LEVERAGING PSYCHOLOGICAL FRAMING OF WARNING MESSAGES TO IMPROVE USER SECURITY ON SMARTPHONES

BY DAVID LAMBROPOULOS

A thesis submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Janne Lindqvist

and approved by

_____

_____

_____

New Brunswick, New Jersey

October, 2018

ABSTRACT OF THE THESIS

Leveraging Psychological Framing of Warning Messages to Improve User Security on Smartphones

by David Lambropoulos

Thesis Director: Janne Lindqvist

The current way in which Android warns users about the capabilities their apps request, do not effectively warn users about their security risks. This work focused on the design, implementation and evaluation of smartphone warning system. We also utilized psychological framing for design of warning messages. These warning messages were delivered to the user in a field study ($N = 41$) using their personal smartphones. We present effective strategies for warning smartphone users. These resulted in average longer viewing/interaction times, higher on average rate at which they consider making changes to their permission settings and were more likely to view apps requesting their permissions.

Acknowledgements

Thank you to Dr. Meghan Mclean for all the work you put in and all the helpful advice.

This material is based upon work supported by the National Science Foundation under Grant Numbers 1228777 and 1750987. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

There are many facets that go into relaying information to users and coming up with a methodology as a means to effectively warn a user in order to motivate them to consider the security risks of their smartphone settings. Particularly, the smartphone settings focused on in this thesis are the **permissions** that the user has the ability to grant or deny access to applications that are requesting them. These permissions are termed **dangerous permissions** as they are able to access personal and harmful information about the user.

1.1 Motivation

The focus of this thesis was the development of a system that would prove to be an effective method of creating and delivering warning messages to the users about their potential security risks regarding their smartphone devices. The design principles of this warning system are ease of use, non-evasiveness, and informative to the user.

Users typically are not aware of their security risks[22, 30, 34, 41, 48, 62, 71, 35] regarding the personal information that is being accessed, and possibly collected, by **applications** (termed **apps**) on their smartphone devices. Therefore, the purpose of this thesis is twofold; that which is focused on the design of warning messages that will inform the user about changes that can be performed to improve their privacy and security, and that which is concerned about design of a system or method to deliver said warnings.

One question that played a major role as a design factor: How should a warning be designed in order to maintain user attention while reducing how prone a user is to get habituated to the warnings? **Habituation** is defined as a attenuation in behavioral response as a result of a reoccurring stimulus[56]. Essentially, habituation is a naturally occurring simple form of learning in which people deem a reoccurring response as less essential in order to focus on a response that calls for attention. In relation to this work, habituation would refer to a user being continuously presented with warning messages and over-time dismissing them as unimportant.

The objectives for this warning system are as follows. One, the system will release warnings to the user after the user accesses their phone. Two, the user's device will be monitored for behaviors and decisions made while interacting with these warnings such as what options are chosen and how long the user read the warning. Three, the user's device will have information scanned periodically such as apps installed on their phones and permissions granted to those apps. Four, all warnings use a simple design that has clear information and options. These objectives are aligned with the design principles stated.

The system, once running on a user's device, runs without the knowledge of the user and only requires interaction or acknowledgement of the user upon presenting the user with a warning message. Once a decision to a warning has been made by the user, no further action is required by the user until the next warning. Furthermore, user time and attention being at a premium holds that the design of the warnings and understandability should remain clear to the user.

The warning system is synergistically designed to make more security informed users out of the average day-to-day smartphone users. The Android operating system warns users when an app on their device is requesting permission to access information or a facet of their device to perform its tasks. Android warns a user by simply stating that

the app is requesting a permission, and presents the user with the option to **'allow'** or **'deny'** the permission that is being requested[9, 8]. The warning contains no pertinent information about the permission. This current method just prompts the user with a generic statement such as "Allow **App** to access the camera on your device" and therein lies their limitation. There is no guarantee that the user will not become accustomed to the warning message and instinctually ignore this warning message as the content is static across messages and offers no additional information or repercussions for their security actions.

In order to reduce the chance that users ignore warnings, the warning system employed in this work applies the use of framing tactics. Specifically, the novel security warnings presented in this thesis employ one of three types of techniques that are believed to be important in designing effective warning messages. The warning messages either use fear framing, prosocial framing, or are designed to resemble the prototypical form of warning messages currently in Android. As opposed to todays current implementation in Android which does not provide the user with anything outside of the generic message, the warning system presented in this thesis applies these novel framing techniques to the warning content and provides the user with the necessary information to make changes to their security settings. After reading the novel warnings, the user has the tools to make future decisions regarding their security and privacy. Decisions such as revoking access to secure content deemed inappropriate by the user, via permissions, for a specific app to request personal information. The framing of these messages will motivate users to update their app permission settings.

These novel security warnings presented in this thesis employ one of three types of techniques that we believe to be most important in designing an effective warning message. The messages either use fear framing, prosocial framing, or are designed to educate the users regarding risks that come forth from settings to their security.

These novel security warnings will be beneficial to many users as they will be more likely to attend to the warning being given to them. Although this warning system was only tested on users of smartphones with Android 6.0 and later, older models are believed to also benefit from this system. Furthermore, the concept of framing warning messages and providing informative content can be generalized to multiple platforms such as iOS, smart TVs, smart watches and even internet browsers. Individuals are increasingly faced with decisions that they must make regarding their security and privacy and sometimes not even away that this is the case.

1.2 Background

1.2.1 Framing Techniques

Prosocial behaviors are defined as behaviors with the goal to selflessly do things that are beneficial to others[23]. People have been shown to be more likely to participate in an action if they believe by doing so will be beneficial to others. One such example would be that of getting vaccinations being perceived as beneficial to others[37, 47]. Although there has been work showing people's ability to influence the security behaviors of others'[27]. There has been very little work in the field of useable security. Only one prior work that had addressed the efficacy of prosocial framing in regards to warning message design[52].

Fear is defined as an emotion that is caused by the belief that something is dangerous and or threatening. It has been described as an unpleasant physiological state that will motivate people to engage in behaviors that will lead to its reduction[51]. Fear can affect a user's ability to make decisions and can be used to motivate them to act in a way that would be beneficial to themselves. There has been uses of fearful language in security warnings[63, 33] and in images that could have incited fear by malware warnings[17] but only a few studies actually intended to use fear[39, 16, 40].

1.2.2 Current Android Permission Model

Mobile devices are becoming increasingly ubiquitous with every passing day. As of the first quarter 2017 report via the IDC, there are 344.3 million smartphones used worldwide with 85.0% of volume running Android and 14.7% of volume running iOS[38]. The number of apps available to these smartphones is anything but attenuating. The main resource used on Android for acquiring these apps is the Google Play Store and on iOS the Apple App Store. The Google Play Store had offered only 30,000 apps back in December 2009, but by December 2017, this number has scaled to 3,500,000 apps[65] with over 65 billion downloads as of 2016[21].

The Android operating system has evolved over the course of its history. Today permissions are separated via their protection level of which there are three. Normal, which corresponds to access of data and resources that are of minimal risk to the user. Dangerous, which corresponds to access to data or resources that deal with the user's private information. Special, which corresponds to data or resources that are sensitive and above dangerous[9]. The user doesn't have the ability to adjust the permissions in that fall under the Normal category.

All versions prior to Android 6.0 Marshmallow used a one-time notice to their users upon installation where the user was presented with an exhaustive list of permissions requested by their apps where it was an all or nothing situation in which the user had to accept everything in order to install the desired app. This often led user to blindly give away permissions simply because of the want for app. The user was presented with a notice similar to the one below:

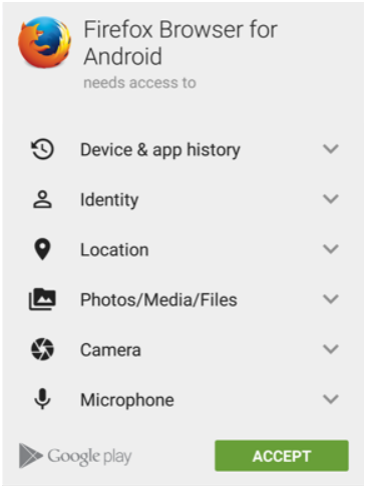
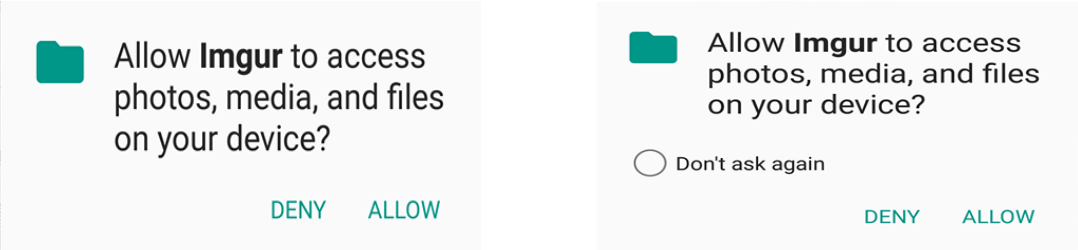


Figure 1.1: Android 5 and below install time permissions message

After the introduction of Android 6.0, Android changed how they handled permission warnings by implementing a just-in-time notice at run time. The just-in-time notification appears when an app requests a permission. These permissions are referred to as run-time permissions and only warned about permissions considered to be dangerous (grouped under the previously mentioned dangerous category). An example of run-time permissions that are encountered by the user is shown below:



(a) Type 1 (b) Type 2

Figure 1.2: Android 6.0+ Just-In-Time Permission Notification

The installation time permission screen is still available through the Google Play Store hidden at the bottom under **"Permission details"** as seen in 1.3. However, this is not clear or in plain sight to the users.

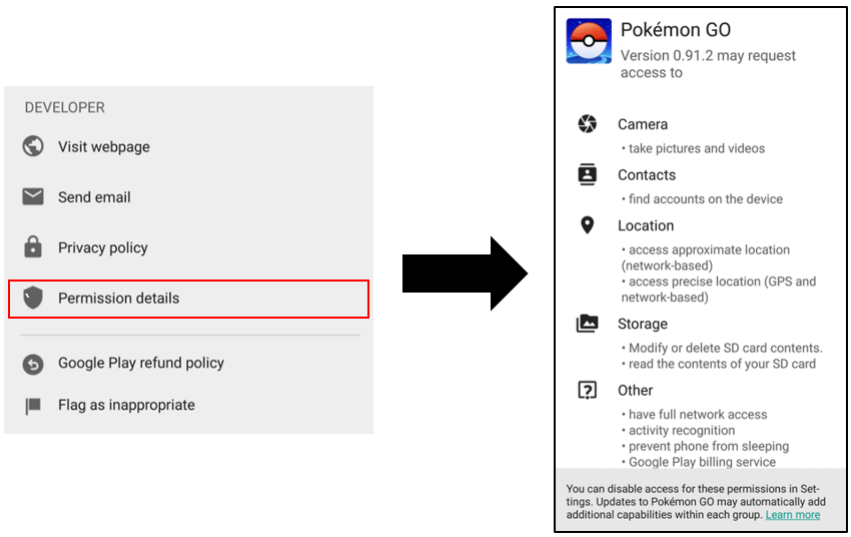


Figure 1.3: Google Play Store Permission Details

Android separates permissions into a list of permission groups in which they consider to be the most dangerous to your personal data. However, the user must accept or deny a whole group and they can't pick and choose items from a group, they must take it as is. All or nothing. These permission groups and their corresponding access to dangerous permissions are as follows:

Android organizes permissions into categories of related permissions called permission groups. There are nine permission groups defined by Android: Calendar, Camera, Contacts, Location, Microphone, Phone, Sensors, SMS, and Storage. A visualization of a these permission groups can be seen in the figure 1.4.

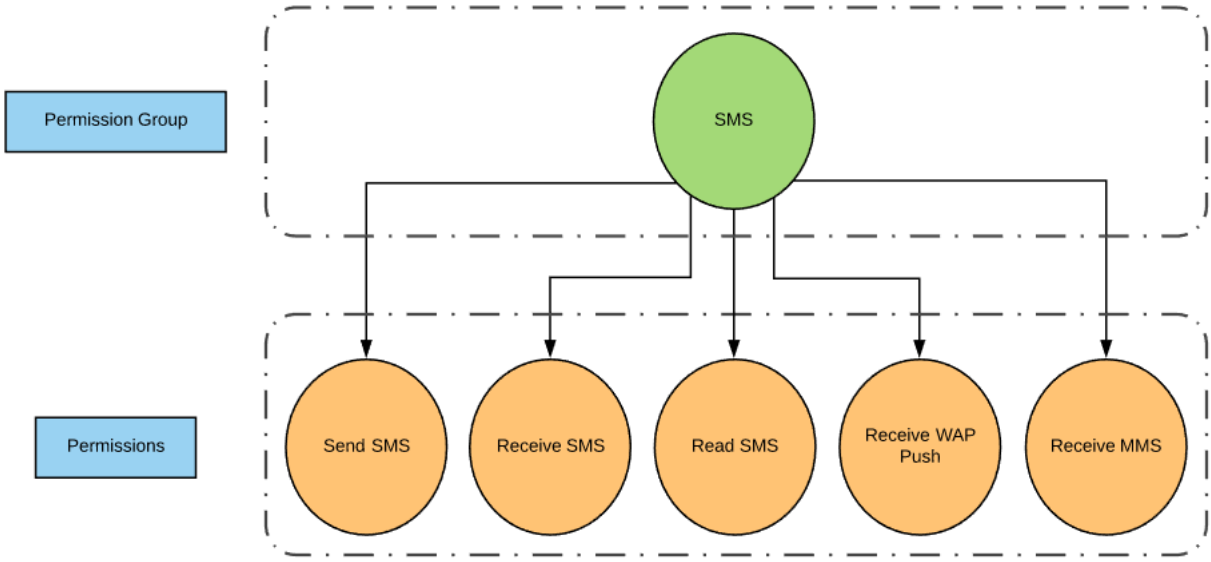


Figure 1.4: Android Permission Group Example: The dangerous permissions associated with the SMS permission group. Granting the SMS permission group to an application gives that application the right to all information related such as reading, sending, receiving SMS/MMS, Receive WAP Push.

As shown in 1.4, SMS contains the five dangerous permissions SEND_SMS, RECEIVE_SMS, READ_SMS, RECEIVE_WAP_PUSH, and RECEIVE_MMS. A full table description of the other eight permission groups can be seen in 1.5. As we cannot provide the typical Android users with fine grain control of their dangerous permissions, this thesis is focused primarily warning the users on the 9 permission groups.

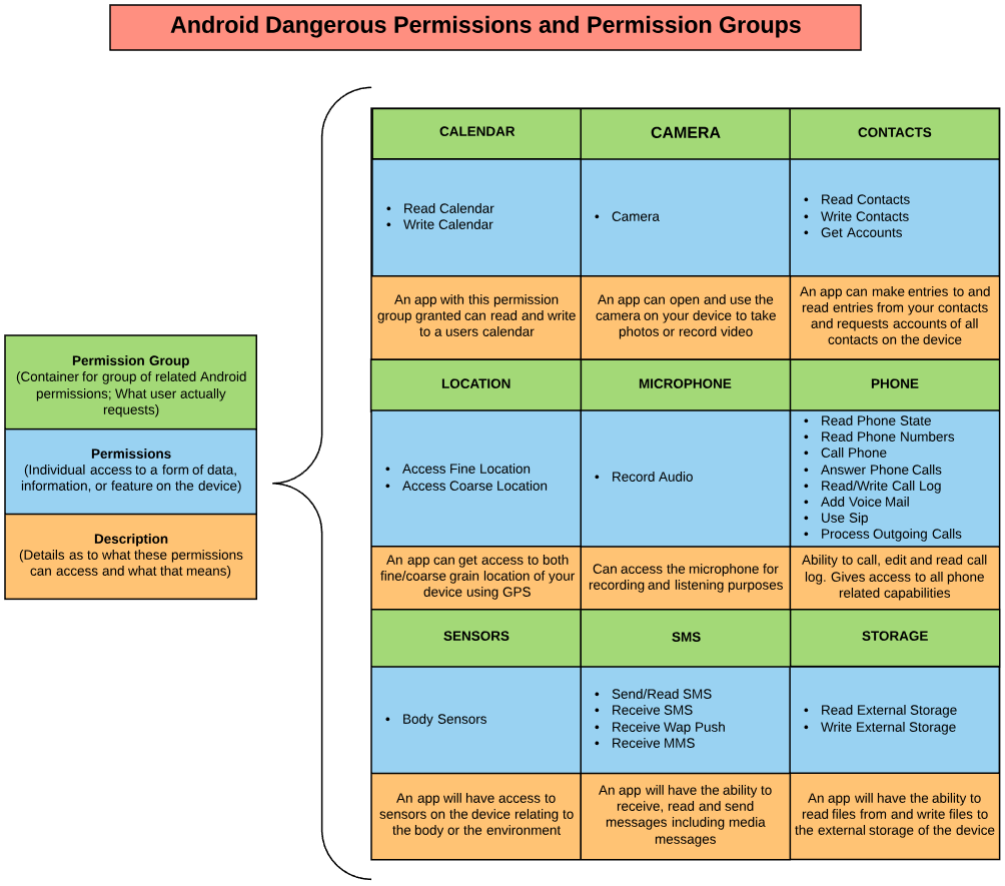


Figure 1.5: Android Permission Groups and Permissions. There are nine separate permission groups. Each permission group grants access to a few subpermissions that allow that permission group to accomplish what is in its description.

Chapter 2

Related Work

Past research had focused around notifying the users with respect to sensitive information that could be dangerous and personal to the users such as content focusing around permissions on smartphone devices and SSL/TLS warning messages that occur in browsers. Many studies mentioned that although they were effective in notifying, sometimes referred to nudging, a user they note that these nudges could fall victim to habituation. There has been a lot of research that focused on the design of privacy notices. It has been noted in the past that few people read notices[26].

Prior to this thesis, my colleagues and I had conducted another study with a similar focus but different approach[52]. This study's similarities were in that it focused around using fear and prosocial framing of warning messages to test their effectiveness. The previous study was a survey-based study that was available to participants through the Amazon Mechanical Turk platform in which we has 365 participants who had successfully completed the survey.

This study had demonstrated that framing can be effective in security warnings presented to participants on their smartphones. This study had also used fear, prosocial, and educationally framed warnings and set out to determine which style of framing would make for the most effective security warnings. All of the methods employed proved to be effective conditions for improving security behaviors and intentions of users. Effectiveness of the warning messages were judged based on how participants reacted and made changed their permission settings. Prosocial framing was found to

be the most effective, with respect to fear and education, in the framing of the security warning messages.

This study helped show why and how these types of framing techniques have an impact on user security. This study had participants read warning messages and record their responses on different scales gauging their feelings. The findings in this study were significant to say the least but it was just a static study in which users were exposed to images via an MTurk study and were not studied in a dynamic context. For example, users may react differently had these warnings been displayed to them on their devices. Furthermore, how would they react to seeing these warnings on a personal level containing apps that they have installed. Many further questions would be of interest to explore such as: How long do users typically read these messages? Do the users actually make the changes to their permissions, as they say they do, or are users simply just stating that they would like to or feel that they should make changes to their permissions but are not actually motivated enough to do so? Will a different framing technique take the lead as most effective in the context of a field study?

A similar study that was conducted by Liu et al.[49] focused on nudging users with recommendations that were geared towards motivating people to review and make changes to their security settings. They did this by using personalized information about the user's device in their recommendations. This study did not use any types of framing to motivate user to make such changes. Although this study showed how effective applying information that is relevant to the participants personalized information, this thesis used personalized information of the participants to design framed warning messages. The personalized information employed in this thesis is that of the apps that the participants have installed on their device and the permission settings of those apps.

Similarly, Almuhimedi et al.[18] had also focused around generating privacy nudges

to the user. This study focused specifically on warning the user periodically about the aggregate total of location accesses by the apps on their device. In his study he had found that it was effective to unintentionally use fearful language to help users understand the gravity of how many times apps have accessed their location. Others such as Fu et al[36] had also focus around apps requesting the location permission on a user's device and suggested using a per app run-time notification of location access.

Fear framing has been used in numerous studies relating to prevention of substance abuse[44, 45, 46, 53, 54, 57, 59, 58, 60, 66, 72, 79]. These studies found varying levels of effectiveness in regards to using fear to warn a participant and motivate them towards a desired behavior such as stopping the abuse of drugs. Fear has also been proven to encourage other healthy behaviors in individuals[73, 50]. Only a few studies actually intended to use fear[39, 16, 40] in the field of usable security and outside of [52], none have used fear intentionally as a framing technique in privacy notices.

There have been studies in the past that aid realizing how unaware users are regarding the data the is collected from their devices via the apps that they have installed[35, 62]. Furthermore, there have been studies that looked into methods of aiding users in an effort to manage their privacy from apps that they have installed on their devices[18, 22, 36]. In the case of Tan et al.[68] users had a higher chance of granting access to a permission given that there was an explanation provided. Put another way, Shih et al[61] showed that providing the user with an explanation that is unclear could lead to a decrease in the user's chance of granting the permission that is being requested.

Chapter 3

Design & Implementation

Designing a system that is both feasible and effective is key. Many factors contribute to the design decisions. To analyze the complete design one must look at all relevant components such as how the system works, how the user is to be warned, when the user is warned, how warnings are designed, and how storage of user data is handled.

The main design constraint taken into account was that of designing a warning system that did not motivate the user, via the warnings provided, to make changes to their smartphones such that this warning system would no longer function properly. As our warnings are focused around the security risks associated with dangerous permission groups, this app was designed to not require any of the dangerous permission that the user was warned about. This constraint was chosen as a safety measure to ensure that the user would not be able to interfere with the functionality of the warning system while being warned about permissions and given the option to disable them.

3.1 System Architecture

The app is composed of multiple modules that asynchronously receives signals broadcasted by a user's device. It does this to monitor different facets of the user's device such as the behaviors and the actions performed by the user on the device. The modules include five broadcast receivers (one, for handling the warning of the user; two, for handling the scanning of the user's device; three, for the monitoring of the power state of the user's device; four, for handling the transfer of data from the local database to

the remote database; five, for reviving the app after a reboot) and two services (one, for initializing the system; two, for monitoring the screen activity and device usage) that are at the core of how this app performs its tasks and analysis.

The interactions between the modules can be seen in 3.1. After installation, the app launches its initialization process, followed by launching all other subsequent modules including the module used to bring the app back to life if it were to be destroyed via the operating system. All data is stored both locally and remotely via an SQLite and Firebase database respectively. This ensures a backup of all data collected during the use of this program.

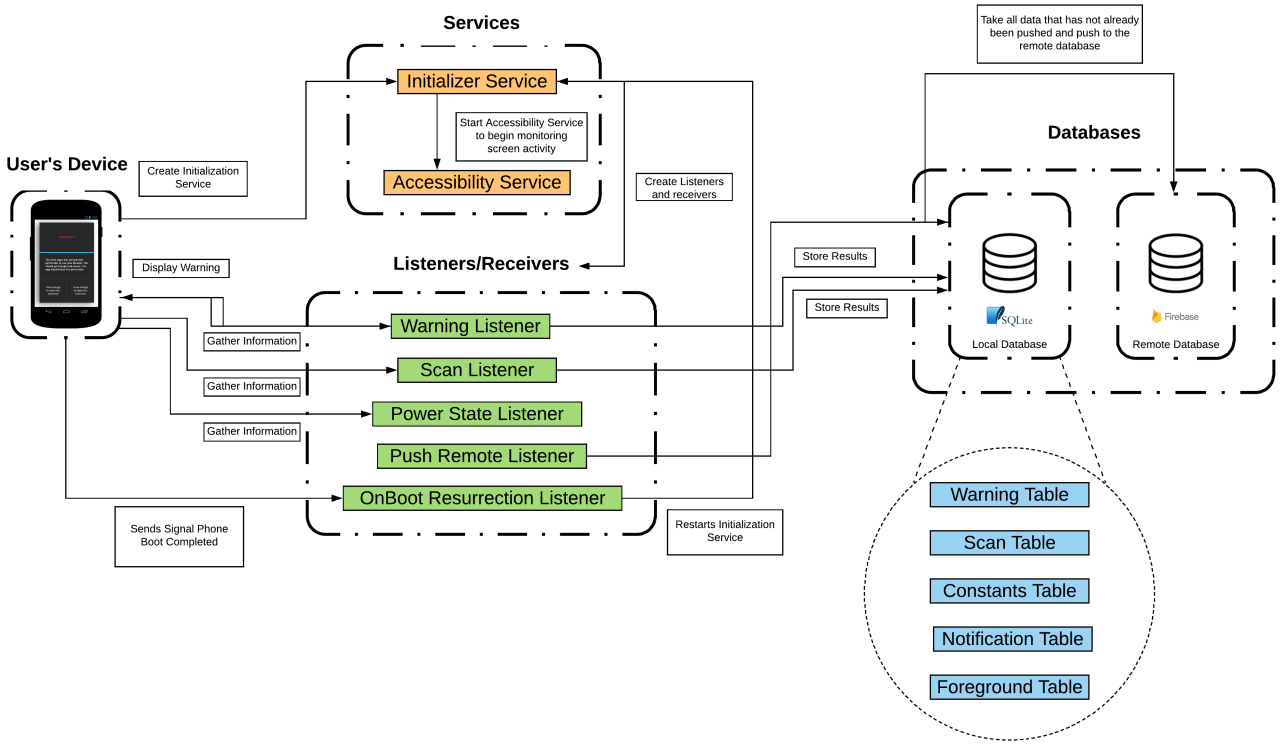


Figure 3.1: System Architecture Diagram. The system consists of three components: 1) The Services, 2) The Listeners/Receivers, and 3) Databases. These three separate components interact with one another in order to warn the user and gather data regarding app permissions

The following sections will review the justifications for and the designs of all of the main components that are demonstrated in the system architecture such as the initialization process of the system, the scanning of a user's device, all steps taken in warning the user (design choices, generation, and display), and the setup of the remote and the local database and the method in which data is transferred from the local to the remote database.

3.2 The Initialization Process

The app begins by launching a service with the sole responsibility of initializing the environment. This service proceeds by grabbing all user constants relating to their device such as the treatment group, the epoch date of the app installation, the SDK number, the model, and the manufacturer. The treatment group is an integer (0-2) that is set by the developer before installation and will assign the user to the appropriate treatment group shown in table 3.1 below.

Number	**Treatment Group**
0	Control
1	Fear
2	Prosocial

Table 3.1: Treatment Group Mappings

Each group is either specialized to control, fear framing, or prosocial framing. The participants in this study were randomly assigned to one of these groups at the beginning of the study. This treatment group variable will only affect the content of the warnings presented to the user. The other facets of the system should function identically across treatment groups.

Following this step, six different modules are then launched consisting of 5 broadcast receivers and an additional service. The receivers all have unique tasks to listen to specific user behaviors and act upon their individual tasks accordingly. The tasks are as follows. One, scanning the user's device periodically for information relevant to the study such as a list of apps on the user's device, a list of permissions granted for each of the apps in the apps list, how many times in total did the user open their settings, all information that was gathered by the power state receiver such as the total number restarts, shutdowns, screen being turned on and screen being turned off. Two, generating and displaying warnings for the user to interact with. Three, monitoring the power state of the user's device and maintaining a count of how many times the user restarts their device, shuts down their device (SHUTDOWN), becomes present on their device (USER_PRESENT), turns the screen on (SCREEN_ON), or turns the screen off (SCREEN_OFF). Four, rebooting the initialization service upon reboot of the user's device (detected by ON_BOOT_COMPLETED signal from the device). Five, periodically checking the local database and collecting all items that have not been pushed to the remote database, staging these items, pushing them to the remote database, and finally marking them as moved. The screen activity service module is responsible for monitoring screen activity of the user. The screen activity service receives signals alerting to changes on the screen of the user's device such as notifications received by the user, apps entering the foreground (used to calculate how long apps were in the foreground of a user's device) and keeps a count as to how many times in total the user enters the settings menu.

This initialization service is also responsible for boosting the persistence of the code. By persistence, I am referring to the ability of this warning system to outlive the activity that had created it. This is required because the life span of a receiver is that of its creator. Therefore, if the creating activity dies all receivers created from

that activity will also die. This is because the life span of the contextually registered receiver will continue to receive broadcasts until the registering context is no longer valid[3]. However, a service can live even after the death of an activity. The context of the service will remain valid as it continues to run in the background indefinitely regardless of whether the starting component has been destroyed[10]. Essentially, this means that even if the user closes the app (destroys the activity) that created the initialization service, the initialization service will continue to run on the user's device and all modules created by that service will continue to run.

3.3 Scanning the User's Device

The user's device will be periodically scanned upon the user becoming present. A user is defined as being present after the device wakes up and the keyguard is gone. Upon becoming present, the user accessing their phone, if it has been at least one minute since the last time in which the user's device has been scanned the user's device will be scanned again. Information from these scans will be stored and used to determine whether the users have been following and/or making changes with the content displayed to them with respect to the warning messages.

Let us first define an app as an object consisting of a name, package type (normal or system level) and a list of permissions. Furthermore, let us define permissions as a list of requested permissions groups that are requested by that app. Each permission group has a name, whether the permission group is granted, and a list of actual requested dangerous permissions in that permission group. An app can be visualized as the following.

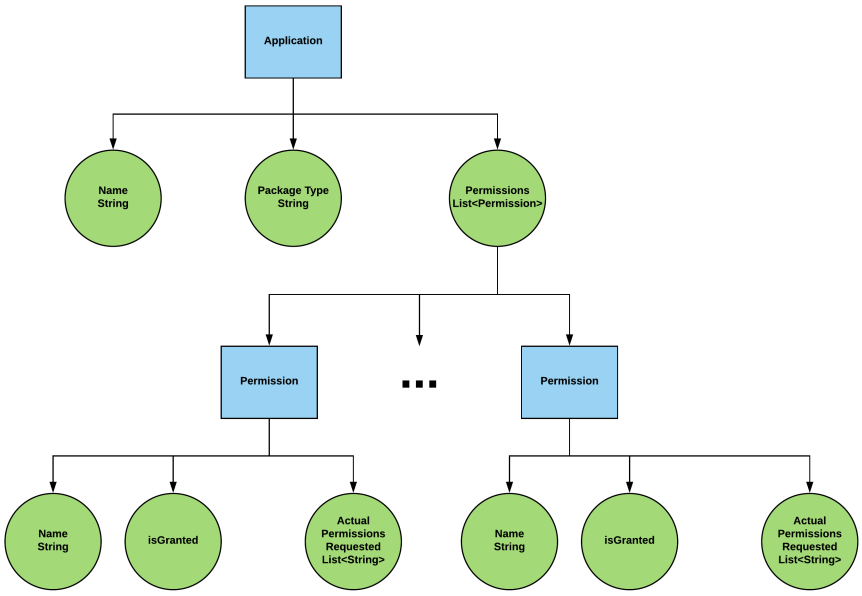


Figure 3.2: Object hierarchy of an Application object. Applications have names, types and permissions. Permissions have a name, a flag representing whether access has been granted and a list of individual permissions that are actually requested that fall under that permission group.

Using this object structure, the main work performed during a scan of the user's device would be populating the fields and attributes. This repeated across all apps on the user's device will give us a description of permissions granted to their applications. Changes to the user's permission settings can be identified by looking at the difference that occurs between each scan.

3.3.1 How Scans are Performed

The flow of operations that take place in order for the scanning of the user's device to be executed is shown in figure 3.3.

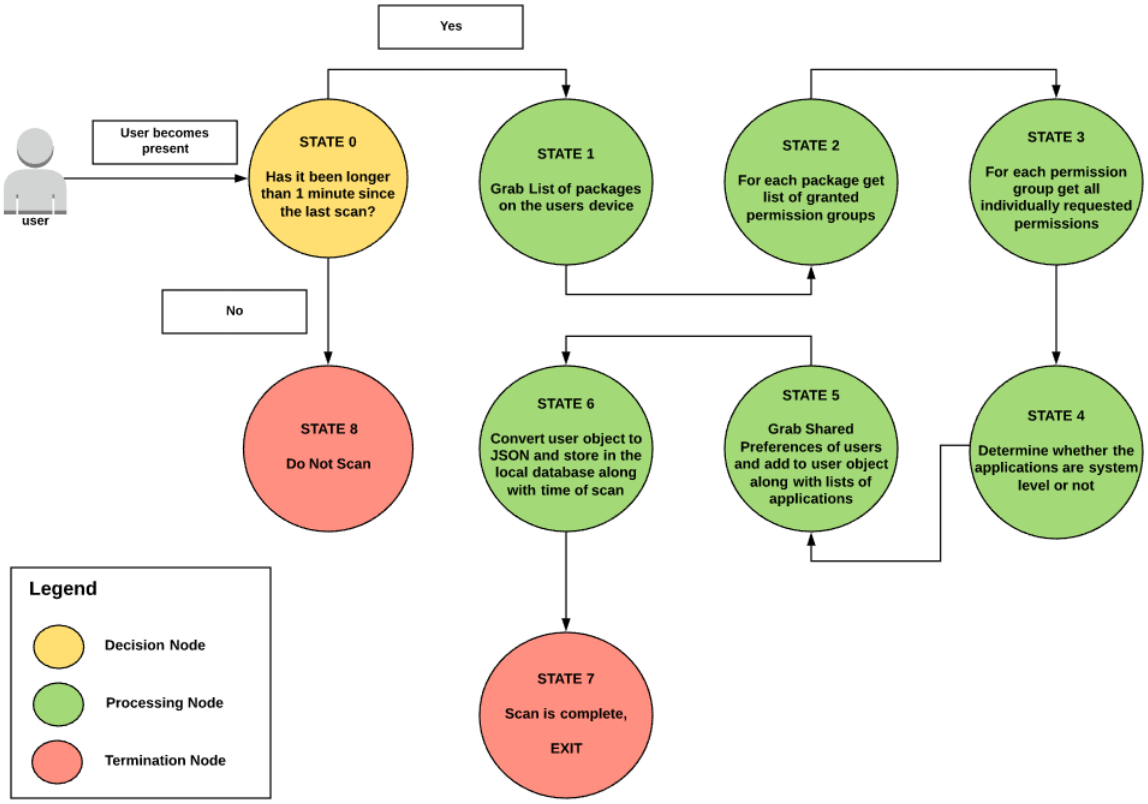


Figure 3.3: Flow chart depicting the steps taking during deciding when to scan and performing the actual scan of a user's device

As shown in STATE 1, the app gets a list of packages on the user's device by using getPackageManager()[5] to return a PackageManger[12] object which is a class for retrieving various kinds of information related to the app packages that are currently installed on the device. From that PackageManger object, we can now create a list of appInfo[2] objects via getInstalledapps()[12]. The getInstalledapps() method returns a list of all app packages (List<appInfo>) that are installed on a device. These appInfo objects contain retrievable information regarding the application in which it is in reference to. Using the GET_META_DATA[12] ComponentInfo[4] flag returns the metaData Bundles that are associated with that component. getInstalledPackages()[12]

returns a PackageInfo[7] List (List<PackageInfo>) of all packages that are installed on the device. A PackageInfo object is in the list, one for all installed packages. Each of these objects contain information about the packages.

Once a list of packages is obtained STATE 2 and STATE 3 are then initiated, the permissions are collected. This begins with the system taking the recently created list of PackageInfo objects and for each PackageInfo object getting a list of names of requested permissions and a list of flags corresponding to those requested permissions. This is either an integer values equal to the value of REQUEST_PERMISSION_GRANTED if the permission is granted or will not be equal to if the permission is not granted. If the list is null that means that this app has requested no permissions. From here one could check which permission groups these permissions belong to. All non dangerous permissions were ignored as these were not adjustable by the user and therefore would not provide useful information in regards to our warnings.

The final step, STATE 4, in grabbing the user's applications is to determine whether the given package is a system level package. This is done by looking at meta data of the app and the ApplicationInfo flags in order to sift out apps that the user would not be able to see or adjust.

In the finalizing state, STATE 5, all counts kept by other receivers and services are then added onto this scan such as total times settings has been accessed, phone has been shutdown, phone had screen turned off, phone had screen turned on, phone has been shutdown, user has become present on their device. All scan data is then packaged up and sent to the local database in which it will be processed into an entry in the SQLite scan table.

Now, as the data has now been passed to the local database in STATE 6 the data is packaged up into a JSON string representation, via the Google GSON API[11], and stored along with the time at which the scan occurred as an entry into the local database

and marked as not pushed to the remote database.

This whole process can be summarized by the following algorithm

Algorithm 1: GrabPermissions – Function used for acquiring the user's apps and permissions on their device

```
   output: A user object U containing data of a user's device
1  Function GrabPermissions ()
2      U ← create blank user object
3      PM ← Get object for retrieving package information
4      P ← get list of installed packages from PM
5
6      foreach package p ∈ P do
7          A ← get application from p
8          𝒫 ← get permission groups from p
9
10         if 𝒫 ≠ ∅ then
11             foreach 𝓹 ∈ 𝒫 do
12                 if 𝓹 is dangerous then
13                     if 𝓹 if granted then
14                         A ← 𝓹 marked as granted
15                     else
16                         A ← 𝓹 marked as not granted
17
18                     A ← add actually requested permissions in 𝓹
19
20         if A not visible to user then
21             A ← mark as not visible
22
23         if A is system level then
24             A ← mark as system level
25         else
26             A ← mark as non system level
27
28         U ← Add application A to user
29
```

3.4 Warning the User

3.4.1 Design of Warnings

Warnings in the context of this warning system are privacy nudges in the form of warnings that have the main role of communicating potential risks to users regarding their security and privacy. Prior research has suggested that warning messages are ineffective[15, 25, 28, 29, 67]. However, the inefficacy of these messages lie primarily in their design and release. Warning messages on smartphone devices often are ineffective because they are frequently ignored and prone to habituation[29, 19, 31]. Other reasons for this lack of effectiveness include obstacles to user comprehension. Similarly, as noted by Bohme[24] and by Wogalter[43], habituation may lead users to disregard warnings.

Thompson and Spencer in their 1966 publication[69] addressed 9 characteristics regarding habituation which are as follows:

1. Given that a particular stimulus elicits a response, repeated applications of the stimulus result in decreased response (habituation). The decrease is usually a negative exponential function of the number of stimulus presentations.

2. If the stimulus is withheld, the response tends to recover over time.

3. If repeated series of habituation training and spontaneous recoveries, the response decrement becomes successively more rapid (this phenomenon might be called potentiation of habituation).

4. Other things being equal, the more rapid the frequency of stimulation, the more rapid and/or more pronounced is habituation.

5. The weaker the stimulus, the more rapid and/or more pronounced is habituation. Strong stimuli may yield no significant habituation.

6. The effects of habituation training may proceed beyond the zero or asymptotic response level.

7. Habituation of response to a given stimulus exhibits stimulus generalization to other stimuli.

8. Presentation of another (usually strong) stimulus results in recovery of the habituated response (dishabituation).

9. Upon repeated application of the dishabituatory stimulus, the amount of dishabituation produced habituates (this phenomenon might be called habituation of dishabituation.

Can we design effective warning messages that not only improve security behaviors but also minimize habituation? The figure below depicts a sketch of how warning messages were constructed. Looking at figure 3.4, we are going to walkthrough all the the individually labeled components and the design designs that went into them.

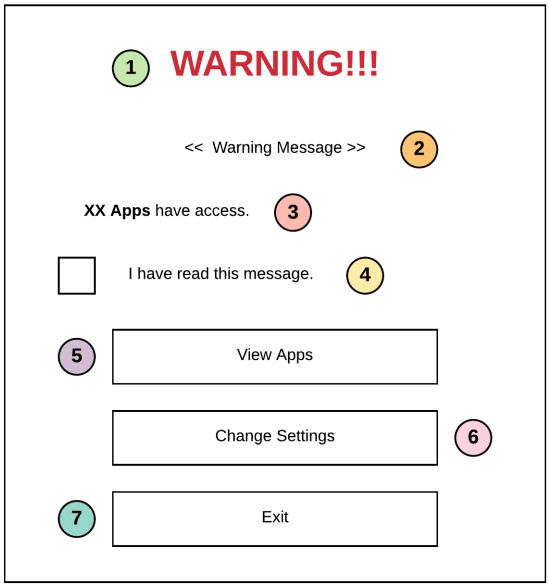


Figure 3.4: Proposed sketch/outline of warning message layout.

Starting with component one of 3.4, the title of the warnings. The use of signal word, "WARNING" and alert symbols such as "!" were used to increase the user's attention and increase perceptions of importance and threat[75]. Furthermore, the use of alert symbols has been found to result in faster response times to the warnings[77]. Also, both size and color of the content in the warnings have been shown to impact the rate at which users become habituated[76]. Color in general has been shown to affect our cognitive systems[42, 32] and specifically, the color red has been shown to be more stimulating[70] and is associated with danger[75, 64]. Another study even found that red warnings improve adherence rates to messages about the consumption of alcoholic beverages[1]. Therefore, red may reduce habituation by increasing users' attention to the message. Taking all of this into consideration, we had chosen to not only choose red for the coloring of this title but also to make it much larger than the body text as our goal is to grab and prevent the user from easily becoming habituated.

The second component of this layout is the content of the warning message. The messages were designed to alleviate habituation by interchanging the content of the messages. The content changes with every presentation of the warning so that the user will never be presented with the same warning more than once. The warning content is never repeated because varying stimuli reduces habituation to the warning.

The "I have read this message" checkbox is included to measure whether the user read the warning. This option is similar to the "I understand the risks" checkbox on Mozilla Firefox SSL warnings that has been shown to reduce click-through rates Akhawe et al[15]. This checkbox measures whether users actually read the warnings; an often overlooked measure of responses to notifications (e.g., [78]).

The third was added to the warning to make the user aware and communicate the number of apps that have access granted to them for the particular permissions groups. These were put into place to give the cognizance of how much their private

information is given away and to whom this information is being delivered to. The design specifically was to present the user with the wording "At least xx app(s) have access to this information. Click the button below to view."

The fifth, sixth, and seventh components are actions able to be performed by the user consisting of neutral (informative), affirmation, and dismission respectively. The neutral button, view apps, will not take the user away from the warning being presented to them. Rather, it will display a list of apps to the user that are requesting that permission group and the user can close and reopen that to their own digression. The button uses the wording "View App(s) Accessing This Information." The fifth and six component were designed to use similar wording that had been used Almuhimedi's paper[18] where the user was allowed to change their settings via the "Let me change my settings" button and dismiss the message via the "Keep sharing my Location" button. As this thesis was not primarily focused around "Location", this statement was generalized to "Keep sharing my private information" as this could then be applicable to all permission groups.

Finally, not all Android devices are the same and vary greatly in screen sizes and pixel densities. The warning should be designed to appear proportionally similar on all devices[14]. As we could not control the screen sizes or pixel densities of our participants devices, we could guarantee the same warning and with the appear similar across the devices of the users. Users receiving warnings of different formatting may lead to different results as this would result in confounding of factors (e.g., different text sizes, text positioned differently, possibility of not fitting or appearing too small, etc.) that could be giving the acquired outcome. In regards to the typography and the layout, Android provides both scale-independent pixels (sp) and density-independent pixels (dp) respectively[13]. Taking a look at figure 3.9, as noted before size matters when getting the users attention the warning title was chosen to be roughly double that of

main content of the warning message. All sizes of the components can be seen labelled in the figure.

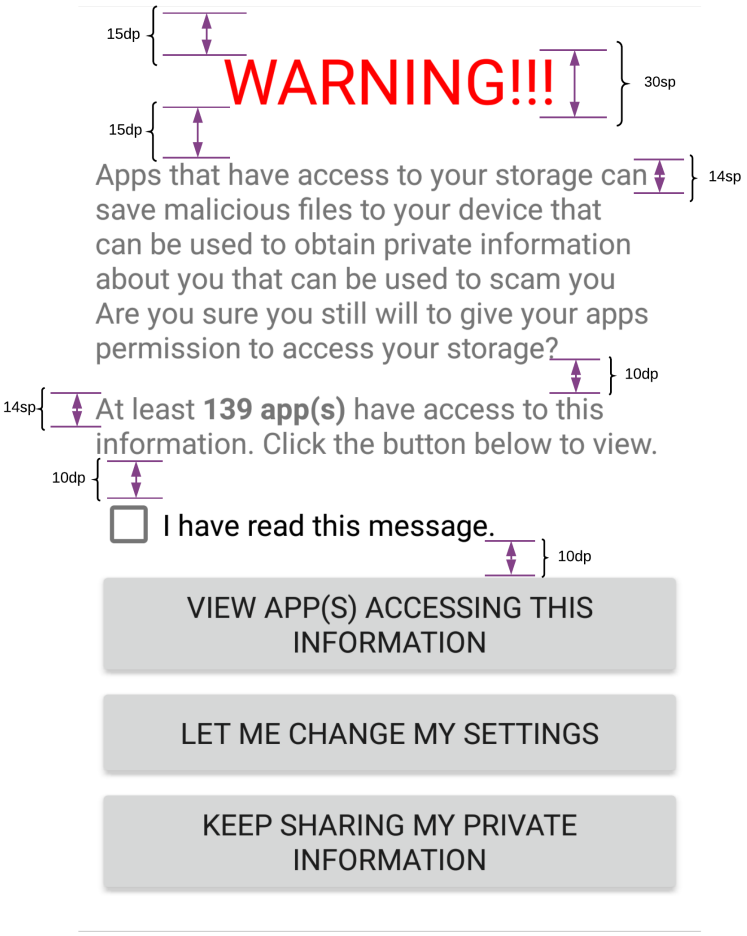


Figure 3.5: Dimensions of a Warning Message

3.4.2 Deciding When to Warn

Deciding when to warn the user is an important element of the design. Given the second and fourth characteristic of Thompson and Spencer (1996) that states that not only does the recovery (dishabituary) process of the user occur during the absence of a stimulus but also that the habituation that is incurred by a user happens when that user is presented with a high frequency of notices or warnings. These two characteristics

show the usefulness of adjusting the frequency and thereby adjusting the down time, defined as the time in which the user is not being warned, one could speculate that this would cause positive effects in the user's performance all while keeping them from becoming habituated to the warnings that are being presented by the system.

Studies have shown that increasing the frequency of a repeated stimulus will lead to habituation[20], also referred to as warning fatigue[15]. The system was chosen to display a warning with a least a 3-hour separation between each display of the message. This 3-hour period reduces the chance that habituation occurs because messages will not appear in rapid succession. Additionally, previous work on warning messages that has found habituation effects released the warnings on a more rapid release schedule, with less than 3 hours in between the message presentations. The system will run a check when a user is determined to be present on their device. If the user is present, a check is made to determine if the user is between the appropriate warning times. If they are, the system performs a check as to whether it has been at least 3 hours since the last time the user was warned. If it has been longer than 3 hours, the user receives a warning. This decision flow is shown below.

Now that it has been decided how often a user is warned, one other factor remains. That is, when is it appropriate to warn a user? The timing of the warnings should be the interval of time in which users are typically awake. Studies[55, 74] have shown the typical sleeping patterns of adults. Based on these statistics, the 12 hour interval between 11a.m. and 11p.m. was chosen as the most appropriate period to present warning messages to the user as they will not be immediately waking up or going to bed. Our consideration was that a user either really in the morning or late at night would not respond as they normally would and this could skew how the users react to the warnings.

One final consideration on the topic of choosing the most appropriate time in which

to warn was that of not warning the users if it is their first time turning on their phone during the prior stated interval in which the warnings are given. This was decided to be the least interruptive method of presenting warnings to the user as they would not be forced to look at the warning the first chance they get to use their phone that day.

3.4.3 Generating Warnings for the User

Generating the warnings for the users is one of the most crucial elements of design. Following the first characteristic of Thompson and Spencer (1966) the warnings were designed as polymorphic as possible as this has been shown to impede habituation[20]. Polymorphic is referring to the ability of the messages to take on many forms by changing the content of the messages from warning to warning. The user is not presented with a warning of the same type that they were just presented with and when they are presented again with the same warning message type the content will be different than they got before.

Warnings have to attributes to them: type of warning and type of message. The type of the warning will be that of one of the dangerous permission groups on Android. If the warning is about location, then the warning type is location. As for the type of message, each warning type will have multiple types of messages that all focus around that warning type but with different approach. For example, one might get a warning dealing with their location and a message focused on how their location can be used for blackmailing purposes and the next time they receive a warning regarding their location they will receive a warning focused on how their location can be used to track them.

Throughout the study, the user will never be presented with the same warning message twice, only the same type. As mentioned the warning types refer to the dangerous permission groups that the warning message corresponds to. The user will never be warned about the same permission group in a row and messages will never repeat.

Given that there are nine dangerous permission groups that the user has the ability to manipulate through their settings as seen in figure 1.5. The app assigns a random order to the user, this order will correspond to the order in which they receive warnings about each permission group and they will be received in a cyclic fashion. Each dangerous permission group has two warnings that highlight the loss of security information, one that highlights a scam, one about impersonation, one about blackmail, and one about tracking, for a total of six warnings for each permission group. The content was consistent across permission groups to ensure that the content was not more impactful in one permission group compared to another. Additionally, the content was consistent across the fear and prosocial conditions such that the users still received warnings that highlighted the same subjects but were just framed differently.

As can be seen from the example in figure 3.6, in which the user was assigned an order where they would receive warnings relating to SMS, Camera, Location, Calendar, Sensors, Contacts, Phone, Storage, and Microphone in that order. And, each permission group has their own order of highlighted warnings as can be seen for the Microphone permission group in this example. Essentially, the user will get the first warning for each of the 9 permission groups, then the second warning for each of the permission groups and so on until all warning lists have been exhausted. Given that there is 9 dangerous permission groups and 6 highlighted warning per group, there will be a total of 54 warnings received by the users throughout the duration of this study.

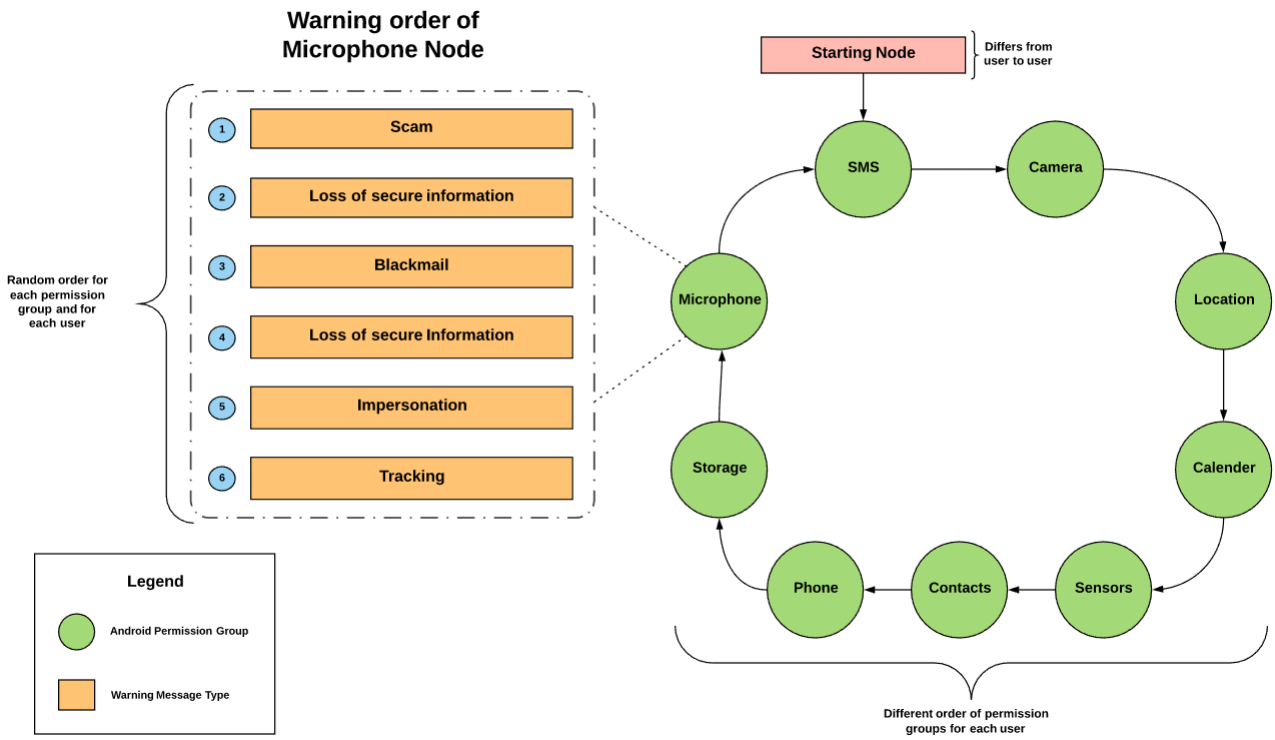


Figure 3.6: Example of the warning order randomly assigned to a user. Every user will have a different order for permission groups and for each permission group they will receive a different warning order.

There are a number of conditions that must be satisfied to determine that it is the appropriate time to present the user with a warning. To summarize all of the prior mentioned conditions, the following flowchart illustrating the process taken to determine if it is appropriate to generate a warning to the user and the process of generating the warning can be seen at figure 3.7. This process of determining whether a warning should be generated is taken everytime the user becomes present on their device.

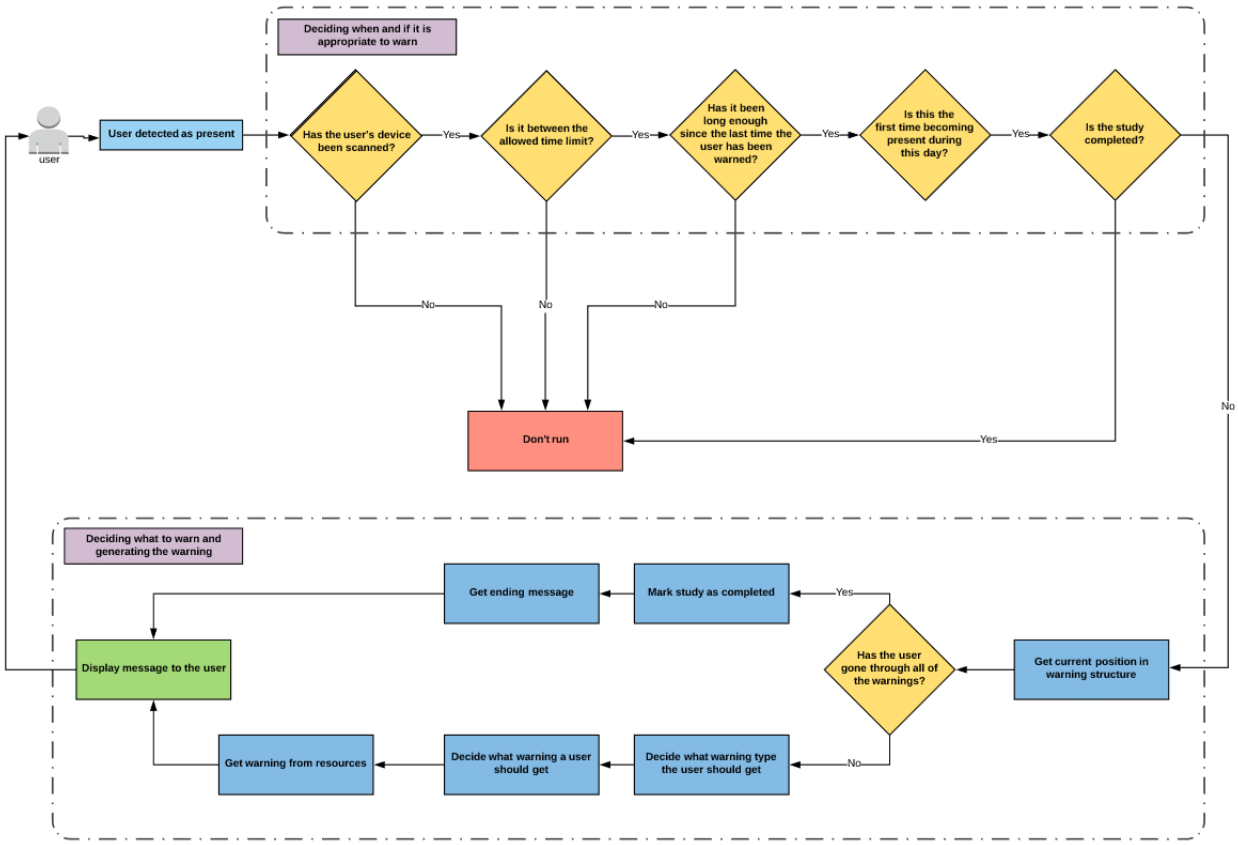


Figure 3.7: Warning Generation Process Flow

There are essentially five conditions that must be satisfied before a warning can be generated and then to be displayed to the user. One key step is the check whether the user's phone has already been scanned. If the user's phone has never been scanned yet since the installation of the app, a warning can't be sent to the user. This is due to the fact that the warnings utilize the information that is gathered during a scan in the generation of the warnings. Information such as how many apps are granted access to the relevant permission group that is currently being warned about and names of relevant apps needed for displaying in the view list. Note, this is only relevant at the beginning of the study as the user will be scanned almost immediately after the app is finished being installed on their devices. If the user has been scanned, the next

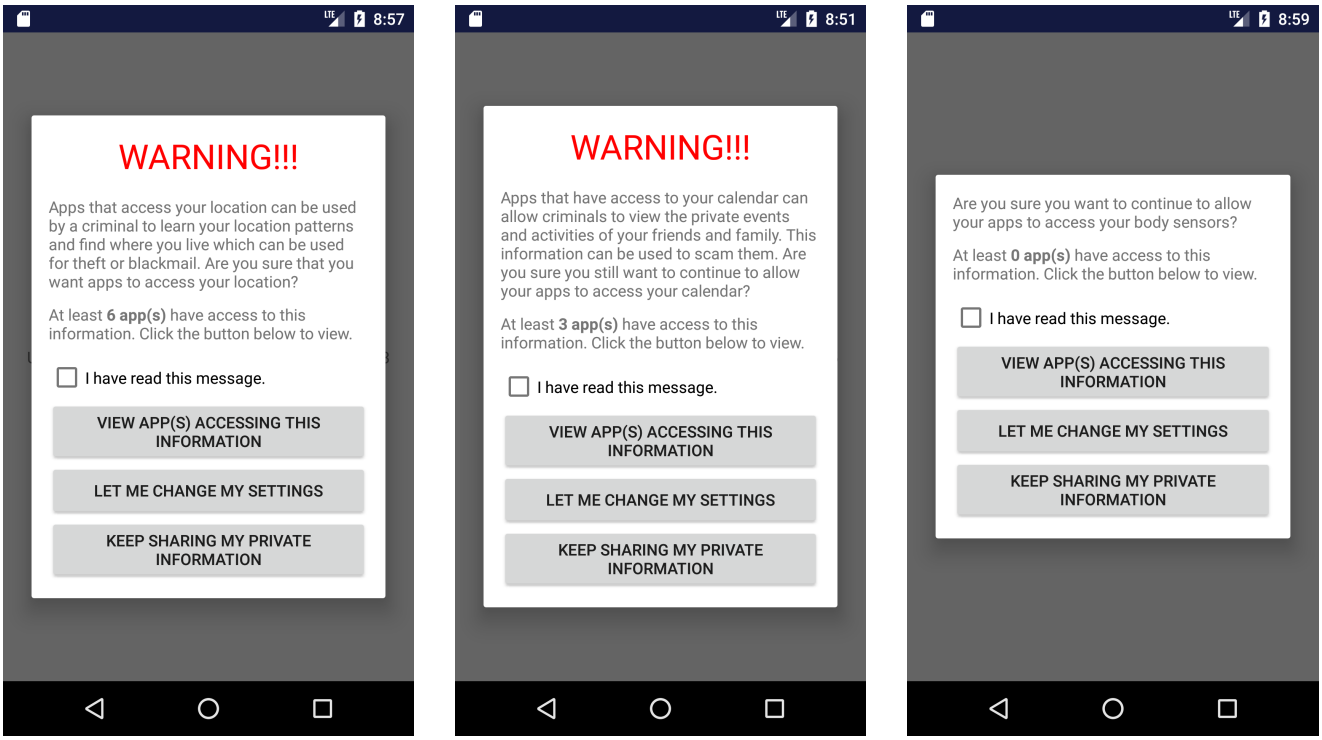
condition that must be satisfied is that it has to perform a check to determine whether the time of day is between the allowed time limit of 11a.m. and 11p.m. If it is between the appropriate time of day, it then has to check if it has been long enough since the last time the user has been warned. That is, has it been at least three hours since the last time in which the user received a warning? Then the system will check if it is the first time in which the user has become present on their device that day. If it is the first time that the user had got to this point today, the user will be marked as present today and on their next time getting to this condition they will pass this check. Finally, the system runs a check to determine if the study has been marked as completed, if the study has not been marked as complete the system can continue.

Once those five conditions mentioned are satisfied, the warning system goes to get the next warning type that is in the data structure presented in 3.6. If there are no more warnings left, this would mean the user has gone through all of the available warnings and is now done with the study. The user is then marked as completed and an ending message telling the user they have completed the study is then sent to be displayed. If the user has not exceeded all of the warnings in the structure, then it is decided which is the next permission group to warn about and what is the warning message that should be displayed from that permission group. The system will then get that warning message from the resources on their device and then send that message to be displayed to the user.

3.4.4 Displaying Warnings to the User

When a warning is generated by the warning receiver it is then passed to the main activity whose responsibility is then to force an alert dialog into the user's foreground. This approach was taken as it is the most natural way in which we could present the warning to the user that is similar to the way in which the Android operating system

presents its permission warnings. An example of these warnings that are presented to the user can be seen in figure 3.8. From the second that the warning is visible in the user's foreground a timer will record, in milliseconds, how long the warning is open before the user makes a decision.



(a) Fear Warning Message (b) Prosocial Warning Message (c) Control Warning Message

Figure 3.8: Example Warning

Once the warning is displayed to the user, the user will be able to make their choice as to how they would like to proceed. The following flowchart in figure 3.9 depicts the possible decisions and the consequences from making those decisions. One such option the user can view the apps on their device. This view menu is shown in figure 3.10. The user can access this view menu multiple times while the warning message is displayed to them and a count of how many times they accessed this view menu is maintained.

If the user chooses the option to keep sharing their private information the app will close and the results of this warning will be stored in the local database. Finally, if the user decides to choose the change my settings option they will then be directed to an onboarding tutorial as shown in figure 3.11 and then redirected to their settings menu.

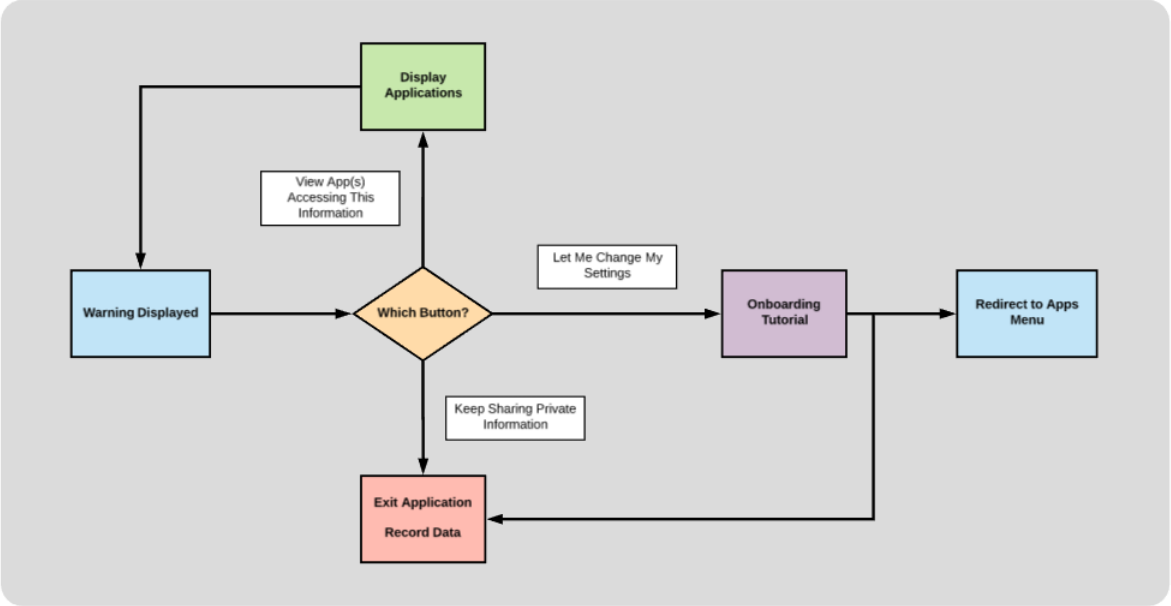


Figure 3.9: Warning Flow chart

The 'View Apps' button when pressed by the user will trigger a builder to construct an additional alert dialog that will be displayed on top of the warning alert dialog. The content of this 'View Apps' dialog is populated by the names of the apps that were gathered by the scan of the user's device. This menu can either be escaped/closed by the user by simply clicking off of the apps list or pressing the back-press button on their device. Once the 'View Apps' dialog has been closed the warning alert dialog will still be open to the user and the user can choose to review this apps list again if they deem so as necessary. I should note that at this point we are now seeing why we need to scan the user's device at least once before they can receive these warnings. It is necessary in order to display such a list to the user.

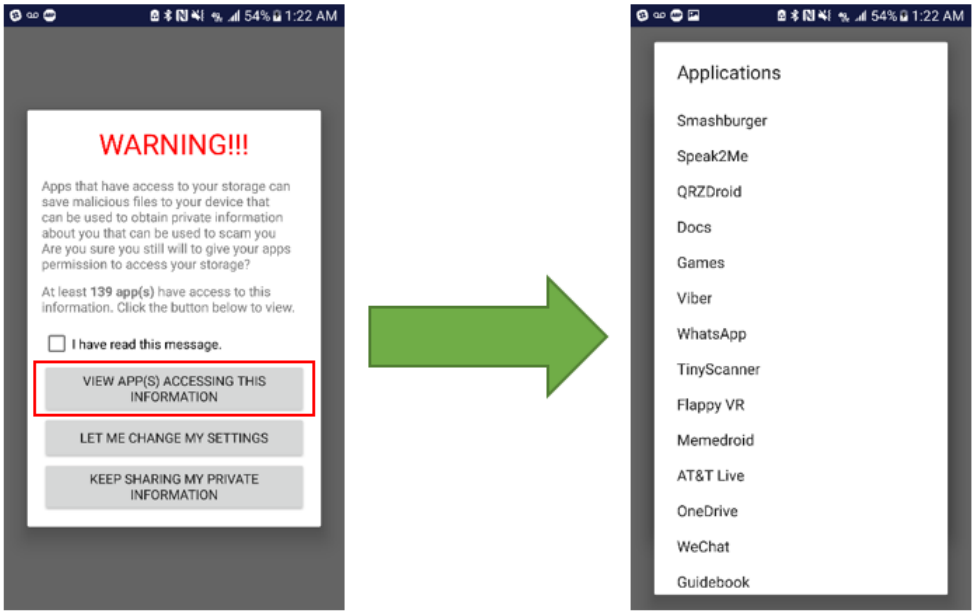


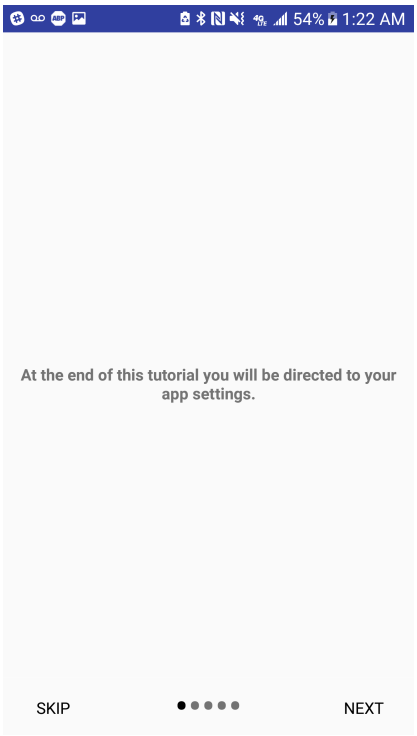
Figure 3.10: User presses “View Apps” button. When pressed, a closable window displaying the apps granted this permission will appear. This button can be pressed multiple times during a single warning. The number of times this button is pressed is recorded.

When the user selects the ’Let Me Change My Settings’ button the warning dialog will then be closed an instance of the Warning Instructions Activity will be created. This activity uses Android ViewPager layout manager in order to present the user with the tutorial that is depicted in 3.11. On this tutorial the user will be presented with five slides that will walk them through the basic of adjusting their permissions on their device. The screen shots that are used in the onboarding tutorial are taken from an Android 7.0 Google Nexus device.

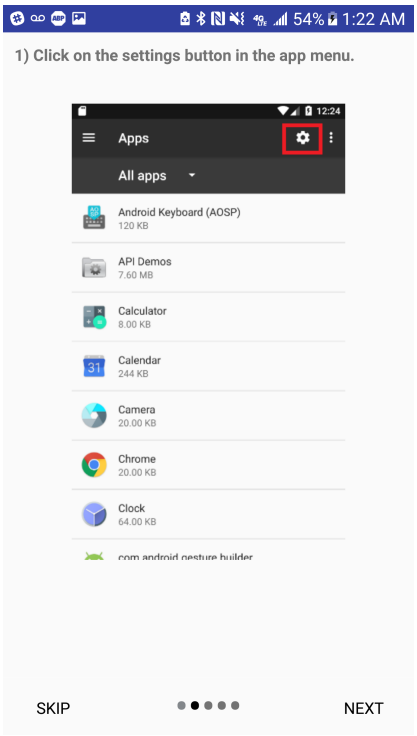
On each of the slides, at the bottom, the user will see two options: 1) Skip the tutorial to last slide via a ‘SKIP’ button, 2) Proceed to the next slide via a ‘NEXT’ button (unless you happen to be on the last slide of the tutorial, this next button will be a ‘GOT IT’ button and will bring the user to their permission settings). The user

will also see their current position in the tutorial via the dot indicators. The 'SKIP' button serves the purpose of allowing the users who have already completed the tutorial in the past and no longer require the information contained within the option to skip the tutorial and not have to go through the entirety everytime. This was added to help mitigate user fatigue as users will be less motivated to change their permission if everytime they went to change them via one of our provided warning messages they would be forced to go through each slide one by one.

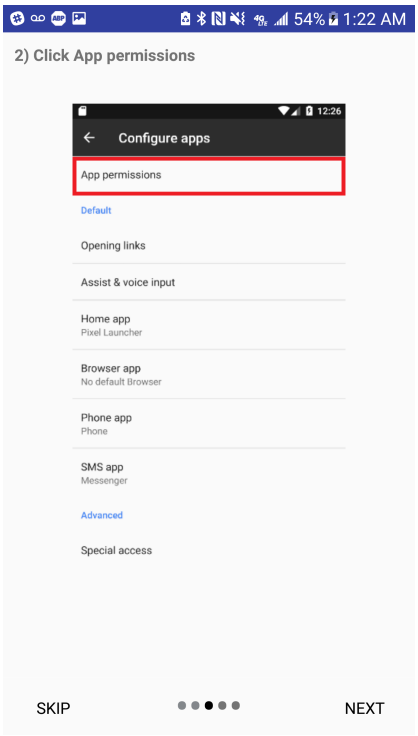
As mentioned, at the end of the onboarding tutorial the user will have a new button presented ('GOT IT') to them that will bring them to their Apps settings menu. This menu was chosen due to the fact that it is the closet that a user can be brought into their settings via an app on a non-rooted phone. The purpose of this tutorial is to walk the user through the required steps that they will have to perform after they have been redirected in order to make any changes to their app permission settings.



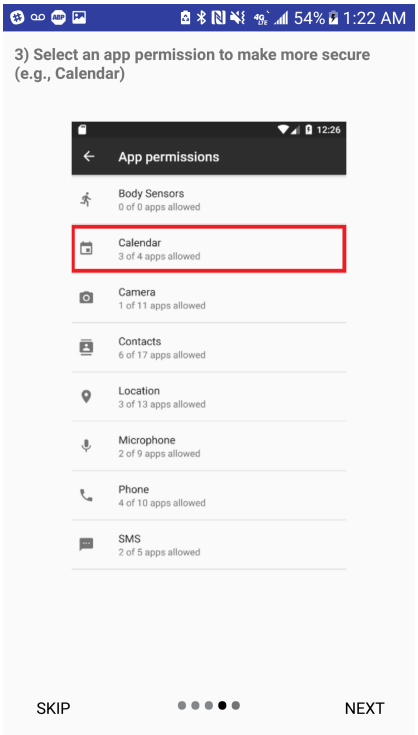
(a) Tutorial Slide 1



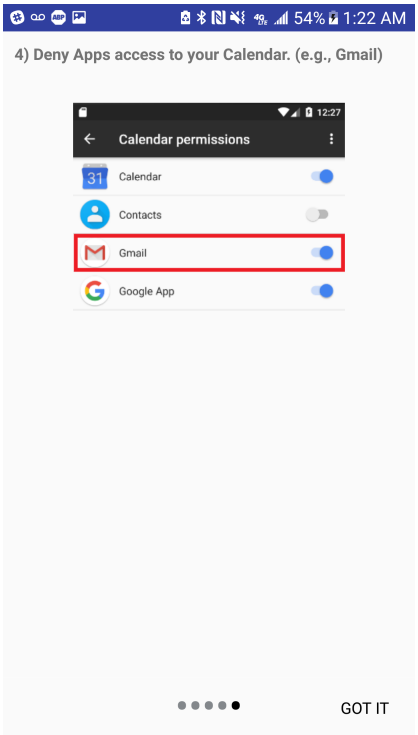
(b) Tutorial Slide 2



(c) Tutorial Slide 3



(d) Tutorial Slide 4



(e) Tutorial Slide 5

Figure 3.11: Onboarding User Tutorial

3.5 Handling User Data

3.5.1 Database Design/Diagrams

The local database has five tables: a warning table, a scan table, a constant table, a notification table, and a foreground apps table. Each table can be visualized as having the following columns

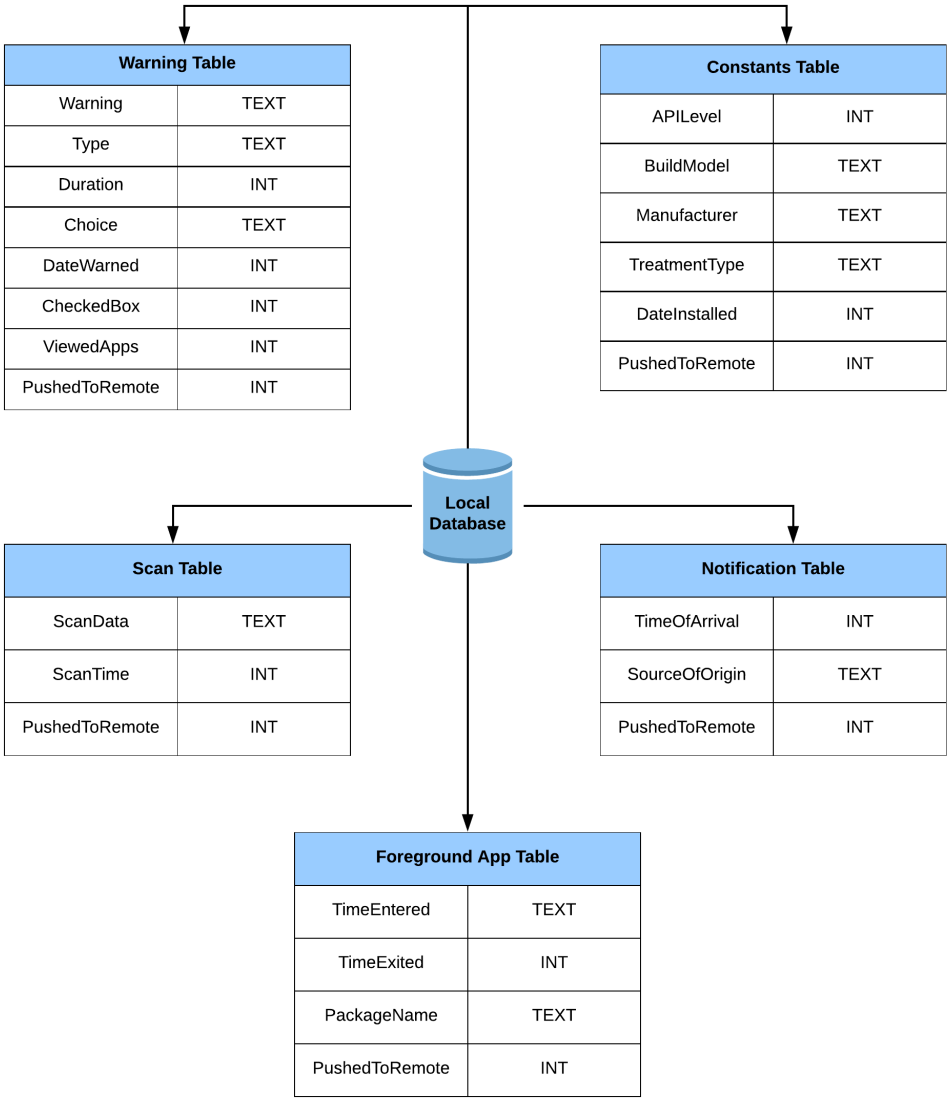


Figure 3.12: Local Database Table Diagram

3.5.2 Local & Remote Database Implementation

Data is stored both locally on the Android device via an SQLite Database and is periodically pushed to the remote Firebase database. This helps mitigate the frequent network requests [6] which entail not only wake up costs but also drained power by keeping the connection alive for 20-60 seconds waiting. This decision was not only made due to concerns for their devices battery life from draining and causing a skew in how they participate within the study but also how congested the remote database had gotten when multiple devices had a constant hold/stream.

If there was only the remote database given that the warning system performs a scan of the user's device every minute then it would have to push to the remote database every minute. As mentioned this connection can be held for up to a minute in search for additional connection and can face the potential case in which it just stay connected and drains the battery of the users device. Battery is a big concern since we did not want the users device to go into power saving mode and kill off consuming processes which could affect how the app performs on the user's device. Also, if the user's device frequently dies it could affect how the user will receive warnings.

The methodology used to store into the local database and then pushed to remote database is as follows. All entries in the local database have an extra entry in their row (PushedToRemote) and are originally marked as not pushed to the remote database by having the value 0. When the time is determined to be appropriate, the push remote receiver will query all un-pushed entries (all entries from each table that are value 0) and push them one at a time to the remote database and as each entry is pushed to the remote database the entry is then updated in the local database as being pushed (setting PushedToRemote as the value 1) so they do not appear in during the next time in which the local database is queried.

Chapter 4

Experimental Evaluation

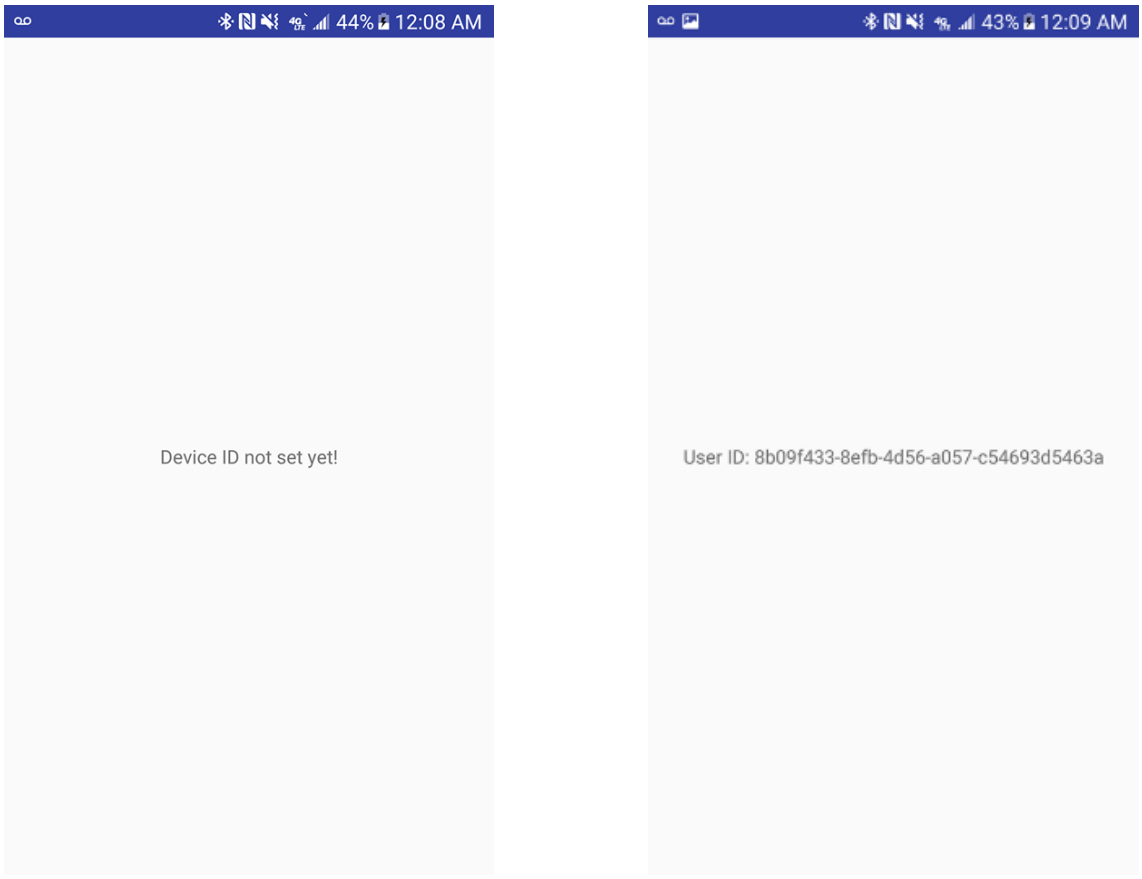
4.1 Design & Procedure

Tying together all design decisions discussed in the previous chapter. The app was designed with all the constraints and set guidelines from prior chapters. The app when on a user's device should run seamlessly without intervention from the user except for when the user is required to interact with one of the warning messages. All gathering, and management of data is performed without the knowledge of the user.

As mentioned prior, a similar study had been conducted using a survey based medium in which we gauged the users as to how they felt in certain scenarios. This continuation was to see if such findings also hold up in a field study that occurs on the user's personal device and during the user's everyday life across an extended period of time. This app was an apparatus used to perform the field study.

The participants in this field study were divided into three different treatment groups: fear, prosocial, and control. The treatment groups that the participants are assigned to decide what type of framing the user will receive for warnings. The fear and prosocial groups will receive framed warning message with fearful language and altruistic language respectively. The control group presents messages that resemble current Android permission messages. The only difference between the treatment groups is the warning messages received.

The participants are randomly assigned to one of the three defined treatment groups and then the specified app is then installed and set up on their device the same exact way for each group. We anonymized the participants by assigning a random UUID (unique user ID) to each participant. This ID is known by the participants and recorded in the database. This ID can be accessed for the participant's convenience on the app as shown below.



(a) Before ID was assigned

(b) After ID was assigned

Figure 4.1: The Device ID. The device ID is generated by the application at the beginning of the study when the system performs the initial scan of the user's device. The device ID is used to relate participants to the data that is being stored in order to assign credit upon completion. The device ID is always available and displayed to the user when they open the application.

Over the span of 2.5-3 weeks, the users will be prompted with the warning messages. Messages will be distributed to the user when they become present on their device AND certain criteria is met. By this I mean that when a user becomes present on their device certain checks will be made and if all checks pass the user will be presented with a warning message.

- The user cannot be warned until their phone is scanned at least once (usually scanned within few minutes of installing the app). This is because the warning needs information from the scan of a user's device to generate certain fields of the warning such as the functionality of the view apps button or how many apps relate to whatever permission they are being warned about

- Is this the first time the user turned on their device today? If so, do not warn them. This condition was added in to alleviate the inerrability aggravation on the user and make the study flow nicer and not annoy the participants.

- Is the time between the hours of 11am-11pm? If not, do not warn them. We chose this specific interval after going over sleeping patterns [2] of university students and fitting the study to a time slot when most participants would be awake and not just waking up or just going to sleep.

- Have you been warned in the past 3 hours? If so, do not warn. The reason for this guideline is we wanted to limit the exposure to the warning as not to dampen the response. So the warnings were spaced out such that at least three hours from the last viewing of a warning message would have to pass before you would be able to get the next warning message.

The app installed on each of the user's devices are the same the only thing that results in the different behavior is an internal variable that is set, and they are compiled to point to a specific database that is meant for each treatment group. Three versions of the APK were generated and stored on a Google Drive and accessible to download for installation via a bit.ly link.

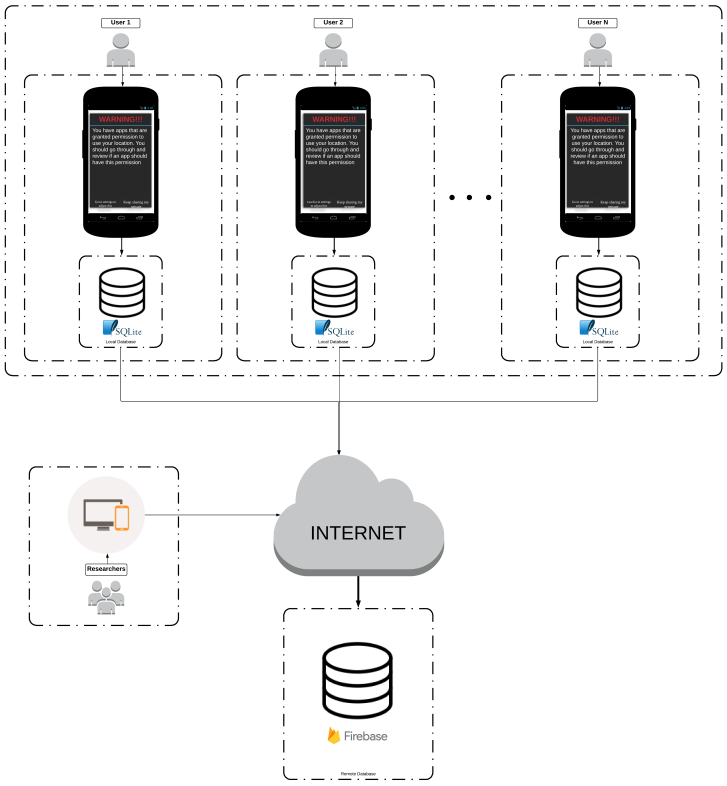


Figure 4.2: Study Architecture Diagram. The study is designed so all participants perform on their devices locally as data is received on remote database and visible to the researchers.

4.2 Results

Each participant of the treatment groups on average interacted with a majority of the warnings that were presented to them. A warning was counted as interacted with if the user chose one of the available options and completed their interaction with that warning. The control ($Mdn = 42$, $SD = 6.56$), fear ($Mdn = 45$, $SD = 4.58$), and prosocial ($Mdn = 47$, $SD = 18.9$) group responded on average to 41.9, 46.6, and 37.6 warnings out of the total 54 warnings presented to them through out the course of this study respectively. The control ($Mdn = 2.61$, $SD = 0.66$), fear ($Mdn = 3.00$, $SD = 0.37$), and prosocial ($Mdn = 3.00$, $SD = 0.82$) groups received an average of 2.80, 3.06, and 2.95 warnings per day respectively.

4.2.1 Warnings of Particular Permission Group Ability to Motivate Users to Consider Changing their Settings

Given all of the treatment groups, if one were to look at the total count in which users chose the option 'Let Me Change My Settings' and then further separate that count by which permission group that warning belonged to. One could then speculate as to which permission group was most motivating in making the users consider changing their permission settings for the treatment groups. As can be seen from the following graph 4.3, warnings related to phone ($n = 17$) permission group seemed to be the most motivating in getting the user to consider making changes to their permission settings. The next three most motivating were warnings relating to the microphone ($n = 14$), contacts ($n = 13$), and sms ($n = 13$) permission groups.

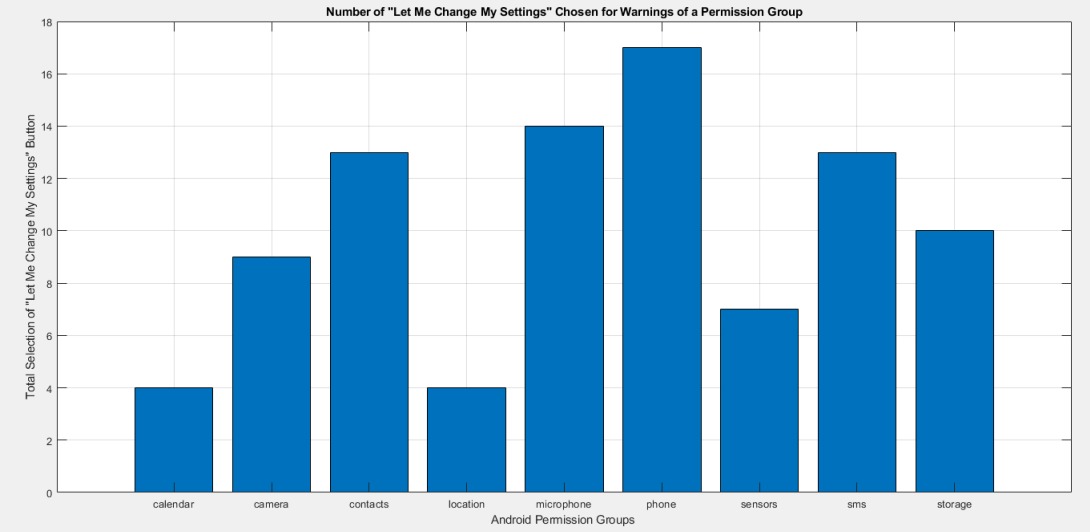


Figure 4.3: Number of Times Users Selected 'Let Me Change My Settings' across all Treatment Groups. This graph shows the total times users in all of the treatment groups selected 'Let Me Change My Settings' for the nine permission groups

Breaking this result down further, the phone permission seemed to be the most motivating permission for the Control ($n = 6$) and Prosocial ($n = 8$) treatment groups. Whereas the sms ($n = 5$) and storage ($n = 5$) permission groups were the most effective for the Fear treatment group. The least effective warnings for all of the treatment groups were those pertaining to the calendar ($n = 4$) and the location ($n = 4$) treatment groups.

4.2.2 Average duration of time in which users viewed warnings

Observing the durations in which users spent viewing and interacting with the warning messages from the time in which they were presented varied across the treatment groups. In specific, the prosocial and control group had warnings that held the users' attention almost twice as long as the warning that were presented to participants in the fear treatment group. On average users in the prosocial ($Mdn = 0.23$ minutes, $SD = 2.62$ minutes) and control ($Mdn = 0.33$ minutes, $SD = 3.14$ minutes) treatment groups

interacted with the warnings for 1.67 and 1.58 minutes respectively whereas the fear group ($Mdn = 0.32$ minutes, $SD = 0.89$ minutes) had an average viewing/interacting time of 0.79 minutes.

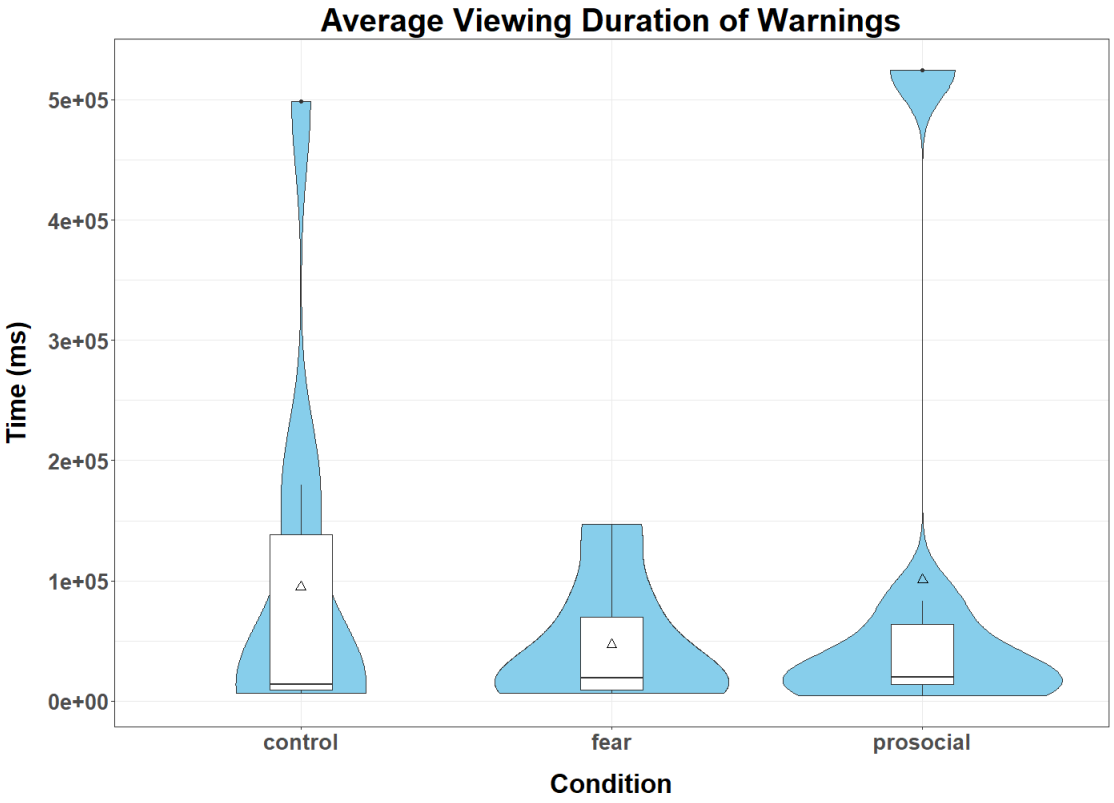


Figure 4.4: Average viewing time by condition. The prosocial and the control conditions had higher average viewing time than the fear condition. The control condition resulted in a much wider spread of times for users.

4.2.3 Keep Sharing vs Let Me Change My Permissions

Further analysis on how users interacted with the warnings in regards to the choices made, one can see that out of all three treatment groups the prosocial treatment group yielded the highest rate of warnings that were effective in motivating the user to make changes to their settings.

The prosocial treatment group ($Mdn = 11.11$, $SD = 10.38$) yielded the highest average of 14.69% for all warnings having users select *'Let Me Change My Settings'*. This ranged from 2.17% to 95.45% of user choosing this option.

The control treatment group ($Mdn = 4.41$, $SD = 5.62$) yielded the lowest average of 6.16% for all warnings having users select *'Let Me Change My Settings'*. This ranged from 0% to 58.70% of user choosing this option.

The fear treatment group ($Mdn = 4.65$, $SD = 7.22$) had an average 7.13% of all warnings having users select *'Let Me Change My Settings'*. This ranged from 0% to 67.31% of user choosing this option.

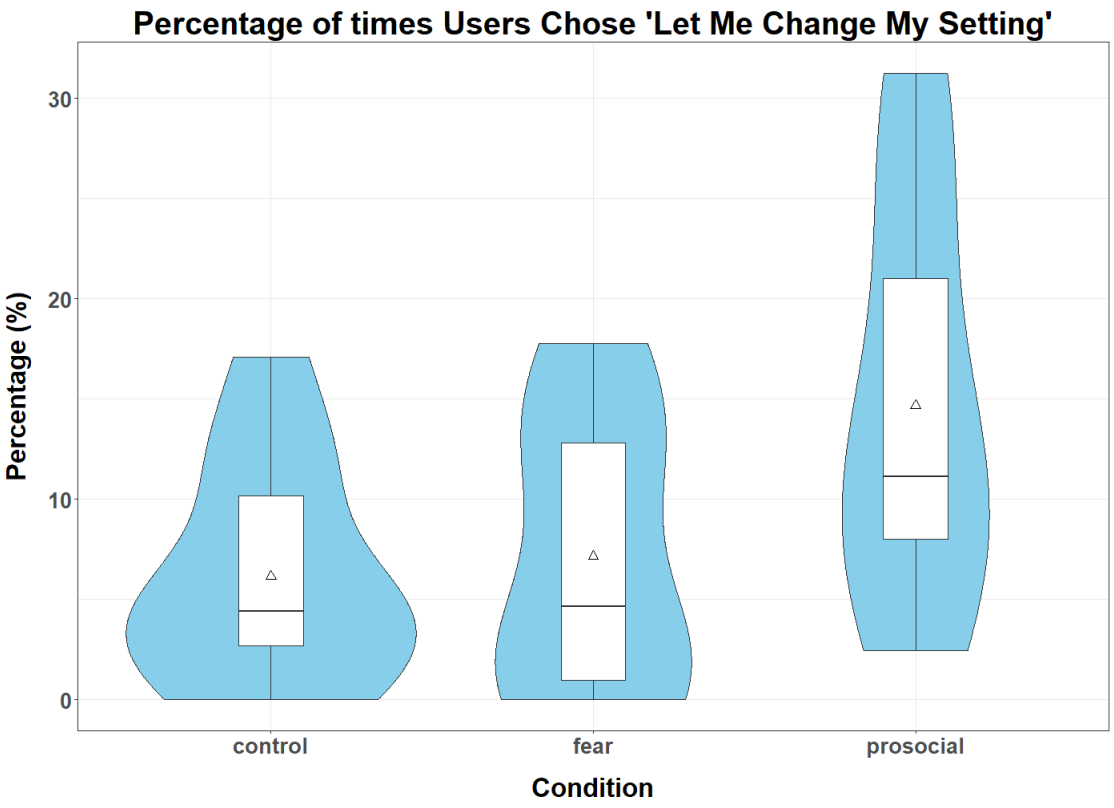


Figure 4.5: Percentage of times users chose *'Let Me Change My Settings'*. As can be seen from the plot prosocial yielded the highest percentages whereas the fear and control group are very similar with fear being slightly higher percentage.

A One-way ANOVA was performed on the percentages in which users chose 'Let Me Change My Settings'. There was a not a significant difference between conditions for users selecting the *'Let Me Change My Settings'* option at the $p > 0.05$ level for the three conditions $[F(2, 38) = 1.25,\ p = 0.30]$ with standard deviations of 8.2015%, 4.8889%, and 8.4043% for the control, fear, and prosocial treatment group respectively.

	Treatments			
	Control	Fear	Prosocial	Total
N	17	12	12	41
ΣX	76	47	100	223
Mean	4.4706	3.9167	8.3333	5.439
ΣX^2	1416	447	1686	3549
σ	8.2015	4.8889	8.4043	7.6421
Source	SS	df	MS	F
Between-Treatments	144.2789	2	72.1395	1.2507
Within-Treatments	2191.8186	38	57.6794	
Total	2336.0976	40		
$p = 0.297824$				

Table 4.1: One-Way ANOVA on the percentages in which users chose 'Let Me Change My Settings'

4.2.4 Average Number of times users selected *'View My Apps'* across treatment groups

The prosocial treatment group had an average of users viewing their applications via the *'View My Apps'* button 42% of the time. The control and fear treatment group were roughly the same with 24% and 20% users would view their applications. The prosocial and the control treatment group were the only treatment groups in which users utilized this button more than once during warnings.

A One-way ANOVA was performed on the average number of times in which users selected the option to view their apps requesting and granted the currently warned

about permission. There was a not a significant difference between conditions for users selecting the *'View My Apps'* option at the $p > 0.05$ level for the three conditions [$F(2, 38) = 2.13$, $p = 0.13$] with standard deviations of 0.2846, 0.1472, and 0.4127 for the control, fear, and the prosocial treatment groups respectively.

	Treatments			
	Control	Fear	Prosocial	Total
N	17	12	12	41
ΣX	3.8326	2.5191	5.1693	11.521
Mean	0.2254	0.2099	0.4308	0.281
ΣX^2	2.1603	0.7673	4.1004	7.0281
σ	0.2846	0.1472	0.4127	0.3078
Source	SS	df	MS	F
Between-Treatments	0.3823	2	0.1911	2.13086
Within-Treatments	3.4084	38	0.0897	
Total	3.7907	40		
$p = 0.132707$				

Table 4.2: One-Way ANOVA on the average number of times in which users selected the option to view their apps requesting and granted the currently warned about permission

4.2.5 Percentage of Users who Checked the 'I have read this message' box

The non-mandatory 'I have read this message' check box was checked by more users in the fear treatment group, followed directly by control and the prosocial group. The volume of the fear ($Mdn = 97.9\%$, $SD = 12\%$) participants ranged from 67% to 100%

of the time regarding users who checked the box with an average of 93.7%. Whereas for the control ($Mdn = 95.6\%$, $SD = 25.9\%$) and the prosocial ($Mdn = 88.9\%$, $SD = 35.6\%$) treatment groups the ranges were 14% to 98% and 0% to 100% respectively with averages of 87.8% and 76.4% respectively. From the figure one can tell the the volume of the participants were more spread around in how they selected this option.

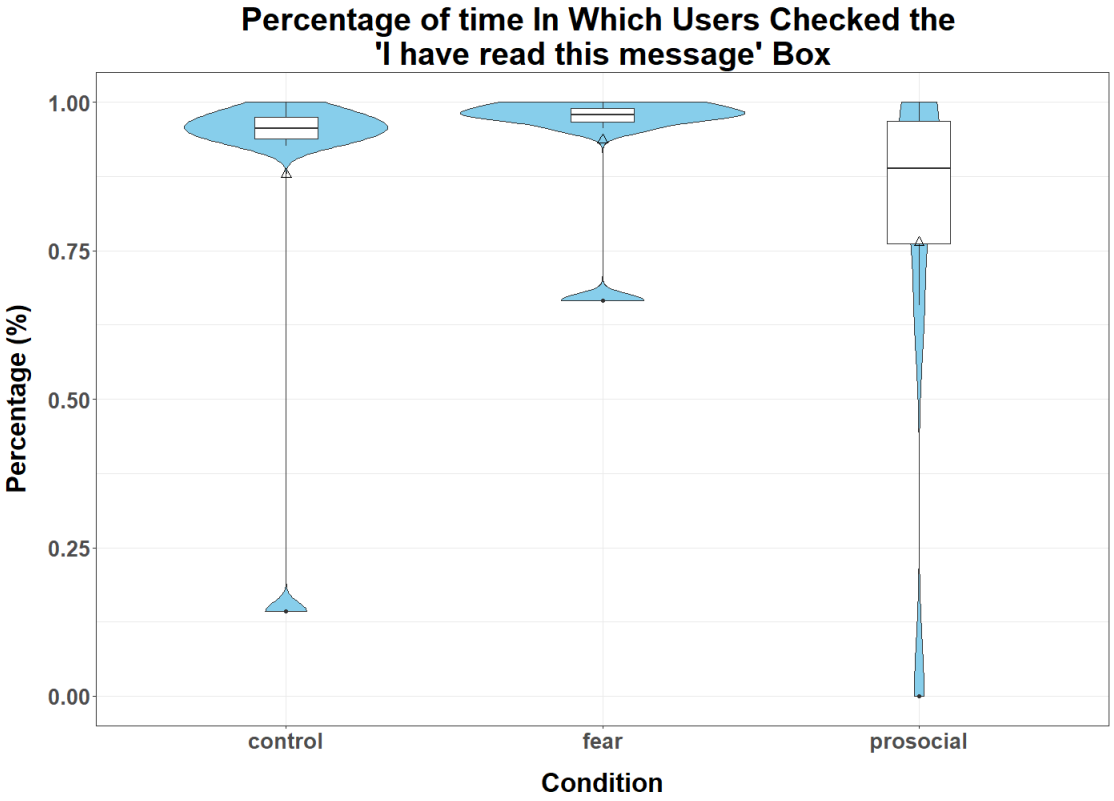


Figure 4.6: Percentage of time in which users checked the 'I have read this message' box. The control and the fear conditions yielded the highest percentage whereas the prosocial was the lowest percentage of time. The prosocial also had more of a spread of users selecting the check box.

4.2.6 Number of Permissions Deactivated by users across treatment groups

There 17 participants in the control treatment group. On average, 4.56 permissions were deactivated throughout the course of this study ($Mdn = 0$, $SD = 17.13$). The range of these permission deactivations were from 0 to 73. On average, participants adjusted 2.84% of their permission settings over the course of the study ($Mdn = 0$, $SD = 9.96$).

In the fear treatment group, there were a total of 12 participants that had completed this study. On average, 1.23 permissions were deactivated throughout the course of this study ($Mdn = 0$, $SD = 2.20$). The range of these permission deactivations were from 0 to 7. On average, participants adjusted 2.84% of their permission settings over the course of the study ($Mdn = 0$, $SD = 10.11$).

Finally, looking a the prosocial treatment group, there were a total of 12 participants that had completed this study. On average, 4.70 permissions were deactivated throughout the course of this study ($Mdn = 0.5$, $SD = 12.79$). The range of these permission deactivations were from 0 to 41. On average, participants adjusted 9.28% of their permission settings over the course of the study ($Mdn = 0.53$, $SD = 18.74$).

A One-way ANOVA was performed on the number of permissions that were deactivated by the users during the course of the study. There was not a significant difference between conditions for users deactivating permissions throughout the course of the study at the $p > 0.05$ level for the three conditions [$F(2, 38) = 0.31$, $p = 0.73$] with standard deviations of 17.6218, 2.2043, and 12.7893 for the control, fear, and the prosocial treatment groups respectively.

	Treatments			
	Control	Fear	Prosocial	Total
N	17	12	12	41
ΣX	82	16	47	145
Mean	4.8235	1.2308	4.7	3.625
ΣX^2	5364	78	1693	7135
σ	17.6218	2.2043	12.7893	13.0181
Source	SS	df	MS	F
Between-Treatments	110.4967	2	55.2484	0.31455
Within-Treatments	6498.8783	38	175.6454	
Total	6609.375	40		
$p = 0.732054$				

Table 4.3: One-Way ANOVA on the number of permissions that were deactivated by the users during the course of the study

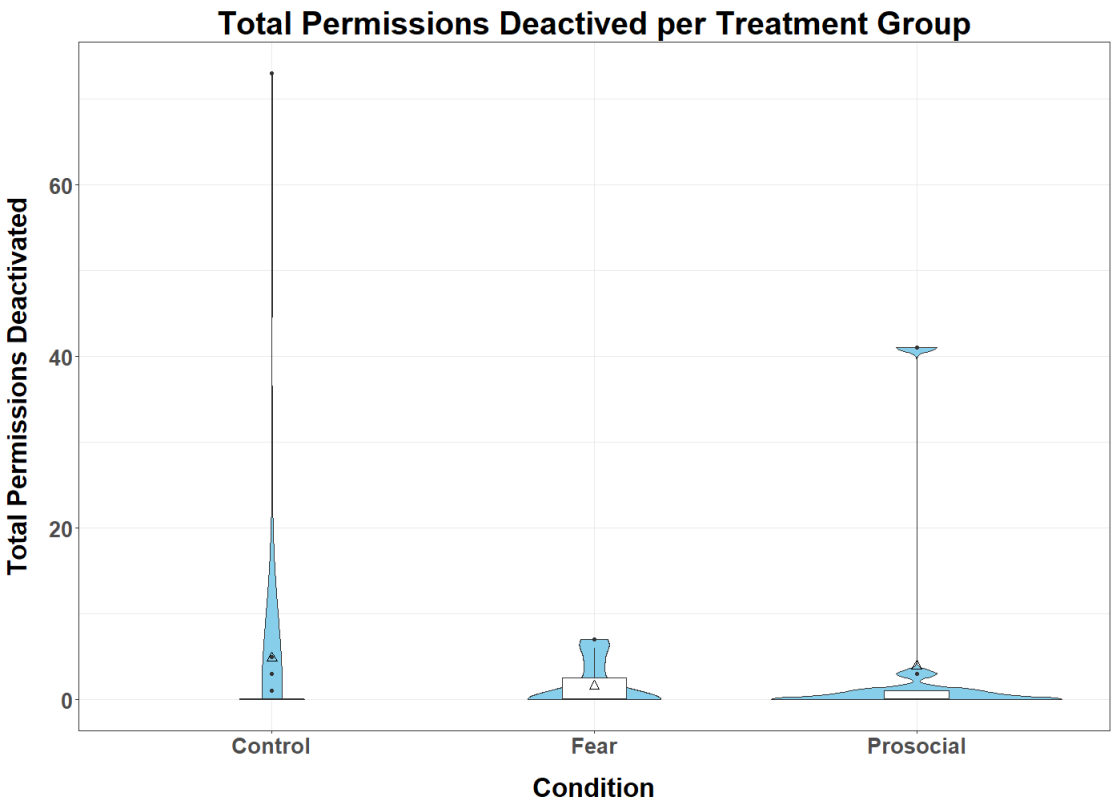


Figure 4.7: Total permissions deactivated per treatment group. This plot shows that users in control and prosocial conditions had larger ranges and medians as opposed to the fear treatment group.

Chapter 5

Conclusions

5.1 Thesis Summary

This thesis focused on the novel design of effective warning messages that employed fear and prosocial framing in an effort to determine which would be most effective in motivating the user to consider making changes to their security settings. Participants interacted with warnings that were displayed on their devices and the way in which they interacted with these warnings was recorded. Furthermore, their device was periodically scanned in order to see when they made changes to their permission settings. The scanning was not apparent to the user and happened in the background while they performed their daily use of their devices.

These warnings were tested on participants in a field study that was performed using the participants personal devices. This field study lasted for 2.5-3 weeks for each of the participants. The warnings were released to the participants periodically throughout the day over the course of this study in which no warning was presented within a three hour interval of the last warning.

5.2 Discussion

The study performed yielded results that showed the varying levels of effectiveness of the proposed novel warning messages. Overall, out of the three treatment groups that were tested the prosocial treatment group participants were more likely to respond in

ways that were trending towards the results of the previous study[52]. This proved to be the case in multiple aspects. More participants in the prosocial treatment group were likely to disable permissions throughout the course of the study, were more likely to choose the option to change their settings, more likely to view applications requesting this permission, and had higher average durations in which they had viewed the warning message before deciding as compared to participants in the fear and control groups.

The fear treatment group seemed to be the worst performing treatment group out of the three. Participants in the fear group were more likely to choose the 'Keep Sharing My Private Information' option from the warning message rather to exit the warning rather than the 'Let Me Change My Settings' option. Furthermore, warning messages presented to the participants in the fear treatment group also had the lowest average viewing/interacting time of all the treatment groups.

The control and the prosocial treatment groups had highest overall average number of permissions deactivated throughout the course of this study. An interested note is that although the fear treatment group did not have the highest average number of permissions deactivated, they did have the largest number of participants who deactivated permissions.

Participants in the prosocial group selected the button to view apps that were granted permissions groups that they were being warned about sometimes even twice. This result was twice as many times on average as compared to the control treatment group. The fear treatment group had the lowest overall average of number of times users viewed the apps during all of the warnings they received. This suggests that participants in the prosocial treatment group were more likely want to see apps that relate to the permission group being warned about and pose a potential security risk.

In regards to the checkbox that was provided, keeping in mind that this box was not mandatory for selection, the control and the fear group had a much higher selection rate

of the check box as can be seen from 4.6. This suggests this suggests that people were more likely to read the control and fear warnings than the prosocial warnings. Although, there was a higher rate in which participants had selected the box in the control and fear treatment groups, there was a lower average viewing/interaction duration of the warning messages. Especially for the fear group with the lowest viewing/interaction duration and the highest average for checking the 'I have read this message' checkbox. Also, members of the prosocial treatment group had the inverse of this result holding the highest average for viewing/interaction duration and the lowest average for checking the 'I have read this message box'. In regards to the fear treatment group, the high rate of checking the box and the much lower average viewing/interacting duration of the warning messages suggests that although participants may have found the warnings to be initially more attention-grabbing, the warnings did not influence security settings as effectively.

Another interesting note was how warnings about certain permission groups yielded varying results. Specifically, what permission group being warned about was most likely to get the user to consider changing their permission settings via the 'Let Me Change My Settings' Button. As can be seen from 4.3, warnings that were most related to the user's phone had the most selections of the 'Let Me Change My Settings' button to go view their settings. This is followed by microphone, contacts, and sms permissions. This is reasonable as these four permissions are very close to the user on a personal level such as who they know, what they say, what they write, and who they call.

None of these results turned out to statistically significant, as can be seen by the One-Way ANOVA tables 4.1 4.2 4.3. However, the results obtained in this study still trended towards that of the prior study in which prosocial framing proved to be an effective novel framing technique for warning messages.

5.3 Limitations

Two common issues arose from the fact that the participants that were in this study had non-rooted Android devices and it would require root privileges to access certain information on their device. We could not completely simulate the way Android users receive just-in-time permission warnings because it is an administrative capability to view which apps are access which permission at the time of request. A second issue that arose from this lack of administrative capability was the fact that participants could not be directed completely to where they would be able to make the permission changes. Rather, they were directed to a menu as close as possible and then presented with instructions to carry on from where we could direct them to.

Bibliography

[1] The noticeability of warnings on alcoholic beverage containers. *Journal of Public Policy and Marketing*, 12(1):38–56, 1993.

[2] Applicationinfo. 2017. `https://developer.android.com/reference/android/content/pm/ApplicationInfo.html`.

[3] Broadcasts. 2017. `https://developer.android.com/guide/components/broadcasts.html`.

[4] Componentinfo. 2017. `https://developer.android.com/reference/android/content/pm/ComponentInfo.html`.

[5] Context. 2017. `https://developer.android.com/reference/android/content/Context.html#getPackageManager`.

[6] Optimizing app-initiated network use. 2017. `https://developer.android.com/topic/performance/power/network/action-app-traffic.html`.

[7] Packageinfo. 2017. `https://developer.android.com/reference/android/content/pm/PackageInfo.html`.

[8] Patterns – permissions. 2017. `https://material.io/guidelines/patterns/permissions.html`.

[9] Permissions overview. 2017. `https://developer.android.com/guide/topics/permissions/overview.html`.

[10] Services. 2017. `https://developer.android.com/guide/components/services.html`.

[11] google-gson. 2018. `https://github.com/google/gson`.

[12] Packagemanager. 2018. `https://developer.android.com/reference/android/content/pm/PackageManager.html`.

[13] Support different pixel densities. 2018. `https://developer.android.com/training/multiscreen/screendensities.html`.

[14] Support different screen sizes. 2018. `https://developer.android.com/training/multiscreen/screensizes.html`.

[15] Devdatta Akhawe and Adrienne Felt. Alice in warningland: A large-scale field study of browser security warning effectiveness. *Usenix*, 22:257–272, 2013.

[16] Yusuf Albayram, Mohammad Maifi Hasan Khan, Theodore Jensen, and Nhan Nguyen. "...better to use a lock screen than to worry about saving a few seconds of time": Effect of fear appeal in the context of smartphone locking behavior. *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 49–63, 2017.

[17] Hazim Almuhimedi, Andrienne Porter Felt, Robert W. Reeder, and Sunny Consolvo. Your reputation precedes you: History, reputation, and the chrome malware warning. *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 113–128, 2014.

[18] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Aquisti, Joshua Gluck, Lorrie Cranor, and Yuvraj Agarwal. Your location has been shared 5,398 times! a field study on mobile app privacy nudging. *CHI*, pages 787–796, 2015.

[19] T. S. Amer and Jo-Mae B. Maris. Signal words and signal icons in application control and information technology exception messages–hazard matching and habituation effects. *Journal of Information Systems*, 21(2):1–25, 2007.

[20] Bonnie Brinton Anderson, C. Brock Kirwan, Jeffrey L. Jenkins, David Eargle, Seth Howard, and Anthony Vance. How polymorphic warnings reduce habituation in the brain—insights from an fmri study. *CHI*, pages 2883–2892, 2015.

[21] App Annie. App annie retrospective. 2017. `http://go.appannie.com/app-annie-2016-retrospective`.

[22] Rebecca Balebako, Jaeyeon Jung, Wei Lu, Lorrie Faith Cranor, and Carolyn Nguyen. "little brothers watching you": Raising awareness of data leaks on smartphones. *In Proceedings of the Ninth Symposium on Usable Privacy and Security (SOUPS '13)*, (12), 2013.

[23] C. Daniel Batson and Adam A. Powell. Altruism and prosocial behavior. *Handbook of psychology*, pages 463–484, 2003.

[24] Rainer Bohme and Jens Grossklags. The security cost of cheap user interaction. *New Security Paradigms Workshop (NSPW)*, pages 67–82, 2011.

[25] Christian Bravo-Lillo, Lorrie Faith Cranor, Julie Down, and Saranga Komanduri. Bridging the gap in computer security warnings: A mental model approach. *IEEE Security & Privacy*, 9(2):18–26, 2010.

[26] Fred Cate. The limits of notice and choice. *IEEE Security & Privacy*, 8(2):59–62, March 2010.

[27] Sauvik Das. Social cybersecurity: Understanding and leveraging social influence to increase security sensitivity. *it – Information Technology*, 58(5):237–245, 2016.

[28] Rachna Dhamija, J. D. Tygar, and Hearst Marti. Why phishing works. *CHI*, pages 581–590, 2006.

[29] Serge Egelman, Lorrie Faith Cranor, and Jason Hong. You've been warned: An empirical study of the effectiveness of web browser phishing warnings. *CHI*, pages 1065–1074, 2008.

[30] Serge Egelman, Adrienne Porter Felt, and David Wagner. Choice architecture and smartphone privacy: There's a price for that. *The economics of information security and privacy*, pages 211–236, 2013.

[31] David Egilman and Susanne Rankin Bohme. A brief history of warnings. *Handbook of Warnings*, pages 11–20, 2006.

[32] Andrew J. Elliot, Markus A. Maier, Arlen C. Moller, Ron Friedman, and Jorg Meinhardt. Color and psychological functioning: The effect of red on performance attainment. *Journal of Experimental Psychology: General*, 136(1):154–168, 2007.

[33] Adrienne Porter Felt, Alex Ainslie, Robert W. Reeder, Sunny Consolvo, Somas Thyagaraja, Alan Bettes, Helen Harris, and Jeff Grimes. Improving ssl warnings: Comprehension and adherence. *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*, pages 2893–2902, 2015.

[34] Adrienne Porter Felt, Serge Egelman, and David Wagner. I've got 99 problems, but vibration ain't one: A survey of smartphone users' concerns. *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, 2012.

[35] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android permissions: User attention, comprehension, and behavior. *Symposium on Usable Privacy and Security (SOUPS)*, (3), 2012.

[36] Huiqing Fu, Yulong Yang, Nileema Shingte, Janne Lindqvist, and Marco Gruteser. A field study of run-time location access disclosures on android smartphones. *Proc. USEC*, 2014.

[37] John C. Hersey, David A. Asch, Thi Thumasathit, Jacqueline Meszaros, and Victor V. Walters. The roles of altruism, free riding, and bandwagoning in vaccination decisions. *Organizational Behavior and Human Decision Processes*, 59(2):177–187, August 1994.

[38] International Data Corporation (IDC). Smartphone os market share, 2017 q1. 2017. `https://www.idc.com/promo/smartphone-market-share/`.

[39] Jeffrey L. Jenkins, Mark Grimes, Jeffrey Gainer Proudfoot, and Paul Benjamin Lowry. Improving password cybersecurity through inexpensive and minimally invasive means: Detecting and deterring password reuse through keystroke-dynamics monitoring and just-in-time fear appeals. *Information Technology for Development*, 20(2):196–213, 2014.

[40] Allen C. Johnston and Merrill Warkentin. Fear appeals and information security behaviors: an empirical study. *MIS quarterly*, pages 549–566, 2010.

[41] Jaeyeon Jung, Seungyeop Han, and David Wetherall. Short paper: Enhancing mobile application permissions with runtime feedback and constraints. *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '12)*, pages 45–50, 2012.

[42] Naz Kaya and Helen H. Epps. Relationship between color and emotion: A study of college students. *College Student Journal*, 38(3):396–406, 2004.

[43] Soyun Kim and Michael Wogalter. Habituation, dishabituation, and recovery effects in visual warnings. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 53(20):1612–1616, October 2009.

[44] Moon J. Lee and Mary Ann Ferguson. Effects of anti-tobacco advertisements based on risk-taking tendencies: Realistic fear vs. vulgar humor. *Journalism & Mass Communication Quarterly*, 79(4):945–963, 2002.

[45] Moon J. Lee and Mija Shin. Fear versus humor: The impact of sensation seeking on physiological, cognitive, and emotional responses to anti-alcohol abuse messages. *The Journal of Psychology*, 145(2):73–92, 2011.

[46] Glenn Lesner, Paul Bolls, and Erika Thomas. Scare'em or disgust'em: The effects of graphic health promotion messages. *Health Communications*, 24(5):447–458, 2009.

[47] Meng Li, Eric G. Taylor, Katherine E. Atkins, Gretchen B. Chapman, and Allison P. Galvani. Stimulating influenza vaccination via prosocial motives. 2016. Art. no. e0159780.

[48] Jialiu Lin, Shahriyar Amini, Jason I. Hong, Norman Sadeh, Janne Lindqvist, and Joy 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)*, pages 501–510, 2012.

[49] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun Zhang, Norman Sadeh, Alessandro Acquisti, and Yuvraj Agarwal. Follow my recommendations: A personalized privacy assistant for mobile app permissions. *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*, pages 27–41, 2016.

[50] George Loewenstein, Troyen Brennan, and Kevin G. Volpp. Asymmetric paternalism to improve health behaviors. *JAMA*, 298(20):2415–2417, 2007.

[51] James E. Maddux and Ronald W. Rogers. Protection motivation and self-efficacy: A revised theory of fear appeals and attitude change. *Journal of Experimental Social Psychology*, 19(5):469–479, 1983.

[52] Meghan McLean, Demetrios Lambropoulos, David Lambropoulos, and Janne Lindqvist. The psychological and security effects of warnings using prosocial and fear framing.

[53] Sara Moscato, David R. Black, Carolyn L. Blue, Marifran Mattson, Regina A. Galer-Unti, and Daniel C. Coster. Evaluating a fear appeal essage to reduce alcohol use among greeks. 2001.

[54] Hye-Jin Paek, Kyongseok Kim, and Thomas Hove. Content analysis of antismoking videos on youtube: message sensation value, message appeals, and their relationships with viewer responses. 2010.

[55] Walter C. Buboltz Jr PhD, Franklin Brown MA, and Barlow Soper PhD. Sleep habits and patterns of college students: A preliminary study. *Journal of American College Health*, 50(3):131–135, 2001.

[56] Catherine H. Rankin, Thomas Abrams, Robert J. Barry, Seema Bhatnagar, David F. Clayton, John Colombo, Gianluca Coppola, Mark A. Geyer, David L. Glanzman, Stephen Marsland, Frances K. McSweeney, Donald A. Wilson, Chun-Fang Wu, and Richard F. Thompson. Habituation revisited: An updated and revised description of the behavioral characteristics of habituation. *Neurobiology of Learning and Memory*, 92(2):135–138, 2009.

[57] Annesa Flentje Santa and Bryan N. Cochran. Does the impact of anti-drinking and driving public service announcements differ based on message type and viewer characteristics? 2008.

[58] Carol L. Schmitt and Thomas Blass. Fear appeals revisited: Testing a unique anti-smoking film. *Current Psychology*, 27(2):145–151, 2008.

[59] Denise D. Schoenbachler and Tommy E. Whittler. Adolescent processing of social and physical threat communications. *Journal of Advertising*, 25(4):37–54, 1996.

[60] Omar Shehryar and David M. Hunt. A terror management perspective on the persuasiveness of fear appeals: A terror management perspective. 2005.

[61] Fuming Shih, Ilaria Liccardi, and Daniel Weitzner. Privacy tipping points in smartphones privacy preferences. *CHI '15 Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 807–816, 2015.

[62] Irina Shklovski, Scott D. Mainwaring, Halla Hrund Skuladottir, and Hoskuldur Borgthorsson. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. *Proceedings of the 32nd Annual ACM COnference on Human Factors in Computing Systems (CHI '14)*, pages 2347–2356, 2014.

[63] Mario Silic, Justin Barlow, and Dustin Ormond. Warning! a comprehensive model of the effects of digital information security warning messages. *The 2015 Dewald Roode Workshop on Information Systems Security Research*, 2015.

[64] Mario Silic and Dianne Cyr. Colour arousal effect on users' decision-making processes in the warning message context. pages 99–109, 2016.

[65] Statista. *Number of available applications in the Google Play Store from December 2009 to December 2017*. `https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/`, 2018.

[66] James W. Sturges and Ronald W. Rogers. Preventive health psychology from a developmental perspective: An extension of proctection motivation theory. 1996.

[67] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor. Crying wolf: An empirical study of ssl warning effectiveness. *Proceedings of the 18th Conference on USENIX Security Symposium (SSYM'09)*, pages 399–416, 2009.

[68] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negron-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. The effect of developer-specified explanations for permission requests on smartphone user behavior. *CHI*, pages 91–100, 2014.

[69] Richard F. Thompson and William A. Spencer. Habituation: a model phenomenon for the study of neuronal substrates of behavior. *Psychological Review*, 73(1):16–43, 1966.

[70] Jean Walters, Michael J. Apter, and Svebak Sven. Color preference, arousal, and the theory of psychological reversals. *Motivation and Emotion*, 6(3):193–215, 1982.

[71] Na Wang, Pamela Wisniewski, Heng Xu, and Jens Grosslags. Designing the default privacy settings for facebook applications. *Proceedings of the Companion Publication of the 17th ACM Conference on Computer SUpported Cooperative Work & Social Computing (CSCW Companion '14)*, pages 249–252, 2014.

[72] Keith Weber, Megan R. Dillow, and Kelly A. Rocca. Developing and testing the anti-drinking and driving psa. 2011.

[73] Dawn K. Wilson, Scot E. Purdon, and Kenneth A. Wallston. Compliance to health recommendations: a theoretical overview of message framing. *Health Education Research*, 3(2):161–171, 1988.

[74] Brian Wilt. How university students sleep. 4 2016. `https://jawbone.com/blog/university-students-sleep/`.

[75] Michael Wogalter. Purposes and scope of warnings. *Handbook of Warnings*, 2006.

[76] Michael S. Wogalter and William J. Vigilante. Attention switch and maintenance. *Handbook of Warnings*, pages 245–265, 2006.

[77] Stephen L. Young. Increasing the noticeability of warnings: Effects of pictorial, color, signal icon and border. *Proceedings of the Human Factors Society Annual Meeting*, 35(9):580–584, 1991.

[78] Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. How busy are you? predicting the interruptibility intensity of mobile users. *CHI*, pages 5346–5360, 2017.

[79] Rick S. Zimmerman, Pamela K. Cupp, Melissa Abadi, Lewis R. Donohew, Carla Gray, Leonard Gordon, and Bailey A. Grossl. The effects of framing and fear on ratings and impact of antimarijuana psas. 2014.