would cover multiple permissions, rather than select individual permissions. Three of the apps did not request any permissions at all, which does question the efficacy of the app. Removing these outliers showed the minimum that was requested was 4.

To test future large numbers of apps the generic method PEMP in Chapter 7 was created to minimize download and extraction times via automation so that research time was spent on analysis of the results rather than obtaining data. Testing of the generic method was performed by another researcher to download, extract and test First person shooter games (FPS).

This generic method was then used to extract and test 60 children's apps. The apps were chosen for their popularity and the top 20 were selected from the 3

age groups (Ages 5 & Under, Ages 6-8 and Ages 9 & Over). The permission frequency for these apps and the requested permissions in each age category was recorded. The privacy quotient was then created for each app. The privacy quotient is determined by recording how many anti-privacy permissions are requested as compared to the number that are available and whether these are rated as high, medium or low. These quotients are graded as High (too many requested and not required by the app to perform its function), Medium (too many requested but required for the app to function), Low (acceptable number requested for the app to perform its function). apps with a rating of {High, High} are to be avoided as they totally contravene the child's privacy.

One observation was that the large number of requested permissions that apps were requesting, that contravened the user's privacy, but did not add to the app's functionality, is not being controlled or regulated.

10.3 Guidance for Regulators

This observation prompted the question "What are the regulators doing to protect users and what do they need to have to be able to review apps developers as well as the marketplaces that sell apps?"

The regulator cannot operate at such a detailed level as the apps themselves but would have to regulate at a higher level. The optimal way to do this would be to regulate at marketplace level and encourage or enforce the marketplace companies to regulate the developers.

The regulatory control is depicted in Figure 10-1. This shows the flow of standards or government requirements of marketplace providers and how they should be managing the developers.

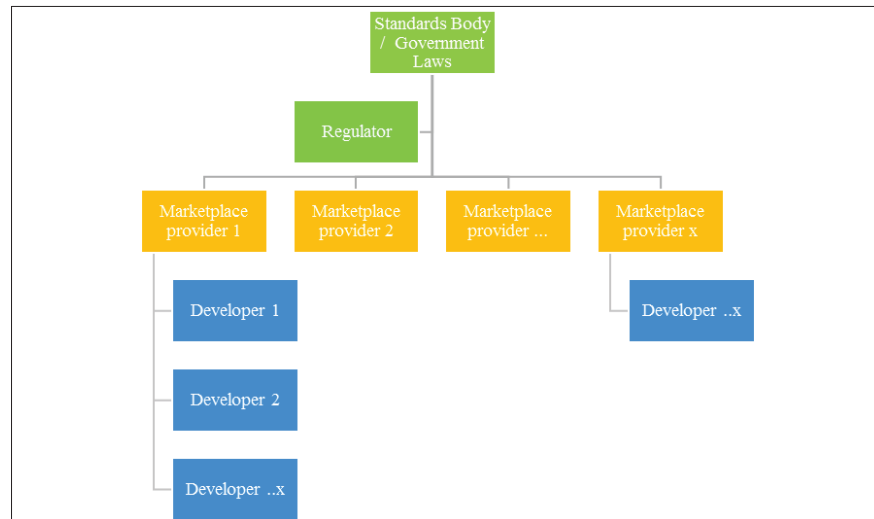


Figure 10-1 A summary of regulatory control

The regulatory bodies are in situ to interpret laws for the providers. However, the number of providers/developers and their global locations prove difficult to regulate as many of the regulator's requirements are not consistent. To simplify, I propose that the Marketplace provider should be regulated and the responsibility to regulate the developer should be with the Marketplace provider.

Previously this was difficult to enforce as most of the Marketplace providers were based in different countries, although they had subsidiaries across the globe. The implementation of GDPR by May of 2018 will be fortuitous as any company doing business with the EU or countries signed up to GDPR must adhere to its requirements (9.4 GDPR - EU Privacy Regulation). This will provide the in-country regulator with the ability to enforce the privacy requirements or fine the perpetrator.

Chapter 11. Contribution

The contribution of the study is twofold. Firstly, the development of PEMP to provide a solution to a generic problem extracting and processing Android Permissions and an application method for the extraction and process. Included is a method, developed to enable researchers to download commercial apps for testing at no cost.

The research makes use of a unique historical dataset containing security apps from 2011 to 2015. This database provides a research opportunity to be able to compare security apps over the 4 years for analysis. The database provides data on the evolution of the security apps in the marketplace, the emergence of new developers and new apps and the perceived requirements for the user's security.

Secondly the research crosses the boundary between technology and psychology (mainly assessing mobile apps as they relate to social and psychological contracts). This has resulted in the provision of a Privacy Impact Model which provides a method of analysing requests to Android smartphone users and determining which requests are beneficial to the user and which are detrimental. Emphasis is placed on protecting the user's privacy and alerting the user if the permission requests made by an application will adversely affect the user's privacy. The Privacy Impact Framework model was created to provide a simple visual output to illustrate the privacy impact of the app at a

glance. The method for determining the impact to the user is based on psychological and social contract theory.

A summary of the research and results follow showing how the creation and implementation of the Privacy Impact Framework Model has aided and enhanced the representation of the resultant conclusions.

11.1 Privacy Impact Framework Model Evolution

The initial idea of a fuel gauge was conceived to illustrate the permission results of the antivirus apps in 2011 and to enable a quick view of the relevance of the permissions requested and if they include all the permissions to perform the antivirus function (efficacy).

11.1.1 The Antivirus Efficacy Gauge

The first designs were very crude and attempted to show the permissions, using a Goldilocks method, too few, too many and just right (Figure 11-1). This model was used to create the initial antivirus framework model (Figure 11-2).

Contribution

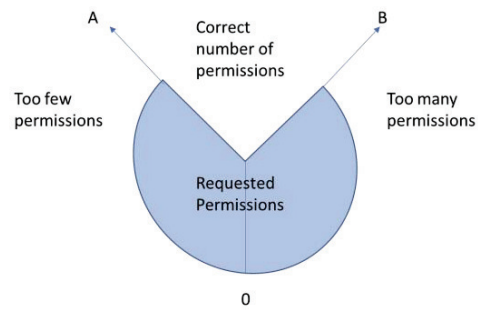


Figure 11-1 Initial Design using the Goldilocks Method

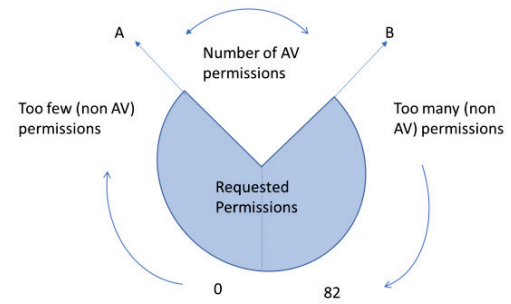


Figure 11-2 Framework using antivirus as a base.

Creation of the diagram, although simple, would show the state of an antivirus app but it was difficult to provide comparisons between apps. The main problem was that some of the apps did not request enough permissions to include the antivirus permissions and those that did, did not request all available permissions (maximum requested was 27). This meant that the diagram would not be suitable to illustrate comparisons between apps.

An additional issue was in defining what was too many or too few permissions.

The gauge needed to evolve to provide the status at a glance. This was necessary once the 2015 apps were analysed and compared to their earlier 2011 apps. Attempts were made to show this as a pie chart with exploding slices to emphasise the antivirus portions of the permissions (Figure 11-3).

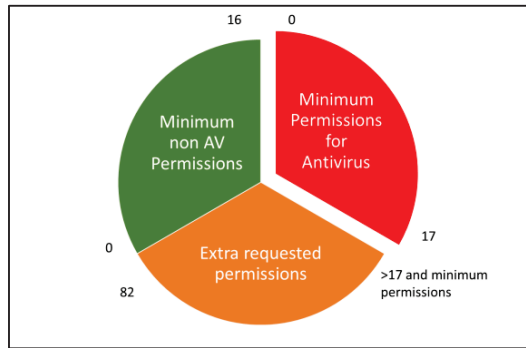


Figure 11-3 Initial antivirus framework as a pie chart with exploding slice

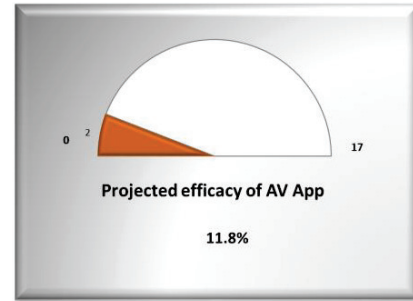
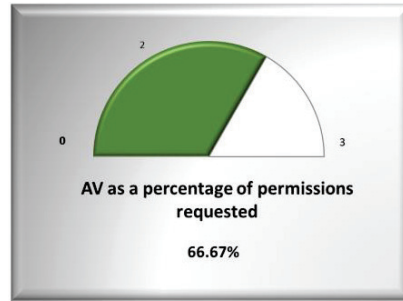
However, the gauge was more responsive to the total permissions requested rather than the antivirus permissions requested. When minimum and antivirus permissions requested were very small the extra requested permission overwhelmed the chart.

The current model provides an overview of the app antivirus function, designated antivirus permissions requested (efficacy) and the antivirus permissions as a percentage of the whole request.

The framework model was applied to the 2011 antivirus apps comparing the free with their commercial variants. These developers and their apps that have permission request differences are in the following diagrams, Figure 11-4, Figure 11-5 and Figure 11-6.

Aegislab

Aegislab
Free



Aegislab

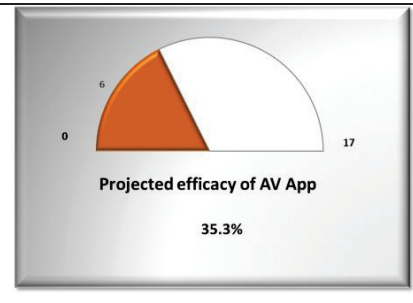
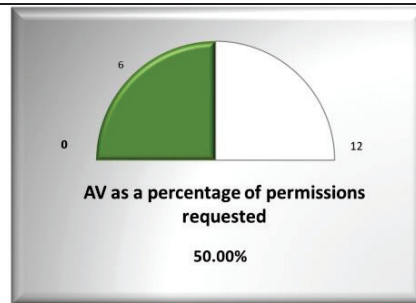
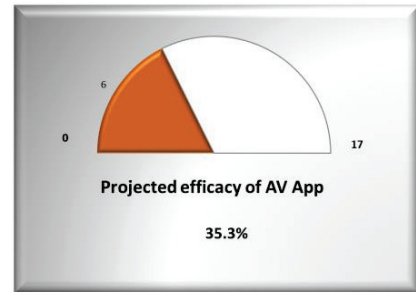
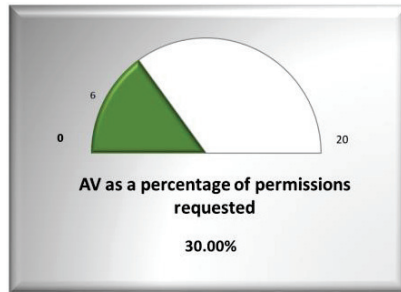


Figure 11-4 Antivirus framework gauge for Aegislab apps

The framework gauges show a clear improvement of the projected efficacy of the commercial app, despite the app requesting apparently less permissions. However, further analysis confirms that there was an increase in antivirus permissions and total permissions requested which is reflected in the efficacy improvement.

Bluepoint

Bluepoint
free



Bluepoint

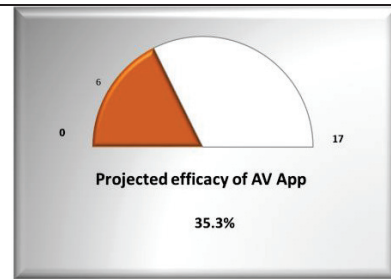
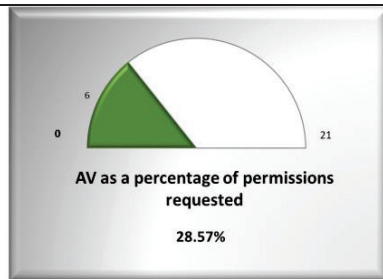
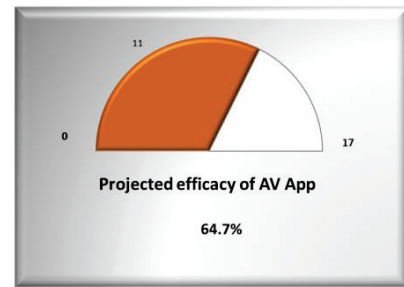
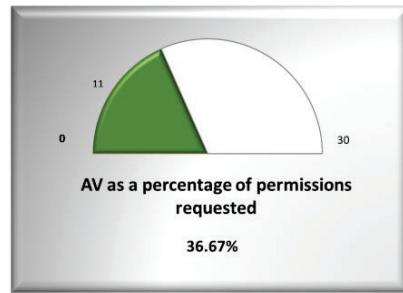


Figure 11-5 Results of Bluepoint antivirus apps comparison

In this case the efficacy does not change, although the percentage of antivirus permissions reduces. Further analysis shows that an additional permission was requested by the commercial app, but the number of antivirus permissions did not change. The increase of permissions by 1 was not significant to affect the efficacy.

Lookout

Lookout
Free



Lookout

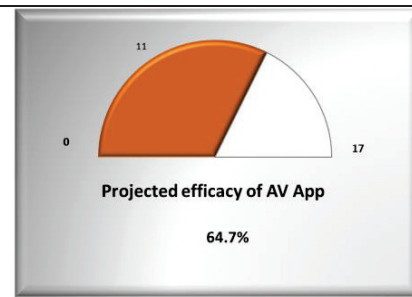
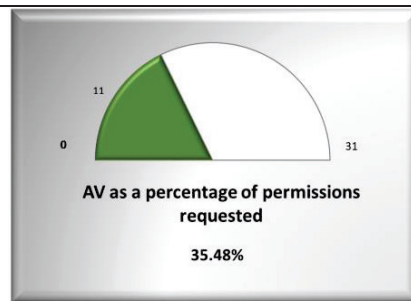


Figure 11-6 Lookout antivirus apps comparison

In this case, the Lookout commercial app is requesting more permissions, but the efficacy remains the same, showing that there is no difference in the number of antivirus permissions requested.

This demonstrates that the projected efficacy framework gauge is useful as a comparison tool in determining the efficiency of the antivirus app, irrespective of the number of non-Antivirus permissions requested.

The model was then used to compare the 2011 antivirus apps with their updated 2015 antivirus variants. This is used to show instantly if the app efficacy improves over the 4 years (Table 11-1). Six of the eight apps, available in an updated version in 2015, had improved effectiveness. The only two apps showing no improvement were both by Bluepoint Inc. Their apps contained

Table 11-1 Antivirus Products Efficacy in 2011 and 2015

Company	Product	2011 version	2015 version
AV Free	AV antivirus free trial	<p>Projected efficacy of AV App 41.2%</p>	<p>Projected efficacy of AV App 76.5%</p>
AVG Mobilatio n	AntiVirus Free AVG	<p>Projected efficacy of AV App 41.2%</p>	<p>Projected efficacy of AV App 76.5%</p>
Bluepoint Security	BluePoin t Antivirus	<p>Projected efficacy of AV App 35.3%</p>	<p>Projected efficacy of AV App 35.3%</p>
Bluepoint Security	BluePoin t Antivirus free	<p>Projected efficacy of AV App 35.3%</p>	<p>Projected efficacy of AV App 35.3%</p>

Contribution

Company	Product	2011 version	2015 version
Dr Web Ltd	Dr.Web Antivirus light	<p>Projected efficacy of AV App 23.5%</p>	<p>Projected efficacy of AV App 35.3%</p>
Dr Web Ltd	Dr.Web Android light	<p>Projected efficacy of AV App 23.5%</p>	<p>Projected efficacy of AV App 76.5%</p>
Lookout Inc	Lookout Mobile Security	<p>Projected efficacy of AV App 64.7%</p>	<p>Projected efficacy of AV App 76.5%</p>
NetQin Mobile Inc.	Nq mobile manager Trial	<p>Projected efficacy of AV App 47.1%</p>	<p>Projected efficacy of AV App 76.5%</p>

the same number of AV_Permis although the number of requested permissions had grown.

In all cases the number of Antivirus permissions recorded in the Manifest file had increased.

11.1.2 Privacy Impact Analysis

In 2012 the initial procedures and guidelines for GDPR was proposed. The draft was released on the 25th January 2012 and reviewed by various Law Groups (Law Patent Group, 2012). In 2016 the implementation date was agreed to be 25th May 2018. Therefore, the model was updated to incorporate these future privacy guidelines and re named as the Privacy Impact Framework Model. The objective of this new model was to incorporate the privacy impacts of antivirus apps in relation to the proposed guidelines.

Existing research has concentrated on the physical or software actions of apps, namely; the tracking performed by the mobile device, the API calls of the apps, the malware that has been introduced into apps or onto mobiles and the security of the mobiles. Little research has been performed on the protection of a user's privacy, other than GPS location tracking.

Privacy and security are affected by new technology and are not necessarily considered during the development process as their addition often impedes the "first-to-market" requirements of stakeholders.

The rapid development of technology prompted many questions.

- What apps are in existence to review the permissions on the apps? The apps request an all or nothing approach, i.e. accept the app permissions so the user can download the app and install it or refuse the permissions and are not permitted to download and install the app.

- How does this relate to the social peer pressure to run apps and play multi user apps? This clearly applies pressure to the user to accept the permissions.
- What happens then with multi app players?
- How much data does the developer collect that not only is from each user but their interaction and their relationship to each other?

All of this is a concern for Security and Privacy specialists.

A tool to measure the privacy impact of an app was required that displayed the influencers on the permissions requested during the lifetime of the app. Initially the permissions were requested on a best guess basis and future requests are determined by a variety of influencers. These influences are; variants (changes) to the code (fixes or improvements - e.g. more levels), technological advances, commercial differentiators, human, regulatory and competition.

A method to check the privacy status of an app grew from the initial research extracting, analysing and assessing app's permissions to reviewing the output in relation to the impact on the user's privacy. This method evolved into the Privacy Impact Framework Model. The framework produces an overview of the current permission status as related to the app and their impact on the user's privacy. The format used is that of a fuel gauge and shows if the permissions requested are privacy related and if too many or too few permissions are requested for the app to function as described¹¹.

¹¹ There are no groups or sub-groups to describe the privacy impacts within the 9 Android permission groups.

With the emphasis now on the privacy aspects of Antivirus apps and previous research showed that the apps were not effective and did not request the appropriate permissions to perform their function. If this theme continues then what confidence is there that the same consideration is used to safeguard a user's privacy. Therefore, the next step was to use the model from the Efficacy Gauge to create an Antivirus Privacy Impact Framework Model.

11.1.3 Antivirus Privacy Impact

The Antivirus Privacy Impact Model was created to analyse and present the privacy status of Security and Antivirus apps on the Google Store, but can be adapted to other genres and marketplaces.

The model was created and used initially to analyse and compare the Antivirus apps' permissions from 2011 and 2015 apps. During 2015 Google introduced "protection normal", see 9.1.3, a permission set which did not require the user's acceptance and was included by default to any app's permission set.

The model therefore had to be adapted so that it could be used on its own or combined with the Base-line privacy impact identification to provide an overall picture of the privacy impact. This would be for apps created or updated after 2015.

During the research in 2011 and 2015, there were 17 Android permissions identified as being necessary to perform the Antivirus function of Security apps, of which 6 impacted the user's privacy. Table 11-2 contains the required permissions and evaluate them into low or no impact ('L'), medium or some

impact ('M') or high impact ('H') and the reason for the medium and high ratings. The low or no impact permission activities are read or view activities.

Table 11-2 Permissions required for Antivirus function

Antivirus permissions	Privacy impacts	Impact Activity
ACCESS_NETWORK_STATE	L	
CALL_PHONE	H	Phone can be used to dial premium numbers without user intervention.
CHANGE_NETWORK_STATE	H	User's network connectivity can be changed
CLEAR_APP_CACHE	L	
DELETE_PACKAGES	M	Able to remove installed apps (packages)
GET_ACCOUNTS	M	All user's account details can be read.
GET_TASKS	L	
INTERNET	M	Switches Internet access off or on.
KILL_BACKGROUND_PROCESSES	L	
MANAGE_ACCOUNTS	H	Update access to user's accounts
READ_CONTACTS	H	Access to all user's contacts on the device.
READ_EXTERNAL_STORAGE	L	
RECEIVE_MMS	M	Reads MMS messages
RECEIVE_SMS	M	Reads SMS text messages
WRITE_CALENDAR	L	
WRITE_CONTACTS	H	Add contacts to user's list. This can be used to circumvent caller blocking
WRITE_EXTERNAL_STORAGE	H	Add or update user information on storage cards

Earlier research of the 2011 Antivirus apps (Chapter 6) established that none of the tested apps requested all 17 permissions necessary to perform the function, with the maximum requested being 6, and 10 of the apps (45% of the 22 analysed) did not request any permissions with a privacy impact. Whereas, in 2015, the maximum number of Antivirus permissions was 10 and 25 of the 67 apps (37%) did not request permissions that had been designated as having a privacy impact. This showed a slight improvement from 2011.

11.1.4 Protection_Normal Privacy Impact

In 2015, Google had introduced the concept of Protection Normal permissions (see 9.1.3). Therefore, a new model was required to evaluate the later (2015) Antivirus apps which incorporated the privacy impact of the base protection normal permissions with the Antivirus privacy impact permissions.

Before being able to analyse apps to determine the privacy impact, a baseline impact was required. Initially an analysis of the Protection_normal permissions was performed to identify the permissions which had an impact to the user's privacy and to determine the Base Privacy Impact (Base_PI). The results of the analysis displayed in Table 11-3.

Table 11-3 Base-line permissions and their privacy rating

Permission	Privacy Impact	Impact Activity
ACCESS_LOCATION_EXTRA_COMMANDS	L	Tracking
ACCESS_NETWORK_STATE	L	
ACCESS_NOTIFICATION_POLICY	L	
ACCESS_WIFI_STATE	L	Obtain WiFi status
BLUETOOTH	M	Control access between device and other Bluetooth devices
BLUETOOTH_ADMIN	M	Control Bluetooth admin, like pairing names and codes
BROADCAST_STICKY	L	
CHANGE_NETWORK_STATE	M	Control access and possibly make device accessible
CHANGE_WIFI_MULTICAST_STATE	M	Broadcast device name to WiFi networks
CHANGE_WIFI_STATE	M	Switch WiFi on/off
DISABLE_KEYGUARD	L	
EXPAND_STATUS_BAR	L	
GET_PACKAGE_SIZE	L	
INSTALL_SHORTCUT	L	
INTERNET	M	Internet access activate or disable
KILL_BACKGROUND_PROCESSES	L	
MODIFY_AUDIO_SETTINGS	L	
NFC	M	Control transfer of data including payment details and make payments
READ_SYNC_SETTINGS	L	
READ_SYNC_STATS	L	
RECEIVE_BOOT_COMPLETED	L	
REORDER_TASKS	M	Change task priorities
REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	L	
REQUEST_INSTALL_PACKAGES	L	
SET_ALARM	L	
SET_TIME_ZONE	L	
SET_WALLPAPER	L	
SET_WALLPAPER_HINTS	L	
TRANSMIT_IR	L	
UNINSTALL_SHORTCUT	L	
USE_FINGERPRINT	L	
VIBRATE	L	
WAKE_LOCK	L	
WRITE_SYNC_SETTINGS	L	

Contribution

There are 34 base-line permissions of which 8 permissions are marked as an impact to privacy and are rated 'M' (medium), these are referenced and labelled as the Base_PI permissions. The other permissions either review, obtain or control status of the operating system for the app with one providing extra tracking commands, these are labelled as Base permissions and are rated as 'L' (low or little impact).

The author decided to illustrate the privacy impact of an app in red to demonstrate that the user needs to heed the warning that their privacy is being abused.

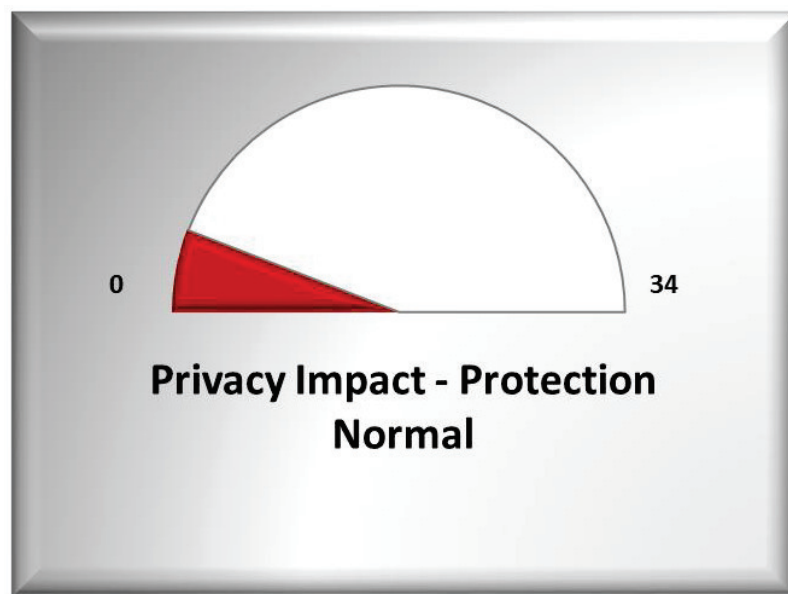


Figure 11-7 Base-line Privacy Impact Status using the Privacy Impact Framework Model

The gauge illustrates that the Base_PI (base-line privacy impact) permissions is small, less than 5.2% of the total available permissions within the protection_normal group. The impact level will increase as additional

permissions are requested by an installed app. The user needs to decide if this is an acceptable level of impact.

11.2 Privacy Impact Framework Model

With the addition of the Protection normal or Baseline privacy impact, the privacy impact had to evolve to incorporate the impact of newer apps as well as to display the impact of earlier apps.

A matrix was created to determine the level of privacy impact depending on the complexity and number of Privacy Impacted permissions (PI_perms).

The protection normal Base_PI_perms were then combined with the Antivirus AV_perms and PI_perms list resulting in 27 minimum permissions for an Antivirus app of which 14 had Privacy Impacts (Table 11-4).

Table 11-4 Resultant list of permissions to perform basic Antivirus function

<i>Permission</i>	<i>Activity</i>	<i>Rating</i>	<i>Base_PI_perm</i>
ACCESS_NETWORK_STATE	AV	Low	Y
BLUETOOTH	Control access	Medium	
BLUETOOTH_ADMIN	Control access	Medium	
CALL_PHONE	PI	Medium	
CHANGE_NETWORK_STATE	AV	Medium	Y
CHANGE_WIFI_MULTICAST_STATE	Control access	Medium	
CHANGE_WIFI_STATE	Control access	Medium	
CLEAR_APP_CACHE	AV	Low	
DELETE_PACKAGES	AV	Low	
DISABLE_KEYGUARD	Change status	Low	
EXPAND_STATUS_BAR	Change status	Low	
GET_ACCOUNTS	PI	Medium	
GET_PACKAGE_SIZE	Obtain status	Low	
GET_TASKS	AV	Low	
INSTALL_SHORTCUT	Change status	Low	
INTERNET	AV	Medium	Y
KILL_BACKGROUND_PROCESSES	AV	Low	Y
MANAGE_ACCOUNTS	PI	Medium	
NFC	Control access including payment details	Medium	
READ_CONTACTS	PI	Medium	
READ_EXTERNAL_STORAGE	AV	Low	
RECEIVE_MMS	AV	Low	
RECEIVE_SMS	AV	Low	
REORDER_TASKS	Change status	Medium	
WRITE_CALENDAR	PI	Medium	
WRITE_CONTACTS	PI	Medium	
WRITE_EXTERNAL_STORAGE	AV	Low	

The base Privacy Impact Framework Model for an Antivirus app with no additional requested permissions is shown in Figure 11-8.

The Privacy Impact Framework Model gauge (PI_gauge) for Antivirus apps was updated to use the defined basic Antivirus permissions available in the 2015 version of Android (27), 14 of these base permissions have a privacy impact. Additional requested PI_perms indicate that there is some risk to the user’s privacy and extra requested permissions relate to additional

functionality. Having this snapshot view of the app enables the user to decide if the additional functionality is worth the additional impact to their private information.

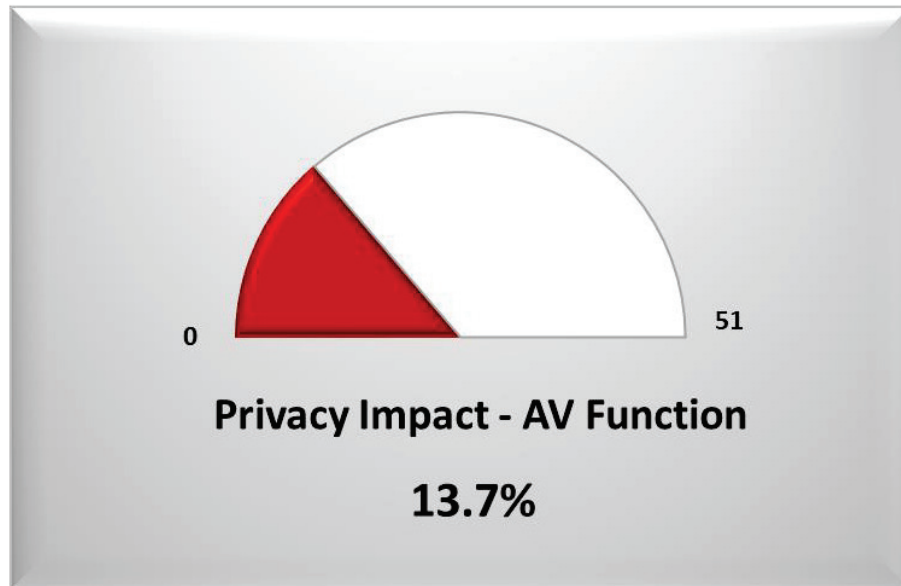


Figure 11-8 Privacy Impact Framework Model for Antivirus Function

The antivirus Base Privacy Impact Framework Model provides the graphical representation of the base permissions required to perform the Antivirus function. In the 2015 version of Android, a minimum of 27 specified AV_perms are required to perform the Antivirus functions of which 4 are already included in the set of Base_PIs. Of these 27 permissions, 14 are defined as Privacy Impact permissions (4 of which are included in the Base-line set of permissions. The app can request more than the specified 27 permissions which are the extra requested permissions but if the Privacy Impact permissions increase then the red portion of the model increases.

Here the Privacy Impact Framework Model was used to display the results of the two Antivirus apps from 2011, which had differences between the free and commercial versions.

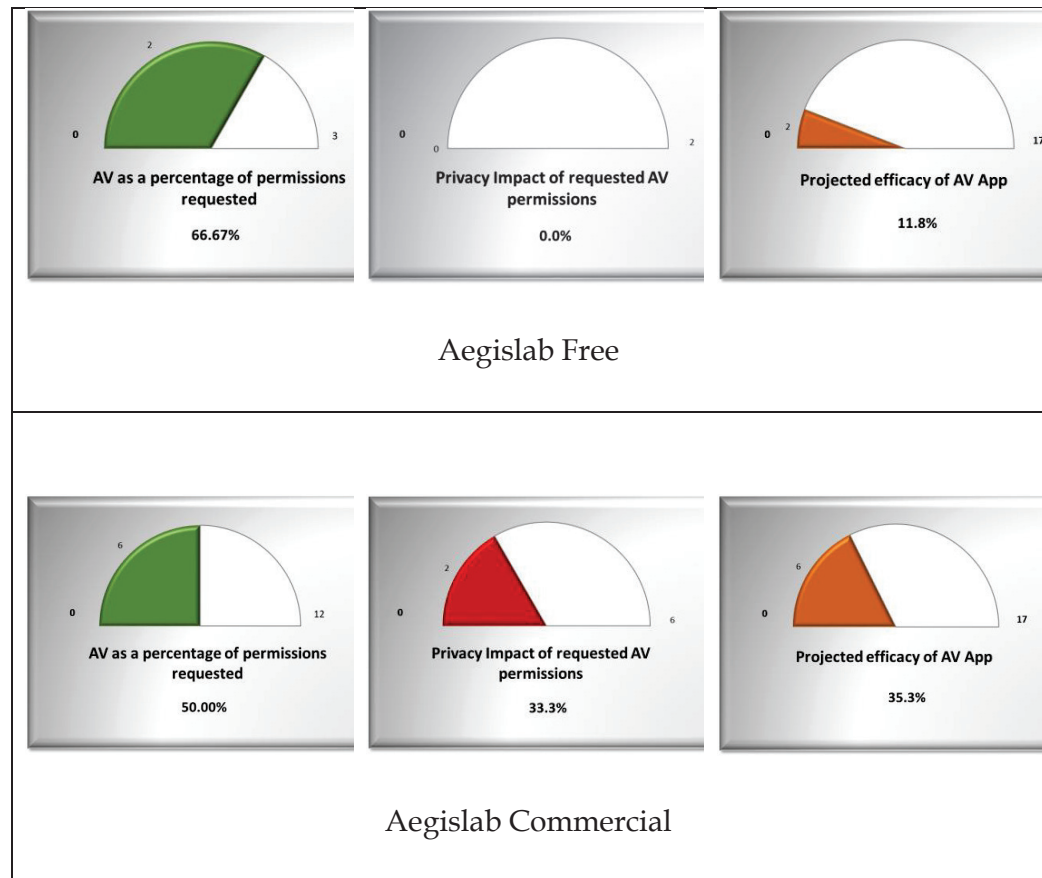


Figure 11-9 Privacy Impact analysis of a free app with its commercial version

The colours chosen for the display are in respect to the effect on the user. These are; green for information, amber for a notification and red to show server impact.

The free app appears to be the better app with regards to the antivirus permissions requested and has a low or zero privacy impact. However, the low efficacy score indicates that there is something not right here and closer inspection shows that the number of permissions requested were abnormally small (3). The commercial app, although having a higher privacy impact is more effective. Here the commercial app requested 12 permissions in total. The comparison also demonstrates that displaying the antivirus permissions as a

percentage is meaningless if the apps are not requesting the same number of total permissions. This gauge has been removed from the model.

11.2.1 Privacy Impact of Children’s apps

The Privacy Impact Framework Model was then used to evaluate the privacy around children’s apps.

In 2015 the number of available permissions had grown to 169. Of which 26 were designated as Privacy Impacted.

Table 11-5 The main 11 Privacy Impact permissions

<i>Android Permission</i>	<i>Definition</i>
<i>ACCESS_COARSE_LOCATION</i>	Allows an app to access approximate location.
<i>ACCESS_FINE_LOCATION</i>	Allows an app to access precise location
<i>CAMERA</i>	Required to be able to access the camera device
<i>CAPTURE_AUDIO_OUTPUT</i>	Allows an application to capture audio output
<i>CAPTURE_SECURE_VIDEO_OUTPUT</i>	Allows an application to capture secure video output
<i>CAPTURE_VIDEO_OUTPUT</i>	Allows an application to capture video output
<i>READ_SMS</i>	Allows an application to read SMS messages
<i>READ_VOICEMAIL</i>	Allows an application to read voicemails in the system
<i>RECEIVE_MMS</i>	Allows an application to monitor incoming MMS messages.
<i>RECEIVE_SMS</i>	Allows an application to receive SMS messages
<i>RECORD_AUDIO</i>	Allows an application to record audio

The main 11 privacy impacting permissions provide an app the ability to track, eavesdrop and spy on the user.

The top 20 children’s apps in each age group were evaluated to determine if children’s privacy was being impacted.

Contribution

In the 0-5 age group, 5 apps requested at least 1 of the permissions from Table 11-5, these were Barbie magical fashion, BBC Cbeebies storytime, Cbeebies Playtime, Peppa's activity maker and Disney color and play.

The Privacy Impact Framework Model was used to evaluate the impact for these 5 apps and the results are in Figure 11-10.

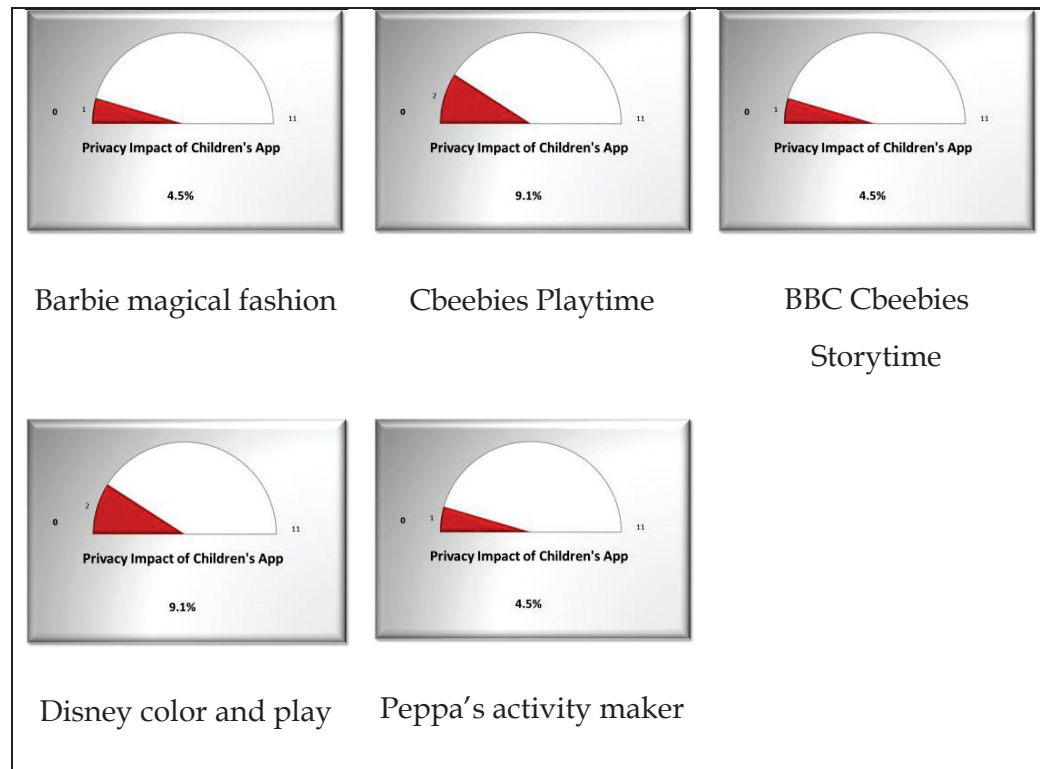


Figure 11-10 Childrens apps 0-5 age group privacy impact

The Privacy Impact Framework Model shows a minimal impact less than 10% impact to the children for these apps.

The Privacy Impact Framework Model was used to evaluate the other two age groups, 6-8 years (Figure 11-11) and the 9+ age group (Figure 11-12).

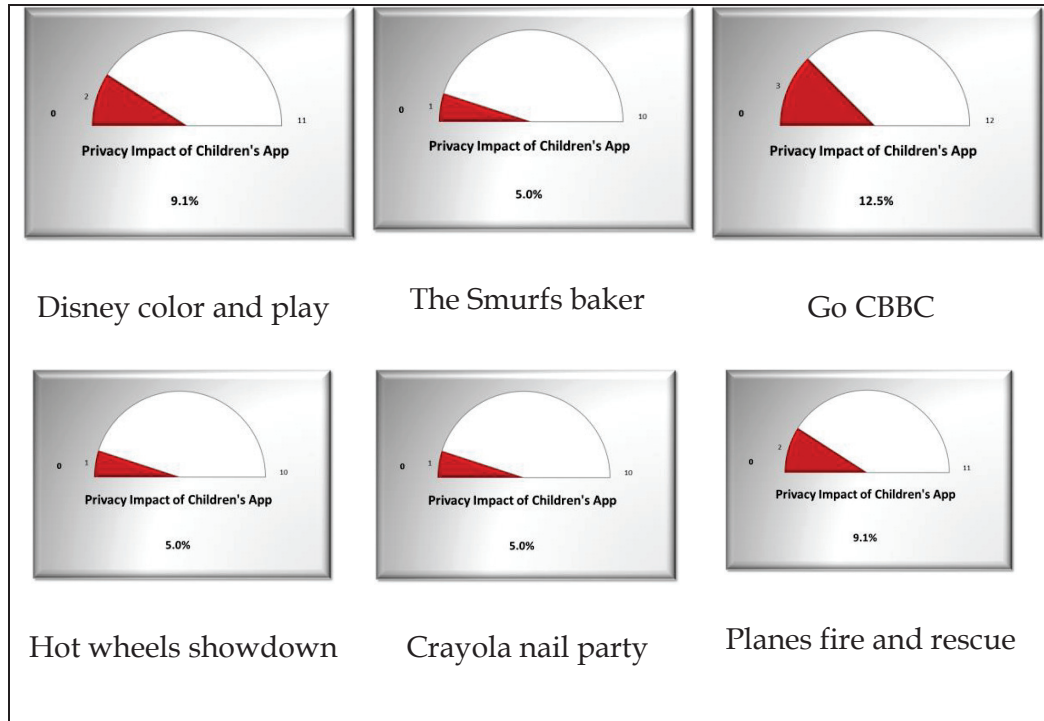


Figure 11-11 Childrens apps 6-8 age group privacy impact

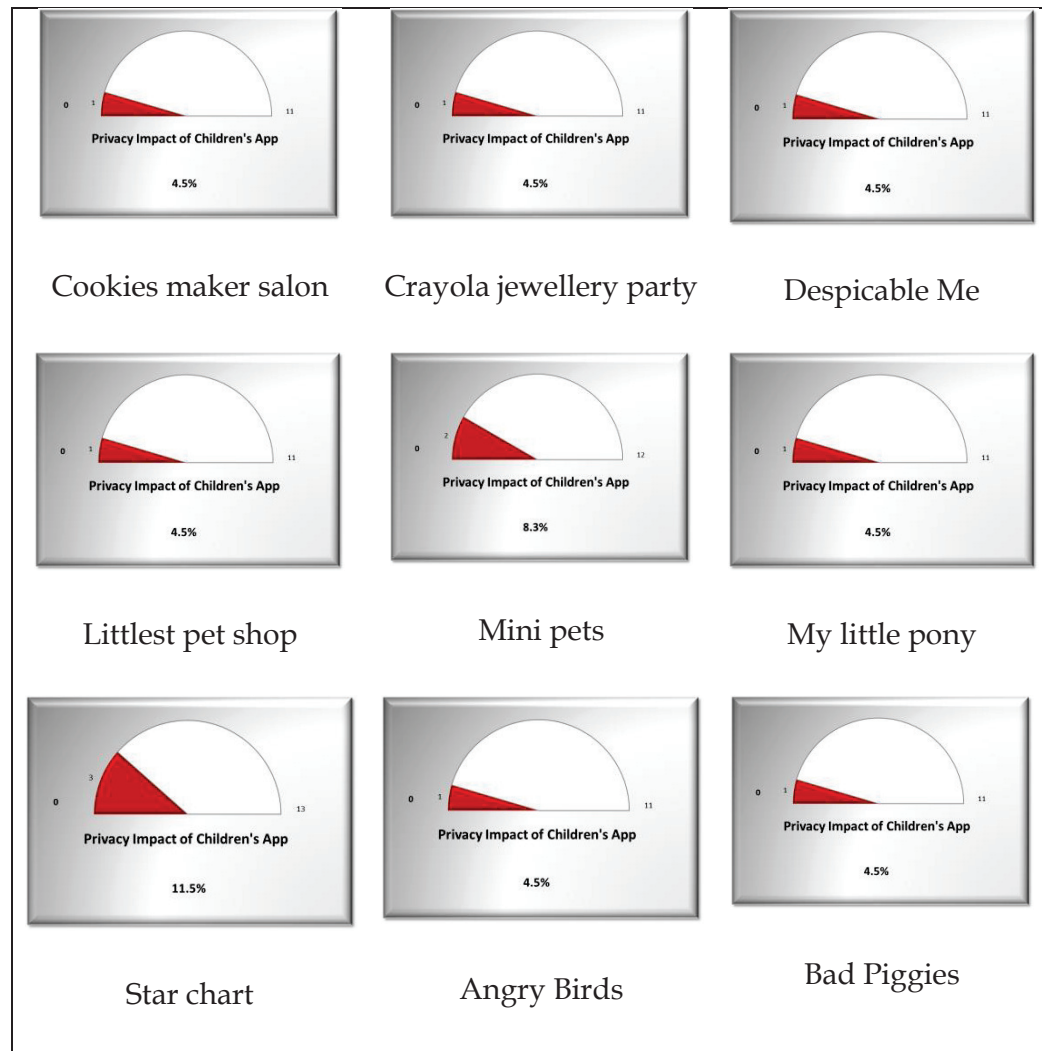


Figure 11-12 Childrens apps 9+ age group privacy impact

The Privacy Impact Framework Model simplifies the analysis output of the privacy impact. The models confirm the previous manual analysis that the children's privacy impact is minimal.

11.2.2 Antivirus app Privacy Impact 2011 vs 2015

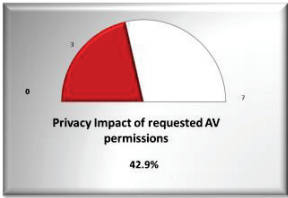
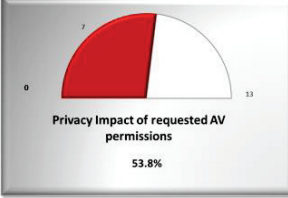
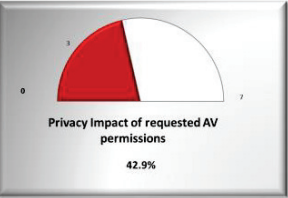
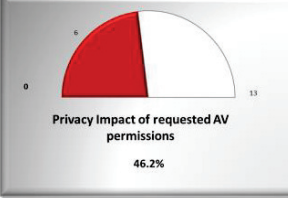
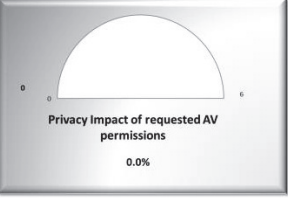
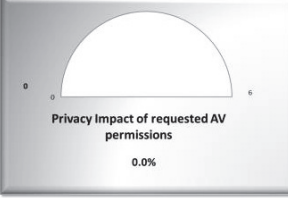
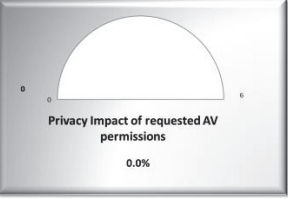
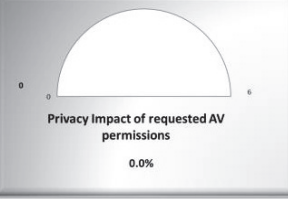
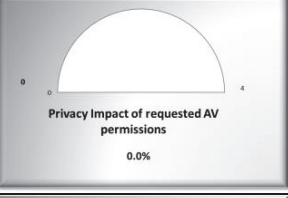
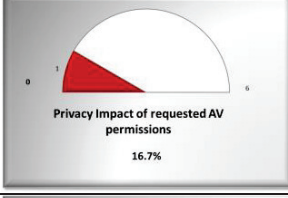
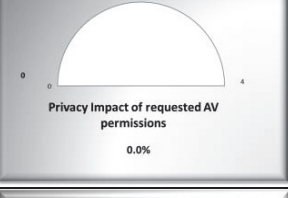
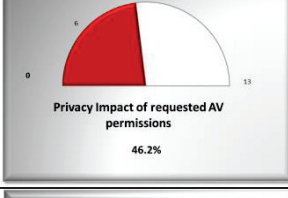
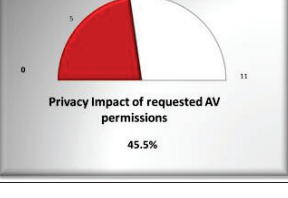
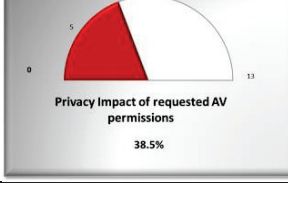
The Privacy Impact of Antivirus apps from 2011 was low. This was mainly due to the few permissions requested and even fewer Antivirus permissions. By 2015 the market had matured. The new privacy guidelines, GDPR, was also agreed in his year and the Privacy Impact was assessed for the apps that were available in 2011 and 2015 to determine if the developers were ready for increased privacy restrictions and if there were any differences between the two versions.

The Privacy Impact Framework Model was used after the comparator analysis to display the apps privacy impact. Google also introduced their protection_normal permission base in 2017 and this was factored into the Privacy Impact Framework Model to show any additional impact with unapproved permissions for newer¹² apps.

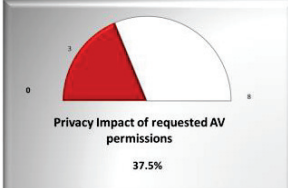
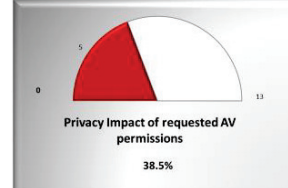
Using the Privacy Impact Framework Model to compare the privacy impact between the 2011 and 2015 apps the output in Table 11-6 show that Bluepoint apps are not requesting any PI_Perms in either 2011 and 2015. Lookout is the only developer whose request of PI_Perms has decreased.

¹² The Apps used in this study were extracted in January 2015.

Table 11-6 Privacy Impact changes for antivirus apps 2011 vs 2015

Company	Product	2011 vs 2015 version	
AV Free	AV antivirus free trial	 Privacy Impact of requested AV permissions 42.9%	 Privacy Impact of requested AV permissions 53.8%
AVG Mobilation	AntiVirus Free AVG	 Privacy Impact of requested AV permissions 42.9%	 Privacy Impact of requested AV permissions 46.2%
Bluepoint Security	BluePoint Antivirus	 Privacy Impact of requested AV permissions 0.0%	 Privacy Impact of requested AV permissions 0.0%
Bluepoint Security	BluePoint Antivirus free	 Privacy Impact of requested AV permissions 0.0%	 Privacy Impact of requested AV permissions 0.0%
Dr Web Ltd	Dr.Web Antivirus light	 Privacy Impact of requested AV permissions 0.0%	 Privacy Impact of requested AV permissions 16.7%
Dr Web Ltd	Dr.Web Antivirus	 Privacy Impact of requested AV permissions 0.0%	 Privacy Impact of requested AV permissions 46.2%
Lookout Inc	Lookout Mobile Security	 Privacy Impact of requested AV permissions 45.5%	 Privacy Impact of requested AV permissions 38.5%

Contribution

Company	Product	2011 vs 2015 version	
NetQin Mobile Inc.	Nq mobile manager Trial	 <p>Privacy Impact of requested AV permissions 37.5%</p>	 <p>Privacy Impact of requested AV permissions 38.5%</p>

The Privacy Impact Framework Models show clearly the impacts of each app.

11.3 Summary

The research began with an analysis of Antivirus apps to determine if they were effectively protecting the user against malware. The hypothesis was that the Antivirus apps protected the user against malware and that the commercial versions provided more protection than their free counterparts. The initial investigation was to determine what differences (if any) there were between the free and commercial versions of the app. The areas reviewed were the specified features and permissions, the sizes of the installed packages and if there was any correlation between the number of downloads of the app and the user rating.

The initial testing of the apps involved a laborious process, first downloading the app to a suitable device to test the efficacy of the app and to transfer the app to a test rig to obtain the values for comparison. Initially each app took approximately 30 minutes to perform the download, transfer to the test rig and prepare the app in Java source code to perform the analysis. Each step was performed manually and required multiple software to be installed to perform each step. This manual process was inappropriate to analyse large quantities of apps and an automated process was developed. Python code was written that would perform the two phases of disassembly from Davlik executable code to Java source code. Additional code then analysed the source code and extracted the Android Manifest file for inputting into a database. Using this process all 22 Antivirus apps were downloaded and analysed in 30 minutes in total. The main time was taken up with the download of the app to the device and extracting the executable and storing it on the test rig. The preparation of the 22 apps for analysis took less than 5 minutes.

Contribution

This process was developed into the P.E.M.P Method which was used in all future extractions and preparation of the app and now included preliminary analysis of the app's permissions of those available in the Froyo permission set and which Antivirus permissions were requested.

The result of the research revealed that although the commercial apps offered more features, these were not included in the app executable code. The MD5 hashes of the source code was identical, which indicated that the additional features were not built into the app and were probably online functions. The correlation between the features and permissions was significant indicating that the different number of features and number of permissions were linked. Therefore, the conclusion was that the hypothesis was incorrect and there was no difference in Antivirus functionality between the free and commercial apps. Although, this was not reflected by the user rating which rated the commercial version of the app higher than the free version.

The next step was to determine the efficacy of the Antivirus app. In 2010 the Antivirus testing organisation AV-test.org was still concentrating mainly on the PC infrastructure marketplace and had started testing apps in the mobile marketplace. They initially tested four apps over four operating systems (Windows mobile, Symbian, Android and iOS), of which only two apps were available for Android. In 2011 the testing had grown to test 6 apps on Android. However, only the feature set of the products were tested¹³.

To determine a baseline for the efficacy, the permissions required to detect and remove malware needed to be defined. Using many years of experience as a

¹³ AV-test.org eventually started testing Antivirus permissions in 2014.

Contribution

Security professional I could describe the various functions of an Antivirus Product and select the appropriate permissions required to permit the app to perform each of the functions. These have been named and referred to as the Antivirus permissions (AV_Perms). There were 17 permissions that performed the primary Antivirus roles of which 6 contravened the user's privacy.

To evaluate the efficacy and to eliminate outliers, a larger testbed was needed and all 22 Antivirus apps in the Android marketplace in 2011 were tested. For each product the app was installed onto a clean device, then the app was tested to detect the malware at download or via scans, if the app could either quarantine or remove the malware, detect if the device was rooted and if any rootkits were installed and finally review the malware signature update files. All these results were recorded and used to determine the efficacy of the Antivirus app. The hypothesis was that the apps that requested the most AV_Perms fulfilled the most functions and were therefore more effective. However, this was not the case, Aegislab Antivirus apps performed the best during the tests with a 90% overall score, but their efficacy rating was 35.3%, whilst Lookout Mobile had the better efficacy rating (64.7%) but achieved a slightly lower score of 70% overall, mainly due to reduced virus signature update functionality, which was not permission related. None of the apps tested passed all the tests.

In 2015, the Antivirus market had grown, but only five developers and their products had lasted over the intervening period. The 22 apps in 2011 had reduced to 7 as the apps evolved and matured. New apps and developers had entered the market and there were 57 developers with 67 products. The number of permissions available in the newer version of Android had also increased from 82 to 154. The median number of permissions had also increased but the maximum had reduced. Three of the apps did not request any permissions at all which cast doubts on their ability to perform the

Antivirus function. During the superseding years the commercial antivirus testing company started testing on mobile devices, from two Android Antivirus apps in 2011 to 12 in March 2015, although there were 67 Antivirus apps on the marketplace.

The Antivirus apps from 2015 were analysed and compared to their 2011 versions. In each case the number of requested permissions had increased. The Antivirus permissions were checked, and the 2015 apps were tested to determine if there was an improvement in the efficacy rating. There was a strong positive correlation between the Antivirus and Privacy permissions. As the apps had matured, more Antivirus permissions were requested and therefore so had the number of privacy impacting permissions.

To enable higher numbers of apps to be reviewed the 2011 process was automated and named the P.E.M.P. process (Permission Extraction Method and Process). PEMP automates the 2011 process and reduces the processing time. The automation was written using opensource code and has been tested on a variety of app genres and across multiple versions of Android. The apps from 2011 were re-tested using the PEMP process with the same output results.

The protection of children has a high priority and the next genre of apps to be tested for privacy were a sample of the free children's apps over 3 age groups. The mean number of permissions requested increased with age as did the privacy impact permissions. Analysis of the permissions showed that the child using some of the apps could be tracked, overheard or seen. The impact though was low as these types of permissions were not used in conjunction with each other.

The previous app analysis was concentrating on the protection the user was receiving. The next section reviewed the apps from the user perspective, what the user expected or what the user perceived the developer providing. Using

Contribution

Social and Psychological contracts to evaluate this expectation, showed that the developers were not considering the user's needs.

With the agreement of GDPR implementation the marketplace providers separated their privacy terms and conditions from their main agreements. The privacy fairness of these terms and conditions are yet to be tested by the regulators. An overview of how the regulator could implement the controls are defined.

A new concept of Privacy Impact is introduced and how it can be applied to the app marketplace.

Earlier testing of the apps had produced results which were detailed but difficult to show at a glance what the results were. A framework model was created to evaluate the input and display the results in a clearer, unambiguous output. The gauge evolved from simple charts to become the Privacy Impact Framework Model.

The Privacy Impact Framework Model was first used to evaluate and compare the efficacy of Antivirus apps, number of permissions requested and if the permission was an Antivirus permission. The model evolved to be able to display the privacy ratings of apps and was tested on the previous app genres; 2011 and 2015 Antivirus apps and the 2015 Children's apps.

Once the privacy permissions are designated and recorded in the master database, any genre of apps (or just apps) can be downloaded and analysed using the PEMP process and the output provided to the Privacy Impact Framework for display.

The model has been successfully tested on multiple genres and across multiple versions of Android.

Contribution

Currently the Privacy Impact Framework Model display is only visible on a PC, but the intention is to port the Model to a mobile environment as an app and have the display immediately available to a user prior to using or starting the app, thereby giving the user full control over their own data requirements. There are now tools in the marketplace which permit users to switch permissions off or on but, they do not provide guidance on which permissions to switch or whether these permissions used in conjunction with others affect their privacy. An initial privacy matrix was created and is evolving to enable greater detail of the privacy impact for the user.

The problem of communicating to the billions of mobile users the importance of their data and keeping it private is huge. I intend this research to be incorporated into an app which will be provided by the mobile operators and freely available to users. This will encourage use and gradually other educational techniques will aid in understanding data privacy and permit the user to take more control of their data.

Chapter 12. Addendum

Since this research was performed, Google brought app Ops out in Android 4.3, but removed the feature from Android 4.4.2, claiming it was released accidentally. This app provided the ability to switch permissions off once an app had been downloaded and installed. It is now available on the Google Play Store (<https://play.google.com/store/>) on Android 5.0. It will work on earlier releases of Android, but the device is required to be rooted. How to use this utility is described in 12.1.1.

There are also other apps available that allow the user to display and revoke permissions for an app. Some are described below.

Permission Explorer allows the user to filter apps and permissions by categories, giving more details about the permissions that were granted at installation time.

Permissions Observatory and app Permissions perform a similar function. These apps assist in determining if there are any apps with problematic permissions that need to be revoked or perhaps even uninstalled completely.

These are just a few guides that can be used.



12.1.1 Permission Control

Once an app is authorised to access a permissions group, the app may use any of the individual permissions that are part of that group. The user does not need to manually approve individual permissions updates that belong to a permissions group that are already accepted.

Subsequently Google has introduced a setting that permits a user to switch a permission on or off. The following process should be used (Android V6.0).

When you use an instant app

When you use an instant app, you can allow or deny permissions. To see what permissions an instant app has:


- On your device, open the Settings app .
- Go to Google  > **Instant Apps**.
- Tap the app you want to see more about.
- Look under “**Permissions**” to see what permissions the app has.

Turn permissions on or off

You can change the permissions that apps can access in the main Settings app on your device at any time. Keep in mind turning off permissions may cause apps on your device to lose functionality.



See all permissions for each app

For apps installed on your device:



- On your device, open the main **Settings** app .

- Tap **Apps** or **Application Manager** (depending on your device, this may look different).
- Tap the app you want to update.
- Tap **Permissions**.
- Next to a permission you want to turn on, move the switch to the right until it turns green. If you want to turn a permission off, move the switch to the left until it turns gray.

For instant apps

- On your device, open the Settings app .
- Go to Google  > **Instant Apps**.
- Tap the app you want to see more about.
- Look under “**Permissions**”.

See all apps installed on your device that can access permissions


- On your device, open the main **Settings** app .
- Tap **Apps** or **Application Manager** (depending on your device, this may look different).
- Tap Settings  > **app permissions**. If you can't find **app permissions**, you may need to tap **Privacy and safety** > **app permissions**.
- Tap a permission.
- If you want to turn that permission on for a specific app, move the switch to the right until it turns green. If you want to turn a permission off, move the switch to the left until it turns gray.

Check app permissions if an app isn't working

If a feature within an app isn't working as you would expect, try the steps below.

Step 1: Follow the instructions to contact the developer of the app.

Step 2: Check to see if any permissions have been disabled. To check app permissions:

- On your device, open the main **Settings** app .
- Tap **Apps** or **Application Manager** (depending on your device, this may look different).
- Tap the app you want to review.
- Tap **Permissions**. If a permission is turned off, the switch next to it will be gray.
- You can consider turning permissions on to see if that resolves your issue. To turn a permission on, move the switch to the right until it turns green.
- Try using the app again.

Google has also simplified the list of permissions that are presented to the user to enable them to better decide if the permissions requested is acceptable to them. The permissions are listed in permission groups and show the user the high-level name of the permission and not the more detailed permission. To review the permissions in detail, use the instructions in *See all permissions for each app*.

The permission groups are¹⁴:

- Body Sensors
- Calendar

¹⁴ These permission groups are for the permissions available on Android 6.0 and up. Permissions also vary by device and manufacturer.

Addendum

- Camera
- Contacts
- Location
- Microphone
- Phone
- SMS
- Storage

Each of the groups contain the detailed permission linked to that group.

Body Sensors - Access fingerprint data

Calendar - Read and write access to the calendar

Camera - Access the camera device and take photos and/or videos

Contacts - Read and write access to contacts

Location - Access detailed (fine) location

Microphone - Access audio via the microphone and record the audio.

Phone - Access to view number being dialled, answer incoming calls, manage calls, continue a call started from another app and accept calls.

SMS - Access to receive and send SMS messages

Storage - Read and write to external storage

All the above groups could be dangerous to the user as they permit any app to track, overhear, spy, send SMS texts to premium numbers, make and receive calls without any intervention by the user. Google leaves it to the developer to remove permissions from the groups. The user is not asked to accept these permissions.

12.1.2 RA Remover

For users that do not have a technical background to manage or remove app permissions an app, RA (remover app) is available. The app only displays the permissions from the manifest file as bad or not bad.

As with the Google Permission Control method, the system does not check to see which combination of permissions is harmful to the user's privacy etc. The user cannot define what is important to them, maybe permit a little loss of privacy so that they can play the app as single user as compared to the app as multi user. This would provide a fairer "social contract" for the user.

The users and regulators (and developers) require a system that clearly highlights the effect of the user on the installation/use of the app. A traffic light system is too simple, and a fuel gauge or side bar chart with depth that can show the depth of the effect on "privacy or security" as well as how the combination of the permissions is affecting the user.

Also, what difference does the mobile suppliers skin have on the user's privacy, are the hardware providers also obtaining usage data on the users? When do they request this? Another area for future research.

There also needs to be some sort of visual display for the user to show what benefit they have on top of using the app.

Hardening mobiles - what permissions need to be de-activated. Clearly as demonstrated in the initial research into antivirus apps, some of the apps that were supposed to protect the user were clearly spying on the user.

What detection is being performed on re-packaged apps? Users downloading "beta" or "pre-release" popular apps from third party sites as was the case with the "early release" of a "Guide to Pokémon Go" which contained rooting

malware (“Fake Apps Affect ANDROID OS Users,” 2011). Users have also downloaded Trojanised apps in the belief that these were legitimate app updates as discussed by Oscar Abendan in his report, “Trojanised apps are legitimate Android apps that cybercriminals maliciously altered to serve their own purposes. They download, modify, and upload legitimate apps to the Android Market or other app stores. These apps are usually free, so more users are likely to download them onto their mobile devices”. To be able to detect that the app/update is not legitimate the user would have to have a security product installed on their device that would detect malware at download.

Android app updates can add new “sub-permissions” in a category without requesting acceptance of these new permissions. The user would only be able to detect this by comparing permissions in the manifest files. A shorthand method to determine if there has been a change is to hash the package’s manifest files to ensure that nothing was injected.

References

- Adams, C., & Katos, V. (2005). The ubiquitous mobile and location-awareness time bomb. *Cutter IT Journal*.
- Adams, J. S. (1965). Inequity In Social Exchange. *Advances in Experimental Social Psychology*, 2, 267–299. [http://doi.org/10.1016/S0065-2601\(08\)60108-2](http://doi.org/10.1016/S0065-2601(08)60108-2)
- Angus, J. (2011). 15 Best Websites for Android Apps Downloads. *Tripwire Magazine*. Retrieved from <http://www.tripwiremagazine.com/2011/07/android-apps-download.html>
- APK Downloader. (2014). Retrieved from <http://apps.evozi.com/apk-downloader>
- APK Downloader V2. (2014). Retrieved from <http://codekiem.com/>
- APKtool. (2015). Retrieved from <http://ibotpeaches.github.io/Apktool/>
- AV Test Reports. (2011). *AV Test Org*. Retrieved from http://www.av-test.org/no_cache/en/tests/test-reports/
- Balakrishnan, D., Nayak, A., Dhar, P., & Kaul, S. (2009). Efficient geo-tracking and adaptive routing of mobile assets. In *2009 11th IEEE International Conference on High Performance Computing and Communications, HPCC 2009* (pp. 289–296). <http://doi.org/10.1109/HPCC.2009.79>

References

- Barrera, D., Kayacik, H. Günes, van Oorschot, P. C., & Somayaji, A. (2010). A methodology for empirical analysis of permission-based security models and its application to android. *Proceedings of the 17th ACM Conference on Computer and Communications Security*, (1), 73–84. <http://doi.org/http://doi.acm.org/10.1145/1866307.1866317>
- Bartel, A., Klein, J., Monperrus, M., & Le Traon, Y. (2014). Static analysis for extracting permission checks of a large scale framework: The challenges and solutions for analyzing Android. *IEEE Transactions on Software Engineering*, 40(6), 617–632. <http://doi.org/10.1109/TSE.2014.2322867>
- Browning, J. (2011). Google's Android phones face more attacks via apps. *SF Gate*. Retrieved from <http://www.sfgate.com/cgi-bin/article.cgi?f=/c/a/2011/04/21/BUL51J5G3C.DTL&feed=rss.technology>
- Brunk, B. D. (2002). Understanding the privacy space. *First Monday*.
- Canalys. (2011). Google's Android becomes the world's leading smart phone platform. *Americas The*, 13(January), 1–4. <http://doi.org/10.4324/9780203462027>
- Chrome Browser Download. (2012). Retrieved from https://www.google.com/intl/en_uk/chrome/browser
- Conger, K. (2017). Uber responds to report that it tracked devices after its app was deleted. *Tech Crunch*. Retrieved from <https://techcrunch.com/2017/04/23/uber-responds-to-report-that-it-tracked-users-who-deleted-its-app/>
- Dan Frommer. (2017). These are the 10 most popular mobile apps in America - Recode. Retrieved November 5, 2018, from

References

- <https://www.recode.net/2017/8/24/16197218/top-10-mobile-apps-2017-comscore-chart-facebook-google>
- de Souza e Silva, A. (2013). Location-Aware Mobile Technologies: Historical, Social and Spatial Approaches. *Mobile Media & Communication*, 1(1), 116–121. <http://doi.org/10.1177/2050157912459492>
- Dehghantanha, A., Udzir, N. I., & Mahmud, R. (2011). Towards data centric mobile security. *Proceedings of the 2011 7th International Conference on Information Assurance and Security, IAS 2011*, 62–67. <http://doi.org/10.1109/ISIAS.2011.6122796>
- Duncan, J. (2011). Protecting Location Privacy on Android Smart-Phones. *Security*, (September), 1–70. Retrieved from <http://www.kaso.co.uk/msccyber/JonathanDuncan-MSc-Dissertation.pdf>
- Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., & Sheth, A. N. (2010). TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. *Ossi '10*, 49, 1–6. <http://doi.org/10.1145/2494522>
- Enck, W., Gilbert, P., Chun, B.-G., Cox, L. P., Jung, J., McDaniel, P., & Sheth, A. N. (2014). TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. *Communications of the ACM*, 57(3), 99–106. <http://doi.org/10.1145/2494522>
- Enck, W., Ongtang, M., & McDaniel, P. (2009). On lightweight mobile phone application certification. In *Proceedings of the 16th ACM conference on Computer and communications security - CCS '09* (p. 235). <http://doi.org/10.1145/1653662.1653691>

References

- Fake Apps Affect ANDROID OS Users. (2011). *Trend Micro Inc.* Retrieved from <https://www.trendmicro.com/vinfo/us/threat-encyclopedia/web-attack/72/fake-apps-affect-android-os-users>
- Felt, A., Ha, E., Egelman, S., & Haney, A. (2012). Android permissions: User attention, comprehension, and behavior. *Proc. of SOUPS*, 1-14. <http://doi.org/10.1145/2335356.2335360>
- Felt, A. P., Chin, E., Hanna, S., Song, D., & Wagner, D. (2011). Android permissions demystified. *Proceedings of the 18th ACM Conference on Computer and Communications Security - CCS '11*, 627. <http://doi.org/10.1145/2046707.2046779>
- FlexiSPY™ Unique Android Spy app – Reveals Secrets Others Cannot. (2017). Retrieved June 26, 2017, from <https://www.flexispy.com/en/android-spy-app-flexispy.htm>
- GDPR FAQs. (2017). *GDPR Portal*. Retrieved from <http://www.eugdpr.org/gdpr-faqs.html>
- GDPR Summaries of Articles. (2017). *GDPR Portal*. Retrieved from <http://www.eugdpr.org/article-summaries.html>
- Gibler, C., Crussell, J., Erickson, J., & Chen, H. (2012). AndroidLeaks: Automatically detecting potential privacy leaks in Android applications on a large scale. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7344 LNCS, 291-307. http://doi.org/10.1007/978-3-642-30921-2_17
- Goodin, D. (2010). Android Malware Attacks. *The Register*. Retrieved from http://www.theregister.co.uk/2010/11/10/android_malware_attacks/
- Google Android phone shipments increase by 886%. (2010). Retrieved from

References

- <http://www.bbc.co.uk/news/technology-10839034>
- Henry, R., & Flyn, C. (2012). Tap, tap, tapping us all up - mobile apps invade privacy. *The Sunday Times*, 12-13.
- Horn, L. (2010). AVG Buys DroidSecurity. *PCMAG*. Retrieved from <http://www.pcmag.com/article2/0,2817,2372477,00.asp>
- IDC. (2011). Retrieved from <http://www.idc.com/promo/smartphone-market-share/os>
- Java Download. (2015). Retrieved from <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>
- Katos, V., & Adams, C. (2005). Modelling corporate wireless security and privacy. *Journal of Strategic Information Systems*. <http://doi.org/10.1016/j.jsis.2005.07.006>
- Kellex. (2011). Droid-Life.
- Kelley, P. G., Consolvo, S., Cranor, L. F., Jung, J., Sadeh, N., & Wetherall, D. (2012). A Conundrum of Permissions, 1-12.
- Kelly, G. (2014). Report of Mobile Malware. *Forbes*. Retrieved from <http://www.forbes.com/sites/gordonkelly/2014/03/24/report-97-of-mobile-malware-is-on-android-this-is-the-easy-way-you-stay-safe/#355de8017d53>
- Law Patent Group. (2012). New draft European data protection regime. Retrieved June 6, 2018, from http://mlawgroup.de/news/publications/detail.php?we_objectID=227
- Lella, A., & Lipsman, A. (2017). The 2017 U.S. Mobile app Report - Comscore,

References

- inc. Retrieved November 5, 2018, from <https://www.comscore.com/Insights/Presentations-and-Whitepapers/2017/The-2017-US-Mobile-App-Report>
- Lu, L., Yegneswaran, V., Porras, P., & Lee, W. (2010). BLADE: An Attack-Agnostic Approach for Preventing Drive-By Malware Infections. *Interactions*, 440–450. <http://doi.org/10.1145/1866307.1866356>
- Marforio, C., Francillon, A., & Capkun, S. (2011). Application Collusion Attack on the Permission-Based Security Model and its Implications for Modern Smartphone Systems, 1–16.
- McIlroy, S., Ali, N., & Hassan, A. E. (2016). Fresh apps: an empirical study of frequently-updated mobile apps in the Google play store. *Empirical Software Engineering*, 21(3), 1346–1370. <http://doi.org/10.1007/s10664-015-9388-2>
- Mobile device market to reach 2.6 billion units by 2016 | Canalys. (2013). Retrieved February 22, 2013, from <https://www.canalys.com/newsroom/mobile-device-market-reach-26-billion-units-2016>
- MobileSpy. (2017). Retrieved June 26, 2017, from <http://www.mobile-spy.com/android.html>
- Mobistealth. (2017). Retrieved June 11, 2017, from <http://www.mobistealth.com/android-spy-software>
- Moore, T., Clayton, R., & Anderson, R. (2009). The Economics of Online Crime. *Journal of Economic Perspectives*, 23(3), 3–20. <http://doi.org/10.1257/jep.23.3.3>
- Mylonas, A., Kastania, A., & Gritzalis, D. (2013). Delegate the smartphone

References

- user? Security awareness in smartphone platforms. *Computers and Security*, 34, 47–66. <http://doi.org/10.1016/j.cose.2012.11.004>
- Palfrey, J., Gasser, U., & Boyd, D. (2010). Response to FCC Notice of Inquiry 09 - 94 “ Empowering Parents and Protecting Children in an Evolving Media Landscape. *Berkman Center Research ...*, 7641. Retrieved from http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1559208
- Pilz S, H. & S. (2012). Are free antivirus scanners any good? . *AV-Test.Org. The Independent IT-Security Institute*. . Retrieved from <http://www.av-test.org/en/home>
- Porras, P., Saïdi, H., & Yegneswaran, V. (2010). An analysis of the iKee.B iPhone botnet. In *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering* (Vol. 47 LNICST, pp. 141–152). http://doi.org/10.1007/978-3-642-17502-2_12
- Privacy Blocker. (2017). Retrieved May 17, 2017, from play.google.com/store/apps/details?id=com.xeudoxus.privacy.inspector
- Product Review: Mobile Security - August 2010. (2010). Retrieved from <https://www.av-comparatives.org/mobile-security>
- Protection Normal. (2017). Retrieved from https://developer.android.com/reference/android/content/pm/PermissionInfo.html#PROTECTION_NORMAL
- Python Downloads. (2015). *Python*. Retrieved from <https://www.python.org/downloads/>
- Ramachandran, R., Oh, T., Stackpole, W., & Smartphone, A. P. C. (2012). Android Anti-Virus Analysis, 35–40.

References

- Raywood, D. (2010). Mobiles used by Zeus as SMS messages are used to deliver one time passwords. *SC Magazine UK*.
- Razaghpanah, A., Nithyanand, R., Vallina-Rodriguez, N., Sundaresan, S., Allman, M., Kreibich, C., ... Org, I. C. (2018). Apps, Trackers, Privacy, and Regulators A Global Study of the Mobile Tracking Ecosystem. In *NDSS*. <http://doi.org/10.14722/ndss.2018.23353>
- Rideout, V. (2013). Zero to eight: Children's media use in America 2013. *Pridobljeno*, 1-31. Retrieved from <https://www.common sense media.org/research/zero-to-eight-childrens-media-use-in-america-2013>
- Rousseau, D. M. (1989). Psychological and implied contracts in organizations. *Employee Responsibilities and Rights Journal*, 2(2), 121-139. <http://doi.org/10.1007/BF01384942>
- Schlegel, R., Zhang, K., & Zhou, X. (2011). Soundcomber: A stealthy and context-aware sound trojan for smartphones. *Proceedings of the 18th Annual Network and Distributed System Security Symposium (NDSS)*, 17-33. Retrieved from <https://www.cs.indiana.edu/~kapadia/papers/soundcomber-ndss11.pdf>
- Senior Lab DE Apps. (2017). Retrieved from <https://play.google.com/store/apps/details?id=seniorlabde.allapps>
- Shah, V. (2012). CDCD-5 an Improved Mobile Forensics Model, 2(4), 739-741.
- Smart phones overtake client PCs in 2011. (2012). *Canalys Newsroom*. Retrieved from <http://www.canalys.com/newsroom/smart-phones-overtake-client-pcs-2011>

References

- Snoopwall App. (2017). Retrieved from <https://play.google.com/store/apps/details?id=com.snoopwall.privacyapp>
- Tablet Computers Hold Back PC Sales Growth. (2011). Retrieved from <http://www.bbc.co.uk/news/business-12180679>
- Taylor, K. (2010). Mobility Features. *Tgdaily*. Retrieved from <http://www.tgdaily.com/mobility-features/53287-trojan-can-take-over-android-phones>
- Thomas, R., & Martin, J. (2006). The underground economy : priceless. *USENIX; Login* 1206. Retrieved from <http://www.usenix.org/publications/login/2006-12/openpdfs/cymru.pdf>
- Top ten reviews. (2011). Retrieved from <http://anti-virus-software-review.toptenreviews.com/>
- Tsavli, M., Efraimidis, P. S., Katos, V., & Mitrou, L. (2015). Reengineering the user: privacy concerns about personal data on smartphones. *Information and Computer Security*. <http://doi.org/10.1108/ICS-10-2014-0071>
- Vennon, T., & Stroop, D. (2010). Android Market: Threat Analysis of the Android Market. *Statistics*, 88. Retrieved from <http://threatcenter.smobilesystems.com/wp-content/uploads/2010/06/Android-Market-Threat-Analysis-6-22-10-v1.pdf>
- Wain, K., Au, Y., Zhou, Y. F., Huang, Z., & Lie, D. (2012). PScout : Analyzing the Android Permission Specification. *CCS '12 Proceedings of the 2012 ACM Conference on Computer and Communications Security*, 217–228.

References

<http://doi.org/10.1145/2382196.2382222>

www.cyrket.com/m/android. (n.d.). Retrieved April 16, 2011, from www.cyrket.com/m/android

Xing, L., Pan, X., Wang, R., Yuan, K., & Wang, X. F. (2014). Upgrading your Android, elevating my malware: Privilege escalation through mobile OS updating. *Proceedings - IEEE Symposium on Security and Privacy*, 393-408. <http://doi.org/10.1109/SP.2014.32>

Xu, N., Xu, N., Zhang, F., Zhang, F., Luo, Y., Luo, Y., ... Teng, J. (2009). Stealthy video capturer: a new video-based spyware in 3G smartphones. *WiSec '09: Proceedings of the Second ACM Conference on Wireless Network Security*, 69-78.

<http://doi.org/http://doi.acm.org/www.library.gatech.edu:2048/10.1145/1514274.1514285>

Appendix A Detailed Permission tables

A.1. Antivirus Apps Analysis Input Tables

The extraction of data during the research process produced a variety of input tables. These tables were used as a base for the research. A summary of each of the tables is included in the main text of the thesis.

Table A-1, shows the Antivirus products with the highest downloads according to the Androlib Market site (Androlib Market , 2011) on February 28th, 2011. Beta and trial products have been included in the selection, although applications with downloads lower than 50 and/or with zero (nil) ratings have been excluded.

Table A-1 Security Applications on Androlib Marketplace as at 28/02/2011

<i>Application</i>	<i>Company</i>	<i>Rating</i>	<i># of reviews</i>	<i># of downloads</i>	<i>reviews as a % of download</i>
Lookout Mobile Security	Lookout Inc	4.6	126412	20,587,202	0.61
AntiVirus Free AVG	Droidsecurity - AVG	4.4	80331	13,082,544	0.61
Dr Web for Android light	Doctor Web Ltd	4.6	14655	896,001	1.64
Antivirus Free	Creative Apps	4.33	7689	591,624	1.30
NQmobile Antivirus	NetQin Mobile Inc.	4.5	5566	250,000	2.23
NQmobile Antivirus For 1.5/1.6 Android	NetQin Mobile Inc.	4.4	783	250,000	0.31
Super Security Standard	superdroid.net	4.3	1950	128,798	1.51
ViRobot Mobile	Hauri Inc	4.3	1254	123,425	1.02
AppScan Beta	Aegislab	4.3	1066	80,514	1.32
Super Task Killer 2011	NetQin Mobile Inc.	4.2	1512	75,600	2.00
MyAndroid Protection 2.0+	Mymobile Security	3.9	234	60,622	0.39
MyMobile Protection 2.0+	Mymobile Security	3.6	94	33,571	0.28
MyAndroid Protection 1.5/1.6	Mymobile Security	3.78	55	17,742	0.31
Anti virus	Andro Security	3.7	128	16,410	0.78
MyMobile Protection v.1.5/1.6	Mymobile Security	3.9	63	15,366	0.41
BluePoint Antivirus	Bluepoint security Inc	4.1	231	10,645	2.17
MobiShield	Trustmobi	4.2	71	10,000	0.71
Kinetoo Malware scan	CPU Media Sarl	4.1	65	5,000	1.30

Appendices

<i>Application</i>	<i>Company</i>	<i>Rating</i>	<i># of reviews</i>	<i># of downloads</i>	<i>reviews as a % of download</i>
Antivirus - Risk Detector	eDroid Apps	4	28	1,000	2.80
Onetouch antivirus	Shipwrecktech	3.7	10	1,000	1.00
BlackBelt AntiVirus	UMU Ltd	4	11	49	22.45

The table contains the free security apps that contain an Antivirus component and the developer. Details of the user rating and the number of reviews and downloads are also recorded. The number of reviews as a percentage of the download were calculated to determine if the rating value was a true representative of the users downloading the product. The lower the figure indicated that more users that download the product provided a rating. This was then used to rank the apps.

Table A-2 Companies with free and commercial versions of Antivirus apps

<i>Company</i>	<i>Application</i>	<i>Order in free list</i>	<i>Cost</i>
Lookout Inc	Lookout Mobile Security	1	free
Lookout Inc	Lookout Mobile Security	1	\$29.99
AVG Mobilation	AntiVirus Free AVG	2	free
AVG Mobilation	AntiVirus Pro	2	£6.10
AVG Mobilation	Security Pro	2	£3.05
Doctor Web Ltd	Dr Web Anti-virus	3	\$4.99
Doctor Web Ltd	Dr Web Antivirus light	3	free
Creative Apps	Antivirus Free	4	free
NetQin Mobile Inc.	NQmobile Antivirus	5	free
NetQin Mobile Inc.	NQmobile Antivirus for 1.5/1.6 Android	6	free
superdroid.net	Super Security Standard	7	free
Hauri Inc	ViRobot Mobile	8	free
Aegislab	Aegislab Antivirus free	9	free
Aegislab	AntiVirus Elite	9	£4.88
Bluepoint security Inc	BluePoint Antivirus	16	free
Bluepoint security Inc	BluePoint Antivirus	16	£3.05
Trustmobi	MobiShield	17	free
CPU Media Sarl	Kinetoo Malware scan	18	free
Shipwrecktech	Onetouch antivirus	20	free
Qianjun	Virus Terminator	new	free
MoonBeam Development	Android defender virus protect	new	free

Appendices

Moonbeam Development	Defender Pro virus	new	\$4.99
McAfee	McAfee WaveSecure	trial	\$19.90
Mymobile Security	MyAndroid Protection 2.0+	trial	€ 36
Mymobile Security	MyAndroid Protection 1.5/1.6	trial	€ 36
P.Defender Antivirus	MyAntiVirus Pro	paid	\$10.12
UMU Ltd	BlackBelt Antivirus	trial	£9.95
Kaspersky	Kaspersky Mobile Security	paid	£6.07
DMA	Antivirus	paid	£0.85
UMU Ltd	BlackBelt Security	trial	£19.95
Livezen	Smart Defender Pro	paid	\$1.99
Webroot	Webroot Mobile Security basic	new	free
Webroot	Webroot Mobile Security	new	£9.15

The table orders the security applications by the number of downloads as at 04/05/2011. The price of the product is also included in the currency as specified on the Play Store. The order provides the rank of the free app according to the rating co-efficient. The other values indicate that the product is new (new since 28/02/2011), a trial version, or paid (commercial only).

Table A-3 Comparison of features of Antivirus products in the study

	<i>Lookout</i>	<i>AVG</i>	<i>Dr Web</i>	<i>Aegislab</i>	<i>Bluepoint</i>	<i>Moonbeam</i>
	<i>Mobile</i>					
Scheduled scans	Y	Y	Y	Y	Pro version only	Y
Email support	Y	Y	Y	N	Y	Y
Real Time Protection	Y	Y	Y	Y	Y	Y
Scan Memory cards	N	Y	Y	N	Y	N
Virus definition Update	Y	Y	Y	Y	(Uses cloud)	Y
Real Time Scan of Audio files	N	Pro version only	N	N	Y	N
Real Time scan of Email	Y	Y	Y	N	Y	N
Real Time scan of SMS	Y	Y	Y	Elite version only	Y	N
Real Time scan of Market Apps	Y	Y	Y	Y	Pro Version only	Y

The features of the products, as stated by the developers, were documented and used in the comparisons.

Table A-4 Android permissions requested by each app.

Android Permission	Lookout Inc		AVG Mobilation		Doctor Web Ltd		AegisLab		Blueprint Security Inc		MoonBeam Development	
	Lookout Mobile Security (Premium)	Lookout Mobile Security (Free)	AVG AV Pro	AVGAV free	Dr Web AV	Dr Web AV Light	AV Elite	AV free	Antivirus	AV free	Defender Pro	Android defender virus protect
ACCESS_COARSE_LOCATION	y	y	y	y							y	y
ACCESS_COARSE_UPDATES			y	y								
ACCESS_FINE_LOCATION	y	y	y	y							y	y
ACCESS_NETWORK_STATE	y	y	y	y	y	y	y	y			y	y
ACCESS_WIFI_STATE			y	y			y					
CALL_PHONE												
CHANGE_NETWORK_STATE					y	y						
CHANGE_WIFI_STATE			y									

Appendices

Total Permissions	30	30	27	27	16	8	14	3	11	11	7	7
-------------------	----	----	----	----	----	---	----	---	----	----	---	---

The table contains all the permissions for this version of Android (Froyo). Where an app has requested a permission, it has been marked with a “y”.

Table A-5 Details of non-android permissions requested

Other Permissions	Lookout Inc		AVG Mobilation	
	Lookout Mobile Security (Premium)	Lookout Mobile Security (Free)	AVG AV Pro	AVG AV free
com.android.browser.permission.READ_HISTORY_BOOKMARKS	y	y	y	y
com.android.browser.permission.WRITE_HISTORY_BOOKMARKS	y	y	y	y
com.android.launcher.permission.INSTALL_SHORTCUT			y	y
com.android.launcher.permission.UNINSTALL_SHORTCUT			y	y
com.android.launcher.permission.WRITE_SETTINGS			y	y
com.android.vending.CHECK_LICENSE			y	
com.antivirus.permission.C2D_MESSAGE				y
com.google.android.c2dm.permission.RECEIVE	y	y	y	y
com.htc.launcher.permission.WRITE_SETTINGS			y	y
com.lookout.permission.C2D_MESSAGE	y	y		
com.sonyericsson.homescreen.permission.READ_SETTINGS			y	y
com.sonyericsson.homescreen.permission.WRITE_SETTINGS			y	y
org.antivirus.permission.C2D_MESSAGE			y	
Total permissions	4	4	11	10

Two of the developers requested additional permissions, Lookout Mobile and AVG. The permissions requested by each of these developer's apps are recorded.

The analysis of the ratings was similar irrespective of the number of features of the app (Table 6-2).

As the popularity of the Android OS grew many antivirus and security developers were bought by the mainstream Security software companies. The Antivirus arena on mobiles which was in its infancy in 2011 matured over the four years. The major providers of Antivirus programs from the PC/Laptop arena consolidated their position by purchasing or by merging with other companies, as in the case with AVG entering the mobile Antivirus market by purchasing DroidSecurity (Horn, 2010). This meant that multiple Antivirus products were available from one company, whilst the products were consolidated, incorporated into an existing product or dropped from the marketplace altogether.

The growth of apps with Antivirus components from 2011 to 2015 is shown in Table A-6.

Table A-6 Android Antivirus apps in 2011 and 2015

2011 apps	2015 apps	
Aegis Appscan Beta	360_security_-_antivirus_boost	Dr.Web_v.9_Anti-virus
Aegislab Antivirus free	Advanced_Task_Manager_-_Boost	Dr.Web_v.9_Anti-virus_Light
Aegislab mobile	ALYac_Android	DU_Speed_Booster_Cache_Cleaner
Android defender virus protect	AMC_Security_-_Clean_n_Booster	eScan_-_Mobile_Antivirus
Antivirus droid	Android_Optimizer	Free_Antivirus_2014+_Security
Antivirus Free	Anti_Virus_Android	free_antivirus_2015_security
AntiVirus Free AVG	Antivirus Pro for Android	Free_Antivirus_and_Security(panda)
AV antivirusfree trial	Anti-Virus_Android	Free_Antivirus_and_Security(sophos)
BlackBelt Antivirus	Antivirus_Android(androhelm)	Free_Antivirus_Pro_2014
BluePoint Antivirus	Antivirus_Booster_&_Cleaner	Free_Security_n_Antivirus
BluePoint Antivirus free	Antivirus_Complete_Protection	Free_virus_scan_(Antivirus)
Defender Pro virus	Antivirus_for_Android(A.A)	GuardX_Antivirus
Dr. Web Android light	Antivirus_for_Android(dala)	Hornet_AntiVirus_Free
Dr. Web Antivirus light	Antivirus_for_Android(itus)	kaspersky_internet_security
Lookout Mobile Security	Antivirus_for_Android(lab4)	LINE_Antivirus
Lookout Mobile Security trial	Antivirus_for_Android(moobila)	lookout_security_n_antivirus
MyAndroid Protection	Antivirus_for_Android_FREE	Mobile_Security_and_Antivirus(bullguard)
Nqmobile booster	antivirus_for_android_TM	Mobile_Security_n_Antivirus(avast)
Nqmobile manager Trial	Antivirus_for_androids_2015	Mobile_Security_n_Antivirus(bitdefender)
ScanLife	antivirus_free-mobile_security	Mobile_Security_n_Antivirus(eset)
SmrtGuard Pro Trial	AntiVirus_n_Anti-Adware	norton_security_and_antivirus
Super Security Standard	Antivirus_n_Mobile_Security(trustgo)	NQ_Mobile_Security_&_Antivirus
	antivirus_n_mobile_security_(quickheal)	Secure_Antivirus
	antivirus_Security_-_FREE	Security_-_Free

Armor for Android	Security_n_Antivirus_-_FREE(mcafee)
avira_antivirus_security	Tablet Antivirus Security PRO
bitdefender_antivirus_free	tablet_antivirus_security_FREE
Bkav_Security_-_Antivirus_Free	Virus_Removal_for_Android
BluePoint_Antivirus_Free	virus_scan_(antivirus)
clean_master_(speed_booster)	VIRUSfighther_Antivirus_FREE
Cleaner_Master_Antivirus_Plus	White-Gate_Antivirus
cm_security_antivirus_applock	Zoner_AntiVirus
Comodo_Security_&_Antivirus	Zoner_AntiVirus_-_Tablet
CY_Security_Antivirus_Cleaner	

In 2011 there were 22 apps with Antivirus components. In 2015 the number of apps with Security or Antivirus functions was 240, of which 67 were Antivirus apps. Developers use multiple tags or keywords to provide greater visibility of their apps during searches. The 240 apps contained the keywords “security” or “antivirus” or both. These apps were reviewed to confirm that they did possess an Antivirus component. In total 67 of the 240 apps performed Antivirus functions.

This research added to the initial 2011 research and concentrated on analysing the permissions of the 67 Antivirus apps in 2015. The permissions and features from the initial 2011 Antivirus apps were available to perform comparison testing between the apps that were available in both 2011 and 2015, albeit at a newer release.

Table A-7 List of Security and Antivirus apps in 2015

app Name	Developer
360 Security Antivirus Boost	Qihoo 360 (NYSE:QIHU)
360????	360mobilesafe
70 Antivirus Programs for Free	Lev Well
960 Clean Antivirus Security +	STAR APPS PVT LTD
Advanced Mobile Antivirus Tips	PhanQuocQuan
Advanced Mobile Care Antivirus	Galileo Best Free Download
AegisLab Antivirus Free	AegisLab
AegisLab Antivirus Premium	AegisLab
All About Antivirus	Havana Apps
AMC Security Clean & Booster	IObit Mobile Security
AMS Antivirus Mobile Security	AMS Antivirus Mobile Security
Anti Virus & Mobile Security!	Suzy Software
Anti Virus 2014	Plato Information Best Apps
ANTI VIRUS 2014	Puttarapha LLC.
Anti Virus and Spyware Remover	IZSALA KANTIWONG
Anti Virus Android	PiggiesMaz
Anti Virus Info	Appsplan1
Anti Virus Remover	Stephen Best Free Apps
AntiGen Max Anti Virus	Ian Voorhies
AntiVirus	AndroHelm Antivirus
AntiVirus - Android	AndroHelm Antivirus
AntiVirus	Kevlanche
AntiVirus	Playerum
AntiVirus – Spanish	MyPengo Mobile
AntiVirus & Anti-Adware	SeCore Mobile Security
Antivirus & Mobile Security	Quick Heal Technologies America Inc
Antivirus & Mobile Security	TrustGo Inc.
Antivirus & Mobile Security	Trustlook Mobile Security
Antivirus & Security	AVAST Software
Antivirus 2014 for Android	Wequees
Antivirus 2015 Virus Security	Complete mobile security AntiVirus Free
	Anti virus
Antivirus and Mobile Security	Plato Information Best Apps
Antivirus Android	AndroHelm Antivirus
Anti-Virus Android	AndroHelm Antivirus
antivirus Android phones 2015	Mobile Speed Booster, Clean Free Master
	Antivirus
AntiVirus Android.	AndroHelm Antivirus
Antivirus Auto Remove Virus	Jonesaevan
Antivirus Auto Remove Virus	koogoo

app Name	Developer
Antivirus Auto Remove Virus	MoneyLand
Antivirus Auto Remove Virus	NZ Design
Antivirus Auto Remove Virus	Plato Information Best Apps
Antivirus Auto Remove Virus	Stephen Best Free Apps
antivirus auto remove virus	Tkdevmobile
Antivirus Booster & Cleaner	PSafe Tecnologia S.A.
Antivirus Complete Protection	sagamore
Antivirus Download Free	Stephen Best Free Apps
Antivirus for Android	Android Antivirus
Antivirus for Android	Dala Apps
Antivirus for Android	Itus Mobile Security
Antivirus for Android	Moobila Corporation
ANTIVIRUS FOR ANDROID	New Papa
Antivirus for Android FREE	XipIO
Antivirus for Android Pro	Itus Mobile Security
Antivirus for Android.	Android Antivirus
ANTIVIRUS FOR ANDROID™	Tap Media Inc.
AntiVirus FREE	Kevlanche
Antivirus Free	Wequees
Antivirus Free Phones	Jackson app
Antivirus Free-Mobile Security	NQ Creative Apps
Antivirus guide	Havana Apps
AntiVirus Laser	MyNikko
AntiVirus Laser Pro	MyNikko
Antivirus Manual	Havana Apps
Antivirus Mobile Security Scan	Mohammad Ashraf Hossain
Antivirus Plus	ABV Corporation
Antivirus Plus	Zr technologies
Antivirus Pro	ABV Corporation
Antivirus Pro	NCN-NetConsulting Ges.m.b.H.
Antivirus Pro 2014	NCN-NetConsulting Ges.m.b.H.
Antivirus Pro 2015 Security	Antivirus Pro
AntiVirus PRO Android Security	AVG Mobile
Antivirus Pro for Android	Android Antivirus
Antivirus Programs	Havana Apps
Antivirus Protection	BachTruongSon
Antivirus Protection Gold	sagamore
Antivirus Quiz	theandroidgalaxy
Antivirus Realtime	Blue Master
Antivirus Scanner Security app	Free mobile speed booster, anti virus clean master

app Name	Developer
AntiVirus Security	AndroHelm Antivirus
AntiVirus Security - FREE	AVG Mobile
Antivirus Security Free 2015	koogoo
Antivirus Security Manager	Blue Application
Antivirus security pro	Antivirus Security Complete Virus Protection
Antivirus Security Pro	Appatron Soft
Antivirus Security Scanner	Appatron Soft
AntiVirus Software	ESTSoft
Antivirus Tablet	AndroHelm Antivirus
Antivirus TESTVIRUS	P.Defender Antivirus
Antivirus Tips	Chatura Dange
Antivirus Ultimate	ABV Corporation
AntiVirus VIP	Run+Run+Now
Antivirus*	DMA
Armor for Android™ Antivirus	Armor for Android™
Audio Book Anti Virus	Twayesh Projects
AVG AntiVirus PRO for Xperia™	AVG Mobile
Avira Antivirus Security	AVIRA
AVL Pro Antivirus & Security	AVL Team
Bastiv Security Antivirus	Bastiv Security
Best Antivirus	ru.fo
Best Antivirus Security	AndroidAppTools
Best Free Antivirus	Jonesaevan
Bitdefender Antivirus Free	Bitdefender
Bkav Security – Antivirus Free	Bkav Corporation
BlackBelt AntiVirus Trial	BlackBelt SmartPhone Defence Ltd.
BluePoint Antivirus Free	BluePoint Security, Inc.
BluePoint Antivirus Pro	BluePoint Security, Inc.
Bornaria security (Antivirus)	Ariasecure Corp.
CCleaner	Piriform
Clean Master (Speed Booster)	Cheetah Mobile
Cleaner Booster 360 Antivirus	PLUSStudio
Cleaner Master & Antivirus	Heart Throb
Cleaner Master Antivirus	RED ANDRO SOLUTIONS
Cleaner Master Antivirus Plus	IFSC Code
Cleaner Master AntiVirus Pro	RED ANDRO SOLUTIONS
Cloud Security & Antivirus	Cloud Mobile Apps
Cloud Security AntiVirus FREE	AuroraTeam
CM Security Antivirus AppLock	Cheetah Mobile (AntiVirus & AppLock)
CM Security Antivirus Plus	RED ANDRO SOLUTIONS

app Name	Developer
Comodo Security & Antivirus	Comodo Security Solutions
CoolAntivirus Antivirus	SOR ENTERTAINMENT, S.L.
CY Security Antivirus Cleaner	CY Security
Dr.Mobile Antivirus & Security	SSME
Dr.Mobile PRO Antivirus	SSME
Dr.Web v.9 Anti-virus	Doctor Web, Ltd
Dr.Web v.9 Anti-virus Life lic	Doctor Web, Ltd
Dr.Web v.9 Anti-virus Light	Doctor Web, Ltd
DU Speed Booster?Cache Cleaner	DU Apps
EICAR Anti-virus Test	eXtorian
eScan - Mobile Antivirus	MicroWorld Technologies Inc.
eScan - Tablet Antivirus	MicroWorld Technologies Inc.
Fastscan Anti-Virus	K-TEC Inc.
Fastscan free Anti-Virus	K-TEC Inc.
FREE Android Antivirus	Simple Soft Alliance
Free Antivirus	DavmaTech
Free Antivirus 2015 Security	Antivirus Pro
Free Antivirus 360°	Android Antivirus Free
Free Antivirus and Security	Panda Security
Free Antivirus and Security	Sophos Limited
Free Antivirus for Android	NZ Design
Free Antivirus Pro	NCN-NetConsulting Ges.m.b.H.
Free Antivirus Pro 2014	NCN-NetConsulting Ges.m.b.H.
Free Antivirus Pro 2015	NCN-NetConsulting Ges.m.b.H.
Free Antivirus Protection	Blue Application
Free Antivirus Security 2014	apps for life
Free Antivirus Software	Plato Information Best Apps
Free Cleaner 360 For Antivirus	BallDEVELOPER
Free Mobile Antivirus	Blue Application
Free Tablet Antivirus Security	Best Free of Best Apps
Free virus scan (Antivirus)	Complete mobile security AntiVirus Free
F-Secure Antivirus Test	Anti virus
F-Secure Mobile Security	F-Secure Corporation
G-Protector Anti Virus Utility	F-Secure Corporation
GreenShield Antivirus Suit	Gpc
GuardX Antivirus	Trantor Soft
Hornet AntiVirus Free	QStar
Hornet AntiVirus PRO	Hornet Mobile Security
IKARUS mobile.security	Hornet Mobile Security
Kaspersky Internet Security	IKARUS Security Software GmbH
	Kaspersky Lab

app Name	Developer
KT Antivirus	Katyayini Infotech Private Limited
LabMSF Antivirus beta	LabMSF
LabMSF Antivirus Premium	LabMSF
LINE Antivirus	LINE Corporation
Ma Antivirus	GenieSoftSystem Pvt Ltd.
Malwarebytes Anti-Malware	Malwarebytes
MAX GAMER ANTIVIRUS	Max Mobi Secure
Mobile & Security & Antivirus	Star Cube Applications
Mobile Antivirus AntiBug	ABV Corporation
Mobile Antivirus Security	Blue Application
Mobile Antivirus Security Info	NgoQuocHung
Mobile Cleaner - Antivirus	artbenad
Mobile Cleaner - Antivirus	Azedev
Mobile Cleaner And Antivirus	KITMADE
Mobile Cleaner Antivirus 360	BallDEVELOPER
Mobile Safe Antivirus	Blue Application
Mobile Security & Antivirus	ESET
Mobile Security & Antivirus	Trend Micro EMEA
Mobile Security & Antivirus -Bitdefender	Bitdefender
Mobile Security and Antivirus	BullGuard
Mobile Security and Antivirus	SecuraLive
Mobile Security Antivirus	koogoo
My anti virus	PLAY FUN
My AntiTheft & Antivirus	Mobile Cloud Labs Plc.
My Antivirus	Mobile Cloud Labs Plc.
Netlux Mobile Antivirus	Netlux Systems Private Limited
New Antivirus 2014	Plato Information Best Apps
Norton Security and Antivirus	NortonMobile
NQ Mobile Security & Antivirus	NQ Mobile Security (NYSE:NQ)
Octo Antivirus Free	Octappis
Operation Antivirus	MobiTrail
Othello Anti-virus	webmarkcom
Phone Antivirus	ISawan
Phone Clean Virus	Monoapps
Quick AntiVirus	ONS
Ram Cleaner - Antivirus	Azedev
Ram Cleaner And Antivirus	artbenad
Right Antivirus – Top Security	VcareAll
Secure Antivirus	Secure Antivirus
SecureBrain Antivirus (BETA)	SecureBrain
SecureIT Antivirus & Security	SecurityCoverage, Inc.

app Name	Developer
Security - Free	Webroot Inc.
Security & Antivirus Lookout	Lookout Mobile Security
Security & Antivirus -FREE	McAfee (Intel Security)
Security & Antivirus Guard	Sophos Limited
Shield Antivirus Protection	Gauraw_Yadav
SkyShield Mobile AntiVirus	SmartInstall Sp. z o.o.
Smart Android Antivirus	Samtech Solutions
Smart Antivirus	VcareAll
Smart Antivirus 2014	Deeni Apps
Snap Secure	SnapOne, Inc.
Star Antivirus	Secure Antivirus
Super Antivirus Defender	Mobile DevTeam
Super Security & Antivirus	Innovative & Creative Apps
SVS Antivirus Security Scanner	Mohammad Ashraf Hossain
syncNscan - Security/Antivirus	syncNscan Mobile Security
Tablet AntiVirus Security FREE	AVG Mobile
Tablet AntiVirus Security PRO	AVG Mobile
Test Virus	Itus Mobile Security
Top 10 Mobile Antivirus	mzpassiona
Total Antivirus Defender FREE	Security Defend
ULTIMATE U ANTIVIRUS	MrPaul (Pavel Gutsalov)
VG ??? Web SDK	Infracore Technology, Inc
Video antivirus review	PashaYakushev
Virus Cleaner AntiVirus Prank	Technologizer
Virus Cleaner antivirus(Prank)	Alieman studio
Virus Guard (AntiVirus)	Mob&Me
Virus scan (Antivirus 2015)	Viking Mobile Inc
Virus Scan (Antivirus)	pablosoftware
Virus Scan (Antivirus)	Wequees
Virus Scan(Antivirus)	MoneyLand
VIRUSfighter Antivirus FREE	SPAMfighter aps
VIRUSfighter Antivirus PRO	SPAMfighter aps
White-Gate Antivirus	White Gate
xCore Antivirus Free	xCore LLC
XRIME Mobile Antivirus	XRIME Mobile
Zoner AntiVirus - Tablet	ZONER, Inc.
Zoner AntiVirus	ZONER, Inc.
Zoner AntiVirus Test	ZONER, Inc.
Zoon Mobile Antivirus	Zoon Developers
Zoon Mobile Antivirus Free	Zoon Developers
Zoon Tablet Antivirus Free	Zoon Developers

Appendices

Of the 67 Antivirus apps the 64 free apps were downloaded and prepared for analysis. The app name, package name, developer, rating, number of downloads and size were recorded.

Table A-8 Antivirus apps in 2015 and their properties

app name	Package name	Developer	Rating	Downloads	Size (KB)
360_security_-_antivirus_boost	com.qihoo.security.apk	qihoo 360	4.5	3624762	9681
Advanced_Task_Manager_-_Boost	mobi.infolife.taskmanager.apk	INFOLIFE LLC	4.5	313171	3736
ALYac_Android	com.estsoft.alvac.apk	ESTsoft Corp	4.4	49704	18123
AMC_Security_-_Clean_n_Booster	com.iobit.mobilecare-40601.apk	Iobit Mobile Security	4.5	204349	8602
Android_Optimizer	com.teebik.mobilesecurity-402.apk	Teebik Apps	4.1	9863	4261
Anti_Virus_Android	com.viruskiller.antivirusandroid545-1.apk	PiggiesMaz	4	3615	6245
Antivirus Pro for Android	£3.99	Android_Antivirus	4.2	1192	
Anti-Virus_Android	com.androhelm.antivirus.free-75.apk	AndroHelm Antivirus	4.1	4181	4037
Antivirus_Android(androhelm)	com.androhelm.antivirus.free2-32.apk	AndroHelm Antivirus	4.2	10400	4708
Antivirus_Booster_&_Cleaner	com.psafe.msuite-30.apk	Psafe Technologies S.A.	4.5	1056261	7949
Antivirus_Complete_Protection	com.antivirus_virusscan-46.apk	Sagamore	4.1	1961	1572
Antivirus_for_Android(A.A)	and.anti-8.apk	Android Antivirus	4.1	48992	243
Antivirus_for_Android(dala)	com.antivirusforandroid-11.apk	Dala Apps	4	7629	1242
Antivirus_for_Android(itus)	com.androidantivirus-20.apk	Itus Mobile Security	4.2	11147	1766
Antivirus_for_Android(lab4)	com.lab4apps.antivirus-15.apk	Android Antivirus	4.2	49349	770
Antivirus_for_Android(moobila)	com.moobila.appriva.av.apk	Moobila Corporation	4.2	12886	3602
Antivirus_for_Android_FREE	com.proj.mysafetyscanvirus-2.apk	XipiO	4.8	116	2576
antivirus_for_android_TM	com.androidsantivirus-57.apk	CTG	4	16535	5043
Antivirus_for_androids_2015	com.linchpin.utility.appslocker-11.apk	Free mobile speed booster antivirus	4.1	3759	1835
antivirus_free-mobile_security	com.zrgiu.antivirus-338.apk	clean master nq creative apps	4.3	330131	2035

Appendices

app name	Package name	Developer	Rating	Downloads	Size (KB)
AntiVirus_n_Anti-Adware	com.secure.privacyshield-8309091.apk	SeCore Mobile Security	4.2	3064	1158
Antivirus_n_Mobile_Security(trustgo)	com.trustgo.mobile.security-51.apk	TrustGo Inc	4.6	263521	4286
antivirus_n_mobile_security_(quickheal)	com.quickheal.platform-107.apk	quick heal technologies	4.4	89709	12667
antivirus_Security_-_FREE	com.antivirus.apk	AVG Mobile	4.4	2855255	13354
Armor for Android	£19.62	Armor_for_Android	4.6	1646	
avira_antivirus_security	com.avira.android-3280.apk	Avira	4.3	156479	7607
bitdefender_antivirus_free	com.bitdefender.antivirus-2019194.apk	Bitdefender	4.3	24852	1084
Bkav_Security_-_Antivirus_Free	bms.main-140.apk	Bkav Corporation	4.3	32888	1786
BluePoint_Antivirus_Free	bluepointfree.ad-42.apk	BluePoint Security	4.3	1638	3367
clean_master_(speed_booster)	com.cleanmaster.mguard.apk	Cheetah mobile	4.7	20176504	12362
Cleaner_Master_Antivirus_Plus	com.cm.virus-2.apk	IFSC Code	4.3	4673	2211
cm_security_antivirus_applock	com.cleanmaster.security.apk	Cheetah mobile	4.7	7690699	7902
Comodo_Security_&_Antivirus	com.comodo.pimsecure-293670.apk	Comodo Security Solutions	4.4	16477	16143
CY_Security_Antivirus_Cleaner	com.cyou.security-20602028.apk	CY Security	4.5	28759	6067
Dr.Web_v.9_Anti-virus	com.drweb.pro.apk	Doctor Web ltd	4.4	336750	7186
Dr.Web_v.9_Anti-virus_Light	com.drweb.apk	Doctor Web ltd	4.5	728598	2857
DU_Speed_Booster_Cache_Cleaner	com.dianxinos.optimizer.duplay.apk	du apps	4.5	5515835	8485
eScan_-_Mobile_Antivirus	com.escan.main-10.apk	MicroWorld Technologies	4.2	1098	6128
Free_Antivirus_2014+_Security	antivirus.free-16.apk	apps for life	4.1	30995	2624
free_antivirus_2015_security	com.ctool.antivirus-4.apk	Antivirus pro	4.2	4603	1637
Free_Antivirus_and_Security(panda)	com.pandasecurity.pandaav-17.apk	Panda Security	4.2	7367	3761
Free_Antivirus_and_Security(sophos)	com.sophos.smsec-1433.apk	Sophos Limited	4.3	4813	11306

Appendices

app name	Package name	Developer	Rating	Downloads	Size (KB)
Free_Antivirus_Pro_2014	at.ncn.freeantiviruspro2014-10.apk	NCN-NetConsulting	4	2551	2189
Free_Security_n_Antivirus	com.trustlook.antivirus-82.apk	Ges M.b.H trustlook mobile security	4.3	97701	10771
Free_virus_scan_(Antivirus)	com.antivirusfree-11.apk	Complete mobile security Antivirus Free	4.2	481	2838
GuardX_Antivirus	org.qstar.guardx-14.apk	Anti virus Qstar	4.3	3715	1277
Hornet_AntiVirus_Free	de.security.mobile-95.apk	Hornet Mobile Security	4.3	9161	1072
kaspersky_internet_security	com.kms.free-64.apk	Kaspersky Lab	4.6	743215	27393
LINE_Antivirus	jp.naver.lineantivirus.android-1037.apk	LINE Corporation	4.2	54147	4789
lookout_security_n_antivirus	com.lookout.apk	lookout mobile security	4.5	773412	7983
Mobile_Security_and_Antivirus(bullguard)	com.bullguard.mobile.mobilesecurity-17.apk	BullGuard	3.9	775	5989
Mobile_Security_n_Antivirus(avast)	com.avast.android.mobilesecurity-7875.apk	Avast software	4.4	2260490	9856
Mobile_Security_n_Antivirus(bitdefender)	com.bitdefender.security-2030683.apk	Bitdefender	4.4	36523	6494
Mobile_Security_n_Antivirus(eset)	com.eset.ems2.gp.apk	ESET	4.6	228438	9152
norton_security_and_antivirus	com.sysmantec.mobilesecurity-2245.apk	norton mobile	4.4	417950	7927
NQ_Mobile_Security_&_Antivirus	com.nqmobile.antivirus20-514.apk	NQ Mobile security	4.4	276352	5449
Secure_Antivirus	com.pleap.av.app-249.apk	Secure Antivirus	3.9	2129	15789
Security_-_Free	com.webroot.security-6657.apk	Webroot Inc	4.3	11062	2854
Security_n_Antivirus_-_FREE(mcafee)	com.wsandroid.suite-431001.apk	McAfee Mobile Security	4.3	229368	8410

Appendices

app name	Package name	Developer	Rating	Downloads	Size (KB)
Tablet Antivirus Security PRO	£8.99	AVG_Mobile	4.4	1264	
tablet_antivirus_security_FREE	com.antivirus.tablet-212903.apk	AVG Mobile	4.2	139438	13354
Virus Removal_for_Android	bluetooths.antivirus.super02-2.apk	jit182da1	4.4	496	11366
virus_scan_(antivirus)	com.pablosoftware.virusscan-34.apk	pablosoftware	3.9	7676	2949
VIRUSfighter_Antivirus_FREE	com.virusfighter.android-30.apk	SPAMfighter apps	4.1	9023	2517
White-Gate_Antivirus	org.whitegate.av-53.apk	White Gate	4.4	2775	987
Zoner_AntiVirus	com.zoner.android.antivirus-53.apk	Zoner Inc	4.4	43281	1444
Zoner_AntiVirus_-_Tablet	com.zoner.android.antivirus_tablet-18.apk	Zoner Inc	4.3	6282	1205

A summary of the number of permissions requested by each of the apps included in this study are shown in Figure 6-11 and the detailed tables of permissions requested are provided in Appendix A.

The permissions requested were reviewed to determine if any old permissions were being requested. Old permissions were those designated as no longer valid in this version of Android.

There were six old permissions that were being requested.

Table A-9 Apps that requested “old” permissions.

<i>app Name</i>	<i>Developer</i>	<i>ACCESS COARSE UPDATES</i>	<i>ACCESS LOCATION</i>	<i>ADD SYSTEM SERVICE</i>	<i>RAISED THREAD PRIORITY</i>	<i>READ OWNER DATA</i>	<i>READ SETTINGS</i>
Android Optimizer	Teebik Apps			Y			
Antivirus for Android TM	CTG	Y					
Antivirus & Mobile Security	TrustGo Inc	Y					
Antivirus Security - FREE	AVG Mobile	Y					
Bkav Security - Antivirus Free	Bkav Corporation					Y	
Free Security & Antivirus	Trustlook mobile security		Y				
Lookout Security & Antivirus	Lookout Mobile Security					Y	
Norton security and Antivirus	Norton mobile						Y
Secure Antivirus	Secure Antivirus	Y				Y	
Security & Antivirus - FREE	McAfee Mobile Security	Y			Y		
Tablet Antivirus Security FREE	AVG Mobile	Y					

The requesting of these non-valid permissions could be due to a variety of causes, these include (but are not limited to); backward compatibility, incomplete code review or no code review or updates. The lack of code review indicates that the Antivirus is not being updated and is not protecting the device against new malware.

Appendices

Of the 22 security apps in the marketplace in 2011, 7 had been updated and were available in 2015. Five of the developers from 2011 were still active as developers in 2015, the rest had either gone out of business or had been subsumed by other companies. The table below shows the apps available in 2011 to 2015, the developer name and the number of permissions requested in that year’s variant. Some of the apps names changed between 2011 and 2015, but their package name (installable component) remained constant with version variants.

Table A-10 Developer and their Antivirus apps available in 2011 and 2015

<i>Developer</i>	<i>2011 app name</i>	<i>2011</i>	<i>Developer</i>	<i>2015 app name</i>	<i>2015</i>
		<i>Permissions</i>			<i>Permissions</i>
AVFree	AV antivirusfree trial	27	AVG Mobile	antivirus Security - FREE	41
AVG Mobilation	AntiVirus Free AVG	27	AVG Mobile	tablet antivirus security FREE	42
Bluepoint security Inc	BluePoint Antivirus free	21	BluePoint Security	BluePoint Antivirus Free	29
Bluepoint security Inc	BluePoint Antivirus	20			
Doctor Web Ltd	DrWeb Android light	8	Doctor Web Ltd	Dr.Web v.9 Anti-virus	31
Doctor Web Ltd	DrWeb Antivirus light	8	Doctor Web Ltd	Dr.Web v.9 Anti-virus Light	13
Lookout Inc	Lookout Mobile Security	30	lookout mobile security	lookout security & antivirus	38

Appendices

Lookout Inc	Lookout Mobile Security trial	31	
NetQin Mobile Inc.	Nqmobile booster	15	NQ Mobile Security & Antivirus
NetQin Mobile Inc.	Nqmobile manager Trial	19	37

Note: NetQin Mobile Inc. name changed to NQ Mobile Security between 2011 and 2015, the company combined the booster and manager product into one security product. In 2011 Lookout Inc. produced a full and trial version of their Antivirus product. This was previously consolidated as in 2011. The developer AVFree was purchased by AVG.

A.2. Analysis of Children’s Apps Input Tables

The apps were selected using the default Google Ranking system, this is the order that the app is displayed to the user on the Play Store. The top 20 free apps were selected. The ranking order of the app and the number of downloads, user rating and permissions (Table A-11).

Table A-11 Ranking order and details of top 20 apps in the 0-5 age group by app name

Rank	app name	Package name	Developer	Rating	#Downloads
10	Barbie magical fashion	com.budgestudios.BarbieMagicalFashion-554.apk	Budge studios	3.9	121180
4	BBC Cbeebies storytime	air.uk.co.bbc.cbeebiesstorytime-2000038.apk	Media Applications BBC	3.9	4288
2	Cbeebies Playtime	uk.co.bbc.cbeebiesplaytime-41.apk	Media Applications BBC	4	19881
11	Disney princess palace pets	com.disneydigitalbooks.PalacePets_goo-20.apk	Disney studios	3.9	41608
20	disney_color_and_play	com.disneydigitalbooks.disneycolorandplay_goo.apk	Disney studios	3.7	9301
14	Kids baloon pop game	se.appfamily.balloonpopfree-73.apk	app Family	3.8	9925
17	kids_dinosaur	se.appfamily.dinoadventure.apk	app Family	3.6	16386
12	Lego duplo train	com.lego.duplo.trains-5.apk	Lego group	3.8	109676
3	Lego Juniors create and cruise	com.lego.bricksmore-18.apk	Lego group	3.9	289728

Appendices

Rank	app name	Package name	Developer	Rating	#Downloads
6	Lego juniors quest	com.lego.juniors.quest-14.apk	Lego group	3.8	279557
19	Lego Duplo_food	com.lego.duplo.food.apk	Lego group	3.7	62484
16	Letter_school_free	com.letterschool.lite.apk	Sanoma_media	3.8	2087
18	olafs_adventure	com.disney.digitalbooks.olafs.bestdayever_goo.apk	Disney studios	3.8	11258
8	Pancake tower	net.otouch.cake-7.apk	O!touch	3.9	28104
7	Peppa's activity maker	air.com.peppapig.activitymaker-1002007.apk	Entertainment One	3.1	3061
1	Peppa's Paint Box	air.com.peppapig.paintbox-1002002.apk	Entertainment One	3.6	16127
13	Sweet baby girl - dream house	air.com.tutotoons.app.babyhouse-2002003.apk	TutoTOONS	3.8	39715
9	Thomas and Friends go thomas	com.budgestudios.ThomasAndFriendsGoThomas-24.apk	Budge studios	4	33962
15	Toca kitchen	com.tocaboca.tocakitchen-103.apk	Toca Boca AB	4.1	117474
5	Toca Kitchen 2	com.tocaboca.tocakitchen2-104.apk	Toca Boca AB	4.4	50197

The 20 apps in this category were supplied by 7 app providers. The most popular apps were the ones supplied by the Lego Group with 4 apps. The Disney group was second with 3 apps.

Appendices

The 20 apps in the study for the 6 to 8 age group is shown in Table A-12. The table shows the package name, developer, user rating, user rating, number of downloads and number of permissions requested.

Table A-12 Age group 6-8 top 20 apps selected for the study.

Rank Order	app Name	Package Name	Developer	User Rating	Number of Downloads	Permissions
13	10monkeys multiplication	com.tenmonkeys.multiplication-3.apk	10monkeys.com	3.9	2376	2
14	Caillou house of puzzles	com.budgetstudios.caillouhouseofpuzzles-160.apk	Budge studios	3.6	13529	5
12	Chuggington chug patrol free	com.storytoys.Chuggington.Book1.Free.GooglePlay-16.apk	Story toys	3.8	7490	7
9	Disney color and play	com.disneydigitalbooks.disneycolorandplay_goo-1000408.apk	Disney worldwide	3.7	6234	7
7	Go CBBC	uk.co.bbc.cbbegocbbc.apk	Media Applications	3.8	2364	10
5	Gruffalo & the vanishing wood	com.magiclightpictures.vanishingwood-3.apk	Magic light pictures	3.5	68	6
15	Hot wheels showdown	com.mattel.HWShowdown-8.apk	Mattel	3.8	38632	8
6	king of math junior free	com.oddrobo.komjfree-3.apk	Oddrobo Software	4	3807	0
2	Lego city my city	com.lego.city.my_city-22009.apk	Lego Group	3.9	637612	6
3	Lego friends art maker	com.lego.friends.artmaker-29.apk	Lego Group	4.2	19280	5
4	Lego technic race	com.lego.technic.race-1.apk	Lego Group	3.8	114084	5
10	Miles from tomorrow land	com.disneydigitalbooks.milesfromtomorrowlandmissions_goo-20.apk	Disney worldwide	3.8	5134	8

Appendices

<i>Rank Order</i>	<i>app Name</i>	<i>Package Name</i>	<i>Developer</i>	<i>User Rating</i>	<i>Number of Downloads</i>	<i>Permissions</i>
1	Planes fire and rescue	com.disneydigitalbooks.planesfireescue_goo-6.apk	Disney	4.1	2092	9
8	Strawberry sweet shop	com.budgetstudios.StrawberryShortcakeSweetShop-250.apk	Budge studios	3.7	111398	6
11	the 7d mine train	com.DisneyDigitalBooks.SevenDMineTrain-2.com	Disney worldwide	4.2	37621	5
15	The_Smurfs_baker	com.budgetstudios.SmurfsBakery.apk	Budge_studios			7
17	Crayola_nail_party	com.budgetstudios.CrayolaNailParty.apk	Budge_studios	3.6	39577	7
18	Starfall_free	air.com.starfall.more.apk	Starfall_education	4	6810	3
19	Endless_number	com.originatorkids.EndlessNumbers.apk	Originator_inc	4.2	771	4
20	Lego_elves	com.lego.elves.unithemagic.apk	Lego Group	3.9	6337	5

The 20 apps in this category were supplied by 11 app providers. The most popular apps were the ones supplied by the Disney, Lego and Budge, who supplied 4 each.

The 20 apps in the study for the over 9’s age group is shown in Table A-13. The table shows the package name, developer, user rating, number of downloads and number of permissions requested.

Appendices

Table A-13 Apps selected for this study with number of permissions requested.

Rank Order	app Name	Package Name	Developer	User Rating	Number of Downloads	Permissions
5	Angry birds pop bubble shooter	com.rovio.Abstellapop.apk	Rovio entertainment	4.2	103486	10
2	Big hero 6 bot fight	com.disney.bighero6botfight_goo-267.apk	Disney	4.1	70507	8
11	Burger shop	com.gobbit.burgherhop-15.apk	Gobbit games	4.1	194563	2
6	club penguin	com.disney.cpcompanion_goo-15460.apk	Disney	3.9	233363	7
15	Cookies maker salon	com.libiitech.cookiesmakersalon-1.apk	Libii	4.1	1653	7
10	Crayola jewellery party	com.budgetstudios.CrayolaJewelryParty-6.apk	Budge studios	3.8	5079	7
1	Despicable me	com.gameloft.android.ANMP.GloftDMHM-30120.apk	Gameloft	4.4	7008800	10
8	Frozen free fall	com.disney.frozensaga_goo-101.apk	Disney	4.3	1193824	7
9	Littlest pet shop	com.gameloft.android.ANMP.GloftPEHM-22623.apk	Gameloft	4.2	517918	10
14	Make-up me superstar	com.libii.makeupsuperstar-1.apk	Libii	4.1	16460	5
12	Mini pets	com.miniclip.animalshelter-29.apk	Miniclip.com	4.2	77011	12
7	My little pony	com.gameloft.android.ANMP.GloftPOHM.apk	Gameloft	4.3	738663	11
3	School of dragons	com.KnowledgeAdventure.SchoolOfDragons-25.apk	Knowledge adventure	4	242226	7
13	Star chart	com.escapistgames.starchart.apk	Escapist games	4.2	89180	9
4	Where's my mickey? Free	com.disney.wheresmymickeyfree_goo-4.apk	Disney	3.9	218073	4
16	Masha_search_and_rescue	com.appsministry.mashagame.apk	Apps_ministry	4	172211	9

Appendices

17	Littlest_pet_shop	com.gameloft.android.anmp.gloftpehm.apk	Gameloft	4.2	537436	10
18	Angry_birds_transformer	com.rovio.angrybirdstransformers.apk	Rovio_entertainment	4.3	977789	8
19	Frozen_free_fall	com.disney.frozensaga_goo.apk	Disney	4.3	1215705	7
20	Bad_piggies	com.rovio.badpiggies.apk	Rovio_entertainment	4.3	863707	7

apps in this category were supplied by 10 app providers. The most popular apps being the ones supplied by Disney (5) and Gameloft (4).

A.3. Protection Normal Input Table

To minimise the number of permissions that the user consents to at app download and install, Google introduced the designation “protection_normal” (“Protection Normal,” 2017). This designation applies to permissions which Google has determined that there's “no great risk to the user's privacy or security in letting apps have those permissions”. If the app declares in the manifest file that it needs a normal permission, then the system automatically provides the app with that permission at install time. The user is not prompted at install time to agree to these permissions and is not able to revoke any of them.

Table A-14 Permissions classified as Protection_Normal in Android V6.0

Permission	Activity	Rating
ACCESS_LOCATION_EXTRA_COMMANDS	Track	Low
ACCESS_NETWORK_STATE	Obtain status	Low
ACCESS_NOTIFICATION_POLICY	Review status	Low
ACCESS_WIFI_STATE	Obtain status	Low
BLUETOOTH	Control access	Medium
BLUETOOTH_ADMIN	Control access	Medium
BROADCAST_STICKY	Change status	Low
CHANGE_NETWORK_STATE	Control access	Medium
CHANGE_WIFI_MULTICAST_STATE	Control access	Medium
CHANGE_WIFI_STATE	Control access	Medium
DISABLE_KEYGUARD	Change status	Low
EXPAND_STATUS_BAR	Change status	Low
GET_PACKAGE_SIZE	Obtain status	Low
INSTALL_SHORTCUT	Change status	Low
INTERNET	Control access	Medium
KILL_BACKGROUND_PROCESSES	Change status	Low
MODIFY_AUDIO_SETTINGS	Change status	Low

NFC	Control access including payment details	Medium
READ_SYNC_SETTINGS	Obtain status	Low
READ_SYNC_STATS	Obtain status	Low
RECEIVE_BOOT_COMPLETED	Obtain status	Low
REORDER_TASKS	Change status	Medium
REQUEST_IGNORE_BATTERY_OPTIMIZATIONS	Obtain status	Low
REQUEST_INSTALL_PACKAGES	Obtain status	Low
SET_ALARM	Change status	Low
SET_TIME_ZONE	Change status	Low
SET_WALLPAPER	Change status	Low
SET_WALLPAPER_HINTS	Change status	Low
TRANSMIT_IR	Change status	Low
UNINSTALL_SHORTCUT	Change status	Low
USE_FINGERPRINT	Change status	Low
VIBRATE	Obtain status	Low
WAKE_LOCK	Obtain status	Low
WRITE_SYNC_SETTINGS	Change status	Low

Unlike the permissions request made by apps, these permissions are implicitly accepted as part of using an Android handset, the app permissions are requested for acceptance as normal.

A.4. Detailed Permissions of Antivirus apps in the study

There are 154 permissions defined for Lollipop, Version 5 of Android (V5), which was the latest version the time of this research. Version 5 was available in two releases, version 5.0 which had a market share of 13.1% and version 5.1 which had a market share of 21.9%. The preliminary analysis showed that there were also fifteen (15) permissions requested which were not valid in this

Appendices

version. These non-valid permissions are included here and are highlighted in the tables.

The detailed permissions requested are shown in tables Table A-15 to Table A-19. There were sixty-seven Apps in the study, and they are displayed below in groups of 12 to 15 apps. The number of Apps in each set are;

- Set 1 - 12 Apps (Table A-15)
- Set 2 - 14 Apps (Table A-16)
- Set 3 - 15 Apps (Table A-17)
- Set 4 - 13 Apps (Table A-18)
- Set 5 - 13 Apps (Table A-19)

Appendices

Table A-15 Set one consisting of 12 Apps

Company	Qihoo 360	INFOLIFE LLC	ES soft Corp	Iobit Mobile Security	Teebik Apps	PiggiesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Safe Technologies S.A.	Sagamore	Android Antivirus
Permission (154 valid of 169 requested)	38	7	22	38	35	9	0	21	26	49	8	5
ACCESS_CELL_ID												
ACCESS_CHECKIN_PROPERTIES												
ACCESS_CHECKING_PROPERTIES												
ACCESS_COARSE_LOCATION	y			y	y	y		y		y		
ACCESS_COARSE_UPDATES												
ACCESS_FINE_LOCATION	y			y	y	y		y		y		
ACCESS_GPS												
ACCESS_LOCATION												
ACCESS_LOCATION_EXTRA_COMMANDS												
ACCESS_MOCK_LOCATION								y				
ACCESS_NETWORK_STATE	y	y	y	y	y	y		y		y	y	y
ACCESS_SURFACE_FLINGER												
ACCESS_WIFISTATE	y		y	y	y					y	y	
ACCOUNT_MANAGER												
ADD_SYSTEM_SERVICE					y							
ADD_VOICEMAIL												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
AUTHENTICATE_ACCOUNTS												
BATTERY_STATS				y						y		
BIND_ACCESSIBILITY_SERVICE												
BIND_APPWIDGET												
BIND_DEVICE_ADMIN												
BIND_DREAM_SERVICE												
BIND_INPUT_METHOD												
BIND_NFC_SERVICE												
BIND_NOTIFICATION_LISTENER_SERVICE												
BIND_PRINT_SERVICE												
BIND_REMOTEVIEWS												
BIND_TEXT_SERVICE												
BIND_TV_INPUT												
BIND_VOICE_INTERACTION												
BIND_VPN_SERVICE												
BIND_WALLPAPER												
BLUETOOTH	y			y	y					y		
BLUETOOTH_ADMIN	y			y	y					y		
BLUETOOTH_PRIVILEGED												
BODY_SENSORS												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESTsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
BRICK												
BROADCAST_PACKAGE_REMOVED												
BROADCAST_SMS												
BROADCAST_STICKY			y									
BROADCAST_WAP_PUSH												
CALL_PHONE	y		y	y	y			y		y		
CALL_PRIVILEGED												
CAMERA	y				y	y				y		
CAPTURE_AUDIO_OUTPUT												
CAPTURE_SECURE_VIDEO_OUTPUT												
CAPTURE_VIDEO_OUTPUT												
CD2.MESSAGE												
CD2.MESSAGE.RECEIVE												
CHANGE_COMPONENT_ENABLED_STATE												
CHANGE_CONFIGURATION				y	y							
CHANGE_NETWORK_STATE	y		y	y	y					y		
CHANGE_WIFI_MULTICAST_STATE												
CHANGE_WIFI_STATE	y		y	y	y					y		
CLEAR_APP_CACHE	y		y	y	y				y	y		
CLEAR_APP_USER_DATA												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
CONTROL_LOCATION_UPDATES												
DELETE_CACHE_FILES												
DELETE_PACKAGES												
DEVICE_POWER												
DIAGNOSTIC												
DISABLE_KEYGUARD										y		
DUMP												
EXPAND_STATUS_BAR	y									y		
FACTORY_TEST												
FLAG_ACTIVITY_NEW_TASK												
FLASHLIGHT												
FORCE_BACK												
GET_ACCOUNTS	y		y	y		y			y	y		y
GET_PACKAGE_SIZE	y		y	y	y					y		
GET_TASKS	y	y	y	y	y				y	y		
GET_TOP_ACTIVITY_INFO												
GLOBAL_SEARCH												
HARDWARE_TEST												
INJECT_EVENTS												
INSTALL_LOCATION_PROVIDER												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ES!soft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
INSTALL_PACKAGES												
INSTALL_SHORTCUT												
INTERNAL_SYSTEM_WINDOW												
INTERNET	y	y	y	y	y	y	y	y		y	y	y
KILL_BACKGROUND_PROCESSES	y	y	y	y	y			y		y		
LOCATION												
LOCATION_HARDWARE												
MANAGE_ACCOUNTS	y							y				
MANAGE_APP_TOKENS												
MANAGE_DOCUMENTS												
MASTER_CLEAR												
MEDIA_CONTENT_CONTROL												
MODIFY_AUDIO_SETTINGS				y						y		
MODIFY_PHONE_STATE				y						y		
MOUNT_FORMAT_FILESYSTEMS												
MOUNT_MOUNT_FILESYSTEMS												
MOUNT_UNMOUNT_FILESYSTEMS					y							
NFC				y								
PERSISTENT_ACTIVITY												
PROCESS_OUTGOING_CALLS	y							y				y

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
RAISED_THREAD_PRIORITY												
READ_CALENDAR										y		
READ_CALL_LOG				y				y		y		
READ_CONTACTS	y		y	y	y			y		y		
READ_EXTERNAL_STORAGE			y	y				y			y	
READ_FRAME_BUFFER												
READ_HISTORY_BOOKMARKS												
READ_INPUT_STATE												
READ_LOGS	y		y		y					y		
READ_OWNER_DATA												
READ_PHONE_STATE	y		y	y	y	y		y		y		
READ_PROFILE												
READ_SETTINGS												
READ_SMS	y		y	y	y			y		y		
READ_SOCIAL_STREAM												
READ_SYNC_SETTINGS	y			y						y		
READ_SYNC_STATS												
READ_URI												
READ_USER_DICTIONARY												
READ_VOICEMAIL												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
REBOOT												
RECEIVE_BOOT_COMPLETED	y	y	y	y	y			y		y	y	
RECEIVE_MMS			y							y		
RECEIVE_SMS	y		y	y	y			y		y		y
RECEIVE_WAP_PUSH										y		
RECORD_AUDIO					y	y						
REORDER_TASKS				y								
RESTART_PACKAGES	y	y	y	y	y					y		
SEND_RESPOND_VIA_MESSAGE												
SEND_SMS	y				y			y		y		
SET_ACTIVITY_WATCHER												
SET_ALARM												
SET_ALWAYS_FINISH										y		
SET_ANIMATION_SCALE												
SET_DEBUG_APP												
SET_ORIENTATION												
SET_POINTER_SPEED												
SET_PREFERRED_APPLICATIONS												
SET_PROCESS_LIMIT												
SET_TIME												

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
SET_TIME_ZONE												
SET_WALLPAPER												
SET_WALLPAPER_HINTS												
SIGNAL_PERSISTENT_PROCESSES												
STATUS_BAR												
SUBSCRIBED_FEEDS_READ												
SUBSCRIBED_FEEDS_WRITE												
SYSTEM_ALERT_WINDOW	y			y	y					y		
TRANSMIT_IR												
UNINSTALL_SHORTCUT												
UPDATE_DEVICE_STATS												
USE_CREDENTIALS	y								y	y		
USE_SIP												
VIBRATE	y			y	y					y	y	
WAKE_LOCK	y	y			y			y		y	y	
WiFi_LOCK												
WRITE_APN_SETTINGS										y		
WRITE_CALENDAR										y		
WRITE_CALL_LOG	y			y				y		y		
WRITE_CONTACTS	y			y	y			y		y		

Appendices

Company	Qihoo 360	INFOLIFE LLC	ESIsoft Corp	Iobit Mobile Security	Teebik Apps	PiggesMaz	Android Antivirus	AndroHelm Antivirus	AndroHelm Antivirus	Psafe Technologies S.A.	Sagamore	Android Antivirus
WRITE_EXTERNAL_STORAGE	y		y	y	y	y		y		y	y	
WRITE_GSERVICES												
WRITE_HISTORY_BOOKMARKS												
WRITE_LOGS												
WRITE_OWNER_DATA					y							
WRITE_PROFILE												
WRITE_SECURE_SETTINGS										y		
WRITE_SETTINGS	y			y	y					y		
WRITE_SMS	y			y	y			y		y		y
WRITE_SOCIAL_STREAM												
WRITE_SYNC_SETTINGS	y			y						y		
WRITE_USER_DICTIONARY												
WRITE_VOICEMAIL												
2015	38	7	22	38	35	9	0	21	26	49	8	5

Table A-16 Set two consisting of 14 Apps

Company	Data Apps	Itus Mobile Security	Android Antiviruses	Moobila Corporation	XipIO	CTG	Free mobile speed booster antiviruss clean master	Free creative apps	SeCore Mobile Security	TrustGo Inc	quick heal technologies	AVG Mobile	Armor for Android	Avira
	5	7	4	4	4	26	8	6	4	30	48	41	0	29
Permission (154 valid of 169 requested)														
ACCESS_CELL_ID														
ACCESS_CHECKIN_PROPERTIES														
ACCESS_CHECKING_PROPERTIES														
ACCESS_COARSE_LOCATION						y				y	y	y		y
ACCESS_COARSE_UPDATES						y				y	y	y		
ACCESS_FINE_LOCATION						y				y	y	y		y
ACCESS_GPS														
ACCESS_LOCATION														
ACCESS_LOCATION_EXTRA_COMMANDS														
ACCESS_MOCK_LOCATION														
ACCESS_NETWORK_STATE	y	y	y	y	y	y	y	y	y	y	y	y		y
ACCESS_SURFACE_FLINGER														
ACCESS_WIFI_STATE	y					y	y	y		y	y	y		y
ACCOUNT_MANAGER														
ADD_SYSTEM_SERVICE														
ADD_VOICEMAIL														

Appendices

Company	Dala Apps	Itus Mobile Security	Android Antiviruses	Moobla Corporation	XipLO	CTG	Free mobile speed booster antivirus clean master	mq creative apps	SeCore Mobile Security	TrustGo Inc	quick heal technologies	AVG Mobile	Armor for Android	Avira
WRITE_VOICEMAIL														
	5	7	4	4	4	26	8	6	4	30	48	41	0	29
2015														

Table A-17 Set three consisting of 15 Apps

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
Permission (154 valid of 169 requested)	8	32	29	22	4	37	35	15	31	13	43	40	12	7	14
ACCESS_CELL_ID															
ACCESS_CHECKIN_PROPERTIES															
ACCESS_CHECKING_PROPERTIES															
ACCESS_COARSE_LOCATION		y	y		y	y	y		y						
ACCESS_COARSE_UPDATES															
ACCESS_FINE_LOCATION		y	y		y	y	y		y		y	y		y	
ACCESS_GPS															
ACCESS_LOCATION															
ACCESS_LOCATION_EXTRA_COMMANDS															
ACCESS_MOCK_LOCATION							y					y			
ACCESS_NETWORK_STATE	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
ACCESS_SURFACE_FLINGER															
ACCESS_WIFI_STATE	y	y	y	y		y	y	y	y	y	y	y	y	y	y
ACCOUNT_MANAGER															
ADD_SYSTEM_SERVICE															
ADD_VOICEMAIL															
AUTHENTICATE_ACCOUNTS															

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
BATTERY_STATS				y							y				
BIND_ACCESSIBILITY_SERVICE		y													
BIND_APPWIDGET															
BIND_DEVICE_ADMIN											y	y			y
BIND_DREAM_SERVICE															
BIND_INPUT_METHOD															
BIND_NFC_SERVICE															
BIND_NOTIFICATION_LISTENER_SERVICE															
BIND_PRINT_SERVICE															
BIND_REMOTEVIEWS															
BIND_TEXT_SERVICE															
BIND_TV_INPUT															
BIND_VOICE_INTERACTION															
BIND_VPN_SERVICE									y						
BIND_WALLPAPER															
BLUETOOTH			y			y					y	y			
BLUETOOTH_ADMIN			y			y					y	y			
BLUETOOTH_PRIVILEGED															
BODY_SENSORS															
BRICK															

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
BROADCAST_PACKAGE_REMOVED															
BROADCAST_SMS															
BROADCAST_STICKY				y							y				
BROADCAST_WAP_PUSH															
CALL_PHONE		y				y	y	y	y		y				
CALL_PRIVILEGED															
CAMERA		y	y	y		y	y				y	y			y
CAPTURE_AUDIO_OUTPUT															
CAPTURE_SECURE_VIDEO_OUTPUT															
CAPTURE_VIDEO_OUTPUT															
CD2.MESSAGE															
CD2.MESSAGE.RECEIVE															
CHANGE_COMPONENT_ENABLED_STATE															
CHANGE_CONFIGURATION								y			y				
CHANGE_NETWORK_STATE				y		y	y		y		y	y			
CHANGE_WIFI_MULTICAST_STATE											y				
CHANGE_WIFI_STATE			y	y		y	y	y	y		y	y			
CLEAR_APP_CACHE				y		y	y	y			y				
CLEAR_APP_USER_DATA															
CONTROL_LOCATION_UPDATES															

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
DELETE_CACHE_FILES											y				
DELETE_PACKAGES			y									y			
DEVICE_POWER															
DIAGNOSTIC															
DISABLE_KEYGUARD															
DUMP															
EXPAND_STATUS_BAR				y											
FACTORY_TEST															
FLAG_ACTIVITY_NEW_TASK															
FLASHLIGHT															
FORCE_BACK															
GET_ACCOUNTS		y		y			y		y		y			y	
GET_PACKAGE_SIZE			y	y			y	y	y		y		y		
GET_TASKS		y	y	y			y	y	y		y		y		
GET_TOP_ACTIVITY_INFO															
GLOBAL_SEARCH															
HARDWARE_TEST															
INJECT_EVENTS															
INSTALL_LOCATION_PROVIDER															
INSTALL_PACKAGES			y												

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISc Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
INSTALL_SHORTCUT															
INTERNAL_SYSTEM_WINDOW															
INTERNET	y	y	y	y	y	y	y	y	y	y	y	y	y	y	y
KILL_BACKGROUND_PROCESSES		y	y				y	y	y	y	y	y	y		y
LOCATION															
LOCATION_HARDWARE															
MANAGE_ACCOUNTS									y						
MANAGE_APP_TOKENS															
MANAGE_DOCUMENTS															
MASTER_CLEAR															
MEDIA_CONTENT_CONTROL															
MODIFY_AUDIO_SETTINGS		y					y				y	y			
MODIFY_PHONE_STATE									y		y	y			
MOUNT_FORMAT_FILESYSTEMS			y												
MOUNT_MOUNT_FILESYSTEMS															
MOUNT_UNMOUNT_FILESYSTEMS			y	y			y	y			y				
NFC															
PERSISTENT_ACTIVITY															
PROCESS_OUTGOING_CALLS							y				y				
RAISED_THREAD_PRIORITY															

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
READ_CALENDAR							y					y			
READ_CALL_LOG		y		y					y			y			
READ_CONTACTS		y		y			y		y		y	y			
READ_EXTERNAL_STORAGE	y							y					y		
READ_FRAME_BUFFER															
READ_HISTORY_BOOKMARKS															
READ_INPUT_STATE															
READ_LOGS			y	y			y	y	y		y	y			y
READ_OWNER_DATA		y													
READ_PHONE_STATE	y	y	y	y			y	y	y		y	y	y	y	y
READ_PROFILE															
READ_SETTINGS															
READ_SMS		y		y			y		y		y	y			
READ_SOCIAL_STREAM															
READ_SYNC_SETTINGS											y				
READ_SYNC_STATS											y				
READ_URI															
READ_USER_DICTIONARY															
READ_VOICEMAIL															
REBOOT															y

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security	
RECEIVE_BOOT_COMPLETED	y	y	y				y		y		y	y	y		y	
RECEIVE_MMS		y														
RECEIVE_SMS		y					y		y		y	y				
RECEIVE_WAP_PUSH		y														
RECORD_AUDIO																
REORDER_TASKS		y					y									
RESTART_PACKAGES		y	y	y					y		y	y				
SEND_RESPOND_VIA_MESSAGE																
SEND_SMS		y					y		y			y				
SET_ACTIVITY_WATCHER																
SET_ALARM																
SET_ALWAYS_FINISH																
SET_ANIMATION_SCALE																
SET_DEBUG_APP																
SET_ORIENTATION																
SET_POINTER_SPEED																
SET_PREFERRED_APPLICATIONS																
SET_PROCESS_LIMIT																
SET_TIME																
SET_TIME_ZONE																

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security
SET_WALLPAPER												y			
SET_WALLPAPER_HINTS															
SIGNAL_PERSISTENT_PROCESSES															
STATUS_BAR															
SUBSCRIBED_FEEDS_READ												y			
SUBSCRIBED_FEEDS_WRITE															
SYSTEM_ALERT_WINDOW		y	y	y		y	y		y		y				
TRANSMIT_IR															
UNINSTALL_SHORTCUT															
UPDATE_DEVICE_STATS			y												
USE_CREDENTIALS							y								
USE_SIP															
VIBRATE	y	y	y	y		y	y		y		y	y	y		
WAKE_LOCK		y	y	y		y	y				y	y	y		y
WIFI_LOCK															
WRITE_APN_SETTINGS			y								y				
WRITE_CALENDAR							y		y			y			
WRITE_CALL_LOG		y				y			y			y			
WRITE_CONTACTS		y				y	y		y		y	y			
WRITE_EXTERNAL_STORAGE	y	y	y	y		y	y	y	y	y	y	y	y	y	y

Appendices

Company	Bit-defender	Bkav Corp	BluePoint Security	Cheetah mobile	IISC Code	Cheetah mobile	Comodo Security Solutions	CY Security	Doctor Web ltd	Doctor Web ltd	du apps	MicroWorld Technologies	apps for life	Antivirus pro	Panda Security	
WRITE_GSERVICES																
WRITE_HISTORY_BOOKMARKS																
WRITE_LOGS																
WRITE_OWNER_DATA			y													
WRITE_PROFILE																
WRITE_SECURE_SETTINGS			y									y				
WRITE_SETTINGS		y	y	y		y	y				y	y		y		y
WRITE_SMS		y				y	y		y		y	y				
WRITE_SOCIAL_STREAM																
WRITE_SYNC_SETTINGS											y					
WRITE_USER_DICTIONARY																
WRITE_VOICEMAIL																
2015	8	32	29	22	4	37	35	15	31	13	43	40	12	7	14	

Appendices

Table A-18 Set four consisting of 13 Apps

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ESET
Permission (154 valid of 169 requested)	23	7	30	8	4	0	41	18	38	31	29	31	34
ACCESS_CELL_ID													
ACCESS_CHECKIN_PROPERTIES													
ACCESS_CHECKING_PROPERTIES													
ACCESS_COARSE_LOCATION	y		y				y		y			y	
ACCESS_COARSE_UPDATES													
ACCESS_FINE_LOCATION	y		y				y		y			y	y
ACCESS_GPS													
ACCESS_LOCATION			y										
ACCESS_LOCATION_EXTRA_COMMANDS							y					y	
ACCESS_MOCK_LOCATION							y						
ACCESS_NETWORK_STATE	y	y	y	y	y		y	y	y	y	y	y	y
ACCESS_SURFACE_FLINGER													
ACCESS_WIFI_STATE	y		y	y			y			y	y	y	y
ACCOUNT_MANAGER													
ADD_SYSTEM_SERVICE													
ADD_VOICEMAIL													
AUTHENTICATE_ACCOUNTS										y			

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ESET
BATTERY_STATS							y				y		
BIND_ACCESSIBILITY_SERVICE													
BIND_APPWIDGET													
BIND_DEVICE_ADMIN			y						y	y		y	
BIND_DREAM_SERVICE													
BIND_INPUT_METHOD													
BIND_NFC_SERVICE													
BIND_NOTIFICATION_LISTENER_SERVICE													
BIND_PRINT_SERVICE													
BIND_REMOTEVIEWS													
BIND_TEXT_SERVICE													
BIND_TV_INPUT													
BIND_VOICE_INTERACTION													
BIND_VPN_SERVICE													
BIND_WALLPAPER													
BLUETOOTH												y	
BLUETOOTH_ADMIN													
BLUETOOTH_PRIVILEGED													
BODY_SENSORS													
BRICK													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ESET
BROADCAST_PACKAGE_REMOVED													
BROADCAST_SMS			y				y			y			
BROADCAST_STICKY								y					
BROADCAST_WAP_PUSH			y							y			
CALL_PHONE	y						y			y	y		y
CALL_PRIVILEGED													
CAMERA			y				y		y				y
CAPTURE_AUDIO_OUTPUT													
CAPTURE_SECURE_VIDEO_OUTPUT													
CAPTURE_VIDEO_OUTPUT													
CD2.MESSAGE													
CD2.MESSAGE.RECHIVE													
CHANGE_COMPONENT_ENABLED_STATE													
CHANGE_CONFIGURATION													
CHANGE_NETWORK_STATE							y	y	y			y	y
CHANGE_WIFI_MULTICAST_STATE													
CHANGE_WIFI_STATE							y	y				y	y
CLEAR_APP_CACHE								y					y
CLEAR_APP_USER_DATA													
CONTROL_LOCATION_UPDATES													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ISET
DELETE_CACHE_FILES													
DELETE_PACKAGES													
DEVICE_POWER													
DIAGNOSTIC													
DISABLE_KEYGUARD									y				y
DUMP													
EXPAND_STATUS_BAR							y						
FACTORY_TEST													
FLAG_ACTIVITY_NEW_TASK													
FLASHLIGHT									y				
FORCE_BACK													
GET_ACCOUNTS			y				y		y	y	y	y	y
GET_PACKAGE_SIZE			y					y			y		
GET_TASKS	y		y				y	y	y	y	y	y	y
GET_TOP_ACTIVITY_INFO													
GLOBAL_SEARCH													
HARDWARE_TEST													
INJECT_EVENTS													
INSTALL_LOCATION_PROVIDER													
INSTALL_PACKAGES													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ISET
INSTALL_SHORTCUT													
INTERNAL_SYSTEM_WINDOW													
INTERNET	y	y	y	y	y		y	y	y	y	y		y
KILL_BACKGROUND_PROCESSES	y		y				y	y	y		y		
LOCATION													
LOCATION_HARDWARE													
MANAGE_ACCOUNTS							y		y			y	y
MANAGE_APP_TOKENS													
MANAGE_DOCUMENTS													
MASTER_CLEAR													
MEDIA_CONTENT_CONTROL													
MODIFY_AUDIO_SETTINGS							y		y			y	
MODIFY_PHONE_STATE							y				y		
MOUNT_FORMAT_FILESYSTEMS													
MOUNT_MOUNT_FILESYSTEMS													
MOUNT_UNMOUNT_FILESYSTEMS													
NFC	y												
PERSISTENT_ACTIVITY									y				
PROCESS_OUTGOING_CALLS										y	y		y
RAISED_THREAD_PRIORITY													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ISET
READ_CALENDAR							y			y			y
READ_CALL_LOG	y		y						y	y	y		y
READ_CONTACTS	y		y				y		y		y	y	y
READ_EXTERNAL_STORAGE		y	y	y	y			y		y	y	y	
READ_FRAME_BUFFER													
READ_HISTORY_BOOKMARKS													
READ_INPUT_STATE													
READ_LOGS	y						y	y	y	y	y	y	y
READ_OWNER_DATA									y				
READ_PHONE_STATE	y				y		y	y	y	y	y	y	y
READ_PROFILE			y										y
READ_SETTINGS													
READ_SMS	y		y				y		y	y	y	y	y
READ_SOCIAL_STREAM													
READ_SYNC_SETTINGS							y		y				
READ_SYNC_STATS										y			
READ_URI													
READ_USER_DICTIONARY									y				
READ_VOICEMAIL													
REBOOT													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ISET
RECEIVE_BOOT_COMPLETED	y	y	y	y			y	y	y	y	y	y	y
RECEIVE_MMS	y										y		y
RECEIVE_SMS	y						y	y	y	y	y	y	y
RECEIVE_WAP_PUSH													
RECORD_AUDIO													
REORDER_TASKS													
RESTART_PACKAGES			y				y						
SEND_RESPOND_VIA_MESSAGE			y							y			
SEND_SMS	y						y		y		y	y	y
SET_ACTIVITY_WATCHER													
SET_ALARM													
SET_ALWAYS_FINISH													
SET_ANIMATION_SCALE													
SET_DEBUG_APP													
SET_ORIENTATION													
SET_POINTER_SPEED													
SET_PREFERRED_APPLICATIONS							y						
SET_PROCESS_LIMIT								y					
SET_TIME													
SET_TIME_ZONE													

Appendices

Company	Sophos Limited	NCN-NetConsulting Ges M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qstar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ISET
SET_WALLPAPER													
SET_WALLPAPER_HINTS													
SIGNAL_PERSISTENT_PROCESSES													
STATUS_BAR													
SUBSCRIBED_FEEDS_READ							y						
SUBSCRIBED_FEEDS_WRITE													
SYSTEM_ALERT_WINDOW			y									y	y
TRANSMIT_IR													
UNINSTALL_SHORTCUT													
UPDATE_DEVICE_STATS													
USE_CREDENTIALS			y								y		
USE_SIP													
VIBRATE		y	y	y			y	y	y		y		
WAKE_LOCK	y	y	y	y			y	y	y	y	y		y
WIFI_LOCK													
WRITE_APN_SETTINGS							y						
WRITE_CALENDAR							y		y	y			y
WRITE_CALL_LOG	y		y						y	y	y		y
WRITE_CONTACTS	y		y				y		y	y	y		y
WRITE_EXTERNAL_STORAGE	y	y	y	y			y	y	y	y	y		y

Appendices

Company	Sophos Limited	NCN-NetConsulting Cos M.b.H	Trustlook mobile security	Complete mobile security Antivirus Anti-virus	Qatar	Hornet Mobile Security	Kaspersky Lab	LINE Corporation	lookout mobile security	BullGuard	Avast software	Bitdefender	ESET
WRITE_GSERVICES													
WRITE_HISTORY_BOOKMARKS													
WRITE_LOGS													
WRITE_OWNER_DATA													
WRITE_PROFILE													y
WRITE_SECURE_SETTINGS									y				
WRITE_SETTINGS							y		y	y	y	y	
WRITE_SMS	y		y				y		y	y	y	y	y
WRITE_SOCIAL_STREAM													
WRITE_SYNC_SETTINGS							y		y	y			
WRITE_USER_DICTIONARY									y				
WRITE_VOICEMAIL													
2015	23	7	30	8	4	0	41	18	36	31	29	31	34

Appendices

Table A-19 Set five consisting of 13 Apps

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
Permission (154 valid of 169 requested)	39	37	26	32	46	0	42	5	4	8	10	14	14
ACCESS_CELL_ID													
ACCESS_CHECKIN_PROPERTIES													
ACCESS_CHECKING_PROPERTIES													
ACCESS_COARSE_LOCATION	y	y	y	y	y		y						
ACCESS_COARSE_UPDATES			y	y	y		y						
ACCESS_FINE_LOCATION	y		y	y	y		y					y	y
ACCESS_GIFS													
ACCESS_LOCATION													
ACCESS_LOCATION_EXTRA_COMMANDS	y												
ACCESS_MOCK_LOCATION													
ACCESS_NETWORK_STATE	y	y	y	y	y		y	y	y	y	y	y	y
ACCESS_SURFACE_FLINGER													
ACCESS_WIFI_STATE	y	y	y		y		y			y			
ACCOUNT_MANAGER													
ADD_SYSTEM_SERVICE													
ADD_VOICEMAIL													

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
AUTHENTICATE_ACCOUNTS				y									
BATTERY_STATS	y		y	y			y						
BIND_ACCESSIBILITY_SERVICE	y												
BIND_APPWIDGET													
BIND_DEVICE_ADMIN	y			y	y		y						
BIND_DREAM_SERVICE													
BIND_INPUT_METHOD													
BIND_NFC_SERVICE													
BIND_NOTIFICATION_LISTENER_SERVICE													
BIND_PRINT_SERVICE													
BIND_REMOTEVIEWS													
BIND_TEXT_SERVICE													
BIND_TV_INPUT													
BIND_VOICE_INTERACTION													
BIND_VPN_SERVICE													
BIND_WALLPAPER													
BLUETOOTH	y		y				y						
BLUETOOTH_ADMIN	y		y				y						
BLUETOOTH_PRIVILEGED													
BODY_SENSORS													

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
BRICK													
BROADCAST_PACKAGE_REMOVED													
BROADCAST_SMS					y								
BROADCAST_STICKY	y												
BROADCAST_WAP_PUSH													
CALL_PHONE	y	y		y	y		y					y	y
CALL_PRIVILEGED													
CAMERA	y	y	y		y		y						
CAPTURE_AUDIO_OUTPUT													
CAPTURE_SECURE_VIDEO_OUTPUT													
CAPTURE_VIDEO_OUTPUT													
CD2.MESSAGE													
CD2.MESSAGE.RECEIVE													
CHANGE_COMPONENT_ENABLED_STATE					y								
CHANGE_CONFIGURATION		y											
CHANGE_NETWORK_STATE		y	y										
CHANGE_WIFI_MULTICAST_STATE													
CHANGE_WIFI_STATE		y	y		y		y						
CLEAR_APP_CACHE		y											
CLEAR_APP_USER_DATA		y											

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
CONTROL_LOCATION_UPDATES					y								
DELETE_CACHE_FILES		y											
DELETE_PACKAGES					y								
DEVICE_POWER													
DIAGNOSTIC													
DISABLE_KEYGUARD					y		y						
DUMP													
EXPAND_STATUS_BAR													
FACTORY_TEST													
FLAG_ACTIVITY_NEW_TASK													
FLASHLIGHT		y											
FORCE_BACK													
GET_ACCOUNTS	y	y	y	y	y		y	y					
GET_PACKAGE_SIZE	y	y		y			y						
GET_TASKS	y	y	y	y	y		y				y		
GET_TOP_ACTIVITY_INFO													
GLOBAL_SEARCH													
HARDWARE_TEST													
INJECT_EVENTS													
INSTALL_LOCATION_PROVIDER													

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
INSTALL_PACKAGES													
INSTALL_SHORTCUT													
INTERNAL_SYSTEM_WINDOW													
INTERNET	y	y	y	y	y		y	y	y	y	y	y	y
KILL_BACKGROUND_PROCESSES		y	y	y	y		y						
LOCATION													
LOCATION_HARDWARE													
MANAGE_ACCOUNTS		y		y			y						
MANAGE_APP_TOKENS													
MANAGE_DOCUMENTS													
MASTER_CLEAR													
MEDIA_CONTENT_CONTROL													
MODIFY_AUDIO_SETTINGS	y						y						
MODIFY_PHONE_STATE				y	y								
MOUNT_FORMAT_FILESYSTEMS	y				y								
MOUNT_MOUNT_FILESYSTEMS													
MOUNT_UNMOUNT_FILESYSTEMS	y	y			y								
NFC													
PERSISTENT_ACTIVITY													
PROCESS_OUTGOING_CALLS		y			y							y	y

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
RAISED_THREAD_PRIORITY					y								
READ_CALENDAR	y			y									
READ_CALL_LOG	y				y								
READ_CONTACTS	y	y		y	y		y					y	y
READ_EXTERNAL_STORAGE					y				y				
READ_FRAME_BUFFER													
READ_HISTORY_BOOKMARKS													
READ_INPUT_STATE													
READ_LOGS	y	y	y	y	y		y						
READ_OWNER_DATA			y										
READ_PHONE_STATE	y	y	y	y	y		y			y	y	y	y
READ_PROFILE													
READ_SETTINGS	y												
READ_SMS	y	y		y	y		y						
READ_SOCIAL_STREAM													
READ_SYNC_SETTINGS			y				y						
READ_SYNC_STATS													
READ_URI													
READ_USER_DICTIONARY													
READ_VOICEMAIL													

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jit182da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
REBOOT													
RECEIVE_BOOT_COMPLETED	y	y	y	y	y		y			y		y	
RECEIVE_MMS				y	y							y	
RECEIVE_SMS	y	y		y	y		y					y	
RECEIVE_WAP_PUSH		y											
RECORD_AUDIO													
REORDER_TASKS					y								
RESTART_PACKAGES	y	y		y	y		y				y	y	
SEND_RESPOND_VIA_MESSAGE													
SEND_SMS	y	y		y	y		y					y	
SET_ACTIVITY_WATCHER													
SET_ALARM													
SET_ALWAYS_FINISH													
SET_ANIMATION_SCALE													
SET_DEBUG_APP													
SET_ORIENTATION													
SET_POINTER_SPEED													
SET_PREFERRED_APPLICATIONS													
SET_PROCESS_LIMIT													
SET_TIME													

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
SET_TIME_ZONE													
SET_WALLPAPER													
SET_WALLPAPER_HINTS													
SIGNAL_PERSISTENT_PROCESSES													
STATUS_BAR													
SUBSCRIBED_FEEDS_READ													
SUBSCRIBED_FEEDS_WRITE							y						
SYSTEM_ALERT_WINDOW	y	y	y	y	y		y						
TRANSMIT_IR													
UNINSTALL_SHORTCUT													
UPDATE_DEVICE_STATS													
USE_CREDENTIALS		y		y	y								
USE_SIP													
VIBRATE		y	y		y		y				y	y	y
WAKE_LOCK	y	y	y	y	y		y	y		y	y		
WIFI_LOCK													
WRITE_APN_SETTINGS													
WRITE_CALENDAR	y			y			y						
WRITE_CALL_LOG	y				y								
WRITE_CONTACTS	y	y		y	y		y						

Appendices

Company	Norton mobile	NQ Mobile security	Secure Antivirus	Webroot Inc	McAfee Mobile Security	AVG_Mobile	AVG Mobile	jitl82da1	Pablo software	SPAMfighter apps	White Gate	Zoner Inc	Zoner Inc
	y	y	y	y	y		y	y	y	y	y	y	
WRITE_EXTERNAL_STORAGE													
WRITE_GSERVICES													
WRITE_HISTORY_BOOKMARKS													
WRITE_LOGS													
WRITE_OWNER_DATA							y						
WRITE_PROFILE													
WRITE_SECURE_SETTINGS					y								
WRITE_SETTINGS	y	y	y		y		y						
WRITE_SMS	y	y			y		y						
WRITE_SOCIAL_STREAM													
WRITE_SYNC_SETTINGS			y	y	y		y						
WRITE_USER_DICTIONARY							y						
WRITE_VOICEMAIL													
2015	39	37	26	32	46	0	42	5	4	8	10	14	14

Appendix B Android Operating System

The Android operating system (OS) is a privilege-separated OS and by default applications (apps) or packages are not permitted to perform any operation that would impact another app, the operating system or the user, this is known as sandboxing. The sandbox creates an area for applications to run in and the access that is available to the installed app to a system resource. This access is controlled using a system of permissions. These permissions form part of the application sandbox and provide a modicum of basic security to the operating system. These permissions are defined and declared in an application's manifest file.

The source code of an Android app is written in Java and to run on an Android mobile the code is first compiled into Java Executable (.JAR), installed on the device and converted into Dalvik bytecode. Dalvik bytecode is compact and is suited for systems that are constrained by processor speed and memory, as is the case with mobiles which are limited by size and technology available in the small form factor¹⁵. Dalvik compiles the application to machine code at runtime, which increases power consumption as the app is compiled at every initiation.

Android Architecture

The Android operating system consists of five layers, these are:

¹⁵ Dalvik has since been superseded by Android Runtime (ART), which was first used in beta form in KitKat (Android V4.4).

Application.

Application Framework

Libraries

Android runtime

the Linux kernels

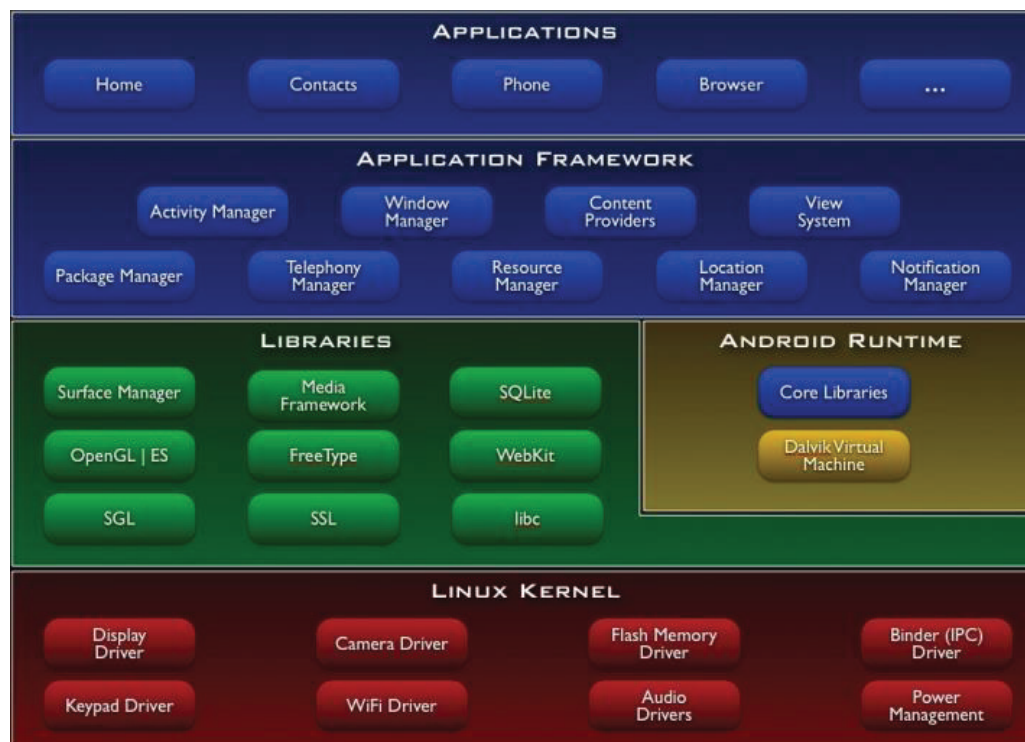


Figure B-0-1 Android Architecture (2011)

The application (app) is written in Java source code, compiled to Java bytecode and then assembled into Davlik. The framework services and libraries are mainly written in Java.

The applications and most framework code executes in the Android Runtime service and the app runs in the Davlik Virtual Machine (DVM). The native libraries, daemons and services are written in C or C++ and the system core libraries also reside in the Android Runtime layer.

The Linux kernel consists of the hardware drivers, networking, file system access and inter-process-communication.

Android Internals

The Android OS is classed as an open system. Internal layers and interconnectivity details are freely available. This type of information is not available on proprietary systems like Apple’s iOS. A pictorial overview of the internals by Constantine Shulupin is shown in Figure B-0-2.

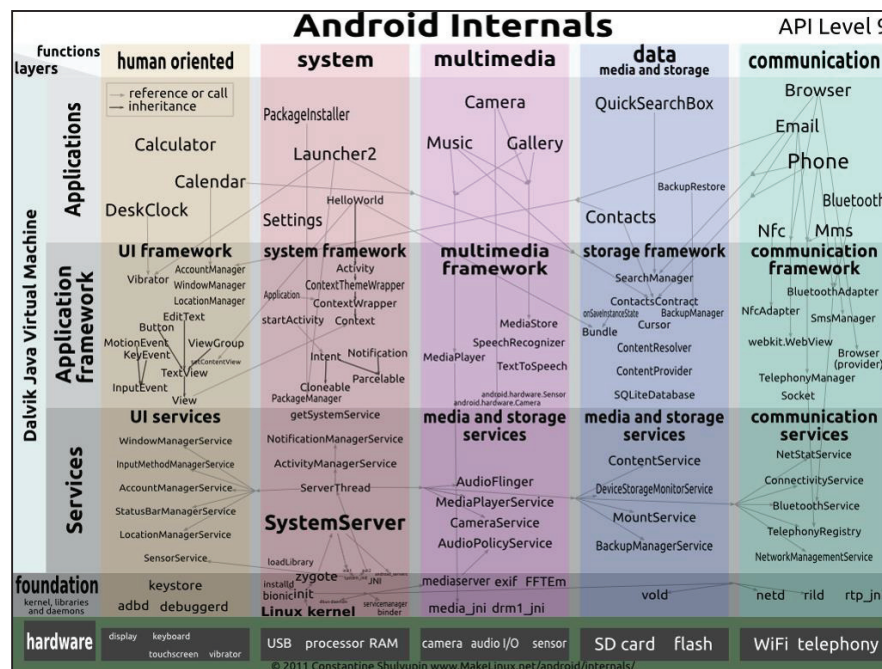


Figure B-0-2 Android Internals for API Level 9

The figure displays the interconnections between the functions in the application layer with the application framework and services layers. The figure also provides examples of the physical location or hardware that the various functions interact with.

The Android system provides individual permissions and permission groups that can be requested in the app Manifest file. The permission groups correspond to the system applications in the application level. Specification of the group enables the app to have control over the individual permissions within that group.

Android Vulnerabilities

The benefit of an open OS is the speed that developers can get apps to the marketplace. However, the openness also provides the internal working of Android open to attack. This simplifies the development of malware to attack internal systems by gaining access to the administrative side of the OS. Users can facilitate access to this vulnerability by “rooting” the device and installing apps from non-legitimate app stores. Some actors injected malware into apps and have them made available on the Google Marketplace (<https://market.android.com/>). To protect the user from these apps, Google introduced Google Bouncer in February 2012 (Albanesius, 2012), an antivirus program that scanned apps before they were made available on the Google PlayStore¹⁶. The article also described how it worked, “once an application is uploaded, the service immediately starts analyzing it for known malware, spyware and trojans. It also looks for behaviors that indicate an application might be misbehaving and compares it against previously analyzed apps to detect possible red flags.” Google stated that “it runs every app in its cloud infrastructure to simulate how it might work on an Android device to look for anything fishy. Developer accounts are also scrutinized to guard against banned individuals making a reappearance”.

Android Permissions

In 2011 an application did not have any associated permissions¹⁷ and declared in the manifest file which permissions it needed. At installation time the user is notified by the installer the permissions that the app is requesting, and the

¹⁶ Google PlayStore is the new name for Google Marketplace.

¹⁷ Google introduce the concept of default permissions called normal_protection in API 23 (Marshmallow) 2015

Appendices

user then has the option to deny (don't install) or accept (continue install) the permissions.

The user is not able to select which permissions the app can receive during the installation process.

To aid developers the permissions have been explicitly mapped to resources and the resource functions as defined in the API level for that version of Android. The Android Developers forum documents the Permissions available to developers of Android apps and provides a list of the permissions grouped by function and with a brief description (Manifest Permissions, 2011). In 2011 Froyo was the most common version of Android and there were 130 available permissions available for definition in the Manifest file as an API. As newer versions of Android were released there were changes in the functionality of the operating system and this is reflected in the API calls available. These API calls are defined by their corresponding permissions. New permissions are added and or deleted in each re-iteration of Android, but most permissions remain constant, albeit with some minor variations or consolidations of sub-functions.

Example

Using the Bluetooth permission as an example; the manifest file would contain the string

android.permission-group.BLUETOOTH_NETWORK

The API which this relates to this permission is android.bluetooth

This lets applications

- Scan for other Bluetooth devices
- Query the local Bluetooth adapter for paired Bluetooth devices

- Establish RFCOMM channels/sockets
- Connect to specified sockets on other devices
- Transfer data to and from other devices

The control is performed through the defined two interfaces and 14 classes of the API.

In 2011 there were 17 API levels and additional APIs being introduced in subsequent levels. The API level provides the developer with program functionality that can be written whilst the level indicates to the user that new features are available¹⁸.

Further information on Android permissions can be found in *Android Permissions Demystified* (A. P. Felt et al., 2011).

¹⁸ New functionality can be added to an API without requesting user permission.

Appendix C Antivirus Function testing

The malware installed consisted of two viruses, an application containing adware and an application that permits root access to the device. Test Viruses were freely available on the marketplace sites to assist in testing AntiVirus products and the following two were downloaded from the Google Marketplace and used in the testing process.



Antivirus TESTVIRUS from P.Defender Antivirus. This file contains code which antivirus products detect as a virus signature.



EICAR Anti-virus Test from Extorian. The file also contains code which is detected by antivirus programs as a virus signature.



The application containing the adware was QR Droid, package name is la.droid.qr

Appendices

The application permitting root access was installed as part of the jail-breaking/rooting of the device and is called Superuser, which provides root access to the device, package name is com.noshufou.android.su.

The testing consisted of the following;

- Downloading and install the app.
 - The app is downloaded to the G1 device from the marketplace and installed.
- Checking for any antivirus database updates.
 - The app is started, and a note is made if the app requires an update to its antivirus/signature database and if the update must be initiated manually or if it is performed automatically.
- Ease of scanning.
 - Is the scanning performed automatically and can it be scheduled, or does it need to be initiated manually?
- Scanning to detect malware.
 - A full scan of the G1 device is performed and the scan results reviewed to verify that the installed malware was detected.
- Removal of malware and rescanning to verify its removal
 - Does the product remove or quarantine the malware automatically or is manual intervention required? Once the item has been removed/quarantined does the product automatically re-scan and has the malware been removed (this is checked by accessing the filesystem of the device as well as using file management programs).
- Downloading malware to verify real time monitoring (protection) and removal or blocking of the malware during download.
 - The two viruses are removed from the device, the app is started, and a test virus is then downloaded from the marketplace. Is the malware detected during the download and prevented from installing or is the user permitted to override the detection and installation?
- Re-scanning of the device to verify that the malware is detectable by the product if not detected in real time mode.
 - If the malware was not detected during the download and installation process, verify that it is detected during the subsequent scan.

The app testing results are shown below with snapshot images taken during the testing.

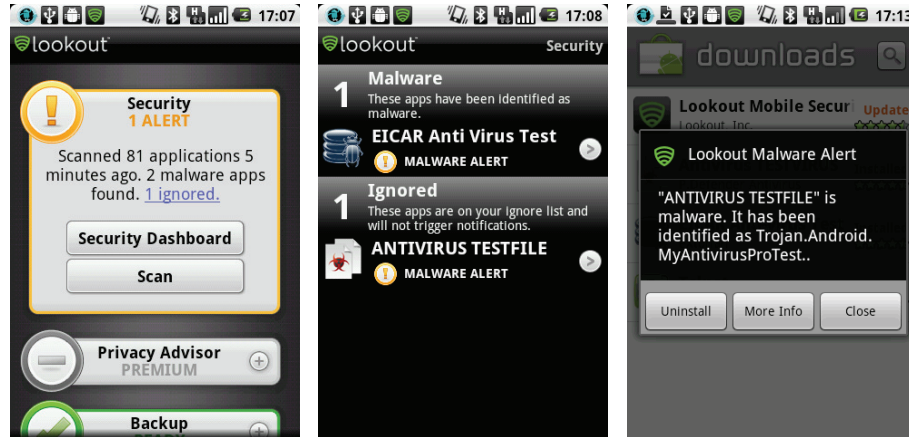


Lookout Free

The app downloaded and installed without any issues. Scanning was performed manually and there were no options for scheduling scans. The product does not have an option to update the malware database but instead notifies the user of additional malware and requests that the user installs the newer version of the app containing the updated signatures. The app detected both installed viruses but did not detect the adware application or the superuser toolkit. The

Appendices

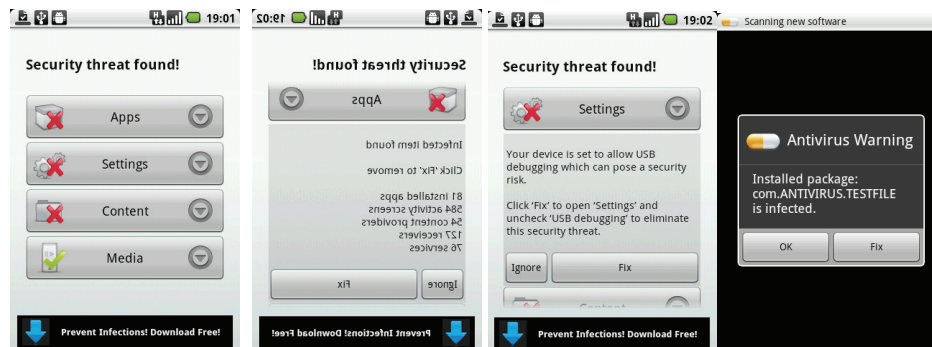
app has a status interface which provides a summary of detected threats and a security dashboard with details of the threats.



The app detected the download and installation of the malware and provided the option to remove (uninstall) the malware.

ANTIVIRUSFree

The app downloaded and installed without any issues. Starting the app, the user is given the option to either protect or configure the device. Scanning was performed manually after selecting the option to protect the device, there were no options for scheduling scans. The app detected only one of the installed virus test files and the root access but did not detect the adware application.



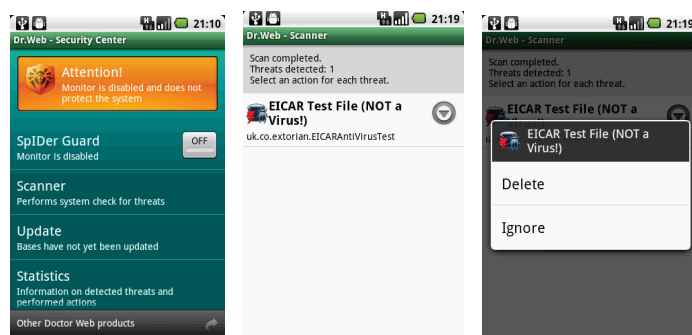
Appendices

The app detected the download and installation of the malware and provided the option to remove (uninstall) the malware.



Dr.Web Anti-virus Light

The app downloaded and installed after the second attempt. Starting the app, the user is presented with the security interface of the product. This interface provides the ability to turn on real time monitoring, run a scan, update the virus database and review threat statistics.



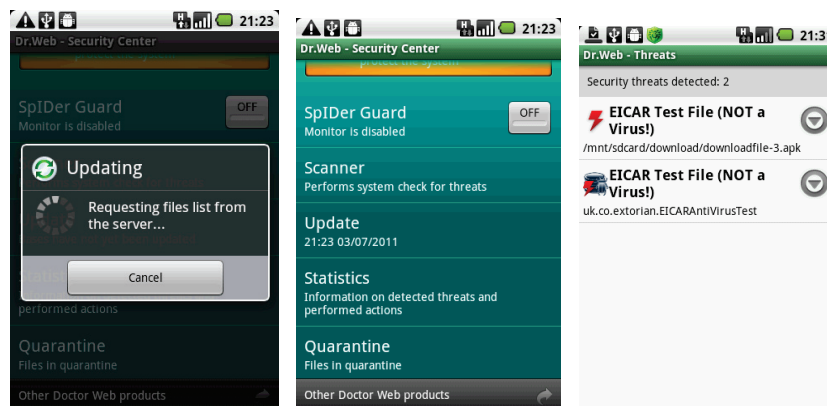
Multiple scan profiles are available to perform one of three types of scan profiles:

1. Quick scan
2. Full scan
3. Custom scan

A full scan was performed. The app detected one of the installed viruses but did not detect the adware application or the root access. Selecting the threat result prompted the user to either remove or ignore the malware. The product also correctly identified the virus as an Antivirus Test file.

Removal of the malware was successful, but also produced a scanning error message. Updating of the database resulted in the new malware signatures being downloaded from a central server and the date and time of the update was recorded on the Security Centre screen.

Appendices



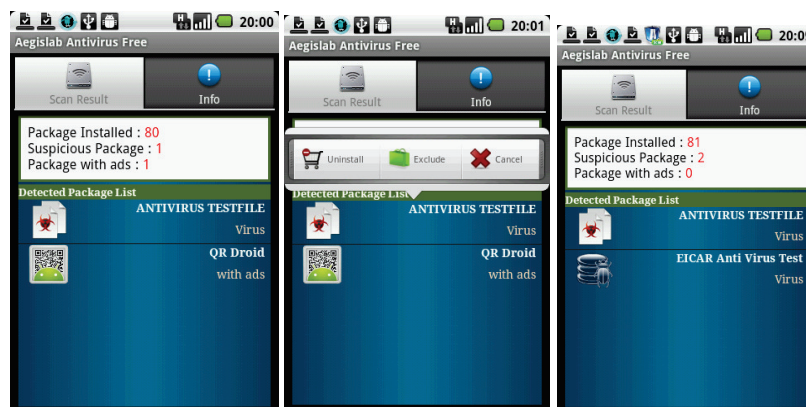
The SpIDer Guard option is switched off by default and had to be switched on to detect the malware being downloaded.

A subsequent full re-scan and a quick scan also detected the malware. The only other area of note was the length of time which the full scan took (over 5 minutes) in comparison to the other products (less than 3 minutes).



Aegislab Antivirus Free

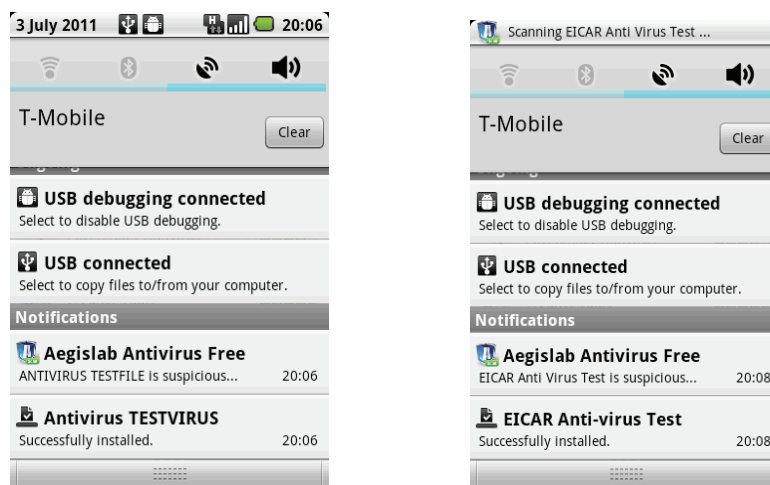
The app downloaded and installed without any issues. When started, the app loads the malware definition file and then presents an initial screen which provided the option to scan, update the malware database, exclude apps or review the network statistics of the device. Scanning was performed manually after selecting the scan option. The app detected both installed viruses and the adware application, but not the root access.



Appendices

Options were provided to uninstall or exclude the detected malware. Removal was successful, and the product immediately re-scanned to ensure that the malware had been removed. Selecting exclude placed the malware into an exclusion list.

The product detected the download of the malware and provided a notification that the product is suspicious. A second product was downloaded and was also detected.

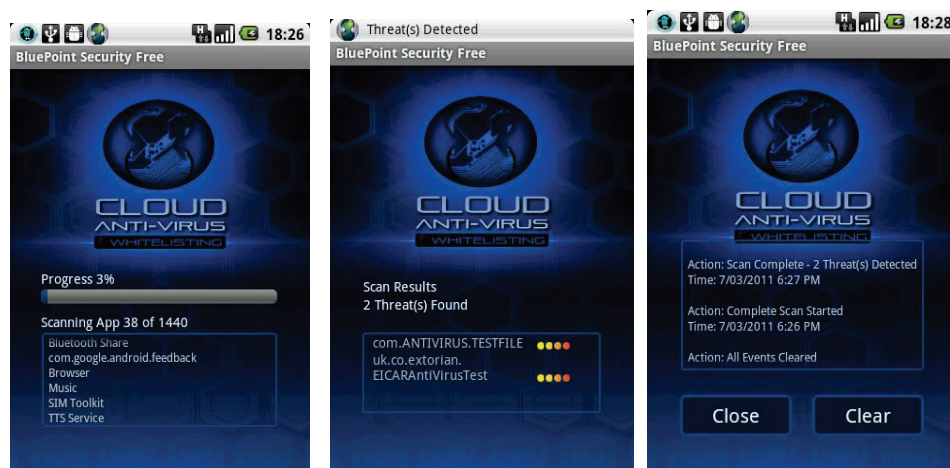


The product also gave the user the option to update the malware definition file, which was downloaded to the device.



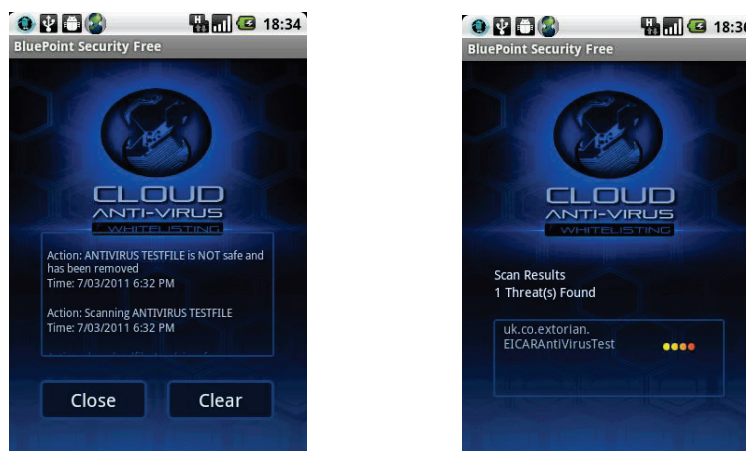
Bluepoint Antivirus Free

The app downloaded and installed without any issues. When started the initial screen displays options to scan, change settings and review events. This app uses a cloud malware database so there is no requirement to update or load a database onto the device. The detection database in the cloud receives the queries in real time when the Antivirus app needs to check a file. Selecting settings displays the status of the product. Scanning was performed manually after selecting the scan option.



The app detected both installed viruses but did not detect either the adware application or the root access.

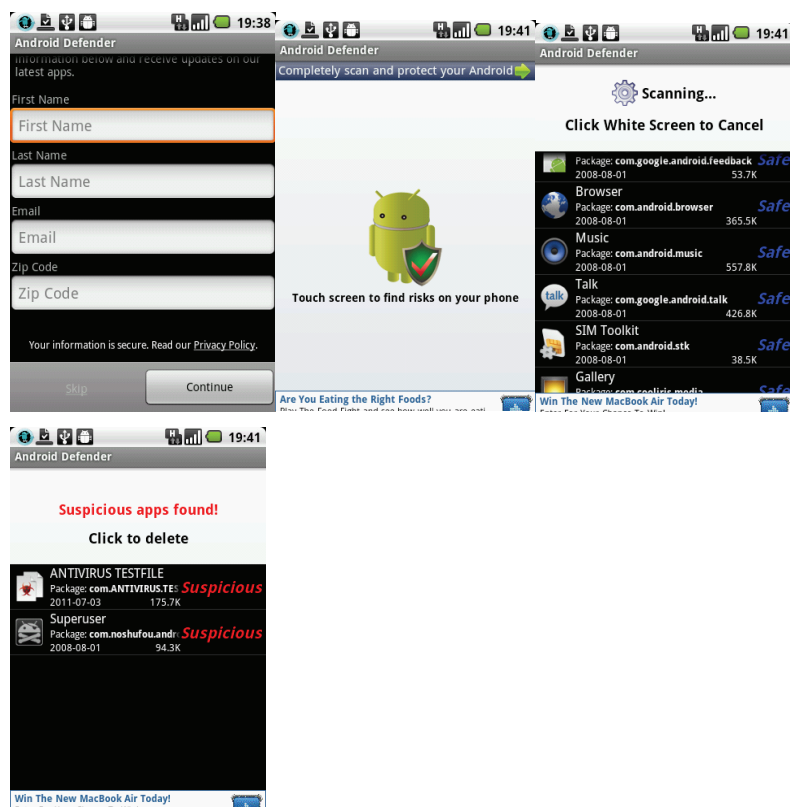
The product detected the download and installation of the malware and provided a notification to the user. Selecting the detected threat provides an option to remove and the product confirms the removal. A re-scan of the device confirms the removal of the malware.



Android Defender Virus Protect

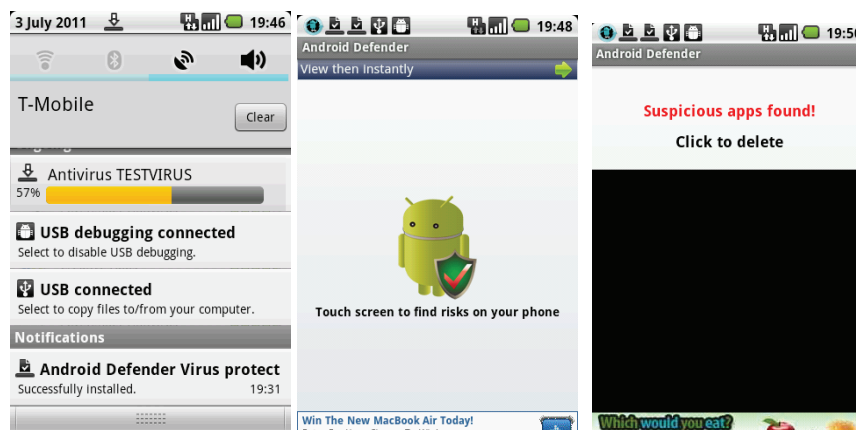
The app downloaded and installed without any issues. When started the app requires the user to enter personal contact details, name, email address and zip code. No checking is performed other than that the zip code entered is a valid US zip code. Once the data is entered, the main screen requires confirmation from the user to perform scanning.

Appendices



Scanning is then performed. The app detected both installed viruses but not the adware application or the root access.

Selection of the malware provided an option to remove (uninstall) the malware.



The app did not detect the malware during download or installation, but only when a scan was run.

Appendix D Evolution of Android Malware

Date	Malware Name	Description
Aug 9, 2010	SMS.AndroidOS.FakePlayer.a	First SMS Android Malware In the Wild: The malicious program penetrates Android devices in the guise of a harmless media player application. Once manually installed on the phone, the Trojan uses the system to begin sending SMSs to premium rate numbers without the owner's knowledge or consent, resulting in money passing from a user's account to that of the cybercriminals.
Aug 17, 2010	AndroidOS_Droisnake.A	This is the first GPS Spy Malware disguised as an Android Snake game application. To the victim, Tap Snake looks like a clone of the Snake game. However, once someone installs this app on a phone, the "game" serves as a front for a spy app that proceeds to run in the background, secretly reporting GPS coordinates back to a server. The would-be spy then pays for and downloads an app called GPS Spy and enters an email address and code to gain access to the victim's uploaded data.
Sep 14, 2010	SMS.AndroidOS.FakePlayer.b	Pornography lands on Android! This malware is a variant of SMS.AndroidOS.FakePlayer.A. The malware poses as a pornographic application whose package name is pornoplayer.apk, and it installs on the phone with a pornographic icon. When the user launches the application, the malware does not show any adult content and, instead, sends 4 SMS messages to short codes, at the end-user's expense.
Oct 13, 2010	SMS.AndroidOS.FakePlayer.c	Pornography back on Android! Third variant of the malware SMS.AndroidOS.FakePlayer.A. New pornographic application, old icon. Sends 2 SMS messages to short codes, at the end-user's expense.
Dec 29, 2010	Android.Getinimi	First example of a Botnet-Like Malware on Android. "Grafted" onto repackaged versions of legitimate applications, primarily games, and distributed in third-party Chinese Android app markets. Once the malware is installed on a user's phone, it has the potential to receive commands from a remote server that allow the owner of that server to control the phone. The specific information it collects includes location coordinates and unique identifiers for the device (IMEI) and SIM card (IMSI).

Appendices

Date	Malware Name	Description
Feb 14, 2011	Android.Adroid AKA Android.HongTouTou	New Malware with Botnet-like Features from China. The trojan compromises personal data such as IMEI/IMSI of the device and sends them back to the remote side to react based on the commands from there. Like Android.Geinimi but with a lower profile (less commands)
Feb 22, 2011	Android.Pjapps	New Trojan horse embedded on third party applications. It opens a back door on the compromised device and retrieves commands from a remote command and control server.
Mar 1, 2011	Android.DroidDream AKA Android.Rootcager AKA AndroidOS_Lootoor.A	The first example of a new generation of Mobile Malware: distributed through the Official Android Market, affected, according to Symantec 50,000 to 200,000 users. Exploits two different tools (rageagainstthecage and exploit) to root the phone
Mar 9, 2011	Android.BgServ AKA Troj/Bgserv-A AKA AndroidOS_BGSERV.A	Trojanised version of the Android Market Security tool released by Google, on March the 6th, to remove the effects of DroidDream. The trojan opens a back door and transmits information from the device to a remote location. It shows more than ever security and reputation flaws in the Android Market Proposition Model. 5,000 users affected.
Mar 20, 2011	Android.Zeahache	Trojan horse that elevates privileges on the compromised device, discovered on a Chinese language app available for download on alternative Chinese app markets. The app has the ability to root an Android device (by mean of the exploit tool called by zHash binary), leaving the device vulnerable to future threats. The app, which provides calling plan management capabilities was found also on the Android Market albeit this version lacked the code to invoke the exploit.
Mar 30, 2011	Android.Walkinwat	Manually installed from non-official Android Markets, the Trojan modifies certain permissions on the compromised device that allow it to perform the following actions: Access contacts in the address book, access network information, access the phone in a read-only state, access the vibrator on the phone, Check the license server for the application, find the phone's location, initiate a phone call without using the interface, open network sockets to access the Internet, read low-level log files, send SMS messages, turn the phone on and off. It gives a message to user trying to discipline users that download files illegally from unauthorized sites.
May 9, 2011	Android.Adsms AKA AndroidOS_Adsms.A	This malware specifically targeted China Mobile subscribers. The malware arrived through a link sent through SMS. The said message tells the China Mobile users to install a patch for their supposedly vulnerable devices by accessing the given link, which leads to a malicious configuration file. The malware then send message to premium numbers.

Appendices

Date	Malware Name	Description
May 11, 2011	Android.Zsone AKA Android.Smslibook	Google removed a Trojan, Zsone, from the Android Market with the ability to subscribe users in China to premium rate QQ codes via SMS without their knowledge. 10,000 users affected.
May 22, 2011	Android.Spacem	A biblical plague For Android! Trojanised version of a legitimate application that is part threat, part doomsayer. The threat was embedded in a pirated version of an app called 'Holy ***king Bible', which itself has stirred controversy on multiple forums in which the app is in circulation. The malware targeted North American Users. After the reboot, it starts a service which at regular intervals, attempts to contact a host service, passing along the device's phone number and operator code. It then attempts to retrieve a command from a remote location in intervals of 33 minutes. In addition to having abilities to respond to commands through the Internet and SMS, the threat also has activities that are designed to trigger on the 21 and 22 of May 2011, respectively (The End of The World).
May 31, 2011	Android.LightDD	A brand new version of Android.DroidDream, dubbed DroidDreamLight, was found in 24 additional apps repackaged and redistributed with the malicious payload across a total of 5 different developers distributed in the Android Market. Between 30,000 and 120,000 users affected.
Jun 6, 2011	Android/DroidKungFu.A AKA Android.Gunfu	Malware which uses the same exploit than DroidDream, ragesgainstheage, to gain root privilege and install the main malware component. Once installed, the malware has backdoor capabilities and is able to: execute command to delete a supplied file, execute a command to open a supplied homepage, download and install a supplied APK, open a supplied URL, run or start a supplied application package. The malware is moreover capable to obtain some information concerning the device and send them to a remote server: The collected information include: IMEI number, Build version release, SDK version, users' mobile number, Phone model, Network Operator, Type of Net Connectivity, SD card available memory, Phone available memory. In few words, the device is turned into a member of a botnet.
Jun 9, 2011	Android.Basebridge	Trojan Horse that attempts to send premium-rate SMS messages to pre-determined numbers. When an infected application is installed, it attempts to exploit the udev Netlink Message Validation Local Privilege Escalation Vulnerability (BID 34536) to obtain "root" privileges. Once running with "root" privileges it installs an executable which contains functionality to communicate with a control server using HTTP protocol and sends information such as Subscriber ID, Manufacturer and Model of the device, Version of the Android operating system. The Trojan also periodically connects to the control server and may perform the following actions:

Appendices

Date	Malware Name	Description
Jun 9, 2011	Android.Uxipp AKA Android/YZHC SMS.A	send SMS messages, remove SMS messages from the Inbox and dial phone numbers. The Trojan also contains functionality to monitor phone usage. Trojan Horse that attempts to send premium-rate SMS messages to predetermined numbers. Again, the threat is as an application for a Chinese gaming community. When executed, the Trojan attempts to send premium-rate SMS messages to several numbers and remove the SMS sent. The Trojan sends device information, such as IMEI and IMSI numbers.
Jun 10, 2011	Andr/Plankton-A AKA Android.Tonclank	This is a Trojan horse which steals information and may open a back door on Android devices. Available for download in the Android Market embedded in several applications, when the Trojan is executed, it steals the following information from the device: Device ID and Device permissions. The above information is then sent to a remote server from which the Trojan downloads a .jar file which opens a back door and accepts commands to perform the following actions on the compromised device: copies all the bookmarks on the device, copies all the history on the device, copies all the shortcuts on the device, creates a log of all the activities performed on the device, modifies the browser's home page, returns the status of the last executed command. The gathered information is then sent to a remote location. Although this malware does not root the phone, its approach of loading additional code does not allow security software on Android to inspect the downloaded file in the usual "on-access" fashion, but only through scheduled and "on-demand" scans. This is the reason why the malware was not discovered before.
Jun 15, 2011	Android.jsmslider	Trojan found in alternative Android markets that predominately target Chinese Android users. This Trojan predominantly affects devices with a custom ROM. The application masquerades as a legitimate one and exploits a vulnerability found in the way most custom ROMs sign their system images to install a secondary payload (without user permission) onto the ROM, giving it the ability to communicate with a remote server and receive commands. Once installed the second payload may read, send and process incoming SMS messages (potentially for mTAN interception or fraudulent premium billing subscriptions), install apps transparently, communicate with a remote server using DES encryption.
Jun 20, 2011	Android.GGTracker	This trojan is automatically downloaded to a user's phone after visiting a malicious webpage that imitates the Android Market. The Trojan, which targets users in the United States by interacting with several premium SMS subscription services without consent, can sign-up a victim to several premium SMS subscription services without the user's consent. This can lead to unapproved charges to

Appendices

Date	Malware Name	Description
Jul 1, 2011	Android.KungFu Variants	<p>a victim's phone bill. Android users are directed to install this Trojan after clicking on a malicious in-app advertisement, for instance a Fake Battery Saver.</p> <p>Repackaged and distributed in the form of "legitimate" applications, these two variants are different from the original one by re-implementing some of their malicious functionalities in native code and supporting two additional command and control (C&C) domains. The changes are possibly in place to make their detection and analysis harder.</p> <p>The repackaged apps infected with the DroidKungFu variants are made available through several alternative app markets and forums targeting Chinese-speaking users.</p>
Jul 3, 2011	AndroidOS_Crusewin.A AKA Android.Crusewind	<p>Another example of a trojan which sends SMS to premium rate numbers. It also acts as a SMS Relay. It displays a standard Flash icon in the application list. The Trojan attempts to download an XML configuration file and uses it to retrieve a list of further URLs to send and receive additional data. The Trojan also contains functionality to perform the following actions: delete itself, delete SMS messages, send premium-rate SMS messages to the number that is specified in the downloaded XML configuration file, update itself.</p>
Jul 6, 2011	AndroidOS_SpyGold.A AKA Android.GoldDream	<p>This backdoor is a Trojanised copy of a legitimate gaming application for Android OS smartphones. It steals sensitive information of the affected phone's SMS and calls functions, compromising the security of the device and of the user. It monitors the affected phone's SMS and phone calls and sends stolen information to a remote URL. It also connects to a malicious URL to receive commands from a remote malicious user.</p>
Jul 8, 2011	DroidDream Light Variant	<p>New variant of DroidDream Light in the Android Market immediately removed by Google. Number of downloads was limited to 1000 - 5000. This is the third iteration of malware likely created by the authors of DroidDream.</p>
Jul 11, 2011	Android.Smsniffer AKA Andr/SMSRep-B/C AKA Android.Trojan.SmsSpy.B/C AKA Trojan-Spy.AndroidOS.Smsr.a	<p>ZITMO arrives on Android! This threat is found bundled with repackaged versions of legitimate applications. When the Trojan is executed, it grabs a copy of all SMS messages received on the handheld device and sends them to a remote location.</p>
Jul 12, 2011	Android.HippoSMS AKA Android.Hippo	<p>Another threat found bundled with repackaged versions of legitimate applications. When the Trojan is executed, it grabs a copy of all SMS messages received on the handheld device and sends them to a remote location.</p>

Appendices

Date	Malware Name	Description
Jul 15, 2011	Android.Fokonge	This threat is often found bundled with repackaged versions of legitimate applications. The repackaged applications are typically found on unofficial websites offering Android applications. When the Trojan is executed, it steals information and sends it to a remote server.
Jul 15, 2011	Android/Sndapps.A AKA Android.Snadapps	Five Android apps found in the official Android Market share a common suspicious payload which upload users' personal information such as email accounts as well as phone numbers to a remote server without user's awareness.
Jul 27, 2011	Android.Nickispy	Trojan horse which steals several information from Android devices (for instance GPS Location or Wi-Fi position). For the first time on the Android Platform a malware is believed to spy conversations.
Jul 28, 2011	Android.Lovetrap	Trojan horse that sends SMS messages to premium-rate phone number. When the Trojan is executed, it retrieves information containing premium-rate phone numbers from a malicious URL then sends premium-rate SMS messages, and attempts to block any confirmation SMS messages the compromised device may receive from the premium-rate number to mask its activities. The Trojan also attempts to gather IMSI and location information and send the information to the remote attacker.
Aug 2, 2011	Android.Premiumtext	This is a detection for Trojan horses that send SMS texts to premium-rate numbers. These Trojan is a repackaged versions of genuine Android software packages, often distributed outside the Android Marketplace. The package name, publisher, and other details will vary and may be taken directly from the original application.
Aug 9, 2011	Android.NickiBot	It belongs to the same NickiSpy family. However, it is significantly different from its predecessor since it is fully controlled by SMS messages instead of relying on a hard-coded C&C server for instructions. In addition, NickiBot supports a range of bot commands, such as for (GPS-based) location monitoring, sound recording and (email-based) uploading, calllog collection, etc. It also has a check-in mechanism to a remote website. his threat is often found bundled with repackaged versions of legitimate applications. The repackaged applications are typically found on unofficial websites offering Android applications. When the Trojan is executed, it steals information and sends it to a remote server.

Appendix E UPR16

FORM UPR16 Research Ethics Review Checklist



Please include this completed form as an appendix to your thesis (see the [Research Degrees Operational Handbook](#) for more information)

Postgraduate Research Student (PGRS) Information		Student ID:	313485
PGRS Name:	Kathryn Carstens		
Department:	SOC	First Supervisor:	Carl Adams
Start Date: <small>(or progression date for Prof Doc students)</small>	Oct 2010		
Study Mode and Route:	Part-time <input checked="" type="checkbox"/>	MPhil <input type="checkbox"/>	MD <input type="checkbox"/>
	Full-time <input type="checkbox"/>	PhD <input checked="" type="checkbox"/>	Professional Doctorate <input type="checkbox"/>
Title of Thesis:	Android Smartphone Apps: Privacy concerns of Unregulated Permissions on Social and Psychological Contracts.		
Thesis Word Count: <small>(excluding ancillary data)</small>	48,873		
<p>If you are unsure about any of the following, please contact the local representative on your Faculty Ethics Committee for advice. Please note that it is your responsibility to follow the University's Ethics Policy and any relevant University, academic or professional guidelines in the conduct of your study</p> <p>Although the Ethics Committee may have given your study a favourable opinion, the final responsibility for the ethical conduct of this work lies with the researcher(s).</p>			
<p>UKRIO Finished Research Checklist: <small>(If you would like to know more about the checklist, please see your Faculty or Departmental Ethics Committee rep or see the online version of the full checklist at: http://www.ukrio.org/what-we-do/code-of-practice-for-research/)</small></p>			
a) Have all of your research and findings been reported accurately, honestly and within a reasonable time frame?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
b) Have all contributions to knowledge been acknowledged?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
c) Have you complied with all agreements relating to intellectual property, publication and authorship?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
d) Has your research data been retained in a secure and accessible form and will it remain so for the required duration?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
e) Does your research comply with all legal, ethical, and contractual requirements?	YES <input checked="" type="checkbox"/>	NO <input type="checkbox"/>	
Candidate Statement:			
I have considered the ethical dimensions of the above named research project, and have successfully obtained the necessary ethical approval(s)			
Ethical review number(s) from Faculty Ethics Committee (or from NRES/SCREC):	184D-53B5-0CBE-A134-E4D8-7C3B-55A0-05E7		
If you have <i>not</i> submitted your work for ethical review, and/or you have answered 'No' to one or more of questions a) to e), please explain below why this is so:			
There is no human or animal participation in the research. Ethics certificate number is: 184D-53B5-0CBE-A134-7C3B-55A0-05E7			
Signed (PGRS):		Date:	21/06/2018

UPR16 – April 2018

END