CHARACTERIZING ANDROID PERMISSIONS AND ANALYZING THEIR PRIVACY-INTRUSION

BY KIRTY PRABHAKAR VEDULA

A thesis submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Prof. Janne Lindqvist

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May, 2014

ABSTRACT OF THE THESIS

Characterizing Android permissions and analyzing their privacy-intrusion

by Kirty Prabhakar Vedula

Thesis Director: Prof. Janne Lindqvist

Several studies have examined Android apps use of permissions from a security point-of-view. However, privacy has been an increasing concern with the increasing number of apps and most users are still not aware of several privacy-intruding permissions. This research investigates a dataset of permissions for over 610,000 apps existing on the Google Play Store on how they can impact users of smartphones. The objective is to better understand the patterns in permissions from a users perspective. We report findings on thousands of reputed apps using users coarse or exact location. About 70% of such apps were free and the major categories were Entertainment and Games. We observe more than 90% positive correlation between the ratings and such permissions. We also analyze device-intrusion where few apps ask for permissions to let the app run in the background, tweak with hardware and deplete energy using rich-media ads. For example, several apps in News & Magazines category were found to consume energy for downloading even when the device is in sleep mode. More than 20,000 free apps were found to receive packet data anonymously. The thesis concludes with discussion on implications obtained by employing k-modes algorithm to cluster dangerous and safe apps.

Acknowledgements

I would like to thank my advisor Prof. Janne Lindqvist, ECE, Rutgers University for providing me with his valuable guidance and help throughout the course of this research. This project could not have been accomplished without you. I would like to thank the researchers Pern Chia, N Asokan, and Mario Frank for providing the open-license datasets on Android permissions. I greatly appreciate your contribution! Lastly, I thank my family and friends for believing in me even when I didn't. Your relentless support has been very important for me. This material was based upon the work supported by the National Science Foundation under Grant Number 1228777. Any opinions, findings, and conclusions or recommendations expressed in this material are those of author(s) and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Problem Setting

Studies on user-experience aim to enrich the cognizance and thereby encourage users to continue using the applications. If an application breeds distrust, it will lose its customers at once. Hence, it is important to know the users' expectation of privacy, and the extent to which they can handle their concerns. In the aspect of designing mobile-user experiences, the privacy issue is often neglected.

In today's world, understanding the complex permissions of Android applications can be an intensive task and many smart-phone users either take them for granted or think that they have no other option other than to allow the apps to intrude their privacy. For such users, there is a necessity to make them knowledgeable regarding how intrusive the app is. An interesting idea would be to provide a neat interface warning to the user when an app tries to cross a limit of intrusion. Our problem, here, deals with analyzing the permissions from a users' perspective.

1.2 Contributions of the thesis

Over 160 permissions for over 650,000 apps from the Android Market or the Google Play Store have been collected and arranged according to categories. These are studied for delving out inferences from the user data. Certain privacy-related concerns are also discussed in particular.

Relatively less work has been done on understanding permissions directly from then dataset from a users' perspective. In this thesis, we explore Android applications to determine whether Android developers are intruding the privacy and into the device of

the user.

The dataset we have obtained is very sparse in nature. Using the mainstream clustering algorithms to divide the permissions into dangerous and safe is not possible. k-modes clustering has been used to account for the sparsity in the dataset and the results are formulated. In the end, we discuss the factors and some speculations which can affect the overall user experience.

1.3 Structure of the thesis

In Chapter 2, we briefly study the related work that has been done in the field of permissions, security and in learning permissions. In Chapter 3, we give the groundwork and literature review on clustering. In Chapter 4 and Chapter 5, the problem design and the results have been discussed respectively. In Chapter 6, we present a discussion on the various results. In Chapter 7, we conclude with pointers on how future work can proceed from this point.

Chapter 2

Android Permissions

Modern mobile operating systems are being designed with an increased awareness on providing the user with secure applications. However, these consist only the majority and several compromises are being made on users' privacy concerns. Humongous amount of data is being made available in this age without the users knowledge. Providing privacy to a user is an often-neglected aspect of designing user experiences. The users must be properly informed of all the violations they are facing. The trust and confidence of the users are critical to turning from one-time visitors into long-term customers.

All mobile OSs are designed with a set of challenges and restrictions like accessing only a small memory footprint, to be conscious of the power consumption, to allow access to personally identifiable information, and a wide array of hardware.

2.1 Android Architecture Overview

Android is an open-source project built on Linux. It composed of isolated modules with extensibility feature, thus making it light-weight and versatile.

2.1.1 Intent system

Being an independent OS, Android provides its own inter-app communication built on its Intent system. They provide an abstract description of an operation which can be applied to multiple apps at the same time. They can describe the operation they perform, without explicitly mentioning a recipient/app which can performing the task. For example, when an intent *ACTION_VIEW* is sent with a URL of a website, say http//yahoo.com , Android OS finds the right application (Browser) that can deliver

this task correctly. In case there are more than one browser apps, it gives a suggestion to the user to pick one from the list.

Intent binds the three application components of Android namely Activities, Services and Content Providers. Activities are the tasks that are currently running or which are running in the background. They have their own specific lifecycles. These hold the data and the interface that connects data in one process. [1]

2.2 Android Permissions Model

Android is implemented through Content Providers, which manage a dataset and provide access to remote services. The motive behind providing permissions is to protect the personally identifiable information. Android entertains the third party app providers with this policy. Since these operations are dangerous and can destroy a device, Android has employed a filter where it has to be passed through Package Manager in order for its features to function. This protects operations from untrusted developers and allows trusted developers to access the phone normally.

As new hardware is made accessible through Androids SDK, new permissions are added to them. However, they do not change drastically as the newer versions of Android emerge.

The Google Play Store, previously known as the Android Market, is a platform which provides extensive third-party APIs for the Android operation system. When used on mobiles, these APIs have access to phone hardware, settings, and user data. A permissions screen is made available to the user while installing. However, the user can access it at anytime. The idea behind the permission screen is to show to the users capabilities of the app that can potentially access sensitive information about the user or affect the functionality of the phone. These potentially sensitive permissions are defined by Google, and currently there are more than 150 of them available [2].

1. Before installing the application

2. By navigating the settings in the application

3. While installing updates for the application

4.0.3	Icecream Sandwich	165
2.3.4	Gingerbread	137
2.2	Froyo	134
2.1	clair	122
1.6	Donut	106
1.5	Cupcake	103

Table 2.1: Permissions for official releases of Android versions

The following table 2.2 shows various options present in Google Play store.

A	App name
B	Developer Name
C	App Rating
D	Number of ratings
E	Date the app was last updated
F	Category in the Google Play Store it falls under
G	Number of installs (range, not exact number)
H	Description of the app
I	Reviews of the app
J	Permissions the app requests

Table 2.2: Various properties of a Google Play Store app page

The following figure 2.1 shows the permissions for the app *Brightest Flashlight*. A sample Google Play Store install screen showing the permissions.

- View Wi-Fi connections receive data from Internet
- Phone calls read phone status and identity
- Storage modify or delete the contents of your USB storage
- Your applications information retrieve running apps
- Camera take pictures and videos

- Development tools change system display settings
- System tools modify system settings test access to protected storage
- Affects battery control flashlight prevent device from sleeping

The user must scroll to see all of them, and click Show All" to see the hidden ones.

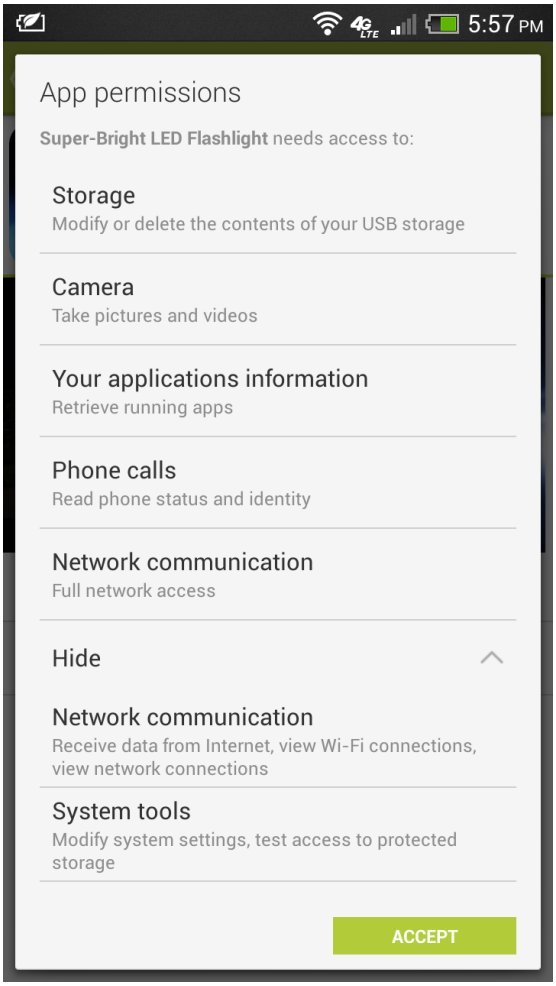

Figure 2.1: This figure shows some permissions asked by makers of Brightest Flashlight app. It is to be noted that even though it is just an app which access the flash on the phone, it also accesses a person's location. It can also affect the user's battery and settings.

Some of the most requested permissions can be seen in Table 2.3; while the rest of the permissions are included in the Appendix. 7.1.

2.3 Permission Footprint

Two apps requesting same set of permissions can be assumed to possess the same capabilities. All these permissions are static they cannot be changed as they cannot be added or removed. These are together called as the permission fingerprint. It uniquely defines what an app has access to. Therefore, the permission footprint establishes the absolute maximum capabilities of an app, even if the system rejects some of them.

2.4 Previous work

Several studies have been done in the field of Android permissions as well as in classification methods. They worked on the idea of what can be done when the privacy concerns are detrimental. Wei et al. have thoroughly analyzed in their paper regarding the applications which ask for permissions which they never use (Eg. Brightest Flashlight doesn't make use of the user's location) and makes the user vulnerable to risks [3]. They have also noted that some application developers use those details for its future versions. The list of permissions has been increasing with its newer versions. This is to offer more functionality to the users. However, it is also necessary to educate the user of what is being offered. [4]

Another study proposes an analysis tool for extracting accurate permission specifications pertaining to the application. [5] and [6]. TaintDroid was a similar tracking app developed by Enck et al. to help users in identifying the application's misuse of used for tracking [7]. This warns users about applications that request blacklisted sets of permissions. It works by noting all the permissions and marking them against the dangerous ones. Zhou et al. [8] found real malware in the wild with DroidRanger, a malware detection system that uses permissions as one input.

Reeder et al. [9] developed permissions manager on Windows operating system by employing expandable grid visualizations. Smetters et al. [10] studied various permission-based architectures for document sharing in organizations.

David Barrera et al. [11] made attempts to understand how the permissions are being used by the user. They have conducted an empirical analysis of the permissions model

and found that very few permissions are actually used by the developers using self-organizing maps. They have developed visualizations to find the relationship between application categories and permission requests and found that most of them are just restricting access to advanced functionality on devices. They have identified application clusters based on requested permissions. However, their focus was more on visualization than on the working with permission requests.

Felt et al. and Chia et al. [12] and [13] surveyed Android applications to identify the most-requested permissions demanded by the apps. They have also found several interesting correlations between the number of permissions, number of installs, average rating, number of applications published by the same developer.

Some other work on permissions has been done in [14], [15], [16], and [17].

2.5 Learning algorithms in permissions setting

Mario Frank et al. [18] have developed a probabilistic model to characterize the permission request patterns from Android and Facebook applications. They have found that the permission requests of low-reputation applications are very different from those of high-reputation applications. They have also found correlations between the app categories and permission request patterns. They have developed a system which can warn the users about applications that do not match the permission request patterns as depicted by the app.

Other research has focused on using machine learning techniques to identify malware. Sanz et al. [19] applied classifiers to the permissions, ratings, and static strings of 820 applications to predict application categories, aimed at malware detection. Shabtai et al. [17] have built a classifier for Android games and tools, as a proxy for malware detection.

2.6 Summary

All the previous work suggests that the current forms of community ratings used in app markets today are not reliable indicators of privacy risks of an app. There are many

apps which are enticing users into granting permissions to intrude their privacy. It was also observed that free applications and applications with mature content request more permissions than the usual ones. Our work follows all this aiming to effectively enforce rules on battery consumption, location access and personally identifiable information,

Permission	Description
INTERNET	Network communication. full Internet access. Allows the app to create network sockets.
WRITE_EXTERNAL_STORAGE	Storage. modify/delete USB storage contents modify/delete SD card contents. Allows the app to write to the USB storage. Allows the app to write to the SD card.
READ_PHONE_STATE	Phone calls. read phone state and identity. Allows the app to access the phone features of the device. An app with this permission can determine the phone number and serial number of this phone, whether a call is active, the number that call is connected to and the like.
ACCESS_FINE_LOCATION	Your location. fine (GPS) location. Access fine location sources such as the Global Positioning System on the tablet, where available. Malicious apps may use this to determine where you are, and may consume additional battery power.
ACCESS_COARSE_LOCATION	Your location. coarse (network-based) location. Access coarse location sources such as the cellular network database to determine an approximate tablet location, where available. Malicious apps may use this to determine approximately where you are.
WAKE_LOCK	System tools. prevent tablet from sleeping prevent phone from sleeping.
READ_CONTACTS	Your personal information. read contact data. Allows the app to read all of the contact (address) data stored on your tablet. Malicious apps may use this to send your data to other people.
CALL_PHONE	Services that cost you money. directly call phone numbers. Allows the app to call phone numbers without your intervention. Malicious apps may cause unexpected calls on your phone bill. Note that this doesn't allow the app to call emergency numbers.
CAMERA	Hardware controls. take pictures and videos. Allows the app to take pictures and videos with the camera. This allows the app at any time to collect images the camera is seeing.
WRITE_CONTACTS	Your personal information. write contact data. Allows the app to modify the contact (address) data stored on your tablet. Malicious apps may use this to erase or modify your contact data.
GET_TASKS	System tools. retrieve running apps. Allows the app to retrieve information about currently and recently running tasks. Malicious apps may discover private information about other apps.
RECORD_AUDIO	Hardware controls. record audio. Allows the app to access the audio record path.
SEND_SMS	Services that cost you money. send SMS messages. Allows the app to send SMS messages. Malicious apps may cost you money by sending messages without your confirmation.
READ_HISTORY_BOOKMARKS	Your personal information. read Browser's history and bookmarks. Allows the app to read all the URLs that the Browser has visited, and all of the Browser's bookmarks.
READ_CALENDAR	Your personal information. read calendar events plus confidential information. Allows the app to read all calendar events stored on your tablet, including those of friends or coworkers. Malicious apps may extract personal information from these calendars without the owners' knowledge.
WRITE_HISTORY_BOOKMARKS	Your personal information. write Browser's history and bookmarks. Allows the app to modify the Browser's history or bookmarks stored on your tablet. Malicious apps may use this to erase or modify your Browser's data
RECEIVE_SMS	Your messages. receive SMS. Allows the app to receive and process SMS messages. Malicious apps may monitor your messages or delete them without showing them to you.
WRITE_CALENDAR	Your personal information. add or modify calendar events and send email to guests without owners' knowledge. Allows the app to send event invitations as the calendar owner and add, remove, change events that you can modify on your device, including those of friends or co-workers. Malicious apps may send spam emails that appear to come from calendar owners, modify events without the owners' knowledge, or add fake events.
MOUNT_UNMOUNT_FILESYSTEMS	System tools. mount and unmount filesystems. Allows the app to mount and unmount filesystems for removable storage.
READ_SMS	Your messages. read SMS or MMS. Allows the app to read SMS messages stored

Chapter 3

Predictive models

3.1 Introduction

Machine learning is the study of learning systems which is an intersection of statistics, computer science, engineering and optimization [20]. These techniques are used for data analysis and decision-making tasks such as classification of categories, estimating probabilities, and data mining.

This chapter introduces the key ideas in unsupervised learning, and focuses on clustering in particular. [21], [22] and [23].

3.1.1 Supervised Learning

In supervised learning, for a sequence of inputs and outputs given to the system, the machine has to learn to produce the correct output matching to the new input. These outputs are called class labels. The goal of this approach is to maximize the output. This is equivalent to performing regression in statistics and is also related to control theory in engineering.

3.1.2 Unsupervised Learning

In unsupervised learning, the machine receives inputs, and does not have any supervised target outputs. Even here, the goal is to maximize the given objective function by finding patterns in the data. It does not account for the noise that is present in the model. This can be eventually used to predict the category of the new samples. Two classic examples of unsupervised learning are clustering and dimensionality reduction. We make use of clustering here.

3.2 Clustering

Clustering is an unsupervised learning technique used widely as a data analysis tool. The context is when we are unaware of the natural groupings of the data, clustering algorithms identify them and produce a representation based on some common characteristics or features. In clustering algorithms, there is very less information about the data. So, we cannot make assumptions about the data.

3.2.1 Applications of Clustering

Cluster analysis has wide ranging applications including computational biology, climatology, psychology and medicine, social network analysis, business and marketing. Clustering can be used for various reasons including data summarization, compression, efficiently finding nearest neighbors, identifying similar objects etc.

3.2.2 Limitations of Clustering

- Clustering is ad hoc in nature, it cannot be applied in every situation
- There is always a trade-off between the data representation and similarity metric
- Results will depend entirely on similarity used
- Computing similarity of mixed-type data is hard

3.3 Clustering Method

The input to a clustering algorithm is called as a tuple or a record. The type of the data is usually numerical, categorical or Boolean. Distance is the usual measure for separating data into clusters. The distance metrics can be Euclidean or Manhattan.

The main goal of these measures is to improve the prediction ability of a clustering algorithm. However, these also affect the error-rate of the clustering algorithm when used for prediction.

Here, we partition the data into training set and test set. The training set will be used to calibrate/train the model parameters. The trained model is then used to make a prediction on the test set.

Several unsupervised methods for comparing clustering, e.g., Jaccard index, Rand index, Fowlkes-Mallows index, Mirkin metric, variation of information etc., exist in the literature. Each algorithm optimizes one of these measures to minimize the intra-cluster distances and to maximize the inter-cluster distance.

3.4 Choosing a clustering algorithm

Choosing a clustering algorithm is a huge task, as there are no concrete objective measures. The crucial choice when deciding on a cluster analysis algorithm is to decide how to quantify dissimilarities between two clusters using some metrics of optimality.

The answer to *What is a good clustering algorithm for my problem?* depends on how much prediction power the clustering step provides.

3.5 k-means Clustering

k-means clustering is a method of partitioning data. It is one of the algorithms coming under into k mutually clusters and returns the index of the cluster to which it has assigned each observation.

3.5.1 Why k-means does not work?

We have a dataset of over 650000 records, and we want to divide them into 5 clusters. The complexity of the original K-Means clustering algorithm is O(n*K*I*f), where n is the number of records, K is the number of clusters we want, I is the number of iterations and f is the number of features in a particular record. It can be clearly seen it will take a long time to execute the algorithm. Also, because our data is sparse, there is no guarantee that they will converge into specific clusters.

3.6 Hierarchical Clustering

Hierarchical clustering algorithms do not provide a single partitioning of the data set, but instead provide an extensive hierarchy of clusters that merge with each other at certain distances. At different distances, different clusters will form, which can be represented using a dendrogram, where the y-axis marks the distance at which the clusters merge, while the objects are placed along the x-axis such that the clusters don't mix. [24]

3.6.1 Linkage Criterion

There are multiple candidates that the cluster can compute its distance to from its centroid. This is called as linkage. Some of the popular linkages are single-linkage clustering, complete linkage clustering and average linkage clustering. Single linkage clustering considers the minimum of the object distances, the complete linkage clustering considers the maximum of object distances and the average linkage clustering considers the average of the distances.

3.7 Feature Selection

External prediction-related quality measures are often used as features for clustering when it is not possible to deal with the data as such. Measures like entropy, normalized mutual information, supervised F-measure etc. have been used in the literature [**?**].

3.7.1 Correlation

In statistics, dependence is any statistical relationship between two random variables or two sets of data. Correlation refers to any of a broad class of statistical relationships involving dependence.

Pearson's correlation coefficient between two variables is defined as the covariance of the two variables divided by the product of their standard deviations. Pearson correlation removes the magnitude effects, and gives the correlation value in the range in the range [-1.0, 1.0]. The value of -1 means the values are anti-correlated, the value

of 0 means that there is no correlation between them and the value of 1 means they are perfectly correlated. These can be generalized to handle categorical data as well.

Other correlation coefficients have been developed to be more robust than the Pearson correlation that is, more sensitive to nonlinear relationships. Mutual information can also be applied to measure dependence between two variables.

3.8 Phi Coefficient

The phi coefficient is a statistical measure which calculates the degree of association between two binary variables. This is similar to correlation coefficient.

It says that two variables are positively associated if most of the data falls in the diagonal of the matrix (typically 2 X 2). If the data falls off the diagonal, they are negatively associated.

Chapter 4

Problem Design

4.1 Dataset

Most previous work was conducted on relatively small, experimental data sets employing at most a few thousand applications.

Our dataset consists of a total of 619250 applications, along with the unique permissions they demand from the user while he is installing the app or accessing it separately. For this, we have obtained permissions and other statistical data from the datasets made available by Chia et al. [12] and Frank et al. [18]. Table 1 lists the statistics of the apps in our dataset as per Google Play Store which makes distinctions between apps and games. Some of the categories include Personalization, Entertainment, Finance, Health, Lifestyle, Multimedia, News and Weather, Productivity, Libraries and Reference, Shopping, Social, Sports and Tools.

4.2 Android Permissions

The idea behind the permission screen is to show to the users capabilities of the app that can potentially access sensitive information about the user or affect the functionality of the phone. These potentially sensitive permissions are defined by Google, and currently there are more than 150 of them available.

4.3 Average User Rating

Chia et al. [12] have devised a formula for finding out the average rating of the users depending on the app's popularity. The average rating is checked for its correlation to the popularity of the app. We also check if they follow any specific heavy-tailed

distribution using this formula.

$$avgra = (avgr - 3) * log(numberofratings) \tag{4.1}$$

4.4 Classifying Permissions

The total number of permissions are listed below according to their categories. Table 4.1 gives the division of permissions divided according to their categories.

Permissions	Number
Storage	1
Phone Calls	3
System Tools	35
Network Communication	9
Your Location	4
Your Personal Information	12
Your Messages	14
Services that Cost You Money	1
Your Accounts	21
Affects Battery	3
Hardware	6

Table 4.1: Common permissions types required by applications. Note that a permission type can include several kind of permissions, e.g. "Your Location" includes both "fine-grained" and "coarse-grained" location permissions.

The following figure 4.1 gives the distribution of apps across different categories. Android segregates all the games separately

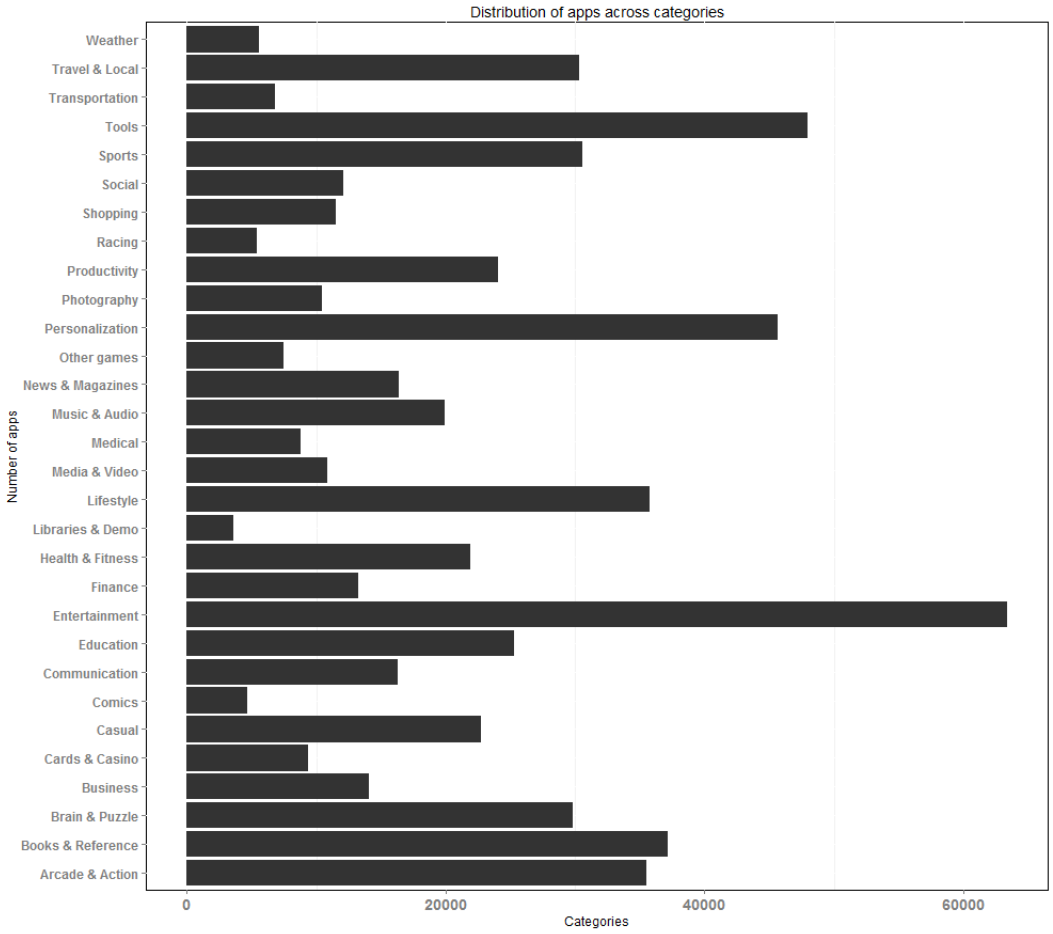

Figure 4.1: This figure shows the statistics of app distribution across all categories.

Category	Number of apps	Dangerous Permissions
Personalization	45635	4985
Weather	5561	1025
Travel & Local	30351	4203
Entertainment	63357	18150
Photography	10454	1274
Tools	47966	10557
Music & Audio	19894	2872
Lifestyle	35772	10369
Libraries & Demo	3574	587
Sports	30596	7167
Transportation	6813	1269
Media & Video	10891	1740
Social	12116	2021
Communication	16311	1465
Books & Reference	37142	8734
Productivity	24033	4301
Shopping	11493	2281
Business	14110	2578
Finance	13278	3771
News & Magazines	16373	3503
Health & Fitness	21925	7445
Education	25281	7296
Medical	8812	2641
Comics	4679	1213
Arcade & Action	35515	8557
Casual	22712	5626
Brain & Puzzle	29838	7674
Racing	5402	621
Cards & Casino	9366	2000
Others	7507	1342

4.5 Research Questions

We explore different aspects of these permissions using sets of research questions using the data with regard to the mobile users experience with the Google Play permissions. This divides our work in exploring five different aspects.

- **RQ1** - *Popularity* - What kind of apps require more permissions? How are the permissions correlated with the popularity of the apps?
- **RQ2** -*Energy* - What permissions do not allow the user to conserve energy efficiently?
- **RQ3** - *Location* - How many apps have the access to the users' network-based coarse location and the users exact location? How does this relate to the apps popularity and cost?
- **RQ4** - *Data Usage* - How many apps make use of the data without knowledge of the user?
- **RQ5** - *Clustering* - Can we segregate the apps on the scale of 1-5 from safe to dangerous?

In the next chapter, we elaborate on these results and answer these research questions.

Chapter 5

Results and Discussion

5.1 RQ1 Popularity and permissions

We have found that the applications which have received favorable reviews from large number of users request permissions different to those which are not reviewed by many users.

Our analysis also shows that the permissions are very sparse in nature. Only a small number of permissions are used frequently, while the others are rarely used.

The following figure 5.1 shows the histogram of the ratings of the apps obtained from our dataset. About 65% of the apps are with 4 or 5 star ratings on the Google Play store. The average rating calculated according to equation (1) also asserts the same. If we analyze category-wise, the *Entertainment* has the most number of dangerous permissions. The figure 5.2 gives the demographic of how apps have dangerous permissions. Some of the dangerous permissions are *Manage Accounts*, *Send SMS*, *Net Access*, *Read contacts*, *Direct Phone Calls*.

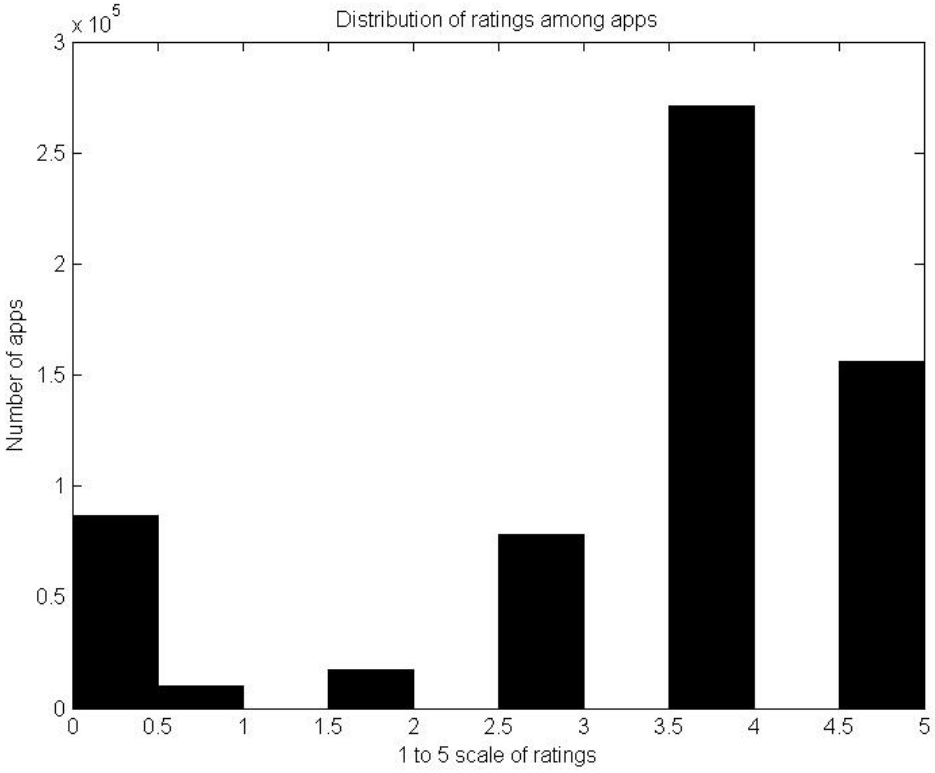

Figure 5.1: This figure shows the histogram of the apps divided ratings wise. More than half of the apps are found to be having either 4 or 5 star rating.

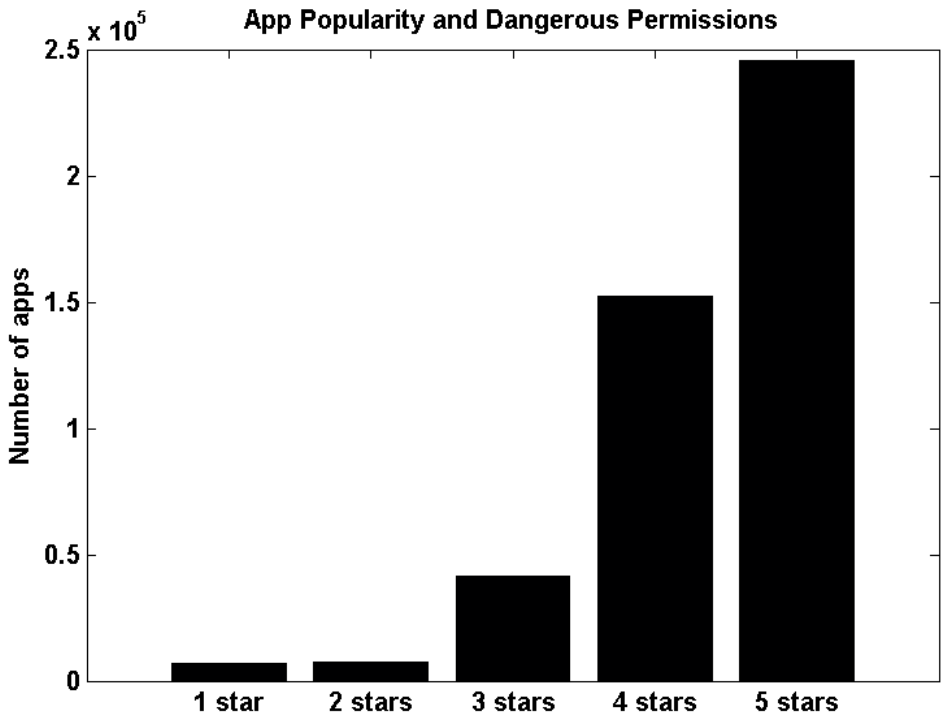

Figure 5.2: This figure shows how the dangerous permissions are divided among the apps, rating-wise. It is found that the more popular and better-rated apps demanded most of the dangerous permissions. They include *Manage Accounts*, *Send SMS*, *Net Access*, *Read contacts*, *Direct Phone Calls*

5.2 RQ2 Location access

We explore to find about the statistics of the apps which access both the network based as well as the exact location. We have correlated this with the cost of the applications, with the average rating of the users as presented in (1). For all the apps, we calculate the phi coefficient for the apps accessing the exact and the network based location.

Table 5.1 gives information on how many apps are accessing exact location and network-based location. While most of the apps do not access the location, the ones that do are found to access both network-based as well as exact location

Table 5.2 provides information on correlation between the apps and the permissions that access the locations. Having most of the values fallen into the diagonal, phi

	Network Location	No Network Location
Exact Location	0.1426	0.0520
No Exact Location	0.0632	0.7422

Table 5.1: This table gives the percentages of the apps accessing the network and the exact locations.

	Location	Ratings
Location	0.923	0.0187
Ratings	0.0187	0.923

Table 5.2: This table gives the correlation between the ratings of the apps and the permissions accessing the network and the exact locations. Since the values are more concentrated in the diagonal, they are positively correlated.

coefficient suggests that there are more apps which access the exact location *and* the network-based location. This is because there are more apps with higher ratings and they all have access to the location.

5.2.1 Findings

From our data, we have found that 19.46% of the Android applications access the network-based location of the user, while 20.58% of the apps have access to the exact location. However, 14.26% of the apps are able to access both the network-based and the GPS-based locations. Most of the apps use *both* the coarse network-based location and the exact location of the user. It was also noted that the category *Entertainment* has dominated even here.

The other categories in which apps access the exact location are Tools, Books and Reference, LifeStyle. Quite surprisingly, about 1000 apps from the category Comics are accessing the users exact location rather than his network-based location. These apps constitute about 21 % in this category.

There are more than 6500 applications which are heavily downloaded with more than 100,000 downloads and rated over an average rating of 4 which had access to either users' coarse location *or* their exact location. Google Maps was the highest downloaded app with over 1,400,000 downloads, and it accesses users' exact location. Some of the dangerous permissions are *Manage Accounts*, *Send SMS*, *Net Access*, *Read*

contacts, *Direct Phone Calls*.

We find that some free apps like Shazam, SoundHound, TuneIn Radio in the category *Music and Audio* request location permissions for identifying the local radio stations and this was done without collecting the users exact location.

Certain free apps gain money from ads, hence they would need to access the location of the devices, and ask more permissions from the users. We have correlated the costs of the apps with the access of location by the apps. It was found that the apps which request only the network-based location are much lesser than the apps which access both the exact and the network-based location. The categories for which the apps were accessing also made sense. The major categories were Travel & Local, and Tools. The *ACCESS_FINE_LOCATION* permission accessed by Travel & Local makes sense as they because that might be relevant to the application's functionality.

The apps which do not access the fine location belonged mainly to the Games category - specifically in racing, arcade, sports. These are present to provide the local-area-network connections feature in many games. Other than games, Entertainment and Media & Video dominated this category.

Figure 5.3 shows the distribution of the apps into the categories, where they access both the network-based and the exact location of the user. This has been analyzed for both paid and free apps.

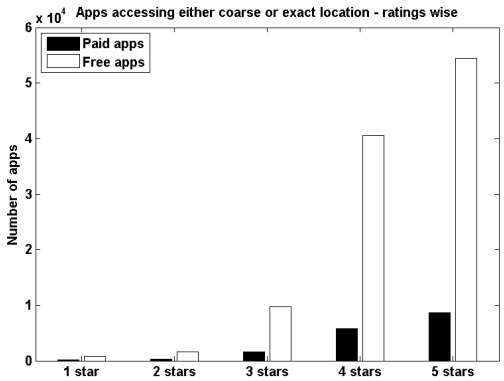

Figure 5.3: This figure shows how the apps access the network-based and the exact location of the users, rating wise.

The following figure 5.4 shows the distribution of the apps into the categories, where they access both the network-based and the exact location of the user. This has been analyzed rating-wise for both paid and free apps.

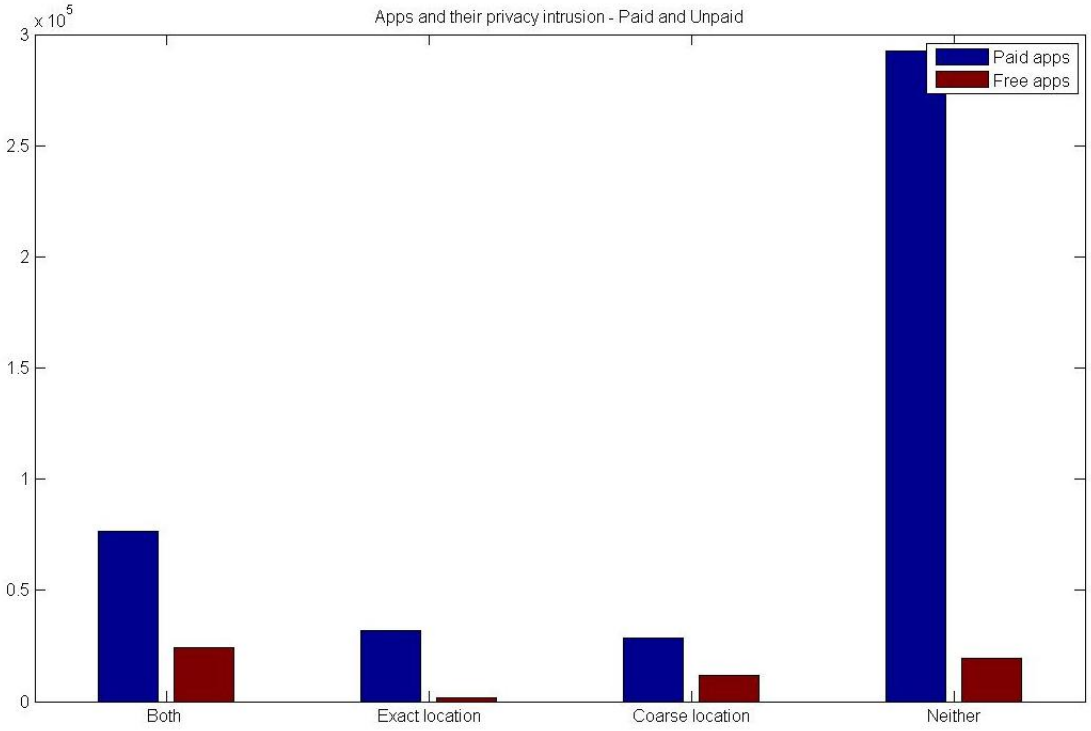

Figure 5.4: This figure shows how the apps access the network-based and the exact location of the users. This also shows the rating-wise split of paid and free apps, and how they are different in intruding the user's privacy.

Figure 5.5 shows the distribution of the apps and how they access permissions that use services that cost the user. They access both the network-based and the exact location of the user. This has been analyzed for both paid and free apps.

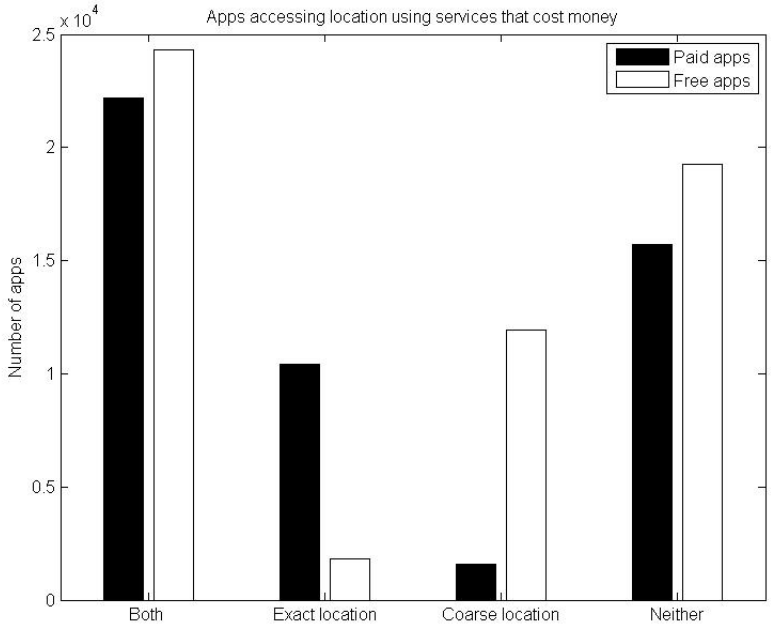

Figure 5.5: This figure shows how the ratings vary for the apps which access the network-based and the exact location of the users. This also shows statistics on paid and free apps have access to services that cost money

5.3 RQ3 Energy Efficiency

We present results of the high-risk apps that are being downloaded and explore the permissions which they pose a threat to the energy consumption of the device. Analyzing this can help us detect the malicious and unethical app development processes and how users can be aware of all such permissions. Identifying the specific permissions which can affect the categories where there is a possibility of a threat can help us pinpoint the problem. Hence, we first analyze the apps which run in the background, apps which demand a lot of hardware control and apps which run even in the sleep mode. Finally, we analyze the games category specifically.

5.3.1 Apps which run in the background

Some apps start automatically as soon as the phone is switched on and keep running in the sleep mode ie. even when the user has stopped using them. Hence, when the user is using his mobile normally, these apps are put in the multi-tasking mode. Since a lot of energy is consumed by multi-tasking, the apps consume more battery. The permissions which the app demands for this purpose are

- Prevent device from sleeping
- Use fine(GPS) location
- Make the application always run

Figure 5.6 shows that many unpaid apps are being built unreliably, and are using up users battery. There is a significant difference between the statistics of free and paid apps while they access the GPS facilities and as they prevent the device from sleeping.

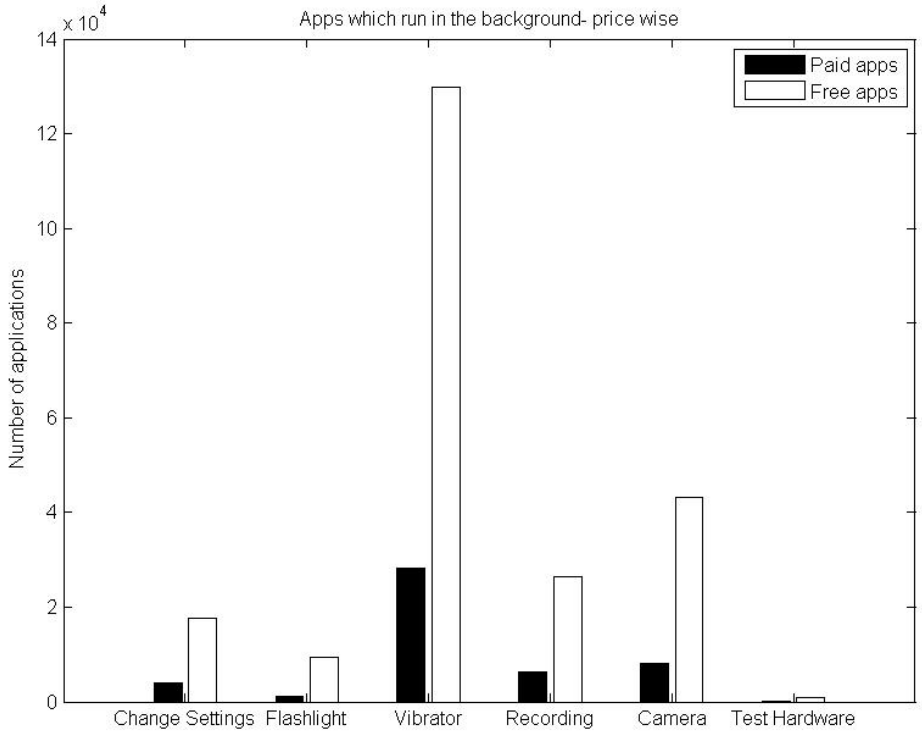

Figure 5.6: This figure shows the energy efficiencies here, and what kind of permissions are making them consume energy. It can be clearly found the free app users are being compromised of their battery as compared to the paid apps

Figure 5.7 shows that for many unpaid apps, there are differences between the highly rated apps as compared to the others.

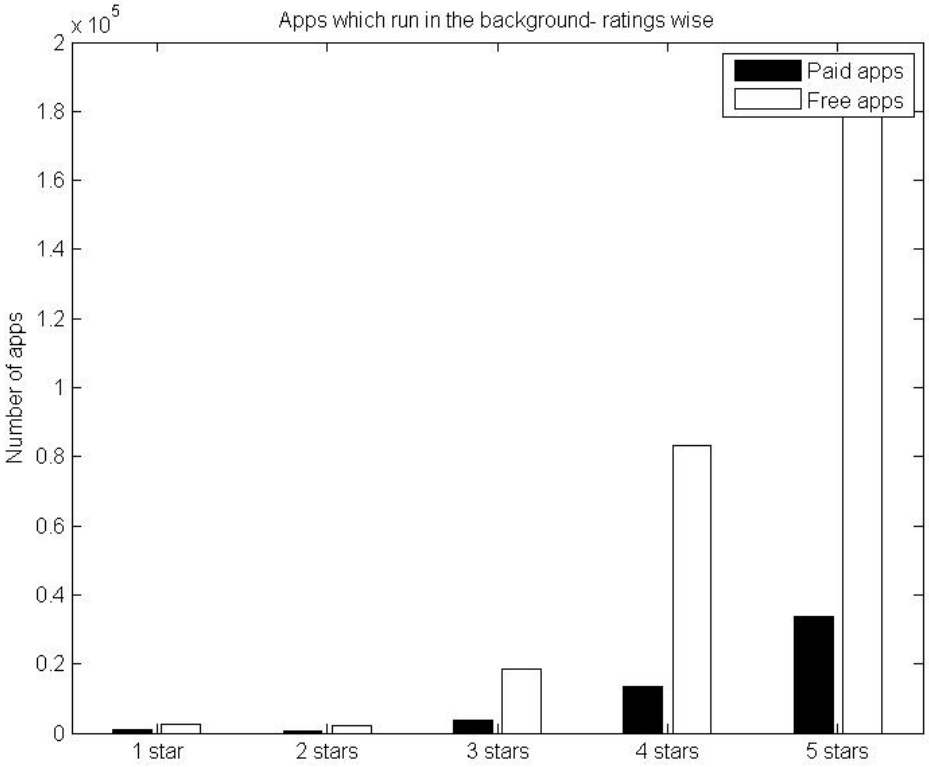

Figure 5.7: This figure shows the energy efficiencies here, and what kind of permissions are making them consume energy. It can be clearly found the highly rated apps are compromising users device-related issues.

5.3.2 Apps which use hardware

Some apps make use of GPS, Camera, Accelerometer and few sensors use the battery extensively. We have analyzed from the permissions that can justify the energy which they are using. The following permissions were considered for analysis.

- Change your audio settings
- Control Flashlight
- Control vibrator
- Record Audio

- Take pictures and video
- Test hardware

The following figure 5.8 shows the energy efficiencies pertaining to hardware-related permissions that are used by the device. It is found that the most-used permission is related to the vibration of the device when giving alerts. They constituted the major part in the free apps.

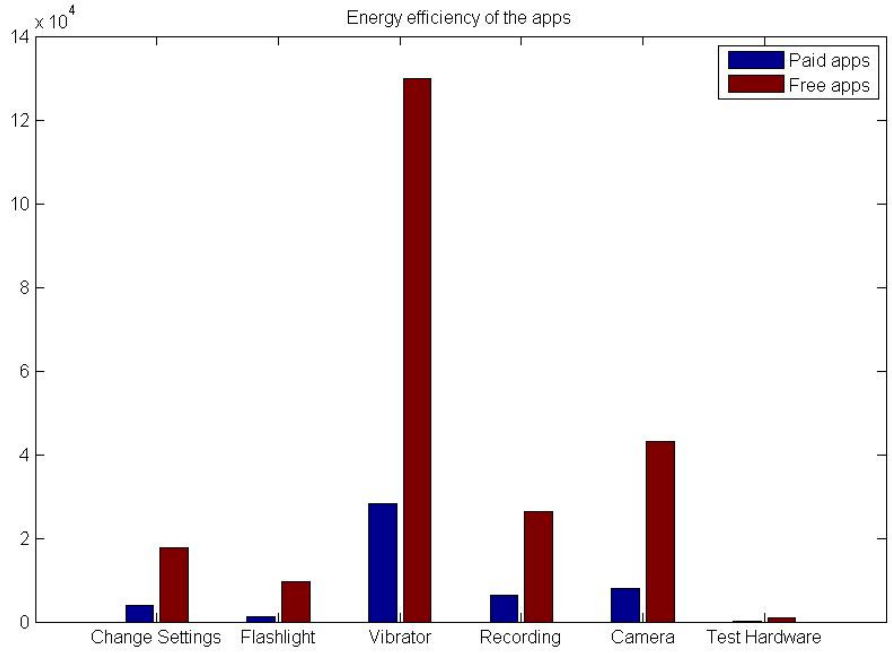

Figure 5.8: This figure shows the energy efficiencies here, and what kind of hardware-related permissions are making them consume energy.

The following figure 5.9 shows the energy efficiencies pertaining to hardware-related permissions in a ratings context.

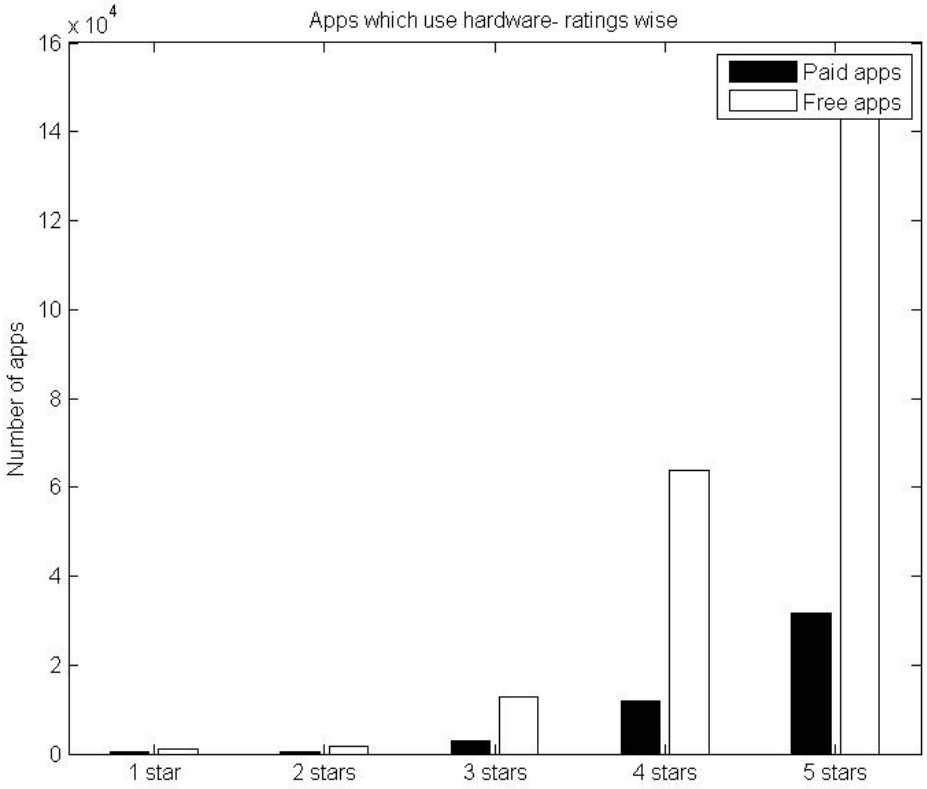

Figure 5.9: This figure shows the distribution of hardware as compared to the ratings of the apps here, and what kind of hardware-related permissions are making them consume energy. It is found that most of the free apps are being compromised of their information

5.4 RQ4 Data Usage

Most of the games would not require Internet connectivity. However, most apps require regular updates and they need to interact online to perform various functions. Apps which are used for news, weather, social networking, messaging and auction sites need to be synced with the websites to download and update the articles, images, messages, offers, prices etc. Along with these, they also sync the advertisements. This can pose some issues in handling users data connections by default. We have analyzed from the permissions that can justify the data which they are using. The following permissions were considered for analysis.

- Broadcast data messages to applications
- Download files without notification
- Receive data from internet
- Change background data usage

The following figure 5.10 shows the data-usage permissions that are used by the device. It is found that the most-used permission is the one which receives data from the internet. Since all data sessions need not to be initiated by the user, apps can request updates to the servers. These requests may be short, but can cause network congestion. This can also be correlated with data consumption and battery power.

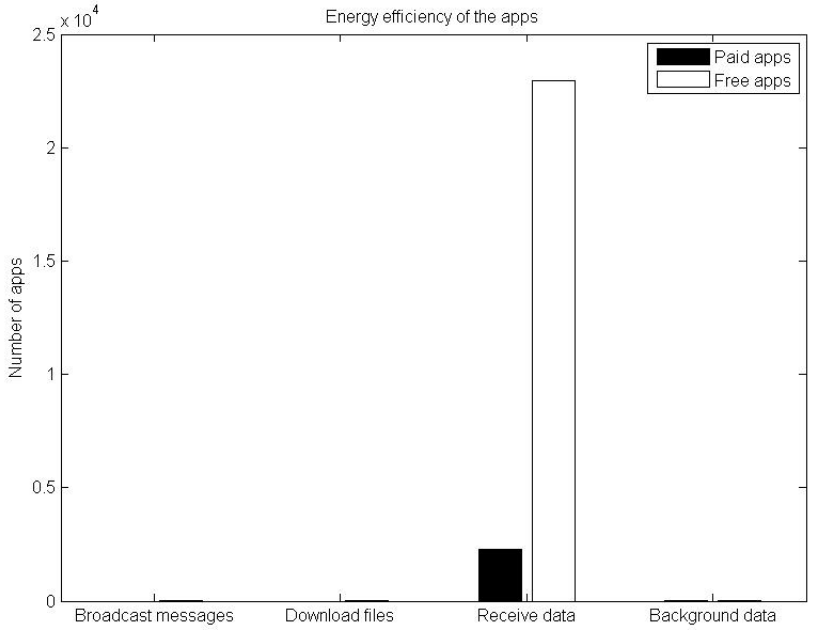

Figure 5.10: This figure shows the energy efficiencies here, and what kind of hardware-related permissions are making them consume energy.

5.4.1 Receive data permission

The following figure 5.11 shows the permission which allows the device to receive data even in sleep. This was the most used permission among all the factors which affect

data usage. These are compared against the ratings of the apps.

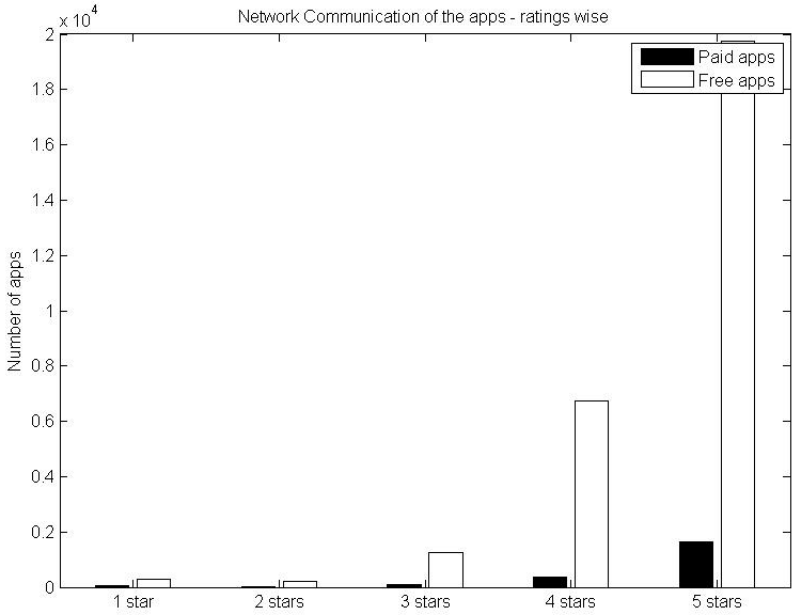

Figure 5.11: This figure shows the statistic of receiving data anonymously, with respect to the ratings from 1 stars to 5 stars. 5-star rated apps are found to be receiving data even in sleep

5.5 RQ5 Clustering

5.5.1 k-modes algorithm

The idea of the algorithm is to represent the dataset using smaller subsets of data. The algorithm takes small batches of the dataset for each iteration and assigns a cluster to each data point in the batch, depending on the previous locations of the cluster centroids.

However, instead of using the mean as the point of reference, we make use of the occurrence as the key factor. Before performing any experiments, an independent train-test split needs to be made. All experiments reported were performed on 5 different train-test splits: 10-90, 30-70, 50-50, 70-30, 90-10. On each train-test split, the k-modes clustering algorithm is performed.

This enabled us to categorize the apps as safe, moderately safe, moderately dangerous and dangerous. We have found that more than 90% of the popular apps have access to dangerous permissions which can intrude users' fine or network-based location.

Chapter 6

Conclusion

The thesis concludes with a discussion on few points from the experiments done.

6.1 Significance of the results

6.2 Accessing fine location

Few apps which do not access the *ACCESS_FINE_LOCATION* but access the network based location belonged to the gaming categories *Brain and Puzzle*, *Card Games* etc. We believe this is to give out the location-based ads as the users complete each level in the games.

6.3 Location-based ads

There are certain apps which cater to entertain people which also access their whereabouts for various reasons, though it is mainly to provide location-based ads to the user.

We believe this because the location information can help the app developers in monitoring the usage of the apps and the number of downloads according to the location. However, this raises privacy concerns to the users as most of them are either unaware of such intrusions, or they are willing to share their location data with their social contacts for various reasons.

6.4 Checking for updates

Some apps devote only a small portion of energy to their core function, while in the rest of the time, they keep checking for updates. Apps that keep waking the smartphone

from sleep mode or prevent the smartphone going into sleep mode, drain the battery. This can happen because of various reasons like checking the server for updates, new content, mail and messages, or monitoring users location.

6.5 Games which consume energy

Games, which include the categories Arcade and Action, Puzzle, Casual, Racing constitute a major portion in the total number of Google Play Store. We also found several games whose name is similar to these apps, and are being downloaded on large-scale despite not being original.

Games like Asphalt 7: Heat, Asphalt 8: Airborne, Babel Rising 3D, Candy Crush Saga, Burger, Deer Hunter are known to consume battery a lot.

Free apps like Angry birds, Words Free, Chess etc. use advertising networks to display video-based ads. Also, only some amount of their energy is focused on the actual game play and the rest is spent on user tracking and ads.

6.6 Battery Consumption

Background applications are invisible to the user, while the application might be using resources. One way to identify such apps is by understanding the notorious permissions they are demanding and clustering them together for ease. Certain major apps like the Facebook Home app was criticized for its battery consumption. Consumers save on battery life by vetting the energy consumption or monitor any unusual activity of apps on their handsets.

Handsets go into deep sleep when inactive to conserve battery life, but most of them have a no-sleep energy bug. This is a condition in which at least one component of the phone is woken up by an app and not put back to sleep due to a programming error so causing severe battery drain.

6.7 Messaging Apps

Messaging apps request permission to access users contacts. However, they do not explicitly tell the user that their contacts list would be uploaded to their servers.

They require precise locations, and request latitude and longitude coordinates should be transmitted less exactly by rounding them off to fewer decimal places. These would damage privacy the most if they turned out to leak personal information.

References

[1] `http://developer.android.com/guide/topics/providers/content-providers.html`.

[2] Steve Mansfield-Devine. Android malware and mitigations. *Network Security*, 2012(11):12–20, 2012.

[3] Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu, and Michalis Faloutsos. Permission Evolution in the Android Ecosystem. (April 2009):31–40, 2011.

[4] James Sellwood and Jason Crampton. Sleeping android: The danger of dormant permissions. In *Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices*, SPSM '13, pages 55–66, New York, NY, USA, 2013. ACM.

[5] Liu Yang, Nader Boushehrinejadmoradi, Pallab Roy, Vinod Ganapathy, and Liviu Iftode. Short Paper : Enhancing Users Comprehension of Android Permissions. 2012.

[6] Yuan Zhang, Bingquan Xu, and X Sean Wang. Vetting Undesirable Behaviors in Android Apps with Permission Use Analysis Categories and Subject Descriptors.

[7] William Enck, Damien Octeau, Patrick Mcdaniel, and Swarat Chaudhuri. A Study of Android Application Security.

[8] Yajin Zhou, Zhi Wang, Wu Zhou, and Xuxian Jiang. Hey , You , Get Off of My Market : Detecting Malicious Apps in Official and Alternative Android Markets. (2), 2011.

[9] Robert W. Reeder, Lujo Bauer, Lorrie Faith Cranor, Michael K. Reiter, Kelli Bacon, Keisha How, and Heather Strong. Expandable grids for visualizing and authoring computer security policies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1473–1482, New York, NY, USA, 2008. ACM.

[10] D. K. Smetters and Nathan Good. How users use access control. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 15:1–15:12, New York, NY, USA, 2009. ACM.

[11] David Barrera, Jeremy Clark, Daniel McCarney, and Paul C. van Oorschot. Understanding and improving app installation security mechanisms through empirical analysis of android. *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices - SPSM '12*, page 81, 2012.

[12] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11*, page 627, 2011.

[13] Q S Ntnu. Is this App Safe ? A Large Scale Study on Application Permissions and Risk Signals. pages 311–320, 2012.

[14] Asaf Shabtai, Uri Kanonov, Yuval Elovici, Chanan Glezer, and Yael Weiss. andromaly: a behavioral malware detection framework for android devices. *Journal of Intelligent Information Systems*, 38(1):161–190, 2012.

[15] Yang Tang, Phillip Ames, Sravan Bhamidipati, Ashish Bijlani, Roxana Geambasu, and Nikhil Sarda. Cleanos: Limiting mobile data exposure with idle eviction. In *Proceedings of the USENIX Conference on Operating Systems Design and Implementation, Berkeley, CA, USA*, 2012.

[16] Lei Liu, Guanhua Yan, Xinwen Zhang, and Songqing Chen. Virusmeter: Preventing your cellphone from spies. In *Recent Advances in Intrusion Detection*, pages 244–264. Springer, 2009.

[17] Yuval Elovici, Asaf Shabtai, Robert Moskovitch, Gil Tahan, and Chanan Glezer. Applying machine learning techniques for detection of malicious code in network traffic. In *KI 2007: Advances in Artificial Intelligence*, pages 44–50. Springer Berlin Heidelberg, 2007.

[18] Mario Frank, Ben Dong, Adrienne Porter Felt, and Dawn Song. Mining Permission Request Patterns from Android and Facebook Applications. *2012 IEEE 12th International Conference on Data Mining*, pages 870–875, December 2012.

[19] B. Sanz, I. Santos, C. Laorden, X. Ugarte-Pedrero, and P.G. Bringas. On the automatic categorisation of android applications. In *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*, pages 149–153, Jan 2012.

[20] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. *ACM Comput. Surv.*, 31(3):264–323, September 1999.

[21] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval: Data Structures and Algorithms*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1992.

[22] Yi-tzuu T. Chien. *Interactive Pattern Recognition*. Marcel Dekker, Inc., New York, NY, USA, 1978.

[23] Raymond T. Ng and Jiawei Han. Efficient and effective clustering methods for spatial data mining. In *Proceedings of the 20th International Conference on Very Large Data Bases*, VLDB '94, pages 144–155, San Francisco, CA, USA, 1994. Morgan Kaufmann Publishers Inc.

[24] Peter Bajcsy. *Hierarchical Segmentation and Clustering Using Similarity Analysis*. PhD thesis, Champaign, IL, USA, 1997. UMI Order No. GAX97-37041.

Chapter 7

Appendix

Permission	Description
INTERNET	Network communication. full Internet access. Allows the app to create network sockets.
WRITE_EXTERNAL_STORAGE	Storage. modify/delete USB storage contents modify/delete SD card contents. Allows the app to write to the USB storage. Allows the app to write to the SD card.
READ_PHONE_STATE	Phone calls. read phone state and identity. Allows the app to access the phone features of the device. An app with this permission can determine the phone number and serial number of this phone, whether a call is active, the number that call is connected to and the like.
ACCESS_FINE_LOCATION	Your location. fine (GPS) location. Access fine location sources such as the Global Positioning System on the tablet, where available. Malicious apps may use this to determine where you are, and may consume additional battery power.
ACCESS_COARSE_LOCATION	Your location. coarse (network-based) location. Access coarse location sources such as the cellular network database to determine an approximate tablet location, where available. Malicious apps may use this to determine approximately where you are.
WAKE_LOCK	System tools. prevent tablet from sleeping prevent phone from sleeping.
READ_CONTACTS	Your personal information. read contact data. Allows the app to read all of the contact (address) data stored on your tablet. Malicious apps may use this to send your data to other people.
CALL_PHONE	Services that cost you money. directly call phone numbers. Allows the app to call phone numbers without your intervention. Malicious apps may cause unexpected calls on your phone bill. Note that this doesn't allow the app to call emergency numbers.
CAMERA	Hardware controls. take pictures and videos. Allows the app to take pictures and videos with the camera. This allows the app at any time to collect images the camera is seeing.
WRITE_CONTACTS	Your personal information. write contact data. Allows the app to modify the contact (address) data stored on your tablet. Malicious apps may use this to erase or modify your contact data.
GET_TASKS	System tools. retrieve running apps. Allows the app to retrieve information about currently and recently running tasks. Malicious apps may discover private information about other apps.
WRITE_SETTINGS	System tools. modify global system settings. Allows the app to modify the system's settings data. Malicious apps may corrupt your system's configuration.
RECORD_AUDIO	Hardware controls. record audio. Allows the app to access the audio record path.
SEND_SMS	Services that cost you money. send SMS messages. Allows the app to send SMS messages. Malicious apps may cost you money by sending messages without your confirmation.
READ_HISTORY_BOOKMARKS	Your personal information. read Browser's history and bookmarks. Allows the app to read all the URLs that the Browser has visited, and all of the Browser's bookmarks.
READ_CALENDAR	Your personal information. read calendar events plus confidential information. Allows the app to read all calendar events stored on your tablet, including those of friends or coworkers. Malicious apps may extract personal information from these calendars without the owners' knowledge.
WRITE_HISTORY_BOOKMARKS	Your personal information. write Browser's history and bookmarks. Allows the app to modify the Browser's history or bookmarks stored on your tablet. Malicious apps may use this to erase or modify your Browser's data
RECEIVE_SMS	Your messages. receive SMS. Allows the app to receive and process SMS messages. Malicious apps may monitor your messages or delete them without showing them to you.
WRITE_CALENDAR	Your personal information. add or modify calendar events and send email to guests without owners' knowledge. Allows the app to send event invitations as the calendar owner and add, remove, change events that you can modify on your device, including those of friends or co-workers. Malicious apps may send spam emails that appear to come from calendar owners, modify events without the owners' knowledge, or add fake events.
CHANGE_WIFI_STATE	System tools. change Wi-Fi state. Allows the app to connect to and disconnect