Using Static Analysis To Display Privacy Properties Of Apps

Maria Paz Velarde

Master of Science
Computer Science
School of Informatics
University of Edinburgh
2016

Abstract

This MSc thesis aimed to improve user comprehension of permissions by combining app analysis and usability; in order to allow for users to make the least dangerous privacy choices. The chosen OS is Android as it holds a share of 82.8% of the smartphone market and past studies have found that user comprehension is alarmingly low.

This project introduces a new approach on how app's privacy related behavior is shown in Android's permission screen. Even though our results did not increased comprehension, we found interesting information about users' attitudes towards privacy and made a thorough evaluation of static analysis tools' state of the art and future work.

User comprehension was evaluated through an Internet survey using our permission screen and Google Play's screen. Our results show that 80% of the times users lacked information, they assumed the worst use of a permission; and that 25% of comprehension questions were answered incorrectly, regardless of the screen shown. Users showed low app behavior comprehension; meaning that in order to successfully incorporate such information in the user interface, future work should evaluate users' mental models on this aspect.

Acknowledgements

First, I would like to thank my supervisor Dr. Kami Vaniea, who proposed this project and gave me the opportunity to work on it. She not only guided us through the research process, but also encouraged collaboration between MSc students, UG students and security experts, allowing us to take the most out of the University's environment.

Thanks to Dr. David Aspinall, Dr. Nassim Seghir, Dr. Wei Chen, Lenka Marekova and all the other students and experts that were part of the AppGuarden meetings through the summer; for providing tools, feedback, data and a space to share our progress and talk about shared research areas.

Finally, thanks to my parents, Manuel and Jacqueline, for making it possible for me to come to The University of Edinburgh and my sister, Andrea, for the unconditional support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Maria Paz Velarde*)

Table of Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

Android OS's presence in the smartphone market keeps growing as well as the amount of apps available for it. Android OS based devices hold a share of 82.8% of the smartphone market, as of August 2015 [6], while Google Play, the official app store, hosts over 2 million apps as of July 2016 [7]. The wide range of features that make applications attractive to the customers can require access to personal information like location, phone number or phone contacts list.

The amount of private information hosted by these devices and the ease of developing and publishing applications introduces new kinds of security and privacy concerns [8]. These new concerns range from "Who is my health application sending information to?", "Is this information encrypted?" to "Have I agreed to being tracked while I exercise?", "Have I allowed ads to know my device's ID?", "Who am I allowing to access my personal information?" [9]. Personal security using apps is of paramount importance and clarity for the users is necessary.

About 90% of the apps available in the Android market are free and most of them rely on ads to generate income, making users' private information available to third parties [7, 10]. Users awareness of what an app can do with their personal information is low, as only 3% of Android users understood permissions in Felt's studies [11], leading to dangerous privacy choices [8, 12, 13]. Privacy is a delicate issue as it is subjective to the user; some people may like ads or they may prefer to have ads rather than paying for an app, others may believe that they have no sensitive information in their phones and may not care about the permissions at all.

Users are not left completely misinformed, before installing an application Google Play shows a screen with the permissions required by it [14]. This screen however, lacks information about how the permissions are used, their purpose or who is requesting them: the app provider or a third party [10, 15]. There's been a lot of research in app analysis, both dynamic and static, [4, 9, 16, 17, 18, 19]; such research takes a more technical approach far from the end-user, or tries to make security decisions for the user. However, it is possible to use these app analysis tools to extract privacy related behavior from apps and display it to the end-users, allowing them to make more informed privacy decisions.

1.1 Related Work

In 2012, Jung *et al.* conducted a study with 20 Android users on sharing personal informaiton in which three aspects of concern were found: (i) context[1], (ii) frequency and (iii) whether the information goes to third parties [20]. In 2012, Felt *et al.* conducted an Internet survey of 308 Android users; their results showed that only 3% of the respondents had full comprehension of the displayed permissions [11]. Concluding that besides the missing information in the permissions screen, users comprehension of the displayed information is low.

In 2012, Lin *et al.* proposed to correct users' mental model by pointing out misconceptions. Mental models define people's perspective towards a problem or situation [12], in this case they define the users expectation of what an app can do with the permissions asked. Users expectations about a permission were crowdsourced and compared with the permission's purpose. They used Taintdroid[2] to find out how sensitive information was used (the purpose). They designed a screen that pointed out the most common misconceptions, displaying the percentage of users that were not expecting certain behavior. Their results showed increased comprehension and awareness about possible privacy threats and reduced the amount of time spent in the screen; however this approach requires crowdsourcing expectations about each app's permissions.

In 2015, Paturi *et al.* proposed a design that aimed to help the user identify where privacy threats came from: app provider or ad libraries. They defined three privacy categories: identity, location and search queries[3]. They extracted the apps information using static and dynamic analysis. Their permissions screen showed the three categories for the app provider and for the third parties. Each category had an safety/threat

[1] Context refers to when in the app lifecycle the permission is used. Jung *et al.* found out that users were concerned about permissions being used in the background, without their knowledge [20].

[2] Taintdroid [16] is a dynamic analysis tool that tracks the flow of sensitive information within an application.

[3] Android applications configuration that allows search with the system's assistance [21]

icon that represented possible privacy threats. Their results showed increased comprehension about the source of privacy threats. They also found that people wanted more details about the information shared and were interested in additional categories like SMS, contacts or calendar.

1.2 Project Goal

This project aims to bridge the gap between app analysis and users' comprehension of permissions. It may be possible to increase users' comprehension by extracting and displaying apps' privacy properties, in an scalable way. At the moment the University of Edinburgh's AppGuarden[22] project has experts working on app analysis and usability, making it an ideal environment to explore the gap between those areas. Using static analysis tools developed by AppGuarden[22] experts, I extracted from the apps information regarding the use of the permissions and their context. With the integration of these tools was possible to generate a permission screen that considered three aspects: (i) context, (ii) category and (ii) whether the information is used by an ad library.

This project's contributions can be classified in two areas: app analysis and human computer interaction (HCI). Testing and working with several static analysis tools we were able to find and understand their limitations. Static analysis tools development was outside the scope of this project, but we found how these limitations related to Android's security architecture and point out possible solutions. Working around them, we proved that is possible to extract and display apps' behavior in an scalable way using static analysis.

In the HCI area, we introduced two aspects to the current permission screen: context and ad libraries information. We evaluated the current permission screen (control group) and our proposal (experiment group) with an Internet survey on 203 app users. The survey considered three aspects: permissions, permissions with context and permissions with ad libraries. Our results show that for both permission screens at least 25% of the users answered incorrectly and 80% of the control group took a more privacy-conservative approach when answering questions that involved context or ad libraries. Besides the low comprehension on permissions there is a research gap on users mental models about app behavior, which needs to be expolred in order to introduce contexts in a comprehensible fashion.

1.3 Paper Outline

Chapter 2 begins with a description of the project's goals and different areas covered. Then, it provides an overview of the methodologies used or related to each area: design and evaluation methodologies, app analysis, android permissions and users comprehension of permissions. *Chapter 3* describes the requirements gathering process; the group affinity diagraming's plan, execution, results, conclusion and recommendations. *Chapter 4* goes into design and implementation. It analyses the tool testing and selection process: test apps' development, tool testing, bug reporting, conclusion and limitations. It delves with the iterative design process: screen's appearance, labels and categories selection. Finally, it covers the integration of the available tools and the development of the display generator. *Chapter 5* contains the evaluation methodology, survey planning and results analysis. Finally, *Chapter 6* discusses the conclusions, critical evaluation and offers recommendations for future research.

Chapter 2

Background

This chapter describes background knowledge needed for the development of this project, previous work and relevant literature. The areas covered lie between Human Computer Interaction and Computer Security, specifically Android Permissions.

The chapter begins by highlighting the importance of information security awareness. The next section describes related work on Android permissions' comprehension and privacy. As this project's HCI component requires the use of design and evaluation methodologies, the following section gives an overview of these methodologies. The next section explains the Android applications architecture and security model. Finally, we describe app analysis methodologies and approaches taken to represent app behavior.

2.1 User Awareness

Past studies have found that security is a concern for users, but they lack sufficient information to make the best choices in that respect [10, 11, 20, 23, 24]. Wash interviewed users to identify their mental models towards information security threats and found that users' perception of security affects how they respond to security expert advice [12]. Bulgurcu *et al.*'s study showed that users' attitude towards security policy compliance depended on their beliefs about the outcomes and an overall assessment of consequences [25]. In Kelley *et al.*'s study some users were not concerned about granting access to their private information due to the belief that Google/Android was watching out for their safety as customers [26].

Security can be improved by secure programming practices, use of encryption or preventing vulnerabilities, but privacy relies largely on what the user agrees to or expects. If a user wants to find out restaurants nearby using Google Maps, they will have to agree with the app having access to their location. In Android apps, users should align their expectations with the permissions' purpose.

The overall permissions model needs improvement. Yee [27] proposed considering security from the beginning of the software design process, aiming to correct mental models through usability. Felt *et al.* [28] proposed a framework on how to choose permission-granting mechanisms, where the user is not overwhelmed with information. My dissertation proposes not to change the permission model but to improve how permissions are displayed. User misconceptions can be clarified, our approach aims to do so through the user interface.

2.2 Users' Comprehension of Permissions

In 2011, Kelley *et al.* conducted interviews to evaluate user comprehension of permissions. In 2012, Felt *et al.* conducted an Internet survey and lab study to evaluate the same. Both independent studies showed that user comprehension was low, only 3% of the surveyed users answered all permission comprehension questions correctly. Balebako *et al.* developed a system that informed the user about privacy leaks through alerts and used it to make a qualitative study of the gap between user expectations and leakages. Study participants were surprised by the information leaked and its frequency [15]. Jung *et al.* performed a similar study that found three particular aspects that concerned users: (i) permissions being used in the background (context), (ii) permission use frequency and (iii) whether private information is used for the app's functionality or if it goes to third parties [20].

2.2.1 Approaches

Approaches taken in this area can be classified in: changes to the permission screen or changes to the permission-granting mechanism.

Modify Permission Screen

Lin *et al.* proposed to educate the user by pointing out misconceptions. They designed a permission screen that showed the permission's purpose and the percentage of users that did not expect it. Expectations were crowdsourced and the permission's purpose was extracted from the apps using static analysis [8]. This solution improved user comprehension and decreased the time spent in the permission screen. Yet, this approach would require crowdsourcing expectations for every apps' permissions making it unscalable.

Kelley *et al.* went further into the implications of comprehension. They state that by the time the user clicks the download button they have already decided to download the app and points out that there is no cancel button in the permission screen [24]. The first proposed solution was to add a permissions checklist in the app's main screen. The checklist showed which private information was requested by the app. This approach helped users to make a more informed decision, however they also found that it was complicated to keep the checklists' terms simple while also providing all the information the user needed.

Kelley *et al.*'s second approach was to change the permission screen. Instead of displaying permissions, two columns of three privacy granules (Identity, Location, Search) would be displayed and a cancel button was added. The first column showed the permissions used by the app provider, while the second column showed the same information but for third parties. The new screen increased comprehension in users and at the same time increased their concern about the permissions within the granules. Users wanted more detailed privacy granules; they showed interest in information like SMS, Contacts and Calendar [10]. However, this design either assumes that all third parties are ad libraries or that users can understand that some third parties are used in the core app functionality and are not necessarily ad libraries. AppBrain[1] shows statistics on libraries and social sdks generally used by apps.

Change Permission Request Mechanisms

Yee proposed an alignment of security and usability from the beginning of the system design through using two methods: security by admonition and security by designation. The main difference between the two is whether to prompt a warning or not. The main goal is to avoid distracting the user from the primary task. In security by admonition, a prompt is shown to the user either warning or requesting confirmation. Security by designation assumes, because of the performed action, that the user agrees with the permission and grants it automatically.

Felt *et al.* treat user attention as a limited resource that should not be wasted. Instead of choosing one permission granting mechanism, they provide a framework for

[1] http://www.appbrain.com/stats/libraries/dev

platform designers on how to choose the most appropriate mechanism. These guidelines follow these principles: conserve user attention by only requesting it when necessary and avoid interrupting the users' primary task. Permission-granting mechanisms used by this framework are: automatic grant, Trusted UI, confirmation dialog and install time warning.

An automatic grant does not require the user interaction to grant permission. A Trusted UI element is an element controlled by the platform rather than the app; it grants a permission through the users' actions e.g. clicking on a "Send SMS" button. Confirmation dialogs are runtime alerts and an install-time warning are Android's approach (except for Android 6.0).

2.3 Design & Evaluation Methodologies

This project involved several evaluations during its development. The requirements-gathering phase involved group affinity diagram with security students, and the screen design phase involved a focus group with security experts. The final evaluation with end users involved A/B testing of Google Play's permission screen and our proposal through an Internet survey. This section provides an overview of these methodologies.

2.3.1 Affinity Diagram

Affinity diagramming allows for meaningful cluster observations and requirements [29]. Observations or requirements are written in sticky notes and clustered by participants. Sticky note authors remain anonymous, which removes biases. Therefore, all of them are fully considered.

This methodology was used for the requirements-gathering stage as it allowed us to find out how users perceived permissions within an app and also identify privacy concerning behavior. The resulting clusters allowed us to identify the users' concerns, their thoughts about privacy and permissions and then to classify such information.

2.3.2 Focus Group

Focus groups are used to attain insight about a product or service; participants should be carefully recruited as they need to identify each other as peers with whom they can express their opinion with confidence [29]. A focus group was made with security experts in order to evaluate if the screen expressed app behavior regarding permissions' use in a comprehensible way.

2.3.3 A/B Testing

A/B testing allows for comparison of two approaches or designs to see which achieves a goal better [29]. Our goal was to increase comprehension. Therefore, our final evaluation was to perform A/B testing on user comprehension of Google Play's permission screen and our proposal. For this project, this testing was done through an internet survey.

2.4 Android

The chosen OS was Android because it occupies 82.8% of the smartphone market [6] and because of the availability of the static analysis tools built for it. This section starts with an overview of android's system architecture, then it describes the major application components and their security implications. Finally, an overview of Android permissions is provided.

2.4.1 System Architecture

Android system has a layered architecture, as depicted in Figure 2.1. Android applications and Framework are developed in Java, this code is compiled and converted to *dex* (Dalvik Executable) files, which are interpreted by the Dalvik virtual machine. The whole system is built on top of the Linux kernel.

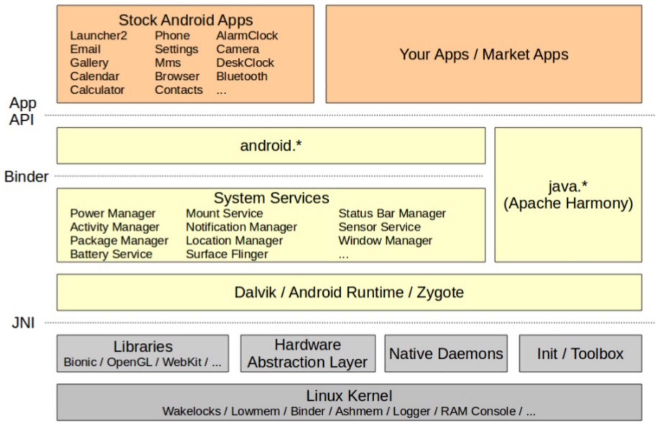


Figure 2.1: Android System Architecture [1, 2]

2.4.2 Permission Model

The system implements two permission models: user-based permissions and application-based permissions. The user-based model is inherited from Linux at the kernel level, it utilizes user ids and groups. The groups control access to protected resources, e.g. processes belonging to the sdcard_rw group will be able to read and write the /sdcard directory. Processes within the system get assigned an Android ID. There is a reserved range of AIDs for apps and the application-based model is built on top of the user-based one [2].

There are three types of Android permissions: API permissions, file permissions and Interprocess Communication (IPC) permissions. API permissions usually work within the Android Framework layer, but there are exceptions. Some permissions can map to functionalities in the kernel layer. When a permission mapped to a kernel level functionality is used, the app's ID is added to the group that has access to that permission. Examples of these are INTERNET and BLUETOOTH [2, 30].

Filesystem permissions work in a similar way but they request access to shared storage. Each app has its own private data storage, when requesting a permission to access a shared data storage, the permissions are added to the group that is allowed to use that directory e.g. WRITE_EXTERNAL_STORAGE grants access to the /sdcard directory.

The Android system handles Interprocess Communication through Intents. Intents can be used to send data within or between apps. IPC permissions allow the sending or receiving of system-wide Intents. IPC endpoints can be Content Providers, Broadcast Receivers or Services.

2.4.3 Application Components

In order to understand the different ways Android permissions can be used we should take a brief look at the application components. Android applications consist of four types of components and Intents. Application components can be: Activities, Services, Content Providers or Broadcast Receivers [3].

Intents

Intents are asynchronous messages that bind the application components. They can be used within an application or system-wide [30]. System-wide intents require the use of permissions, for example applications that use Google Cloud Messages require the

C2D_MESSAGE permission, which will allow binding the app to the com.google.android.c2dm.intent.RECEIVE intent.

Services

Services are background processes that run independently from the app's main threat. Services can send or receive intents and use API permissions; this component does not have user interface.

Content Providers

"Content providers manage access to a structured set of data" [3]. These structured data stores can be SMS Inbox, phone's contact list, etc. Applications use Content Providers to encapsulate access to their data. When an app implements a Content Provider it can allow other apps to use it and define new permissions to request access to the Content Provider.

Broadcast Receivers

This component receives a specific type of Intent. The previously mentioned Google Cloud messaging Intent is associated with a Broadcast Receiver, this specific Intent was implemented to prevent having apps making constant unnecessary requests to a server. Instead, the use of the C2D_MESSAGE permission allows for the server to send an Intent when a change has been made in the cloud. Another example of a Broadcast Receiver is the "Run at start up" permission, which allows for an app to have a broad cast receiver for the ON_BOOT_COMPLETE Intent.

Activities

Activities are the user interface main component, it is the window or screen within an application. Activities have a lifecycle and their visibility status depends on their status within the lifecycle, refer to Figure 2.2. An application can use an API permission at any moment during its lifecycle in a broadcast receiver or in a service.

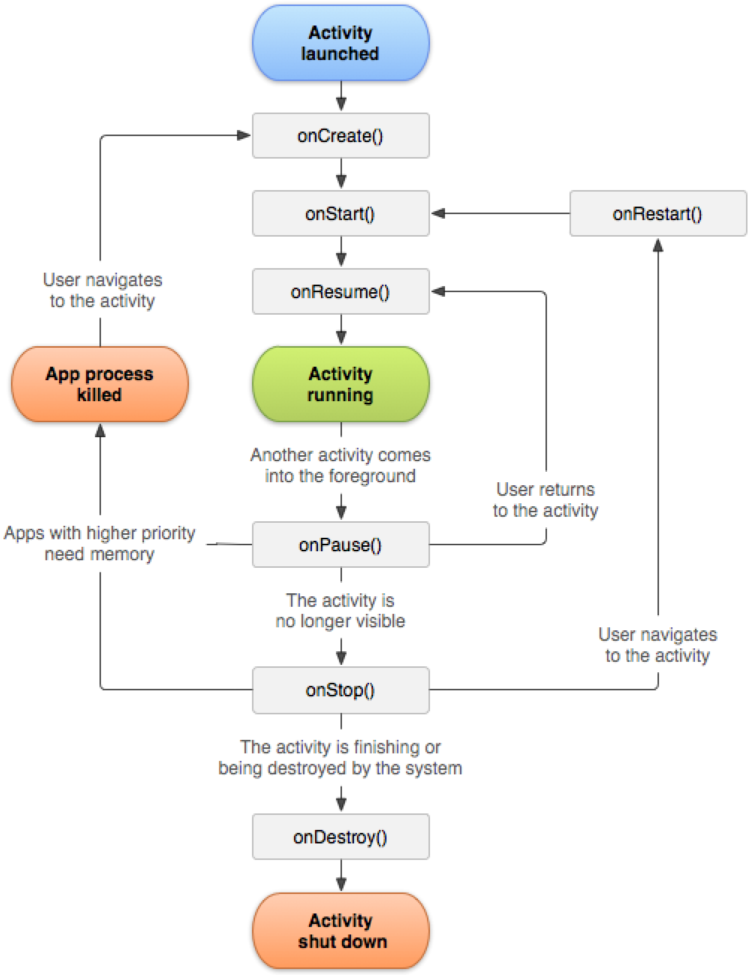

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Figure 2.2: Android Activity Lifecycle [3]

2.5 App Analysis

There are two approaches in app analysis: static and dynamic. Dynamic analysis consists in runtime monitoring the apps' behavior, this approach's results are more accurate, meaning there are no false positives. However malware can detect runtime monitors and malicious behavior could go undetected [4]. This approach needs to test all possible inputs which require several test runs.

Static analysis consists in an evaluation of the source code and all the possible outcomes. This approach's advantages are: requires less time, does not need several runs, easier to scale. Its main disadvantage is that is over conservative. For this project's purposes is an advantage as we need to know what an app may be capable of doing.

2.5.1 Related Work

The AppGuarden project [22] at the University of Edinburgh developed the static analysis tool EviCheck [5], which verifies policy compliance through the analysis of an decompiled *apk*. This tool's functionality is described in more detail in the next section.

TaintDroid is a dynamic analysis tool that identifies sensitive information sources. Once information comes out of those sources, it is followed across program variables, methods or files, until it leaves the system [16]. As private information goes through the program, methods and variables that hold such information are labeled or *tainted*; this type of analysis is called Taint analysis.

FlowDroid is another taint analysis tool that uses static analysis instead of dynamic analysis. It models the apps' lifecycle and follows sensitive information flow. Figure 2.3 provides a example of how static analysis works, specifically FlowDroid. source() is a sensitive information source. FlowDroid follows the now *tainted* resource forward in the method and backward to the parent method. *Tainted* resources are tracked until they leave the system in what is called a sink().

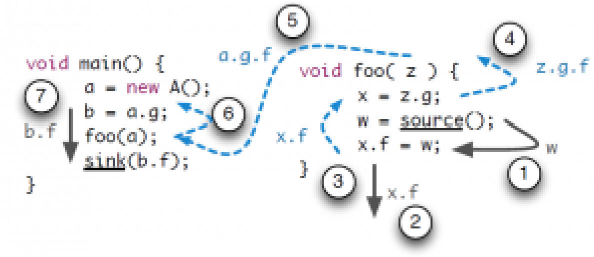


Figure 2.3: FlowDroid Taint Analysis [4]

Finally, we have Pegasus, which combines static analysis, model checking and runtime monitoring. This tool is similar to EviCheck as it verifies policy compliance. In this case, the policy is represented by Permission Event Graphs. These graphs allow for representing the app behavior in terms of policy and sequence of events [19]. We were not aware of this latest tool until the technical evaluation phase of the project, but it would be interesting to attempt using it to generate a permission screen based on a sequence of events.

2.5.2 EviCheck

EviCheck is an static analysis tool created for the verification, certification and generation of security policies. It is built to verify and certify the absence of *bad behavior* through the process shown in Figure 2.4 [5]. The verifier component generates a certificate, while the checker component validates an exiting certificate. EviCheck's normal output, in addition to the generated certificate, is a list of the violated rules and where they were called in the code. It's input parameters are: an *apk* file and a security policy.

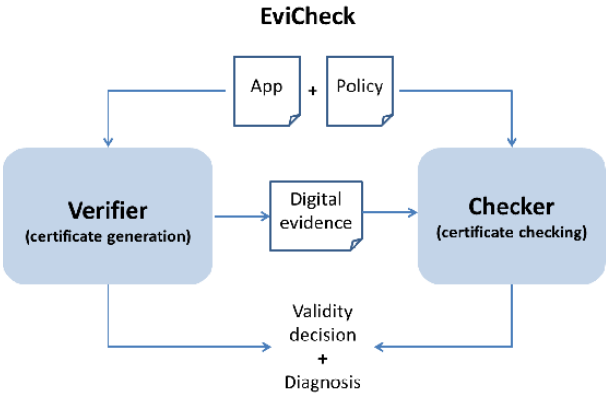


Figure 2.4: EviCheck's verification and certification processes [5]

Security Policies Overview

EviCheck's security policies are defined by one or more rules, where each rule has the following structure:

$$rule := C \quad : \quad P$$
$$\text{where} \quad C : \{\neg context, context\}^*, \quad P : \{\neg permission\}^*,$$
$$\forall context \in \text{EviCheck's contexts},$$
$$\forall permission \in \text{Android permissions}$$

This structure defines contexts where the permissions can not be used, hence the negation of the permissions. Multiple contexts allow for the use of exceptions. For example, the following rule forbids the use of the Record Audio permission in all entry points except onClick.

```
EVICHECK_ENTRY_POINT ~EVICHECK_ONCLICK_HANDLER : ~RECORD_AUDIO
```

The use of one context forbids the use of the permission in that context. The following rule forbids the use of the Camera on a button click.

```
EVICHECK_ONCLICK_HANDLER : ˜CAMERA
```

There are two types of rules: OR-rules and AND-rules. All rules are AND-rules by default, unless : ∨ is used between the *C* and *P*. OR-rules are used to define mutually exclusive permissions within a context. The following rule allows for either Record Audio or Location on a Service, but not both.

```
EVICHECK_SERVICE_METHOD :V ˜RECORD_AUDIO ˜ACCESS_COARSE_LOCATION
```

EviCheck's Output

EviCheck's output shows the broken rule and the method where the permission was used. The following output corresponds to the violation of the rule "not Camera on button click".

```
Rule 155  ==>  set(['EVICHECK_ONCLICK_HANDLER']) : set(['˜CAMERA'])
Policy violated! Tag CAMERA is in Lmariavelarde/cameraapplication6/
MainActivity$1->onClick(Landroid/view/View;)V
```

Instead of using the tool to verify a security policy we decided to use the tool to extract apps' behavior by the definition of over-restrictive policies. *Chapter 4* will describe how this tool was evaluated and used to meet this project's purposes.

Chapter 3

Requirements Gathering

The current Android permission screen has the permissions grouped by resources and faces the two previously mentioned issues: low comprehension and lacking information. The goal of this experiment was to define new groups according to privacy concerns. The chosen methodology was affinity diagraming, which is a qualitative method that allows us to analyze the users' behavior and attitude towards a certain process or design by clustering information [29].

The first section of this chapter covers the affinity diagram planning: definition of sticky notes content and experiment participants. The next section describes the affinity diagram session and its structure. The following section consists of the analysis of the results. Finally, the chapter gives the conclusions and recommendations for future work.

3.1 Methodology

Sticky Notes Content

To find out how users mapped privacy concerns to app behavior, we chose to describe app behavior in terms of context and permissions. Context refers to a method within the application lifecycle that triggers the use of a permission, a permission can be an API call or a permission constant as defined in the Android documentation [31].

All the permission constants to be found in the Android documentation were listed [31], as well as the contexts used by EviCheck [5]. From the list of permission constants were removed deprecated ones and permissions that can only be requested by the system. For each permission and context had description based on Android documentation [3, 31]. To verify the validity of these lists and the descriptions we had a meeting with two App Guarden[22] experts.

At the meeting a first draft of the sticky notes was presented, with the feedback provided by the app analysis experts and Dr. Kami Vaniea the final version of the sticky notes was made. Two types of sticky notes were made, one for context and one for permissions. Both had the type on top, the name of the context or permission in the middle and below a description, as shown in Figure 3.1.

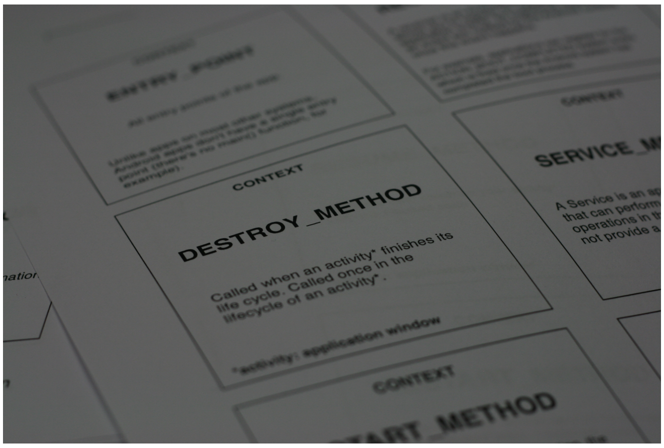

CONTEXT
DESTROY_METHOD
Called when an activity* finishes its life cycle. Called once in the lifecycle of an activity*.
*activity: application window
CONTEXT
SERVICE_M
CONTEXT
START_METHOD

Figure 3.1: Final version of sticky notes design

Participants

The participants were a mixed group of Computer Science students and researchers: six MSc students, three UG students working on security projects and three security experts. Gender wise, the group consisted of five females and eight males.

The target user group for the permission screen are people in general, not necessarily with technical background. Any user can have privacy related concerns, but technical knowledge is required to understand application behavior; making the security students an ideal group for the affinity diagram exercise.

Materials

Materials used included: sticky notes, markers, tape and paper. Two sets of sticky notes were prepared in advanced; contexts and permissions. During the session blank sticky note pads were given to the participants to fill as described in the next section. Markers tape and paper were used to describe clusters.

3.2 Affinity Diagram Group Session

To ease the mental mapping between privacy concerns and permissions the session had the following structure: i) brainstorming, ii) low level clustering, ii) higher level clustering. At the beginning of the session blank sticky note pads and markers were given to each participant.

Brainstorming

First we needed participants to think about permissions and actions that can trigger the use of a permission and concerning app behavior. Three questions were asked at the beginning of the session, each followed by a couple of minutes to write down comments or answers on the blank sticky notes.

Questions asked to participants:

1. Name 3 application permissions.
2. Describe a situation that would trigger or cause a permission to be used.
3. App behaviors you are not comfortable with.

Low Level Clustering

After the brainstorming session sticky notes with contexts and permissions were given to the participants. For this next part participants were asked to paste the sticky notes on the wall and to start grouping them, as shown in Figure 3.2. As is standard in affinity diagraming, no talking was allowed at this stage except for clarifying questions such as handwriting. In case some sticky note seemed to belong to more than one cluster, participants were allowed to write down that permission, context or concern on a blank sticky note and to put it in any applicable cluster.

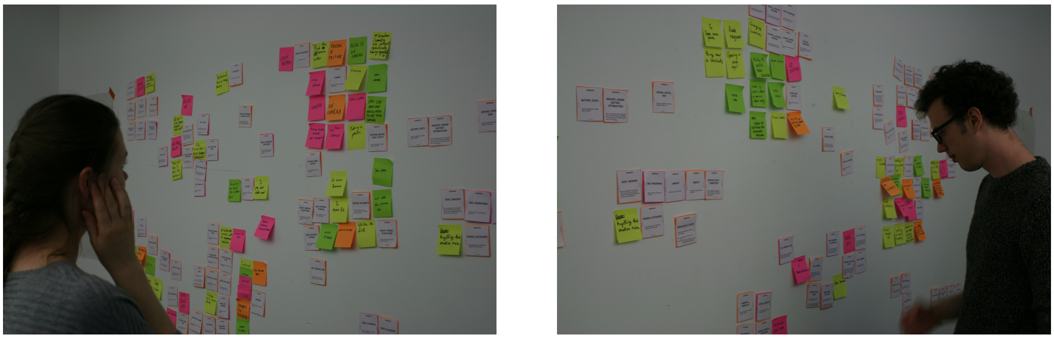


Figure 3.2: Low level clustering

Higher Level Clustering

In the next part of the session talking among the participants was allowed. Paper and tape were provided in order to write descriptions for the clusters and to make sub clusters without damaging the wall, Figure 3.3.

Figure 3.3: High level clustering

3.3 Results

After an hour long affinity diagraming session we got twenty three clusters. Because of the time constraints, not all permissions were clustered and not all clusters got named and sub clustered during the session. Results were analyzed in two ways: individual clusters analysis and with a meeting with experts.

Clusters Analysis

Based on the results, descriptions were added to the clusters and clusters with similar content were merged. Within each cluster concerns and sub clusters were identified. Table 3.1 contains the final results.

Name	**Description**	**Sub clusters**
What is this	Permissions too difficult to understand even for people with computer science background.	
Audio	Permissions to use microphone, modify settings and recording things in background. **Concerns:** Users showed concern about actions happening in backgroung (e.g. with no button push).	• Settings • Trigger (button push or background) • Audio output

Media	Permissions regarding media: access to photos, taking screenshots and access to camera. **Concerns:** Users showed concern about actions happening in backgroung (e.g. with no button push).	• Camera • Background processes
Device settings	Permissions that could read or write settings and use device as root user. **Concerns:** Access to settings without notification.	
Accounts & Device Info	Read/write device information, access to accounts. **Concerns:** Sensitive information getting sent over the internet.	• Device info • Accounts
Contexts	Almost all the contexts ended up in this cluster and the permissions: check license, system alert window, delete cache files, reorder tasks and set always finish.	
Device Stats	Permissions about usage stats, logs and set wallpaper.	
Documents / Storage	Permissions to access files, external storage and to manage removable storage.	• Read files • Storage r/w • Removable storage
Datetime	Read/write calendar, set time.	
Billing	Permission to use Google play billing library. **Concerns:** Make any payment by default	
Location	Permissions to use location: access, save or modify settings. **Concerns:** Store or send location updates.	• Accessing location • Retention • Permission • Modify location • Secondary use of location • Maps
Wi-Fi	Access to Wi-Fi configuration.	
Bluetooth	Pair or find bluetooth devices	

NFC	Permission to use NFC.	
IR	Permission to transmit IR.	
Internet	Use internet. **Concerns:** Unknown reason for internet connection.	
SMS	Read or send SMS. **Concerns:** Cost of sending an SMS.	• Read • Send
Ads	Notifications, status bar, ads. **Concerns:** Information leaked through ad library, install things.	
Calls	Make calls, process outgoing calls.	
Voicemail	Read, write or add voicemail.	
Contacts	Read/write contacts or call log. **Concerns:** Sending contact information.	• Read • Write
Unclustered	Permissions related to sensors, configuration or hardware.	

Table 3.1: Clusters made during the affinity diagraming session

Results Analysis Meeting

A meeting was held with Dr. Kami Vaniea and a student working extensively with apps and EviCheck to further process and interpret the results of the Affinity Diagram. During the meeting, two main privacy concerns were identified: information leaving the device and unwanted advertisement.

Information Leakage

Every concern raised related to private information expressed the users' worries about information leaving the device with or without their permission. Most users used the words "things happening with / without button push", "things happening with / without notification". During the meeting, the clusters and concerns were organized as diagram which led to Figure 3.4.

The diagram shows three contexts: not visible to the user, not necessarily visible but while using the app and visible to the user. Concerns were raised within each context, with a condition or action before the use of the permission: “with/without” a button click or “with/without” my permission. Users did not like at all things happening in the background or without them knowing.

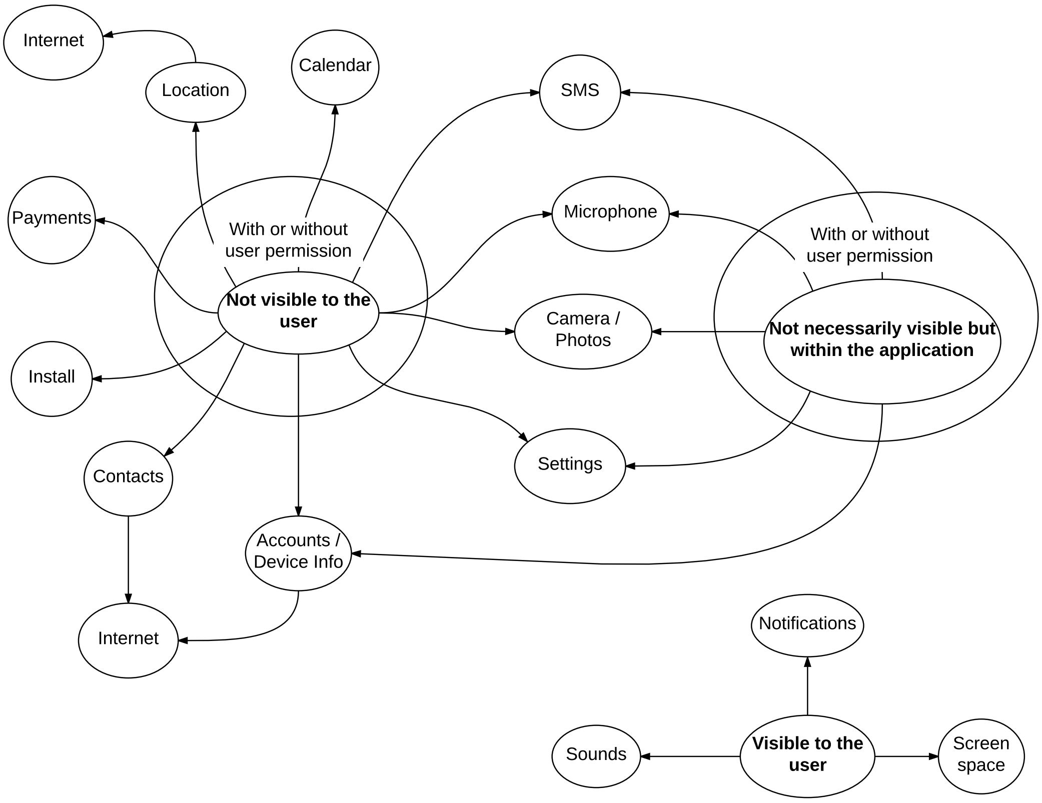


Figure 3.4: Clusters diagram

Each of this contexts can be mapped to a callback in the Android activity lifecycle. Android can do things in the background with: asynchronous tasks, services and broadcast receivers. The only one that requires the app to be open to be executed is the asynchronous task, all the others can be used completely in the background.

Even though the concerns could be mapped to a context, all the context sticky notes ended up clustered together. Also, users were not concerned about the application reading their personal information, but they were concerned about that information leaving the device, which is why Internet was not added on its own but interacting with other permissions.

Advertisements

Ads were mentioned by the users as something annoying and were mapped to the status bar or with notification permissions. In this aspect users cared more about the intrusion to their personal space.

3.4 Conclusions

Permissions more related to security like “install packages” or “use device as root user” raised no concerns during the experiment. Participants seemed more preoccupied about their personal information like contacts or multimedia. Even though several permissions were able to write or change information, participants’ concerns were focused on confidentiality rather than on the integrity of their information, this also could be caused by lack of comprehension of the permissions.

As a result we had two ways of expressing permissions: context grouping and sequence of events.

For the context grouping approach we defined following groups:

- Triggered with button click
- Triggered in background but with application visible
- Triggered in background, without the application visible
- Granted but not used

3.5 Future Work

The purpose of the questions “Describe a situation that would trigger or cause a permission to be used” and “App behaviors you are not comfortable with”, was to map the contexts with the permissions. We were expecting to find the context within the app where those unwanted behaviors happened and to know how users refer to those concerns.

During the affinity diagraming, context sticky notes got clustered together; which means that the participants could not think of contexts as triggers that caused a permission to be used. As a recommendation it would be worth trying to use the same terms used for the brainstorming questions, e.g. “Trigger” or “Action”.

Chapter 4

Design & Implementation

The main goal of this stage was to develop a tool that can extract an app's behavior using static analysis and display such information in a permission screen. To meet this goal the design & implementation process was divided in three parts: (i) technical evaluation, (ii) screen design and (iii) tool's integration & building.

The technical evaluation consisted in testing the static analysis tools available in order to define the information that could be extracted from the apps and select the most optimal tools to do so. Once finished the evaluation, it was possible to start an iterative design process to present such information. The first mock ups of the screen design allowed us to define what information to display and select the tools to use. The final part was tool building, where the goal was to develop a tool that could auto generate the permission screen given an *apk* (Android Application Package).

4.1 Technical Evaluation

Previous work on extracting privacy related information has involved combination of both static and dynamic analysis. However we chose static analysis to make the process scalable and because of the interesting tools available in the University. The AppGuarden project [22] provided EviCheck[5] and a group of static analysis tools to extract the following information from an *apk*: app permissions, API calls, annotated event-API graphs and DFA (Deterministic Finite Automata) graphs. Some of the mentioned tools are still in development which is why the technical evaluation was performed.

This section describes the testing on the following tools:

- **Apk2perm:** Extracts permissions from the apk's manifest file.
- **Apk2api:** Extracts APIs from the apk.
- **Apk2auto:** Generates an annotated event-API graphs.
- **Auto2dfa:** Generates a DFA graph using Apk2auto's output.
- **EviCheck:** Verifies or generates security policies and certificates [5].

To make sure that the tools worked as expected, a test suit of Android apps was built following the Android documentation [32]. Knowing the source code it was possible to define expectations about the tools' outcome and identify API calls or permissions that were not detected properly. Using the affinity diagram results, $\langle context, permission\rangle$ tuples that concerned the users were defined and these became the requirements for the test apps. Each application implemented a tuple of the form $\langle context, permission\rangle$, except for a mixed application that implemented two permissions within one context: $\langle context, \{permission, permission\}\rangle$. Table 4.1 lists the 21 combinations implemented.

Permission(s)	**Context(s)**
Camera	onCreate, onClick
Read Contacts	onCreate, onClick, Service
Internet	onCreate, onClick, Service
Location	onCreate, onClick, Service
Microphone	onCreate, onClick, Service
Write External Storage	onCreate, onClick, Service
Read SMS	onCreate, onClick, Service
Location, Internet	Service

Table 4.1: Context permission combinations implemented in test apps

4.1.1 Test App Example

Record onClick. Listing 4.1 shows a snippet of the implementation of an app that records audio on a button click and saves the recorded audio on the device.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    ...
    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            if(!recording){
                Log.d(LOG_TAG, ”Start recording”);
                startRecording();
            }else{
                Log.d(LOG_TAG, ”Stop recording”);
                stopRecording();
            }
        }
    });
}

private void startRecording() {
    recording = true;
    mRecorder = new MediaRecorder();
    mRecorder.setAudioSource(MediaRecorder.AudioSource.MIC);
    mRecorder.setOutputFormat(MediaRecorder.OutputFormat.THREE_GPP);
    mRecorder.setOutputFile(mFileName);
    mRecorder.setAudioEncoder(MediaRecorder.AudioEncoder.AMR_NB);

    try {
        mRecorder.prepare();
    } catch (IOException e) {
        Log.e(LOG_TAG, ”prepare() failed”);
    }

    mRecorder.start();
}

private void stopRecording() {
    recording = false;
    mRecorder.stop();
    mRecorder.release();
    mRecorder = null;
}
```

Listing 4.1: Record audio on click code

4.1.2 Apk2perm

Using the static analysis tool Apk2perm on the app shown in Listing 4.1, the output in Listing 4.2 was obtained. The tool behaved as expected for all the test suite apps. In addition to generating a file with the permissions used, this tool also lists the app's components: Activities, BroadcastReceivers, Services and Intents.

```
android.permission.WRITE_EXTERNAL_STORAGE
android.permission.RECORD_AUDIO
```

Listing 4.2: Record audio on click permissions

4.1.3 Apk2api

The following listing shows a snippet of the data extracted from the Record audio on click app using Apk2api. As this tool decompiles the *apk* the ouput includes API calls from libraries. As all the apps required calling Android libraries, this output has too much noise and was therefore not used for the screen generation.

```
android.view.MenuItem.getIcon
android.support.v4.view.ViewPropertyAnimatorCompat$LollipopViewPropertyAnimatorCompatImpl.<init>
android.support.design.widget.CollapsingTextHelper.getCollapsedTypeface
android.support.v4.view.ScrollingView.computeHorizontalScrollRange
android.support.v4.util.ArrayMap.put
android.support.v7.widget.Toolbar.getPaddingTop
...
```

Listing 4.3: Record audio on click API calls

4.1.4 Apk2auto

This tool's generates an annotated event-API graph displayed as a JSON (JavaScript Object Notation) object. These graphs have too much noise as they include the information from the Android libraries, Listing 4.4 depicts an example. This tool's output is used as input for Auto2dfa.

```
[{"source": "mariavelarde.recordingapplication1.MainActivity$1.<init>:0", "target": "mariavelarde.recordingapplication1.MainActivity$1.<init>(Lmariavelarde/recordingapplication1/MainActivity;)V:0", "label": ""},
    {"source": "mariavelarde.recordingapplication1.MainActivity.access$100:-1", "target": "mariavelarde.recordingapplication1.MainActivity$1.onClick(Landroid/view/View;)V:20", "label": ""},
    ...
```

Listing 4.4: Record audio on click Annotated event-API graph

4.1.5 Auto2dfa

The ouput of this tool is a DFA graph in JSON format representing the app's behavior, excluding Android or Java libraries. A graphical representation of this output is shown in Figure 4.1 and a snippet of the JSON output is shown in Listing 4.5.

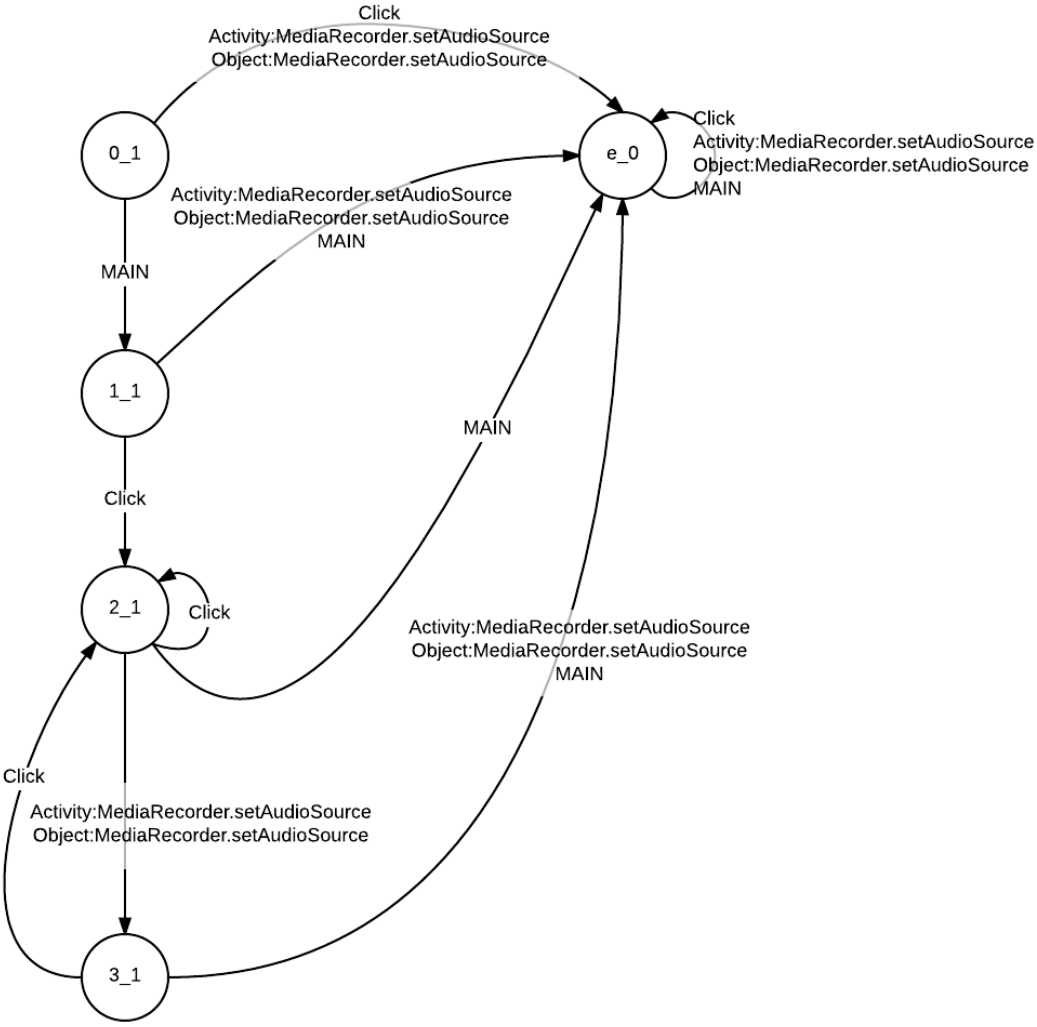

Click
Activity:MediaRecorder.setAudioSource
Object:MediaRecorder.setAudioSource
0_1
e_0
Click
Activity:MediaRecorder.setAudioSource
Object:MediaRecorder.setAudioSource
MAIN
Activity:MediaRecorder.setAudioSource
Object:MediaRecorder.setAudioSource
MAIN
MAIN
1_1
MAIN
Click
2_1
Click
Activity:MediaRecorder.setAudioSource
Object:MediaRecorder.setAudioSource
MAIN
Click
Activity:MediaRecorder.setAudioSource
Object:MediaRecorder.setAudioSource
3_1

Figure 4.1: Record onClick DFA graph

```
[{"source": ":0_1", "target": ":e_0", "label": "Activity:MediaRecorder.setAudioSource
    "},
    {"source": ":0_1", "target": ":e_0", "label": "Object:MediaRecorder.
    setAudioSource"},
    {"source": ":0_1", "target": ":1_1", "label": "MAIN"},
    {"source": ":0_1", "target": ":e_0", "label": "Click"},
    ...
```

Listing 4.5: Record audio on click DFA graph as JSON

Apk2auto and Auto2dfa combined seem to be the right direction to achieve representing app behavior as a sequence of events. Because of the early development stage of these tools, the results were not as expected for all the test apps. API calls to permissions Read SMS, Read Contacts, Write external storage were not detected and no API call to a permission was detected on the app that implemented $\langle Service, \{Location, Internet\}\rangle$.

This latest app's output is shown in Listing 4.6, where the expected API calls LocationManager.requestLocationUpdates and URL.openConnection are missing.

```
[{”source”: ”:0_1”, ”target”: ”:e_0”, ”label”: ”Runnable:ClassLoader.loadClass”},
    {”source”: ”:0_1”, ”target”: ”:1_1”, ”label”: ”MAIN”},
    {”source”: ”:0_1”, ”target”: ”:e_0”, ”label”: ”Activity:ClassLoader.loadClass”},
    {”source”: ”:1_1”, ”target”: ”:2_1”, ”label”: ”Runnable:ClassLoader.loadClass”},
    {”source”: ”:1_1”, ”target”: ”:e_0”, ”label”: ”MAIN”},
    {”source”: ”:1_1”, ”target”: ”:2_1”, ”label”: ”Activity:ClassLoader.loadClass”},
    {”source”: ”:2_1”, ”target”: ”:2_1”, ”label”: ”Runnable:ClassLoader.loadClass”},
    {”source”: ”:2_1”, ”target”: ”:e_0”, ”label”: ”MAIN”},
    {”source”: ”:2_1”, ”target”: ”:2_1”, ”label”: ”Activity:ClassLoader.loadClass”},
    {”source”: ”:e_0”, ”target”: ”:e_0”, ”label”: ”Runnable:ClassLoader.loadClass”},
    {”source”: ”:e_0”, ”target”: ”:e_0”, ”label”: ”MAIN”},
    {”source”: ”:e_0”, ”target”: ”:e_0”, ”label”: ”Activity:ClassLoader.loadClass”}]
```

Listing 4.6: Mixed permissions app Annotated event-API graph

4.1.6 EviCheck

For this dissertation's purposes we were interested in EviCheck's verifier component; which we used to verify the **presence** of *concerning or permission related behavior* instead of the **absence** of *bad behavior*. By making a zero tolerance policy the broken rules in the output represent the app's behavior.

4.1.6.1 Test Policy Definition

For testing purposes a general policy was defined, forbiding all privacy concerning permissions implemented in the test suite within all contexts. The following table lists the permissions and contexts considered for the policy, these list was based on the privacy concerns found on Chapter 3.

Permissions		
INTERNET	BILLING	CAPTURE_AUDIO_OUTPUT
RECORD_AUDIO	MEDIA_CONTENT_CONTROL	MANAGE_DOCUMENTS
READ_EXTERNAL_STORAGE	WRITE_EXTERNAL_STORAGE	CAMERA
CAPTURE_SECURE_VIDEO_OUTPUT	CAPTURE_VIDEO_OUTPUT	READ_CONTACTS
READ_CALL_LOG	WRITE_CALL_LOG	READ_PHONE_STATE
MODIFY_PHONE_STATE	ACCOUNT_MANAGER	GET_ACCOUNTS
GET_ACCOUNTS_PRIVILEGED	ACCESS_COARSE_LOCATION	ACCESS_FINE_LOCATION
ACCESS_LOCATION_EXTRA_COMMANDS	INSTALL_LOCATION_PROVIDER	LOCATION_HARDWARE
CONTROL_LOCATION_UPDATES	CALL_PRIVILEGED	PROCESS_OUTGOING_CALLS
CALL_PHONE	ADD_VOICEMAIL	READ_VOICEMAIL
WRITE_VOICEMAIL	SEND_SMS	SEND_RESPOND_VIA_MESSAGE
READ_SMS	RECEIVE_MMS	RECEIVE_SMS
BROADCAST_SMS		

Contexts		
EVICHECK_RECEIVER_METHOD	EVICHECK_DESTROY_METHOD	EVICHECK_SERVICE_METHOD
EVICHECK_START_METHOD	EVICHECK_STOP_METHOD	EVICHECK_PAUSE_METHOD
EVICHECK_ACTIVITY_METHOD	EVICHECK_RESUME_METHOD	EVICHECK_ONCREATE_METHOD
EVICHECK_RESTART_METHOD	EVICHECK_ONTOUCH_HANDLER	EVICHECK_ONCLICK_HANDLER
EVICHECK_DO_INBACKGROUND		

Table 4.2: EviCheck's General Policy's contexts and permissions

This policy was meant to test both of EviCheck's rule types to figure out which gave the most useful information for the screen generation. Each of the test apps was used to test a $\langle permission, context \rangle$ pair using the AND-rule. The mixed permissions app was used to test the OR-rule, where more than one permission was used within the same context.

The following AND-rule would let us know if the Camera permission was used on a button click:

```
EVICHECK_ONCLICK_HANDLER : ~CAMERA
```

while the violation of the following OR-rule would mean that both Internet and Location permissions were used on a Service.

```
EVICHECK_SERVICE_METHOD :V ~INTERNET ~ACCESS_COARSE_LOCATION
```

4.1.6.2 Results

The use of this policy against the test apps built allowed us to find bugs and report them to EviCheck's developers. For most of the apps, results were as expected; this is described below. Two updates of the tool were provided solving most of the bugs found. For other issues, solutions were provided either using EviCheck or later on the implementation of the display generator.

Location EviCheck outputs the broken rule and the method where the permission was used, which allows us to see if the permission was used by the app or by a third party. OR-rules' output show no location, reason why they were not used in this project's implementation. EviCheck's developer is aware of this behavior.

API Calls Regarding API calls two types of bugs were found: undetected permissions and false positives. EviCheck uses Androguard[1], which is a reverse engineering tool for app analysis. Androguard has an $\langle APIcall, permission \rangle$ map that allows identifying which permission was used once an API call is found in the code.

EviCheck's developer was able to identify the cause of the false positive issue. Read contacts permission was triggered due to a chain of calls in the Android support library, specifically in Landroid/support/v4/print/PrintHelperKitkat. This was identified as an isolated event that should be considered for the implementation stage of the project.

The undetected permissions issue was caused by missing API calls in Androguard's map. EviCheck's developer fixed it by implementing a extend map functionality, which allowed having an additional input file with API calls and their corresponding permission. In order to extend this map is necessary to know the structure of the API call: class, method, parameters type and return type. The following line adds an API call to the Camera permission.

```
Landroid/hardware/Camera->open(I)Landroid/hardware/Camera : CAMERA
```

Content Providers Permissions that read system's databases like Read Calendar, Read Contacts or Read SMS use calls to Content Providers instead of an specific API call for each. Even though the API call for content providers was found in the $\langle APIcall, permission \rangle$ map, Read Contacts and Read SMS were not always detected.

Because of time constrains this bug was not fixed, but it is possible to extract content providers calls using Androguard. The Content Provider API call receiver a Content URI String as parameter, this indicates which data storage is to read by the Content

[1] https://github.com/androguard/androguard

Provider. A possible solution is making a $\langle ContentURI, permission \rangle$ map and use it to identify to what permissions correspond the Content provider calls within the app.

Broadcast Receivers In the development stage of the project we tested the implement tool against *apk*s downloaded from Google Play like: Youtube, WeChat, Whatsapp, Instagram. There were two permissions that were not detected by EviCheck: “Google Cloud Messaging” and “Run at start up”. They do not have an specific API call, instead they allow the app to associate Broadcast Receivers to a system-wide Intent.

When an application requests the “Run at start up” permission, it has a Broadcast Receiver associated with a sysmte-wide Intent that is sent when the phone’s booting process ends; “Google Cloud Messaging” permission faced the same issue. To detect these permissions, we looked for the permission and the mapped Intent in the Manifest file, if both are found the permission was added to the Service context.

In-app Billing The Billing permission was not detected either. We attempted to add the API call that executes the payment but a further evaluation of the In-app billing process is required. This evaluation should identify all possible API calls that can make a purchase and add them to the $\langle APIcall, permission \rangle$ map. This permission may also use Intents, but all the components involved in its functionality should be defined through an evaluation.

4.1.7 Conclusions & Limitations

This chapter provided an overview of static analysis tools. Through their testing we were able to find out to what extend we can use them to extract app behavior, their limitations and propose future work for those limitations.

The affinity diagram results pointed out two directions for app behavior representation: context oriented or as a sequence of events. By the end of the evaluation we were able to extract successfully from an *apk*: permissions, api calls, components, intents, behavior as EviCheck rules. Because of these results we decided to go for the context oriented approach. The sequence of events approach should be consider for future research.

Finally, a list was made of EviCheck’s details or limitations that should be considered during implementation:

- Known false positives, e.g. Android’s PrintHelperKitkat library.

- Location missing when using OR-rule.
- In-app billing is not detected.
- Content providers' issue.
- Permissions that map Intents to Broadcast Receivers, instead of using API calls.

4.2 Screen Design

The goal of this section was to design a permission screen that represents the app behavior in terms of contexts and permissions, incorporating the contexts defined by the end of Chapter 3. Major design changes like colors and fonts were avoided for learnability purposes.

The new permission screen aimed to inform the user in three aspects:

- Context
- Permissions
- Ad libraries

The introduction of ad libraries' information has been already done in past research by Kelley *et al.* [10], but we are taking a different approach for this aspect, described later in this section. Balebako *et al.* introduced context at run time, but not in the permission screen.

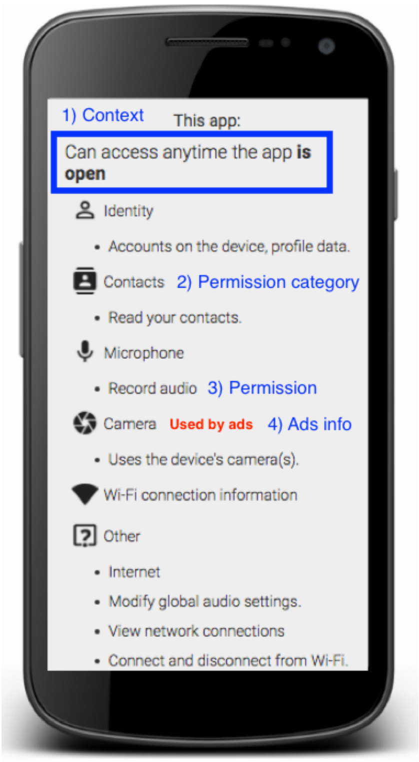
4.2.1 Prototype

Figure 4.2 depicts one of the first screen prototypes, highlighting in blue the different parts of the screen.

Contexts

The four previously defined contexts were labeled as the following phrases:

- **With button click:** "Can access **only with a button click**"
- **In background but with application visible:** "Can access any time the app **is open**"
- **In background, without the application visible:** "Can access any time the app **is not open**"
- **Granted but not used:** "Asks for, but **never uses**".



1) Context This app:

Can access anytime the app **is open**

Identity

- Accounts on the device, profile data.

Contacts 2) Permission category

- Read your contacts.

Microphone

- Record audio 3) Permission

Camera **Used by ads** 4) Ads info

- Uses the device's camera(s).

Wi-Fi connection information

Other

- Internet
- Modify global audio settings.
- View network connections
- Connect and disconnect from Wi-Fi.

Figure 4.2: Screen prototype with highlights of the aspects considered

Permissions

The permissions were grouped following the same categories as Google Play permissions screen, the category icons also remained the same.

Ad libraries

Paturi *et al.* also displayed third parties information in their screen proposal, they chose to display the permissions used by third party libraries [10]. Even though this approach increased comprehension on the source of a possible privacy threat, it assumes that users can acknowledge that not all libraries are ad libraries and that they may be used to implement core functionality. In this aspect we took a slightly different direction by adding an “Used by ads” label to the permissions used by an ad library.

4.2.2 Evaluation

The screen prototype was presented to a group of security experts in one of the AppGuarden project meetings, where the following issues were found:

- Even though the experts were familiar with the contexts within an app, they could not associate the phrases with the contexts.
- The list seemed too long.
- It was difficult to identify the context groups.

4.2.3 Feedback Implementation

Contexts

The labels were replaced for shorter ones, padding between the context and the permissions was increased and a border was added surrounding each context.

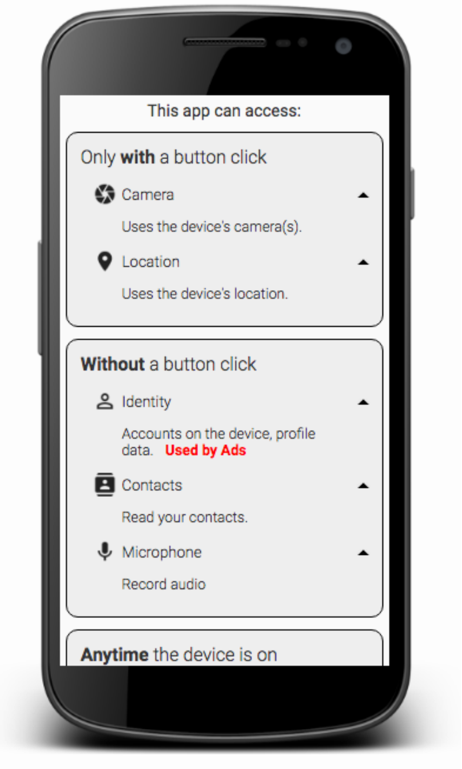
Layout

Arrows with a toggle functionality were added to shorten the list without removing information. The number of permission categories was reduced, to select which categories we would keep we referred to Paturi *et al.*'s study results on granularity level of categories [10].

They classified permissions into Location and Identity. Their results showed that users wanted more detail within those categories, showing more interest in: Contacts, SMS or Calendar. Finally, we obtained the list shown in Table 4.3. The final screen design looked like Figure 4.3.

Categories		
In-app purchases	Identity	Location
Photos/Media/Files	Microphone	Camera
Phone	Calendar	SMS
Contacts	Communication	Other

Table 4.3: Permission Categories used for permission screen



This app can access:

Only **with** a button click

Camera

Uses the device's camera(s).

Location

Uses the device's location.

Without a button click

Identity

Accounts on the device, profile data. **Used by Ads**

Contacts

Read your contacts.

Microphone

Record audio

Anytime the device is on

Figure 4.3: Screen Design

4.3 Tools Integration & Implementation

4.3.1 Overview

To goal of this section was to built a display generator tool and integrate it with the static analysis tools. The display generator was built using the Python programming language and the tools were integrated using a command line script.

Implementation is divided in two parts: (i) data extraction and (ii) display generation. Figure 4.4 represents the data extraction part and Figure 4.5 gives an overview of the display generator tool. Orange boxes in the figures represent the output files obtained from any part of the system, purple boxes are the *apk* file, white boxes are predefined input files, blue boxes are pre-processing, green boxes represent processing and the pink box is the tool's output.

4.3.2 Data extraction

The data extraction part has three steps: 1) *apk*'s permissions and intents are extracted with Apk2perm, 2) app permissions are used as input for the policy generator, which outputs an EviCheck policy and 3) app behavior is extracted using EviCheck with the *apk* and the policy as input.

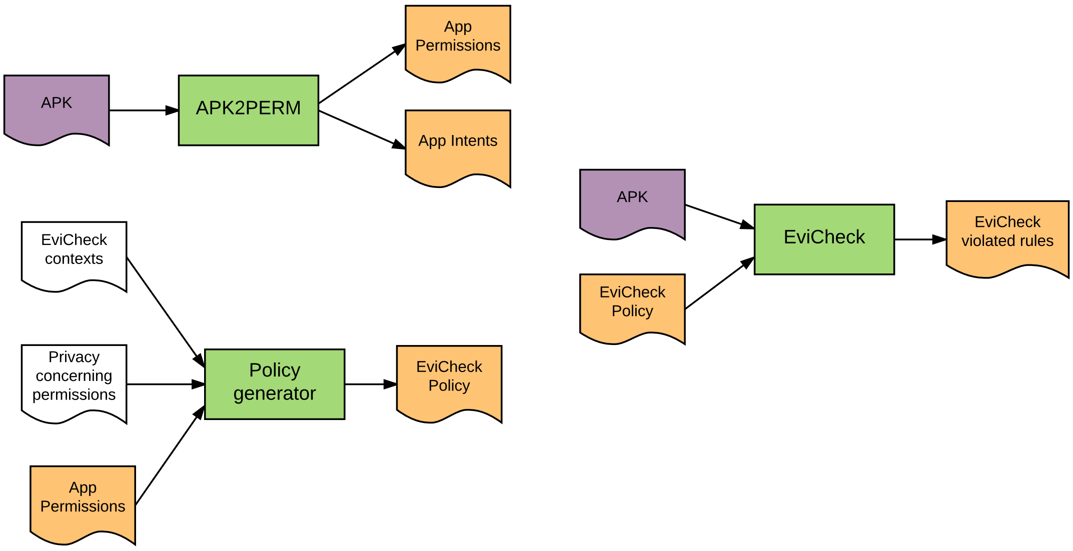


Figure 4.4: Tool Building Overview: Data Extraction

Policy Generation

The lists of concerning permissions and EviCheck contexts were defined as input for the policy generator. To implement the “Asks for, but never uses” context, the permissions found in the app were added to the to the policy. The final output is a policy that detects privacy concerning behavior and the use of permissions found in the app. As the location was needed to verify the use of ad libraries, the generated policy used the AND-rule and only one context, as the following:

```
EVICHECK_ONCLICK_HANDLER : ˜INTERNET
```

The generated policy aims to detect every permission within every context.

4.3.3 Display Generation

Figure 4.5 depicts an overview of the display generator. This tool uses EviCheck rules, app permissions, app intents, ad libraries list, known false positives list, layout information and data information to generate a permission screen in HTML format.

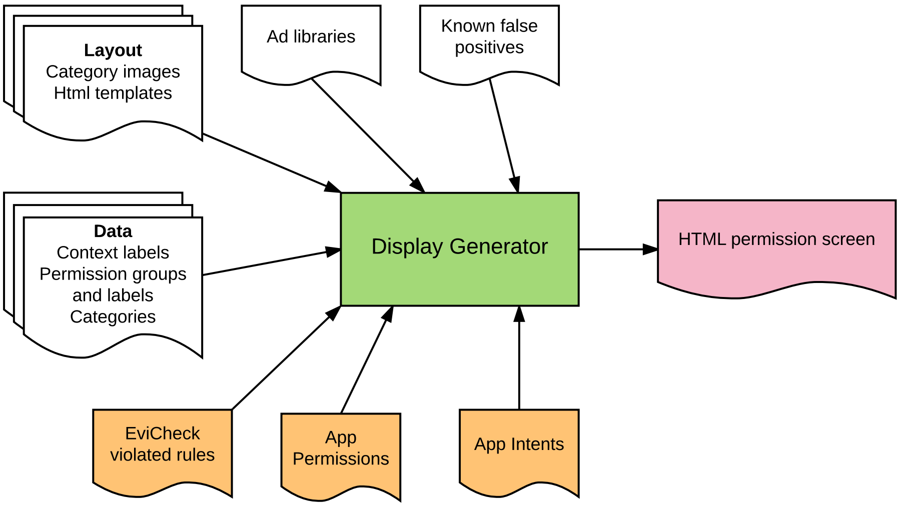


Figure 4.5: Tool Building Overview: Display Generation High Level

Ad Libraries Gathering

The following line was extracted from Youtube’s apk, it shows the use of Internet permission by Google’s Mobile Ads library AdMob.

```
Rule 1  ==>  set([’EVICHECK_ENTRY_POINT’]) : set([’˜INTERNET’])
Policy violated! Tag INTERNET is in Lfst-><init>(Lfsv; Lcom/google/
android/gms/ads/internal/client/AdSizeParcel; Z Z Lfma; Lcom/google/
android/gms/ads/internal/util/client/VersionInfoParcel; Leib;)V
```

With that information it is possible to determine which permission is used by an ad library using the library’s package name. Two MSc students working on Chinese Android markets provided us a list of known ad libraries’ package names. This list was later verified and completed using AppBrain’s list of top ad libraries [2].

AppBrain only provided the readable names of the libraries. Using static analysis on the apps that called those libraries, it was possible to extract the apps’ components and libraries’ package name.

False Positives Gathering

During the implementation phase we found more false positives in the Android support libraries. These were added to the “Known false positives” csv input file. Each false positive was defined by $\{Permission, Context, Location\}$.

[2] http://www.appbrain.com/stats/libraries/ad

Data

The data files had information about the permissions' label, categories' label, $\langle permission, category \rangle$ mapping and categories order (the order in which the categories were to be display in the permissions screen).

Layout

The permission screen was built usng HTML template files and CSS. The category images were obtained from Google Play.

Permssion Screen Generation Process

Before generating the screen, the data extracted from the applications was processed as depicted in Figure 4.6. The whole generation process followed the next steps: (i) load data from files, (ii) remove false positives from detected rules, (iii) check if rule was detected in ad library, (iv) group rules by context, (v) remove repeated rules, (vi) group rules by their permission's category and (vii) generate permission screen.

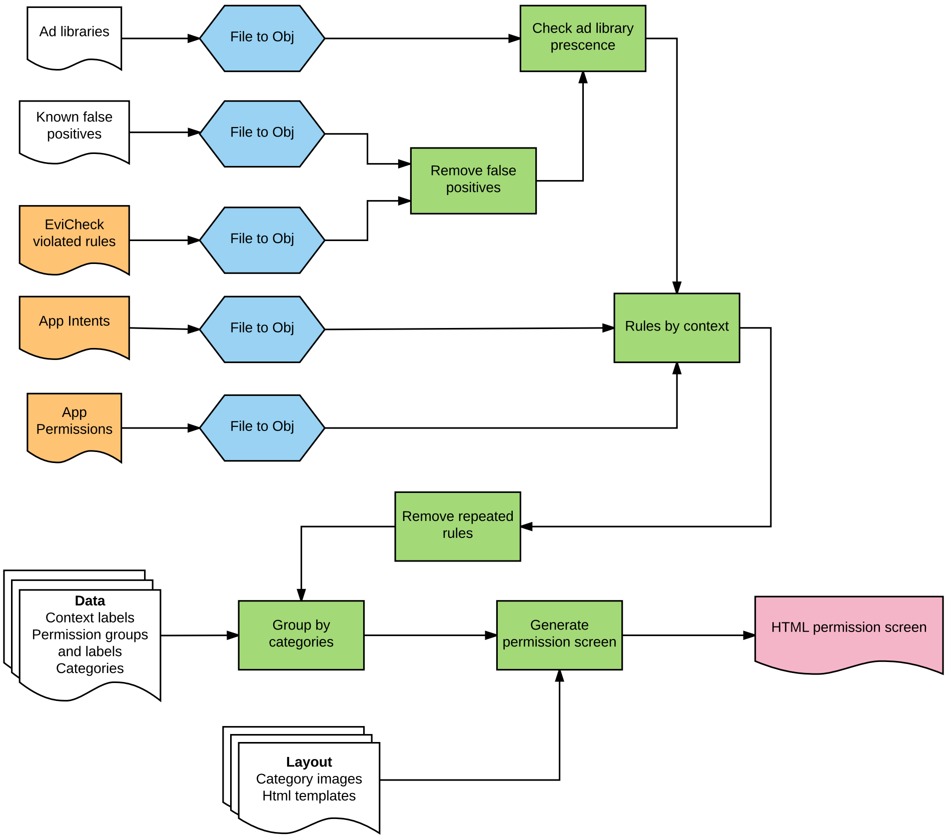


Figure 4.6: Display Generator Low Level

For the screen display we used the four contexts defined in Chapter 4.2. EviCheck's contexts were a assigned to screen contexts, as shown in Table 4.4. This depended on when they were used within the Android Activity lifecycle. Some screen contexts are mutually exclusive, hence step (v).

On button click		
EVICHECK_ONCLICK_HANDLER		
While open		
EVICHECK_DESTROY_METHOD	EVICHECK_START_METHOD	EVICHECK_STOP_METHOD
EVICHECK_PAUSE_METHOD	EVICHECK_ACTIVITY_METHOD	EVICHECK_RESUME_METHOD
EVICHECK_ONCREATE_METHOD	EVICHECK_RESTART_METHOD	EVICHECK_ONTOUCH_HANDLER
EVICHECK_DO_INBACKGROUND		
Anytime		
EVICHECK_SERVICE_METHOD		
Entry point		
EVICHECK_ENTRY_POINT		

Table 4.4: EviCheck Contexts mapped to Screen Contexts

4.3.4 Critical Evaluation

The permission labels were obtained from the Android documentation and Google Play. Future work should include a review of all the permissions' labels in order to make them shorter. Also, this label definition should be evaluated making a comprehension survey.

Some apps implemented deprecated permissions and their newer version, possibly for compatibility reasons. In those cases the screen displayed both permissions, which looked confusing. It is possible to go through the list of permissions, find the tuples of permissions and only show one if both are detected by the app.

Google Play does not show all permissions when clicking on the install button, there is a full permissions list that appears when clicking the "Permission details" button. In the future, two versions of this screen can be considered, a complete one and a shorter one. Parameters to differentiate both should be defined.

Chapter 5

Evaluation

5.1 Methodology

The aim of the final evaluation is to measure end-user comprehension of the screen. The chosen methodology was an Internet survey and its design was based on Felt *et al.*[11], Tam *et al.*[33] and Lin *et al.*[8]'s studies on user comprehension about permissions.

In August 2016, we conducted a between-subject Internet survey on 203 Android users recruited through Amazon's Mechanical Turk (AMT), respondents were paid 1 USD for participating. Users were randomly assigned to the *control group* or to the *experiment group*. Participants assigned to the *control group* were shown Google Play's current permission screen, participants assigned to the *experiment group* were shown our screen proposal.

Once the payments were completed MTurkIDs and IP addresses were deleted. An Ethical Review Procedure was required by the University and can be found in Appendix D.

5.1.1 Survey Structure

The survey had 21 questions divided in five parts: (i) android usage, (ii) comprehension, (iii) Westin index, (iv) demographics and (v) opinion. All questions are close-ended, except for the MTurk ID[1] and the final opinion question.

The *android usage* section included five questions like *"How long have you used an Android phone?"*, *"What factors do you consider before installing an application?"*,

[1] Amazon's Mechanical Turk users' unique identifier.

etc. To limit survey respondents to Android users we asked them the OS version of their device, providing instructions on how to find that information [8, 11]. Comprehension questions will be described in further detail in the next section.

The aim of the Westin index questions was to find out respondents' attitudes towards privacy [11]. The methodology used is called Westin index[2], which provides a set of three segmentation questions to classify users between "Privacy Fundamentalists", "Privacy Pragmatists" and "Privacy Unconcerned" [11, 35]. In the survey, the title displayed for this section was "Opinion Questions".

The *demographics section* included questions like *"Please enter your MTurk ID"*, *"What is your age?"*, *"What is your gender?"*, etc. The final question was optional and open-ended: *"Any additional comments"*. The complete survey can be found in Appendix B.

5.1.2 Questions Design

Comprehension questions were designed to evaluate users understanding on three aspects considered for our screen proposal: permission, context and if the permission had been used by an ad library. Each question showed a permissions screen and asked "Which of the following can this app do?". Below the question, five statements were provided and users had to specify how possible the statement was using a 5-point Likert scale from "Absolutely impossible" to "Absolutely possible". Example statements are shown in Table 5.1. Each experiment question had its corresponding control question, with the same permissions and the same statements.

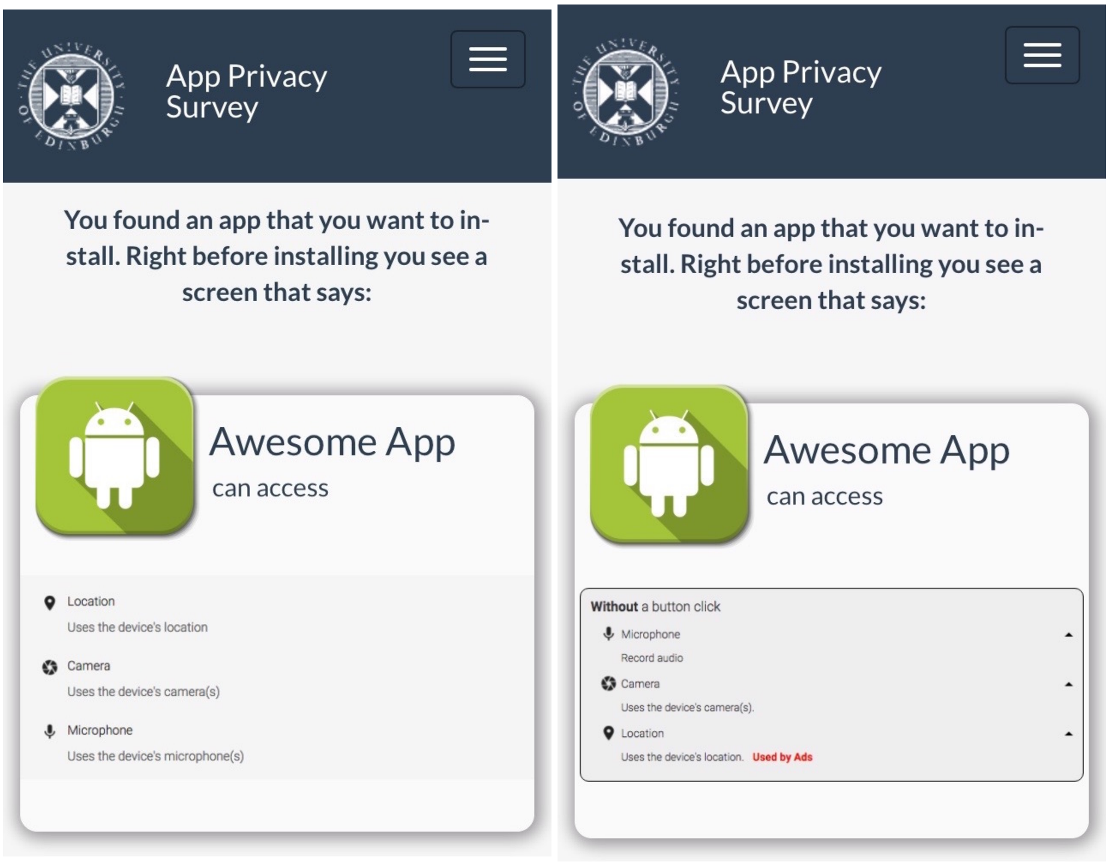
	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Charge purchases to your credit card at any time.	○	○	○	○	○
Get your location.	○	○	○	○	○
Allow ads to know your location.	○	○	○	○	○
Load ads.	○	○	○	○	○
Write on the SD card					

[2] Dr. Alan Westin made a series of privacy surveys and defined indexes to measure his results. This studies and their methodologies have been summarized by Kumaraguru *et al.*[34].

when the app is closed.	○	○	○	○	○

Table 5.1: Survey Question's Statements with 5-point Likert scale answers

Figure 5.1 depicts screenshots of the control and experiment group for one of the 9 survey questions. Both screens display location, camera and microphone permissions. Experiment group screen displays them under the "Without a button click" context.



App Privacy Survey

You found an app that you want to install. Right before installing you see a screen that says:

Awesome App

can access

Location

Uses the device's location

Camera

Uses the device's camera(s)

Microphone

Uses the device's microphone(s)

(a) Control group screen

App Privacy Survey

You found an app that you want to install. Right before installing you see a screen that says:

Awesome App

can access

Without a button click

Microphone

Record audio

Camera

Uses the device's camera(s).

Location

Uses the device's location. **Used by Ads**

(b) Experiment group screen

Figure 5.1: Survey question screens

Each group was shown 9 screens in total, 8 were simple and 1 was more complex. For the experiment group, simple questions had only one context, multiple permissions and could have ad libraries; complex questions had more than one context. For the control group simple and complex questions differ only in the amount of permissions shown.

There were three types of statements: permission, permission and context, permission and ad library. For the experiment group statements could be True or False, this could not be determined for the control group as the screens lacked information about context and ad libraries. Table 5.2 lists the statements corresponding for each question and their answer.

Question Details	Statement	Type	Answer
Q1 **Permissions:** Camera, Location **Contexts:** onButtonClick	Take pictures when you press the button.	Permission and context	True
	Get your location.	Permission	True
	Take pictures at any time.	Permission and context	False
	Get your location while you are using other applications.	Permission and context	False
	Load ads	Permission and ad library	False
Q2 **Permissions:** Identity, Read Contacts, Record Audio **Contexts:** While app is open	Record audio when you open the app.	Permission and context	True
	See who you have called.	Permission	True
	Record audio while you are using other applications.	Permission and context	False
	Read your heart rate.	Permission	False
	Allow ads make your phone vibrate.	Permission and ad library	False
Q3 **Permissions:** External Storage, Internet, Wake lock, Network connections, Vibrate **Contexts:** Anytime the device is on	Write on the SD card even when the application is closed.	Permission and context	True
	Keep your phone's screen on all the time.	Permission and context	True
	Add events to your calendar at any time.	Permission and context	False
	Read your phone number.	Permission	False
	Place phone calls.	Permission	False
Q4 **Permissions:** Read Contacts, SMS, Phone **Contexts:** Never used	Read your phone's contact list while you are not using the application.	Permission and context	False
	Modify your phone's contact list.	Permission	False
	Send text messages.	Permission	False
	Get your location.	Permission	False
	Place phone calls	Permission	False
Q5 **Permissions:**	Write on the SD card while you are using the app.	Permission and context	True

External Storage, Devide ID, Read Contacts, Location **Contexts:** While app is open	Read your phone number. Place phone calls. Allow ads to access your phone's contacts list. Get your location.	Permission and context Permission Permission and ad library Permission	True False False True
Q6 **Permissions:** Internet, Vibrate Network connections, Prevent tablet from sleeping **Contexts:** Anytime the device is on	Load ads Allow ads make your phone vibrate. Allow ads to pair Bluetooth devices Read your calendar anytime. Record audio after pressing "Start recording" button.	Permission and ad library Permission and ad library Permission and ad library Permission and context Permission and context	True True False False False
Q7 **Permissions:** Billing, Accounts, System settings **Contexts:** onButtonClick	Charge purchases to your credit card when you click a button. Allow ads to modify settings. Pair Bluetooth devices Modify settings when you click a button. Charge purchases to your credit card when you open the app.	Permission and context Permission and ad library Permission Permission and context Permission and context	True True False False False
Q8 **Permissions:** Record audio, Camera, Location **Contexts:** While app is open	Get your location. Allow ads to know your location Read your phone's contact list while you are not using the application Get your location while you are not using the application Send text messages.	Permission Permission, ad library Permission and context Permission and context Permission	True True False False False
Q9 **Permissions:**	Get your location. Load ads	Permission Permission, ads	True True

Billing, Device ID, Accounts, Location External storage, Internet **Contexts:** onButtonClick, While app is open, Anytime	Charge purchases to your credit card at any time.	Permission and context	False
	Allow ads to know your location.	Permission, ad library	False
	Write on the SD card when the app is closed.	Permission and context	False

Table 5.2: Survey Statements

5.2 Results

We considered valid responses the ones that were completed in over 6 minutes. The survey's completion time plot showed no outtliers under the first quartile and the outliers over the third quartile were kept. In total we had 185 valid responses, 55.14% of the respondents were Male, 43.24% Female, 0.54% Other and 1.08% preferred not to answer. From the 185 valid responses, 90 belonged to the control group and 95 to the experiment group.

5.2.1 Scores Analysis

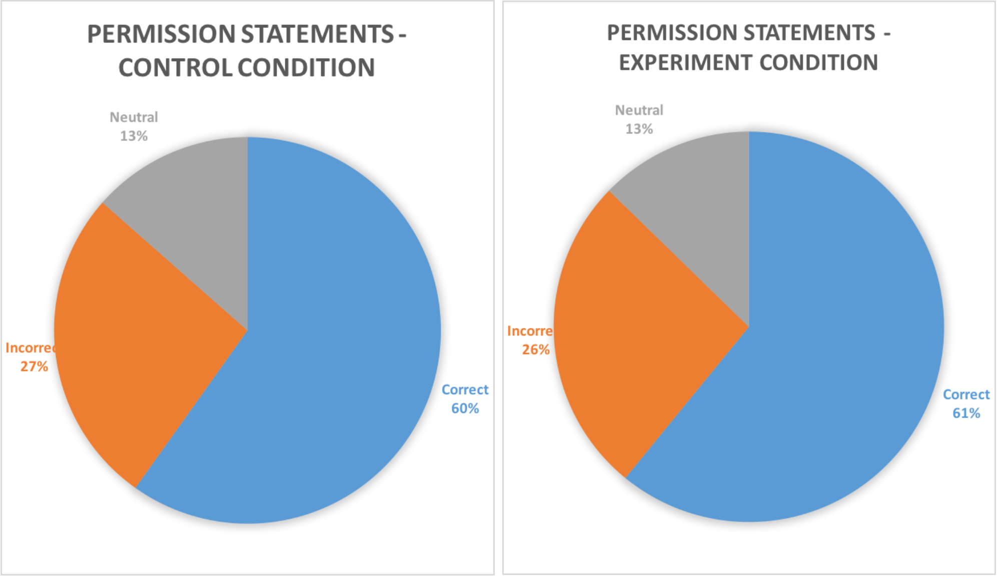
Scores were calculated for each user counting the amount of times they answered correctly, incorrectly or neutral. As the control group screen did not have enough information to answer accurately in terms of context or ad libraries, for each statement with no definitive answer we classified the response as more conservative or less conservative. More conservative responses assumed that the statement was possible even if not enough information was provided, while less conservative responses assumed the statement was not possible.

Each question had 5 statements to be answered in a 5-point Likert scale from "Absolutely impossible" to "Absolutely possible". Each participant answered 9 questions, giving us 45 answered statements per participant. 8325 statement answers were obtained in total. From these 8325 answers, 555 were removed because of inconsistencies found in the phrasing of the permission label between control and experiment group, leaving 7770 in total.

To verify the correctness of the answers "Absolutely possible" and "Possible" where mapped to “Yes”, “Absolutely impossible” and “Impossible” to “No” and “Neutral” was kept. For questions in the control group with no clear answer “Yes” was counted as a “More conservative” answer and “No” as a “Less conservative” one.

Permission Statements

2590 statements belonged to the permission only group, 1330 were from the experiment group and 1260 from the control group. Figure 5.2 shows the percentages of correct, incorrect and neutral answers for these statements. There was a 1% increase in the amount of correct answers in the experiment group. To check if this increase was statistically significant we performed an ANOVA using the number of correct answers as dependent variable and group (control or experiment) as independent variable.



(a) Control Group

(b) Experiment Group

Figure 5.2: Permission Statements Results: Correct, Incorrect and Neutral

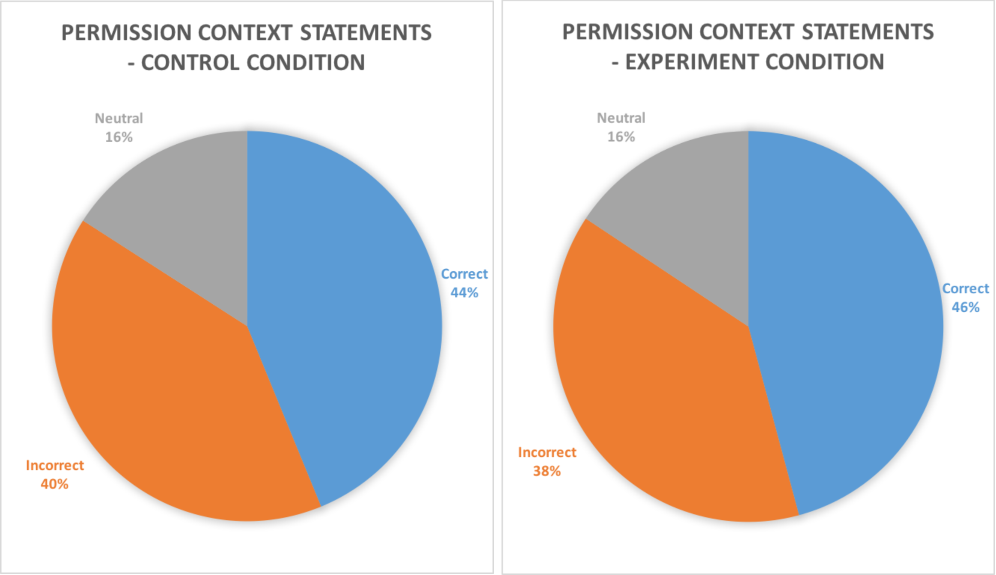
As this statements included no context or library information we did not expected a big difference between control and experiment group. The results of the ANOVA are shown in Table 5.3, as expected, there is no statistically significant difference between the groups. Statements classified as “Less conservative” or “More conservative” were only a 13% of this statement group and they belonged to the screen that showed the “Asks for, but never uses” context.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
GroupType	1	1	1.020	0.17	0.681
Residuals	183	1101	6.016		

Table 5.3: ANOVA Results for Permission Only Statements. Dependent variable: number of correct answers. Independent variable: group type.

Permission Context Statements

Permission-context statements aimed to: increase comprehension regarding $\langle permission, context \rangle$ in the experiment group and find out users' expectations in the control group. Correctness improved in 2%, Figure 5.3. ANOVA test indicated that there was no statistically significant difference between experiment and control group, refer to Table 5.4.



(a) Control Group

(b) Experiment Group

Figure 5.3: Permission Context Statements Results

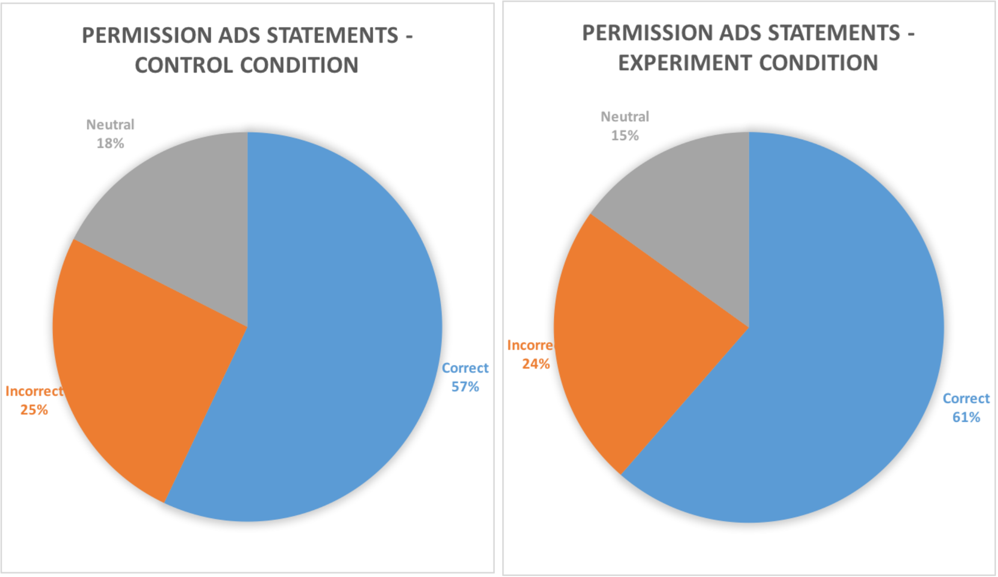
	Df	Sum Sq	Mean Sq	F value	Pr(>F)
GroupType	1	6.8	6.781	0.849	0.358
Residuals	183	1461.4	7.986		

Table 5.4: ANOVA Results for Permission Context Statements. Dependent variable: number of correct answers. Independent variable: group type.

In this case 72% of the control group statements could not be completely True or completely False because of the missing context information. 80% of these undefined statements had “More conservative” answers, which means that users do not trust apps and assume that apps could do anything with the permissions.

Permission Ads Statements

Permission-ads statements had 4% improvement in correctness in the experiment group. In this case statements with no clear answers were 58% of the control group answers and 91% of those were “More conservative”. The results of the survey can be found in Appendix C.



(a) Control Group

(b) Experiment Group

Figure 5.4: Permission Ads Statements Results

ANOVA test shows no statistically significant difference for this 4% improvement, Table 5.5. However, the p-value is close to 0.1, meaning that some improvements in the screen may cause a significant difference. Suggestions for future work will be described in Chapter 6.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
GroupType	1	7.1	7.137	2.698	0.102
Residuals	183	484.1	2.645		

Table 5.5: ANOVA Results for Permission Ads Statements. Dependent variable: number of correct answers. Independent variable: group type.

5.2.2 Frequency Analysis

Even though the increase in comprehension was low, the experiment screen showed no decrease in users comprehension. Both groups had at least 25% of incorrect answers on each question type. Possible causes of these results:

- Users' over conservatism towards missing information.
- Lower than expected correct answers in experiment group.
- Low comprehension of the permissions, independent of context or ad libraries information.
- Low comprehension of the context

To figure out the reason, we obtained the counts for each statement's answer: "Yes", "No" or "Neutral". Full results are in Appendix C. We chose questions that had unexpected results in the experiment group and compared them with the control group. We were looking for improvements in comprehension, to verify if any improvement found was dependent on the group we performed Chi Square Tests.

Question 1 Statement 3

Take pictures at any time. This question showed the Camera and Location permissions, in the experiment group they were displayed under the "Only with a button click" context. The counts in Table 5.6 show an improvement in comprehension as the number of participants that believe the answer to be "Yes" decreases in the experiment group.

	Neutral	No	Yes	Total
Experiment	21	33	41	95
Control	15	13	62	90
Total	36	46	103	185

Table 5.6: Counts for "Take pictures at any time" answers

Chi Square test results in Table 5.7 show a p-value under 0.05, which means that answers for this statement depend on the group.

Person's Chi Square test		
X-squared = 13.852	df = 2	p-value = 0.0009818

Table 5.7: Chi Square test for "Take pictures at any time". Dependent variable: answer. Independent variable: group type.

Question 1 Statement 4

Get your location while you are using other applications. For this statement the expected answer in the experiment group was "No". The "No" answers got duplicated from control to experiment group, but still a big amount of participants answered "Yes".

	Neutral	No	Yes	Total
Experiment	12	16	67	95
Control	8	8	74	90
Total	20	24	141	185

Table 5.8: Counts for "Get your location while you are using other applications" answers.

Chi Square Test's p-value was close to 0.1. There was no statistically significant difference and we were unable to reject the hypothesis that the group and the answer are independent, but this value could be lowered by doing some improvements on the screen.

Person's Chi Square test		
X-squared = 3.6817	df = 2	p-value = 0.1587

Table 5.9: Chi Square test for "Get your location while you are using other applications". Dependent variable: answer. Independent variable: group type.

Question 2 Statement 1

Record audio when you open the app. Question 2 showed Identity, Read Contacts, Record Audio permissions, in the experiment group they were under the "While app open" context.

The expected answer for this statement was “Yes”. In this case an over conservative answer in the control group would be the correct answer. As 80% of the permission,context statements were answered over conservatively, we expected the control group to answer “Yes”. Results in Table 5.10 show that users answered as aspected in both, experiment and control group.

	Neutral	No	Yes	Total
Experiment	3	5	87	95
Control	5	5	80	90
Total	8	10	167	185

Table 5.10: Counts for “Record audio when you open the app” answers.

Chi Square test show no statistically significant difference between control and experiment group answers, Table 5.11.

Person’s Chi Square test
X-squared = 0.65876 df = 2 p-value = 0.7194

Table 5.11: Chi Square test for “Record audio when you open the app”. Dependent variable: answer. Independent variable: group type.

Question 2 Statement 3

Record audio while you are using other applications. This statement’ answer was expected to be “No” in the experiment group, but most of the participants answered that the app was capable of use the permission in the “Anytime” context. This question is an example of the experiment group answering in the same over conservative way as the control group, regarding a context that was not shown in the screen.

	Neutral	No	Yes	Total
Experiment	10	10	75	95
Control	11	9	70	90
Total	21	19	145	185

Table 5.12: Counts for “Record audio while you are using other applications” answers

Chi Square test shows that the answer is independent of group.

Person's Chi Square test		
X-squared = 0.13763	df = 2	p-value = 0.9335

Table 5.13: Chi Square test for "Record audio while you are using other applications". Dependent variable: answer. Independent variable: group type.

Question 3 Statement 1

Write on the SD card even when the application is closed. This question asked for the permissions: external storage, internet, network connections, wake lock and vibrate. In the experiment group the permissions were used under the "Anytime" context. The correct answer for this question was "Yes", but experiment groups participants seemed to be less sure about it. There was no statistically significant difference between control and experiment group answers.

	Neutral	No	Yes	Total
Experiment	27	28	40	95
Control	19	22	49	90
Total	46	50	89	185

Table 5.14: Counts for "Write on the SD card even when the application is closed" answers

Person's Chi Square test		
X-squared = 2.8884	df = 2	p-value = 0.2359

Table 5.15: Person's Chi Square test for "Write on the SD card even when the application is closed". Dependent variable: answer. Independent variable: group type.

Question 3 Statement 5

Place phone calls. This screen did not showed the Phone permission at all. Regardless of the context, the expected answer was "No". Still there seemed to be some confusion.

This answer shows low comprehension on the permissions alone. Chi Square test values show that this comprehension does not depend on the group.

	Neutral	No	Yes	Total
Experiment	14	56	25	95
Control	15	59	16	90
Total	29	115	41	185

Table 5.16: Counts for “Place phone calls” answers

Person’s Chi Square test		
X-squared = 1.9546	df = 2	p-value = 0.3763

Table 5.17: Person’s Chi Square test for “Place phone calls”. Dependent variable: answer. Independent variable: group type.

Chapter 6

Conclusions

This report has discussed issues related to user awareness, comprehension of Android permissions, app analysis and Android's app and security architecture. As privacy is subjective to what the user agrees to, is necessary for developers to provide enough and comprehensible information about the app's behavior towards permissions. We found users' privacy related concerns and proposed two ways of displaying privacy related app behavior.

The project implemented one of those approaches using static analysis tools developed at the University. The whole development of the project involved working with app analysis, computer security and HCI experts; and required several evaluation stages. This project made contributions in two areas: app analysis and usability.

Past studies have combined dynamic and static analysis in order to extract all the information needed from an app, however this approach can not automated [10]. We have evaluated available static analysis tools and found to what extends is app behavior extraction possible. We identified limitations, where they belonged in Android's security architecture and possible solutions were proposed, setting ground floor for future work.

In the usability area we found that at least 25% of comprehension questions were answered incorrectly, showing low comprehension of permissions; which is consistent with past studies [11]. 80% of the answers to questions that lacked information were answered over conservatively, showing that users tend to mistrust the app and assume the worst scenario regarding unprovided information. The low improvement of correct answers in the experiment group showed that users do not understand contexts, this points out a research gap. In order to properly provide context information is necessary to evaluate users' mental models on app behavior. Ad libraries information was close to be statistically significant, which means that with some improvements on the screen there might be a comprehension increase in that aspect.

6.1 Critical Evaluation & Future Work

Requirements Gathering

Contexts' Labels All the context sticky notes ended up in the same cluster, even though the concerns could have been associated with them. This may have been caused by the phrasing of the context. If the affinity diagram exercise were to be done again, context sticky notes should use the same terms as the questions provided at the beginning of the exercise to ease the mental mapping.

Biased Participants We benefited from the participants' Computer Science background but this may have biased the privacy concerns. Probably people that work in an office may show more concern about the Manage Documents permission or information integrity.

Screen Design

Contexts' Labels It would be worth evaluating users mental models on app behavior and on permissions before attempting to combine both aspects in the user interface. This evaluation could fix the context's low comprehension issue. When the design was showed to the group of experts, even though they understood app behavior they used different terms to refer to the contexts provided, an evaluation of mental models could help correcting context labels.

Permission Categories The definition of permission categories was outside of the scope of this project, but it would be worth trying with different approaches.

App Analysis

It should be considered for future work to fix EviCheck's undetected permissions issue: Broadcast Receivers and Content Providers. There is also work to be done in representing app behavior as a sequence of events. Chen *et al.* [19] worked on an approach that allowed expressing app behavior and policies as Permission Event Graphs and the AppGuarden project is working on the development of a tool that shows app behavior as *dfa* graphs.

Evaluation

Survey's Permission Screen

To decrease the attention required for each question and increase comprehension, the survey had only one context per screen except for the "more complicated" question. We did not considered that users would take a more conservative approach about unprovided information, which may have caused some of the incorrect answers in the experiment group. For future work, the use of the whole screen should be considered.

Survey Questions & Results Analysis

To identify if the low scores were due to conservatism or low comprehension, the following aspects should be considered for future surveys: attention (time spent in per question) and level of confidence for the provided answers.

The number of incorrect answers per user should be counted, if all users made mistakes low comprehension is caused by the permission screen. If only a couple of users made lots of mistakes, low comprehension is caused by wrong mental models or laziness. To prevent this bias, Felt *et al.* removed from their study questions with too many incorrect or neutral answers [11].

Bibliography

[1] K. Yaghmour, “Inside Android’s UI,” 2012. [Online]. Available: http://www.slideshare.net/opersys/inside-androids-ui

[2] J. Drake, Z. Lanier, C. Mulliner, P. Oliva, S. Ridley, and G. Wicherski, *Android Hacker’s Handbook*, 1st ed. John Wiley & Sons, 2014.

[3] AndroidDevelopers, “Application Fundamentals,” 2016. [Online]. Available: https://developer.android.com/guide/components/fundamentals.html

[4] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. L. Traon, D. Octeau, and P. Mcdaniel, “FlowDroid : Precise Context , Flow , Field , Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps,” *PLDI ’14 Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pp. 259–269, 2014.

[5] D. Seghir, MN & Aspinall, “EviCheck: Digital Evidence for Android,” *13th International Symposium on Automated Technology for Verification and Analysis*, 2015. [Online]. Available: http://www.research.ed.ac.uk/portal/en/publications/evicheck-digital-evidence-for-android(ab4b47af-77b8-4888-90cc-5021b18c4668).html

[6] InternationalDataCorporation, “Smartphone Os Market-Share,” 2016. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[7] AppBrain, “Number of Android applications,” 2016. [Online]. Available: http://www.appbrain.com/stats/number-of-android-apps

[8] J. Lin, N. Sadeh, S. Amini, J. Lindqvist, J. I. Hong, and J. Zhang, “Expectation and purpose: understanding users’ mental models of mobile app privacy through crowdsourcing,” *Proceedings of the 2012 ACM Conference*

on Ubiquitous Computing - UbiComp '12, p. 501, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2370216.2370290

[9] K. Knorr and D. Aspinall, "Security testing for Android mHealth apps," in *Software Testing, Verification and Validation Workshops (ICSTW), 2015 IEEE Eighth International Conference on*, 2015, pp. 1–8.

[10] A. Paturi, P. G. Kelley, and S. Mazumdar, "Introducing Privacy Threats from Ad Libraries to Android Users Through Privacy Granules," *USEC Workshop*, no. February, 2015.

[11] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: user attention, comprehension, and behavior," *SOUPS '12: Proceedings of the Eighth Symposium on Usable Privacy and Security*, pp. 1–14, 2012.

[12] R. Wash, "Folk models of home computer security," in *Proceedings of the Sixth Symposium on Usable Privacy and Security - SOUPS '10.* New York, New York, USA: ACM Press, 2010, p. 1. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1837110.1837125

[13] N. Tailor, Y. He, and I. Wagner, "POSTER: Design Ideas for Privacy-aware User Interfaces for Mobile Devices," *Proceedings of the 9th ACM Conference on Security #38; Privacy in Wireless and Mobile Networks*, pp. 219–220, 2016. [Online]. Available: http://doi.acm.org/10.1145/2939918.2942420

[14] GoogleDevelopers, "Requesting Permissions," 2016. [Online]. Available: https://developer.android.com/training/permissions/requesting.html

[15] R. Balebako, J. Jung, W. Lu, and L. F. Cranor, "Little Brothers Watching You: Raising Awareness of Data Leaks on Smartphones," *Symposium on Usable Privacy and Security (SOUPS) 2013*, 2013.

[16] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," *Osdi '10*, vol. 49, pp. 1–6, 2010.

[17] D. Barrera, P. C. V. Oorschot, and A. Somayaji, "A Methodology for Empirical Analysis of Permission-Based Security Models and its Application

to Android," *Security*, no. 1, pp. 73–84, 2010. [Online]. Available: http://doi.acm.org/10.1145/1866307.1866317

[18] A. R. Beresford, A. Rice, and N. Skehin, "MockDroid : trading privacy for application functionality on smartphones Categories and Subject Descriptors," *Hot-Mobile '11 Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, pp. 49–54, 2011.

[19] K. Z. Chen, N. Johnson, V. D'Silva, S. Dai, K. MacNamara, T. Magrino, E. Wu, M. Rinard, and D. Song, "Contextual Policy Enforcement in Android Applications with Permission Event Graphs," *Symposium on Network and Distributed System Security (NDSS)*, 2013.

[20] J. Jung, S. Han, and D. Wetherall, "Short paper: Enhancing Mobile Application Permissions with Runtime Feedback and Constraints," in *ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM, 2012.

[21] AndroidDevelopers, "Searchable Configuration," 2016. [Online]. Available: https://developer.android.com/guide/topics/search/searchable-config.html

[22] AppGuarden, "App Guarden Resilient Application Stores," 2016.

[23] P. Dourish, R. E. Grinter, J. D. De La Flor, and M. Joseph, "Security in the wild: User strategies for managing security as an everyday, practical problem," *Personal and Ubiquitous Computing*, vol. 8, no. 6, pp. 391–401, 2004.

[24] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A Conundrum of Permissions:," *In Proceedings of the Workshop on Usable Security (USEC)*, 2012.

[25] B. I. Bulgurcu Burcu, Cavusoglu Hasan, "INFORMATION SECURITY POLICY COMPLIANCE: AN EMPIRICAL STUDY OF RATIONALITY-BASED BELIEFS AND INFORMATION SECURITY AWARENESS," *MIS Quarterly*, vol. 34, no. 3, pp. 523–548, 2010.

[26] P. G. Kelley, L. F. Cranor, and N. Sadeh, "Privacy as part of the app decision-making process," *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, p. 11, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2470654.2466466

[27] K. P. Yee, “Aligning security and usability,” *IEEE Security and Privacy*, vol. 2, no. 5, pp. 48–55, 2004.

[28] A. P. Felt, S. Egelman, M. Finifter, D. Akhawe, and D. Wagner, “How to ask for permission,” *Hotsec*, pp. 7–13, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2372387.2372394

[29] B. Hanington and B. Martin, *Universal Methods Of Design.* Rockport, 2012.

[30] A. P. Felt, “Towards Comprehensible and Effective Permission Systems,” Ph.D. dissertation, 2012. [Online]. Available: http://escholarship.org/uc/item/8h695639

[31] AndroidDevelopers, “Manifest.permission,” 2016. [Online]. Available: https://developer.android.com/reference/android/Manifest.permission.html

[32] ——, “Android Documentation,” 2016. [Online]. Available: http://developer.android.com/develop/index.html

[33] J. Tam, R. Reeder, and S. Schechter, “I’m Allowing What? Disclosing the authority applications demand of users as a condition of installation,” 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.183.2159

[34] P. Kumaraguru and L. Cranor, “Privacy indexes: A survey of westin’s studies,” *School of Computer Science, Carnegie Mellon University*, vol. Tech. rep., no. December, pp. 1–22, 2005.

[35] H. Taylor, “Most People are “Privacy Pragmatists” Who, While Concerned about Privacy,Will Sometimes Trade It Off for Other Benefits,” *Harris Interactive*, 2003. [Online]. Available: http://www.prnewswire.com/news-releases/most-people-are-privacy-pragmatists-who-while-concerned-about-privacy-will-sometimes-trac
html

Appendix A

Sticky Notes Content

Permission	Description
ACCESS_CHECKIN_PROPERTIES	Allows read/write access to the ”properties” table in the checkin database, to change values that get uploaded.
ACCESS_COARSE_LOCATION	Allows an app to access approximate location.
ACCESS_FINE_LOCATION	Allows an app to access precise location.
ACCESS_LOCATION_EXTRA_ COMMANDS	Allows an application to access extra location provider commands.
ACCESS_NETWORK_STATE	Allows applications to access information about networks.
ACCESS_NOTIFICATION_POLICY	Marker permission for applications that wish to access notification policy.
ACCESS_WIFI_STATE	Allows applications to access information about Wi-Fi networks.
ACCOUNT_MANAGER	Allows applications to call into AccountAuthenticators.
ADD_VOICEMAIL	Allows an application to add voicemails into the system.
BATTERY_STATS	Allows an application to collect battery statistics
BLUETOOTH	Allows applications to connect to paired bluetooth devices.
BLUETOOTH_ADMIN	Allows applications to discover and pair bluetooth devices.
BLUETOOTH_PRIVILEGED	Allows applications to pair bluetooth devices without user interaction, and to allow or disallow phonebook access or message access.
BODY_SENSORS	Allows an application to access data from sensors that the user uses to measure what is happening inside his/her body, such as heart rate.
BROADCAST_PACKAGE_ REMOVED	Allows an application to broadcast a notification that an application package has been removed.
BROADCAST_SMS	Allows an application to broadcast an SMS receipt notification.
BROADCAST_STICKY	Allows an application to broadcast sticky intents. Intent: abstract description of an operation to be performed. Sticky intent: used with sticky broadcast, stays around after the broadcast is complete, so that others can quickly retrieve that data.

BROADCAST_WAP_PUSH	Allows an application to broadcast a WAP PUSH receipt notification. WAP messages replace the need to include a link in a message. Instead the user receives an alert that once clicked will direct him to the Web page.
CALL_PHONE	Allows an application to initiate a phone call without going through the Dialer user interface for the user to confirm the call.
CALL_PRIVILEGED	Allows an application to call any phone number, including emergency numbers, without going through the Dialer user interface for the user to confirm the call being placed.
CAMERA	Required to be able to access the camera device.
CAPTURE_AUDIO_OUTPUT	Allows an application to capture audio output.
CAPTURE_SECURE_VIDEO_OUTPUT	Allows an application to capture secure video output.
CAPTURE_VIDEO_OUTPUT	Allows an application to capture video output.
CHANGE_COMPONENT_ENABLED_STATE	Allows an application to change whether an application component (other than its own) is enabled or not.
CHANGE_CONFIGURATION	Allows an application to modify the current configuration, such as locale.
CHANGE_NETWORK_STATE	Allows applications to change network connectivity state.
CHANGE_WIFI_MULTICAST_STATE	Allows applications to enter Wi-Fi Multicast mode.
CHANGE_WIFI_STATE	Allows applications to change Wi-Fi connectivity state.
CHECK_LICENSE	Allows Google Play license check.
CLEAR_APP_CACHE	Allows an application to clear the caches of all installed applications on the device.
CONTROL_LOCATION_UPDATES	Allows enabling/disabling location update notifications from the radio.
DELETE_CACHE_FILES	Allows an application to delete cache files.
DELETE_PACKAGES	Allows an application to delete packages.
DIAGNOSTIC	Allows applications to RW to diagnostic resources.
DISABLE_KEYGUARD	Allows applications to disable the keyguard if it is not secure.
DUMP	Allows an application to retrieve state dump information from system services.
EXPAND_STATUS_BAR	Allows an application to expand or collapse the status bar.
FACTORY_TEST	Run as a manufacturer test application, running as the root user.
FLASHLIGHT	Allows access to the flashlight.
GET_ACCOUNTS	Allows access to the list of accounts in the Accounts Service.
GET_ACCOUNTS_PRIVILEGED	Allows access to the list of accounts in the Accounts Service.
GET_PACKAGE_SIZE	Allows an application to find out the space used by any package.
GLOBAL_SEARCH	This permission can be used on content providers to allow the global search system to access their data.
INSTALL_LOCATION_PROVIDER	Allows an application to install a location provider into the Location Manager.

INSTALL_PACKAGES	Allows an application to install packages.
INSTALL_SHORTCUT	Allows an application to install a shortcut in Launcher.
INTERNET	Allows applications to open network sockets.
KILL_BACKGROUND_PROCESSES	Allows an application to call killBackgroundProcesses(String).
LOCATION_HARDWARE	Allows an application to use location features in hardware, such as the geofencing api.
MANAGE_DOCUMENTS	Allows an application to manage access to documents, usually as part of a document picker.
MEDIA_CONTENT_CONTROL	Allows an application to know what content is playing and control its playback.
MODIFY_AUDIO_SETTINGS	Allows an application to modify global audio settings.
MODIFY_PHONE_STATE	Allows modification of the telephony state - power on, mmi, etc.
MOUNT_FORMAT_FILESYSTEMS	Allows formatting file systems for removable storage.
MOUNT_UNMOUNT_FILESYSTEMS	Allows mounting and unmounting file systems for removable storage.
NFC	Allows applications to perform I/O operations over NFC. Near Field Communication (NFC) is a set of short-range wireless technologies, typically requiring a distance of 4cm or less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and an Android-powered device, or between two Android-powered devices.
PACKAGE_USAGE_STATS	Allows an application to collect component usage statistics. Declaring the permission implies intention to use the API and the user of the device can grant permission through the Settings application.
PROCESS_OUTGOING_CALLS	Allows an application to see the number being dialed during an outgoing call with the option to redirect the call to a different number or abort the call altogether.
READ_CALENDAR	Allows an application to read the user's calendar data.
READ_CALL_LOG	Allows an application to read the user's call log.
READ_CONTACTS	Allows an application to read the user's contacts data.
READ_EXTERNAL_STORAGE	Allows an application to read from external storage.
READ_FRAME_BUFFER	Allows an application to take screen shots and more generally get access to the frame buffer data.
READ_INPUT_STATE	This constant was deprecated in API level 16. The API that used this permission has been removed.
READ_LOGS	Allows an application to read the low-level system log files.
READ_PHONE_STATE	Allows read only access to phone state.
READ_SMS	Allows an application to read SMS messages.
READ_SYNC_SETTINGS	Allows applications to read the sync settings.
READ_SYNC_STATS	Allows applications to read the sync stats.
READ_VOICEMAIL	Allows an application to read voicemails in the system.

REBOOT	Required to be able to reboot the device.
RECEIVE_BOOT_COMPLETED	Allows an application to receive the ACTION_BOOT_ COMPLETED broadcast after the system finishes booting.
RECEIVE_MMS	Allows an application to monitor incoming MMS messages.
RECEIVE_SMS	Allows an application to receive SMS messages.
RECEIVE_WAP_PUSH	Allows an application to receive WAP push messages.
RECORD_AUDIO	Allows an application to record audio.
REORDER_TASKS	Allows an application to change the Z-order of tasks.
REQUEST_IGNORE_ BATTERY_OPTIMIZATIONS	Permission an application must hold in order to use ACTION_REQUEST_IGNORE_BATTERY_OPTIMIZATIONS.
REQUEST_INSTALL_PACKAGES	Allows an application to request installing packages.
SEND_RESPOND_VIA_MESSAGE	Allows an application (Phone) to send a request to other applications to handle the respond-via-message action during incoming calls.
SEND_SMS	Allows an application to send SMS messages.
SET_ALARM	Allows an application to broadcast an Intent to set an alarm for the user.
SET_ALWAYS_FINISH	Allows an application to control whether activities are immediately finished when put in the background.
SET_ANIMATION_SCALE	Modify the global scaling factor.
SET_DEBUG_APP	Configure an application for debugging.
SET_PROCESS_LIMIT	Allows an application to set the maximum number of (not needed) application processes that can be running.
SET_TIME	Allows applications to set the system time.
SET_TIME_ZONE	Allows applications to set the system time zone.
SET_WALLPAPER	Allows applications to set the wallpaper.
SET_WALLPAPER_HINTS	Allows applications to set the wallpaper hints.
SIGNAL_PERSISTENT_PROCESSES	Allow an application to request that a signal be sent to all persistent processes.
STATUS_BAR	Allows an application to open, close, or disable the status bar and its icons.
SYSTEM_ALERT_WINDOW	Allows an app to create windows using the type TYPE_SYSTEM_ALERT, shown on top of all other apps.
TRANSMIT_IR	Allows using the device's IR transmitter, if available. IR Transmitter allows communication with remote controls.
UNINSTALL_SHORTCUT	Allows an application to uninstall a shortcut in Launcher.
UPDATE_DEVICE_STATS	Allows an application to update device statistics.
USE_FINGERPRINT	Allows an app to use fingerprint hardware.
USE_SIP	Allows an application to use SIP service.
VIBRATE	Allows access to the vibrator.
WAKE_LOCK	Allows using PowerManager WakeLocks to keep processor from sleeping or screen from dimming.

WRTE_APN_SETTINGS	Allows applications to write the apn settings.
WRITE_CALENDAR	Allows an application to write the user's calendar data.
WRITE_CALL_LOG	Allows an application to write (but not read) the user's contacts data.
WRITE_CONTACTS	Allows an application to write the user's contacts data.
WRITE_EXTERNAL_STORAGE	Allows an application to write to external storage.
WRITE_GSERVICES	Allows an application to modify the Google service map.
WRITE_SECURE_SETTINGS	Allows an application to read or write the secure system settings.
WRITE_SETTINGS	Allows an application to read or write the system settings.
WRITE_SYNC_SETTINGS	Allows applications to write the sync settings.
WRITE_VOICEMAIL	Allows an application to modify and remove existing voicemails in the system.
BILLING	Allows the user to make purchases within the app.

Table A.1: Permission Sticky Notes Content

Context	Description
ENTRY_POINT	All application's entry points. Unlike apps on most other systems, Android apps don't have a single entry point (there's no main() function, for example).
RECEIVER_METHOD	A receiver is an Android component which allows you to register for system or application events. All registered receivers for an event are notified by the Android runtime once this event happens. For example, applications can register for the ACTION_BOOT_COMPLETED system event which is fired once the Android system has completed the boot process.
DESTROY_METHOD	Called when an activity (app window) finishes its life cycle. Called once in the lifecycle of an activity (app window).
SERVICE_METHOD	A Service is an application component that can perform long-running operations in the background and does not provide a user interface.
START_METHOD	Once the onCreate() finishes execution, the system calls the onStart() and onResume() methods in quick succession. Your activity (app window) never resides in the Created or Started states. Technically, it becomes visible to the user when onStart() is called, but onResume() quickly follows and the activity remains in the Resume state.
STOP_METHOD	When your activity (app window) receives a call to the onStop() method it's no longer visible.
PAUSE_METHOD	When the system calls onPause() for your activity (app window), it technically means your window is partially visible, but most often is an indication that the user is leaving the window and it will soon enter the Stopped state.
ACTIVITY_METHOD	Activity (app window) methods.

RESUME_METHOD	When the user resumes your activity (app window) from the Paused state.
ONCREATE_METHOD	Called when an activity (app window) starts its life cycle. Called once in the lifecycle of an activity (app window).
RESTART_METHOD	When your activity (app window) comes back to the foreground from the stopped state, it received a call to onRestart(). The system also calls the onStart() method, which happens every time your activity becomes visible. The onRestart method, however, is called only when the activity resumes from the stopped state.
PACKAGE_METHOD	Package class contains information about a Java package. This includes implementation and specification versions.
ONTOUCH_HANDLER	Handles Touch events.
ONCLICK_HANDLER	Handles Click events.
DO_INBACKGROUND	Allows a window to perfom background operations.

Table A.2: Context Sticky Notes Content

Appendix B

User Comprehension Survey Questions

We include here the questions asked in the Survey Instrument. Most multiple-choice questions were single answer only unless mentioned otherwise. We mark with "(select all that apply)" questions that respondents could choose more than one answer.

B.1 Android Usage Questions

Q1: How long have you used and Android phone?

( ) Less than a year
( ) 1 - 2 years
( ) 2 - 3 years
( ) 3 - 4 years
( ) Over 4 years

Q2: What is the OS version of your Android phone? *Open your device's Settings. Tap About Phone / About Device / General. Tap Android Version to display your version information.*

( ) Froyo 2.2 - 2.2.3
( ) Gingerbread 2.3.3 - 2.3.7
( ) Ice Cream Sandwich 4.0.3 - 4.0.4
( ) Jelly Bean 4.1 - 4.3.1
( ) KitKat 4.4 -4.4.4
( ) Lollipop 5 - 5.1.1

( ) Marshmallow 6.0 - 6.0.1

Q3: Where you download applications? *Select all that apply*

( ) Google Play

( ) HiMarket

( ) Wandoujia

( ) Other: ____

Q4: What factors do you consider before installing an application? *Rank them from 5 for the important to 1 for least important*

[___] App store reviews

[___] Internet reviews

[___] Screenshots

[___] Permissions

[___] Accessibility features

Q5: Have you ever not installed an app because of permissions?

( ) Yes, I didn't like the permissions

( ) Yes, there were too many permissions

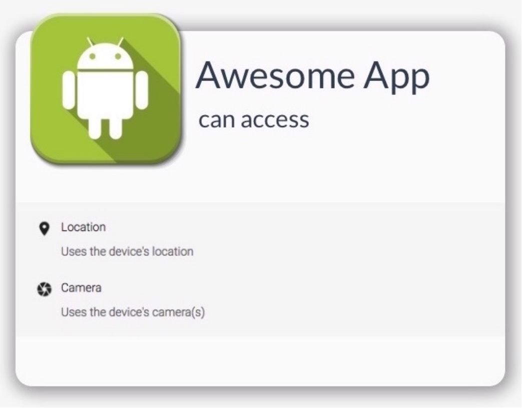
( ) No

( ) I don't know

B.2 Comprehension Questions

For each question all users were shown five statements and one screen. The screen depended on the group they belonged to: control or experiment.

Q1: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

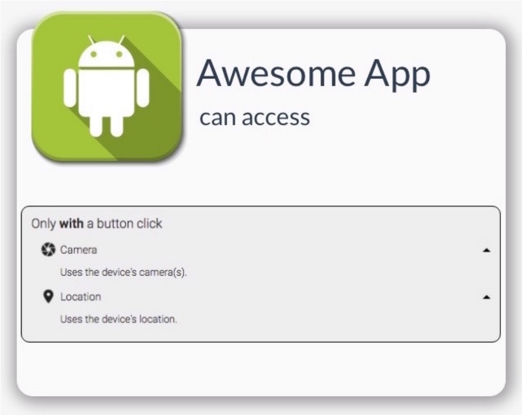
Location

Uses the device's location

Camera

Uses the device's camera(s)

(a) Control



Awesome App

can access

Only **with** a button click

Camera

Uses the device's camera(s).

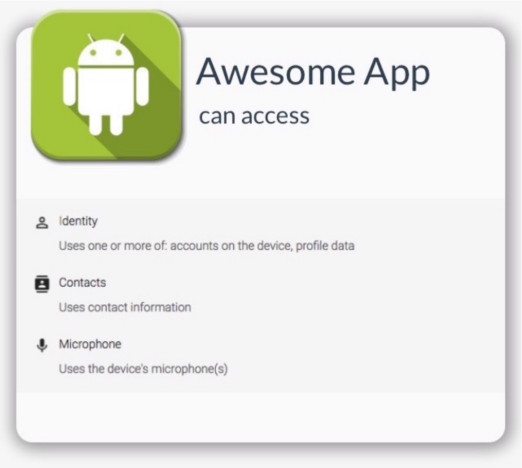
Location

Uses the device's location.

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Take pictures when you press the button.	○	○	○	○	○
Get your location.	○	○	○	○	○
Take pictures at any time.	○	○	○	○	○
Get your location while you are using other applications.	○	○	○	○	○
Load ads.	○	○	○	○	○

Q2: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

Identity

Uses one or more of: accounts on the device, profile data

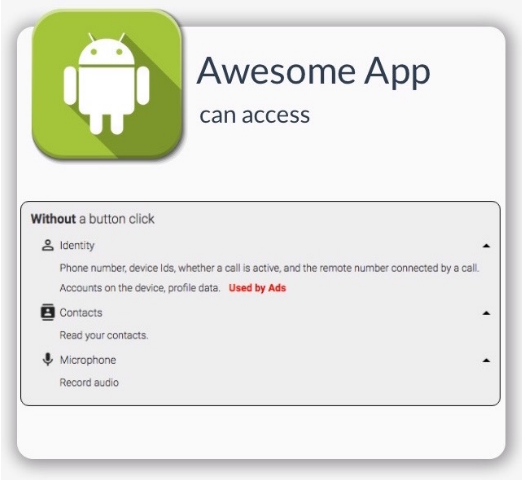
Contacts

Uses contact information

Microphone

Uses the device's microphone(s)

(a) Control



Awesome App

can access

Without a button click

Identity

Phone number, device Ids, whether a call is active, and the remote number connected by a call. Accounts on the device, profile data. **Used by Ads**

Contacts

Read your contacts.

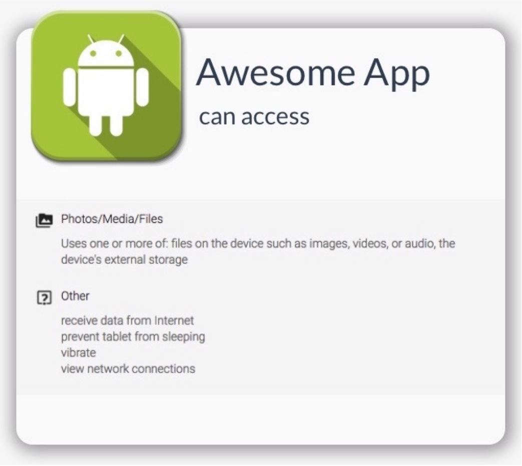
Microphone

Record audio

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Record audio when you open the app.	○	○	○	○	○
See who you have called.	○	○	○	○	○
Record audio while you are using other applications.	○	○	○	○	○
Read your heart rate.	○	○	○	○	○
Allow ads make your phone vibrate.	○	○	○	○	○

Q3: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

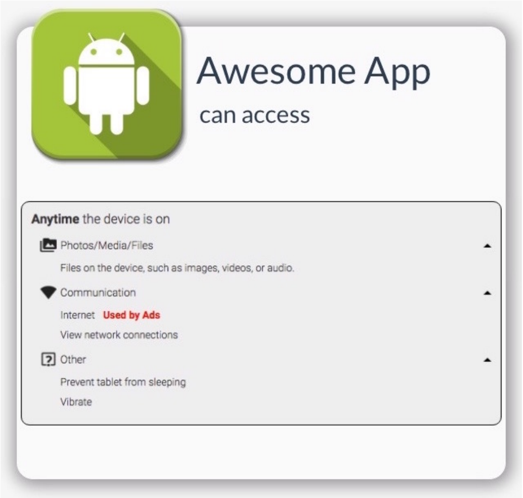
Photos/Media/Files

Uses one or more of: files on the device such as images, videos, or audio, the device's external storage

Other

receive data from Internet
prevent tablet from sleeping
vibrate
view network connections

(a) Control



Awesome App

can access

Anytime the device is on

Photos/Media/Files

Files on the device, such as images, videos, or audio.

Communication

Internet **Used by Ads**

View network connections

Other

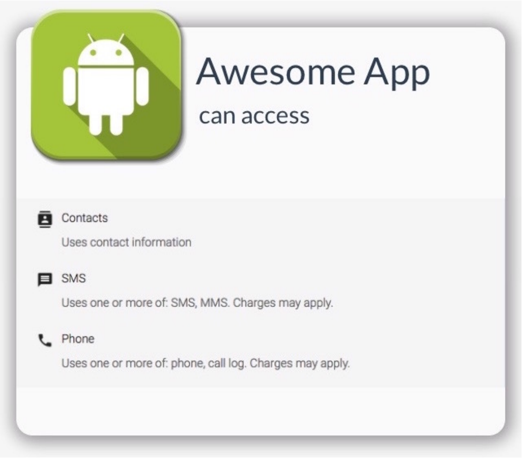
Prevent tablet from sleeping

Vibrate

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Write on the SD card even when the application is closed.	○	○	○	○	○
Keep your phone's screen on all the time.	○	○	○	○	○
Add events to your calendar at any time.	○	○	○	○	○
Read your phone number.	○	○	○	○	○
Place phone calls.	○	○	○	○	○

Q4: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

Contacts

Uses contact information

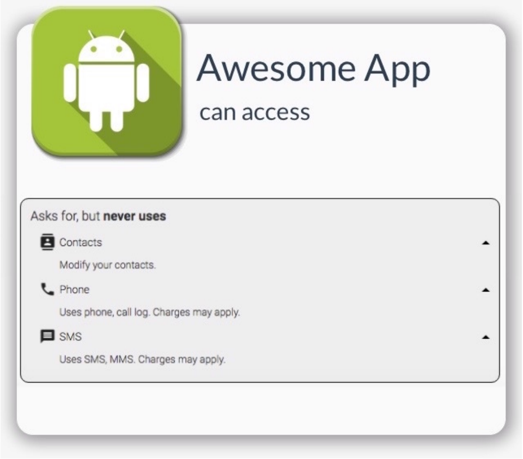
SMS

Uses one or more of: SMS, MMS. Charges may apply.

Phone

Uses one or more of: phone, call log. Charges may apply.

(a) Control



Awesome App

can access

Asks for, but **never uses**

Contacts

Modify your contacts.

Phone

Uses phone, call log. Charges may apply.

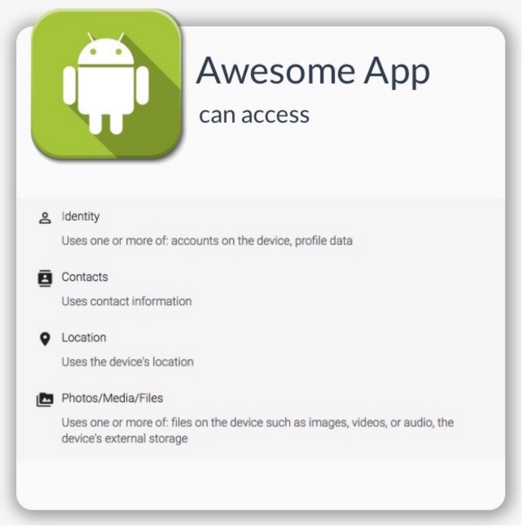
SMS

Uses SMS, MMS. Charges may apply.

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Read your phone's contact list while you are not using the application.	○	○	○	○	○
Modify your phone's contact list.	○	○	○	○	○
Send text messages.	○	○	○	○	○
Get your location.	○	○	○	○	○
Place phone calls.	○	○	○	○	○

Q5: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

Identity

Uses one or more of: accounts on the device, profile data

Contacts

Uses contact information

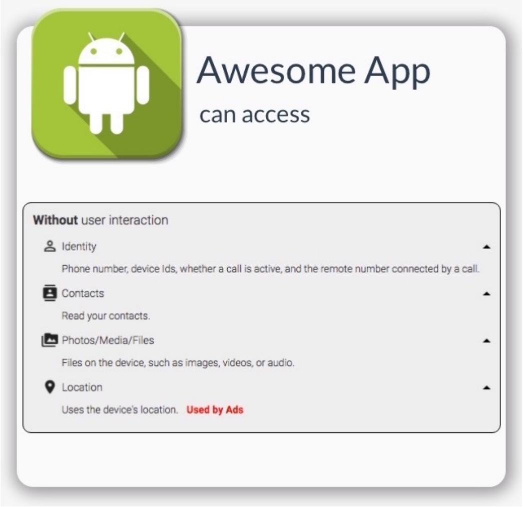
Location

Uses the device's location

Photos/Media/Files

Uses one or more of: files on the device such as images, videos, or audio, the device's external storage

(a) Control



Awesome App

can access

Without user interaction

Identity

Phone number, device Ids, whether a call is active, and the remote number connected by a call.

Contacts

Read your contacts.

Photos/Media/Files

Files on the device, such as images, videos, or audio.

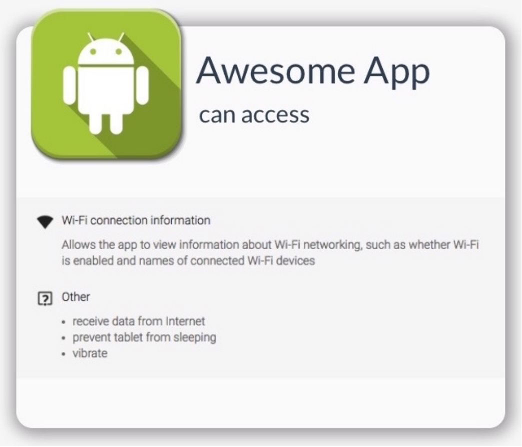
Location

Uses the device's location. **Used by Ads**

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Write on the SD card while you are using the app.	○	○	○	○	○
Read your phone number.	○	○	○	○	○
Place phone calls.	○	○	○	○	○
Allow ads to access your contacts list.	○	○	○	○	○
Get your location.	○	○	○	○	○

Q6: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

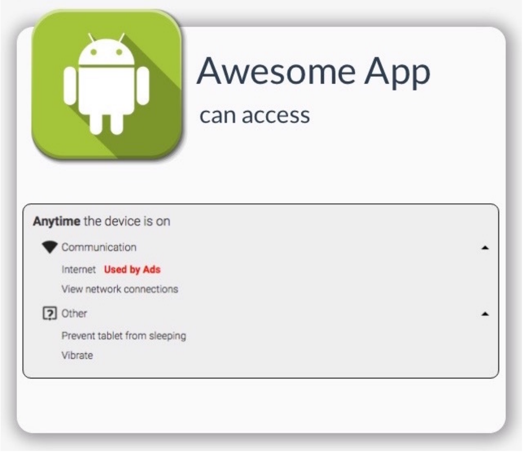
Wi-Fi connection information

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and names of connected Wi-Fi devices

Other

- receive data from Internet
- prevent tablet from sleeping
- vibrate

(a) Control



Awesome App

can access

Anytime the device is on

Communication

Internet **Used by Ads**

View network connections

Other

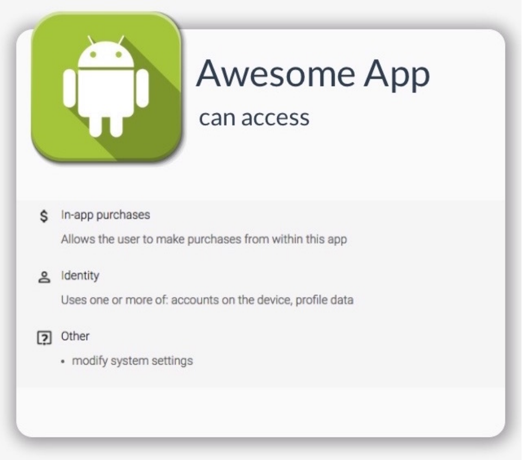
Prevent tablet from sleeping

Vibrate

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Load ads.	○	○	○	○	○
Allow ads make your phone vibrate.	○	○	○	○	○
Allow ads to pair Bluetooth devices.	○	○	○	○	○
Read your calendar anytime.	○	○	○	○	○
Record audio after pressing. “Start recording” button.	○	○	○	○	○

Q7: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

$ In-app purchases

Allows the user to make purchases from within this app

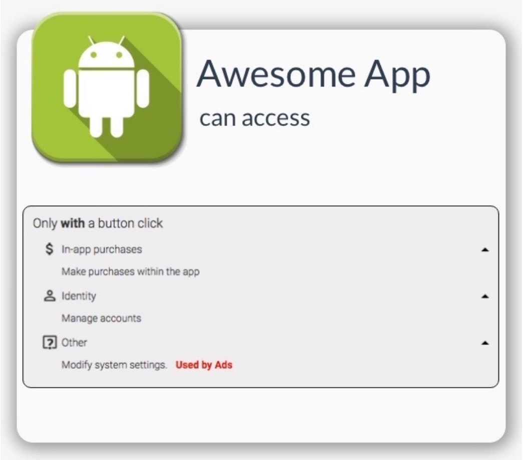
Identity

Uses one or more of: accounts on the device, profile data

Other

- modify system settings

(a) Control



Awesome App

can access

Only **with** a button click

$ In-app purchases

Make purchases within the app

Identity

Manage accounts

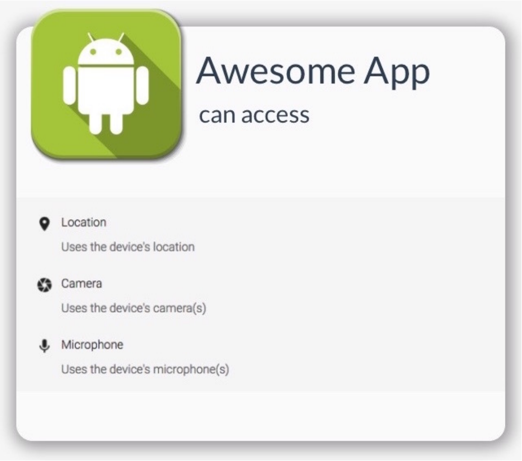
Other

Modify system settings. **Used by Ads**

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Charge purchases to your credit card when you click a button.	○	○	○	○	○
Allow ads to modify settings.	○	○	○	○	○
Pair Bluetooth devices.	○	○	○	○	○
Modify settings when you click a button.	○	○	○	○	○
Charge purchases to your credit card when you open the app.	○	○	○	○	○

Q8: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

Location

Uses the device's location

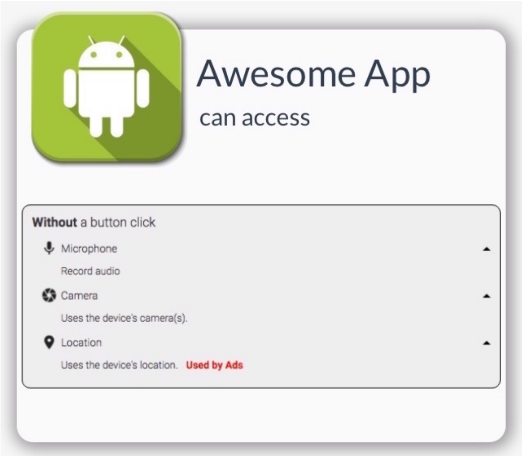
Camera

Uses the device's camera(s)

Microphone

Uses the device's microphone(s)

(a) Control



Awesome App

can access

Without a button click

Microphone

Record audio

Camera

Uses the device's camera(s).

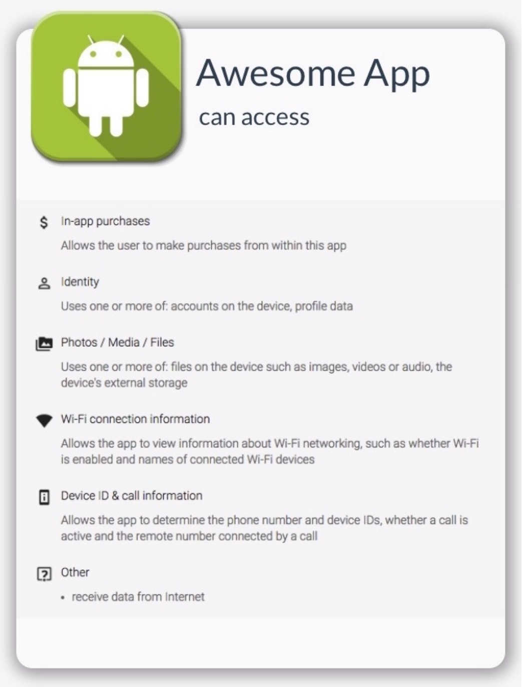
Location

Uses the device's location. **Used by Ads**

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Get your location.	○	○	○	○	○
Allow ads to know your location.	○	○	○	○	○
Read your phone's contact list while you are not using the application	○	○	○	○	○
Get your location while you are not using the application.	○	○	○	○	○
Send text messages.	○	○	○	○	○

Q9: You found an app that you want to install. Right before installing you see a screen that says:



Awesome App

can access

$ In-app purchases

Allows the user to make purchases from within this app

Identity

Uses one or more of: accounts on the device, profile data

Photos / Media / Files

Uses one or more of: files on the device such as images, videos or audio, the device's external storage

Wi-Fi connection information

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and names of connected Wi-Fi devices

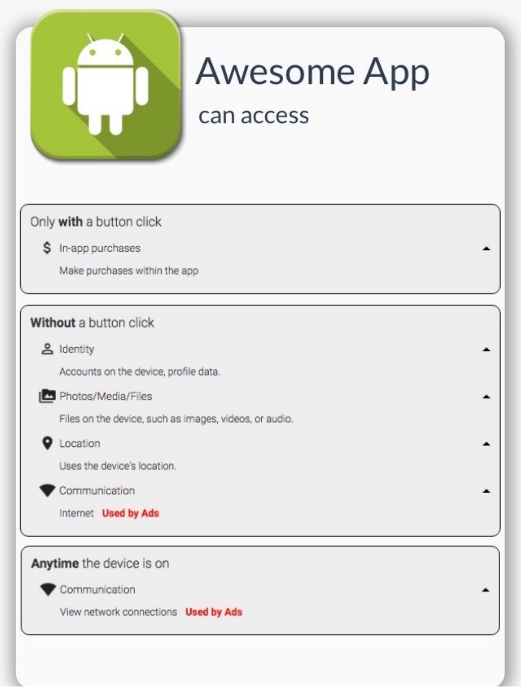
Device ID & call information

Allows the app to determine the phone number and device IDs, whether a call is active and the remote number connected by a call

Other

- receive data from Internet

(a) Control



Awesome App

can access

Only **with** a button click

$ In-app purchases

Make purchases within the app

Without a button click

Identity

Accounts on the device, profile data.

Photos/Media/Files

Files on the device, such as images, videos, or audio.

Location

Uses the device's location.

Communication

Internet **Used by Ads**

Anytime the device is on

Communication

View network connections **Used by Ads**

(b) Experiment

	Absolutely Impossible	Impossible	Neutral	Possible	Absolutely Possible
Get your location.	○	○	○	○	○
Load ads.	○	○	○	○	○
Charge purchases to your credit card at any time.	○	○	○	○	○
Allow ads to know your location.	○	○	○	○	○
Write on the SD card when the app is closed.	○	○	○	○	○

B.3 Westin Index Questions

	Strongly Disagree	Somewhat Disagree	Somewhat Agree	Strongly Agree
Consumers have lost all control over how personal information is collected and used by companies	○	○	○	○
Most businesses handle the personal information they collect about consumers in a proper and confidential way.	○	○	○	○

Existing laws and organizational practices provide a reasonable level of protection for consumer privacy today.	○	○	○	○

B.4 Demographic Questions

Q1: Please enter your MTurk ID:

Q2: What is your age?

( ) 18 - 28
( ) 29 - 39
( ) 40 - 50
( ) 51 - 61
( ) Over 62

Q3: What is your gender?

( ) Male
( ) Female
( ) Prefer not to answer
( ) Other

Q4: Highest level of education achieved

( ) High school
( ) Some college
( ) Bachelors
( ) Masters
( ) Phd

Q5: Have you ever worked in a high tech job? e.g. system administrator, computer programmer

( ) Yes
( ) No

Appendix C

Results of User Comprehension Survey

	Correct	Incorrect	Neutral	More Conservative	Less Conservative
Experiment	913	428	179	0	0
Control	834	413	193	201	46
Total	1747	841	339		

Table C.1: Permission Statements Scores

	Correct	Incorrect	Neutral	More Conservative	Less Conservative
Experiment	827	695	283	0	0
Control	749	689	272	980	251
Total	1576	1384	555		

Table C.2: Permission-Context Statements Scores

	Correct	Incorrect	Neutral	More Conservative	Less Conservative
Experiment	525	201	129	0	0
Control	462	206	142	432	41
Total	987	407	271		

Table C.3: Permission-Ads Statements Scores

	Experiment			Control		
Statement	Yes	No	Neutral	Yes	No	Neutral
Take pictures when you press the button.	88	3	4	84	2	4
Get your location.	85	2	8	88	0	0
Take pictures at any time.	41	33	21	62	13	15
Get your location while you are using other applications.	67	16	12	74	8	8
Load ads	34	30	31	48	18	24
Record audio when you open the app.	87	5	3	80	5	5
See who you have called.	85	2	8	66	7	17
Record audio while you are using other applications.	75	10	10	70	9	11
Read your heart rate.	14	73	8	9	69	12
Allow ads make your phone vibrate.	40	34	21	29	32	29
Write on the SD card even when the application is closed.	40	28	27	49	22	19
Keep your phone’s screen on all the time.	76	7	12	65	7	18
Add events to your calendar at any time.	30	46	19	20	52	18
Read your phone number.	42	34	19	37	34	19
Place phone calls.	25	56	14	16	59	15
Read your phone’s contact list while you are not using the application.	65	18	12	72	7	11
Modify your phone’s contact list.	75	15	5	50	28	12
Send text messages.	71	15	9	76	10	4
Get your location.	40	39	16	31	39	20
Place phone calls	69	16	10	75	8	7
Write on the SD card while you are using the app.	46	30	19	61	15	14
Read your phone number.	86	2	7	74	4	12
Place phone calls.	39	39	17	25	45	20
Allow ads to access your phone’s contacts list.	68	14	13	70	9	11
Get your location.	88	1	6	82	3	5
Load ads	88	2	5	71	4	15
Allow ads make your phone	74	8	13	75	5	10

vibrate.						
Allow ads to pair Bluetooth devices	38	38	19	27	41	22
Read your calendar any	28	49	18	16	50	24
Record audio after pressing “Start recording” button.	25	52	18	20	49	21
Charge purchases to your credit card when you click a button.	81	7	7	70	11	9
Allow ads to modify settings.	78	4	13	68	9	13
Pair Bluetooth devices	30	37	28	41	26	23
Modify settings when you click a button.	87	1	7	75	4	11
Charge purchases to your credit card when you open the app.	35	44	16	44	30	16
Get your location.	87	2	6	86	1	3
Allow ads to know your location	83	5	7	75	7	8
Read your phone’s contact. list while you are not using the application	26	51	18	17	55	18
Get your location while you are not using the application	78	9	8	75	6	9
Send text messages.	12	70	13	11	68	11
Get your location.	88	2	5	52	27	11
Load ads	86	2	7	73	7	10
Charge purchases to your credit card at any time.	26	52	17	33	40	17
Allow ads to know your location.	79	8	8	47	24	19
Write on the SD card when the app is closed.	32	28	35	49	17	24

Table C.4: Statements Results Frequency

Appendix D

Ethical Review Procedures

Ethical Review Procedures: Level 1

Project Details & Self-assessment

This document is closely modelled on documents used in School of Philosophy, Psychology and Language Sciences provided by Ellen Bard and Cedric MacMartin.

This form is to be filled in and submitted at the same time as the project proposal or the funding application it applies to. The form should be submitted by the Principal Investigator, except in the following cases:

- Post-doctoral fellowships – the proposed postdoc mentor.
- UG, MSc, and PhD research projects – the supervisor.
- Visiting researcher – the staff hosting the visitor.

Please submit the completed form by email to: **infkm+ethics@inf.ed.ac.uk**

This address, with appropriate RT number once issued, should be used for all correspondence (including forms and attached documents). This is essential to ensure proper record keeping. No signature is required if the form is sent from a valid University email address.

Project Details

1 **Type Of Project:**

□ Research grant proposal
□ Post-doctoral fellowship
□ Personal research

□ UG final year project
□ PhD project
□ Other: _______________

X MSc project
□ Research performed by visiting researcher

2 **Is there a sponsor/ funding body?** NO

3 **Does the sponsor/funder require formal prior ethical review?** NO
If yes, by what date is a response required ?

4 **Is any other institution and/or ethics committee involved?** NO

If YES, give details and indicate the status of the application at each other institution or ethics committee (i.e., submitted, approved, deferred, rejected):

5 **Title of Project:** App permissions and static analysis

6 **Researchers' names, affiliations, emails**

Include student/supervisor, post-doc/mentor, PI, or visitor/host.

Dr. Kami Vaniea, University of Edinburgh, kvaniea@inf.ed.ac.uk

Maria Paz Velarde, University of Edinburgh, s1556573@sms.ed.ac.uk

7 **State which professional organisation guidelines you are using:**

X School of Informatics research ethics code: http://www.inf.ed.ac.uk/research/ethics/
□ Other ethics code as required by funding body or professional organization:
Title: __________________________________ URL: ______________________________

Self-assessment

Refer to Level 2 form for details on any of the following points.

1. **Protection of research participants' confidentiality**
   Are there any issues of CONFIDENTIALITY which are NOT ADEQUATELY HANDLED by normal tenets of academic confidentiality? NO

   These include well-established sets of procedures that may be agreed more or less explicitly with collaborating individuals/organisations, for example, regarding:
   (a) Non-attribution of individual responses;
   (b) Individuals and organisations anonymised in publications and presentation;
   (c) Specific agreement with respondents regarding feedback to collaborators and publication.

2. **Data protection and consent**
   Are there any issues of DATA HANDLING AND CONSENT which are NOT ADEQUATELY DEALT WITH, and compliant with established procedures? NO

   These include well-established sets of procedures, for example regarding:
   (a) Compliance with the University of Edinburgh's Data Protection procedures (see http://www.recordsmanagement.ed.ac.uk);
   (b) Respondents giving consent regarding the collection of personal data (via consent form).

3. **Significant potential for physical or psychological harm, discomfort or stress**
   Are there any risks of :
   (a) psychological harm or stress for the participants? NO
   (b) physical harm or discomfort for the participants? NO
   (c) any kind to the researcher? NO

4. **Vulnerable participants**

Are any of the participants in the research vulnerable, e.g., children, patients, disabled participants? NO

5. **Moral issues and researcher/institutional conflicts of interest**
   Are there any SPECIAL MORAL ISSUES/CONFLICTS OF INTEREST? These include:
   (a) Conflict of interest: potential benefit to the researcher, friends or family of a particular research outcome which might compromise the researcher's objectivity or independence;
   (b) The need to keep the purposes of research concealed;
   (c) Use of participants who are unable to provide informed consent (e.g., children);
   (d) Situations where research findings would impinge negatively/differentially upon the interests of participants.
   NO

6. **Bringing the University into disrepute**
   Is there any aspect of the proposed research which might bring the University into disrepute? For example, could any aspect of the research be considered controversial or prejudiced?
   NO

7. **Use of animals**
   Does the research involve animals? NO

8. **Developing countries**
   Does the research involve developing countries? NO

9. **Dual use**

Is the research classified or does it have specific adversarial military applications? NO

10. Terrorist or extremist groups
Does your research concern groups which may be construed as terrorist or extremist? NO

Can you stop now?
You may want to assure yourself that your 'NO' answers are correct by checking the detailed form in the next section.

If all the YES / NO answers are NO, the self assessment has been conducted and confirms the ABSENCE OF REASONABLY FORESEEABLE ETHICAL RISKS. This form should be signed by the researchers and submitted. The researchers may retain a copy for their own records.

If any answer is YES, please complete the relevant section in the Level 2 form below.

Ethical Review Procedures: Level 2

Detailed Assessment

This material should help you answer the questions in the self-assessment form.

If any difficulties arise, you should fill in the relevant parts of this form in consultation with a near colleague who is not directly involved with the research. You can also seek advice from members of the School Ethics Panel, or from relevant Ethics Committees of other schools.

You should file a new form if you receive advice on changes from the School or College Ethics Committees. For accountability, the School will view the most recent submission as accurate.

1. Protection of research participants' confidentiality

Refer to the University Data Protection Policy to ensure that the relevant conditions relating to the processing of personal data under Schedule 2 and Schedule 3 are satisfied. Details are available at: http://www.recordsmanagement.ed.ac.uk.

1. If the research requires the collection of personal information from e.g., universities, schools, employers, or other agencies about individuals without their direct consent, what information will be sought and why will written consent for access to this information not be obtained from the participants themselves?

2. If any part of the research involving participants will be recorded using any electronic medium, what medium is to be used and how will the recordings be used?

3. Who will have access to the raw data?

4. If participants will be identified in your records, how will their consent to quotations/identifications be sought?

5. If they will not be identifiable, how will anonymity be preserved?

6. Will the datafiles/audio/video tapes, etc. be disposed of after the study?

7. If not, how long they will be retained and how will they eventually be disposed of?

8. How do you intend the results of the research to be used?

9. If feedback of findings will be given to participants, how and when will this feedback be provided?

2. Data protection and consent

Participants have the following rights over observations and records of their own behaviour:

- If they are engaging in any activity outside their normal daily routine (for example answering a questionnaire, listening for a particular syllable), they must be given some account of what they will be asked to do before they start, and must formally consent to participation;
- In any event, if they will be observed or recorded, they must be informed of and consent to the kinds of record taken;
- They must be assured of anonymity in any publication or dissemination;
- They must consent to how the data will be used;
- They must be free to withdraw from participation at any time.

1. Explain how and when written consent will be obtained from participants or from those responsible for participants unable to consent meaningfully on their own behalf. (If further discussion of this form is needed, please attach a copy of any information sheets and consent forms.)
2. If participants cannot meaningfully provide formal consent in this way, normally someone who is legally able to act on their behalf, for example a parent or legal guardian, must do so. If any of the following cases apply, explain how you will obtain the necessary consent and if you will not, how you can proceed ethically without doing so.

   - administrative consent in lieu of participants' consent

     (Administrative consent may be deemed sufficient:

     i. where the data collection involves aggregated statistical information and where the collection of data presents no invasion of privacy and no potential social or emotional risks:

     ii. where studies focus on the development and evaluation of curriculum materials, resources, guidelines, test items, or programme evaluations rather than the study, observation, and evaluation of individuals.)

   - the consent of parents on behalf of minors,
   - the consent or assent (at least verbal) of minors,
   - the consent of participants who do not share a language with the researcher,
   - the consent of participants with special educational needs.

3. Significant potential for physical or psychological harm, discomfort or stress

If the research could induce any psychological stress or discomfort, state the nature of the risk and what measures will be taken to deal with such problems.

If the research requires any physically invasive or potentially physically harmful procedures, give details and outline procedures to be put in place to deal with potential problems.

If the research involves the investigation of any illegal behaviour, give details.

If there is a real risk of disclosure of activities which should be reported to the authorities, a warning to this effect must be included in the Informed Consent documents. Please provide the wording of this warning.

If there is any purpose to which the research findings could be put that could adversely affect participants, describe the potential risk for participants of this use of the data. Outline any steps that will be taken to protect participants.

If the research could adversely affect participants in any other way, give details and outline procedures to be put in place to deal with such problems.

If the research could adversely affect particular groups of people, describe these possible adverse effects and the protection to be put in place against them.

If the research is expected to benefit the participants, directly or indirectly, give details.

If the true purpose of the research will be concealed from the participants, explain what information will be concealed and why.

If participants will NOT be debriefed at the conclusion of the study, explain why not.

4. Vulnerable participants

What criteria will be used in deciding on the inclusion and exclusion of participants in the study?

If any of the participants are likely to be in any of the following vulnerable categories, indicate the category and describe the measures that will be used to recruit, protect and/or inform participants:

under 16 years of age	in the care of a Local Authority
known to have special educational needs	physically or mentally ill
vulnerable in other ways	members of a vulnerable or stigmatized minority
unlikely to share a language with the researcher	in a student-teacher relationship with the researchers
in any other dependent relationship with the researchers	likely to have difficulty in reading and/or comprehending any printed material distributed as part of the study

If participants will receive any financial or other material benefits because of participation, what benefits will be offered to participants and why?

5. Moral issues and researcher/institutional conflicts of interest

The University has a draft ‘Policy on the Conflict of Interest’. Regarding research the draft states that a conflict of interest would arise in cases where an employee of the University might be

> “ . . . compromising research objectivity or independence in return for financial or non-financial benefit for him/herself or for a relative or friend. . .”

The draft policy also states that the responsibility for avoiding a conflict of interest, in the first instance, lies with the individual, but that potential conflicts of interest should always be disclosed, normally to the line manager or Head of Department. Failure to disclose a conflict of interest or to cease involvement until the conflict has been resolved may result in disciplinary action and in serious cases could result in dismissal.

If your research involves a conflict of interest or any situation which could be construed as a conflict of interest, please give details.

6. Bringing the University into disrepute

If on the level 1 form you have answered that some aspect of the proposed research "might bring the University into disrepute", please elaborate alongside how this might arise, and what steps will be taken by the researcher to mitigate and manage this, to minimise adverse consequences to the University.

7. Use of animals [based on EU FP7 guidelines]

If the proposed research will use animals, please provide the following information:

1. Describe how you have applied the 3Rs: Reduction, Replacement, Refinement.
2. Describe and justify:
   - species and numbers of animals used;
   - humane end points and pain and suffering;
3. Describe how you have explored alternatives to using animals.
4. Answer the following questions:
   - Are those animals transgenic small laboratory animals?
   - Are those animals transgenic farm animals?
   - Are those animals cloning farm animals?
   - Are those animals non-human primates?

8. Developing countries [based on EU FP7 guidelines]

Questions to consider include:

1. Does the research project provide benefit to the local community (in terms of access to healthcare, education, allocation of property rights, capacity to assess and use modern technologies while respecting the population 's own choices and needs, etc.)?
2. Does the research project use local resources (genetic resources, animals, and plants)?

How to deal with research involving developing countries

The categories of issues requiring special attention include:

- A disproportionately heavy burden of diseases (particularly infectious diseases); the breadth and depth of poverty; and high levels of illiteracy
- Wide disparities in health systems and in access to health care; and imbalance between the often-ample resources available for research and the meagre resources available for even basic health care
- Inadequate scientific and ethics infrastructures for the required reviewing process
- The extent of disempowerment of the poor in their personal and communal lives
- Knowledge of the ways in which people of other cultures traditionally view themselves as individuals embedded in communities with respect to the changing boundaries between perceptions of the self that differ from the classical western notion
- The need to understand what it means to be ill in contexts very different from those known to researchers and what can be expected from those one consults for help under those circumstances

9. Dual use [based on EU FP7 guidelines]

1) What is considered as potential dual use

Generally speaking, dual use is a term often used in politics and diplomacy to refer to technology which can be used for both peaceful aims and adversarial military aims. Ethical issues of dual use might arise in cases where:

(d) Classified information, materials or techniques are used in research;

(e) Dangerous or restricted materials e.g. explosives are used in research;

(f) The specific results of the research could present a danger to participants, or to society as a whole, if they were improperly disseminated.

2) How to deal with potential dual use

Regarding implications for the use of and misuse of the research and products, the following measures and strategies can be applied:

(c) The researcher should show awareness of potential risks to participants and society as a whole from inappropriate dissemination of their results;

(d) Appropriate measures to deal with dangerous or restricted materials should be detailed, where applicable;

(e) An appropriate strategy to deal with issues of informed consent and risk management for participants and for society where classified information, materials or techniques are concerned should be demonstrated;

(f) An advisory board should be included in the project, which should identify risks to participants from particular research activities and devise a strategy for minimising and dealing with these risks;

(g) The dissemination and communication strategy of the study results to a wider audience should be controlled by the advisory board, which should report to the relevant funding body on a regular basis.

EU FP7 ethical guidelines can be found at http://cordis.europa.eu/fp7/ethics_en.html.

10. Terrorist or extremist groups

If your research concerns groups which may be construed as terrorist or extremist, please fill in the following form and submit it with your ethics form.

Prevent Duty supplementary form

The Terrorism Act (2006) outlaws the dissemination of records, statements and other documents that can be interpreted as promoting or endorsing terrorist acts.

1. Does your research involve the storage on a computer of any such records, statements or other documents? Yes / No
2. Might your research involve the electronic transmission (eg as an email attachment) of such records or statements? Yes / No
3. If you answered 'Yes' to questions 1 or 2, you are advised to store the relevant records or statements electronically on a secure university file store. The same applies to paper documents with the same sort of content. These should be scanned and uploaded. Access to this file store will be protected by a password unique to you and your School Research Ethics Officer. Please indicate below that you agree to store all documents relevant to questions 1 and 2 on that file store: Yes

3a. Please indicate below that you agree not to transmit electronically to any third party documents in the document store: Yes

4. Will your research involve visits to websites that might be associated with extreme, or terrorist, organisations? Yes / No

5. If you answer ‘Yes’ to question 4, you are advised that such sites may be subject to surveillance by the police. Accessing those sites from university IP addresses might lead to police enquiries. Please acknowledge that you understand this risk by putting an ‘X’ in the ‘Yes’ box.
   Yes
6. By submitting to the ethics process, you accept that your School Research Ethics Officer and the convenor of the University’s Compliance Group will have access to a list of titles of documents (but not the contents of documents) in your document store. Please acknowledge that you accept this by putting an ‘X’ in the ‘Yes’ box. Yes

Countersigned by supervisor/manager