

**PrivacyInformer: An Automated Privacy
Description Generator for the MIT App Inventor**

by

Daniela Yidan Miao

B.A.Sc. Engineering Science, University of Toronto (2012)

Submitted to the Engineering Systems Division

and

Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degrees of

Master of Science in Technology and Policy

and

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2014

© Massachusetts Institute of Technology 2014. All rights reserved.

Author

Engineering Systems Division

Department of Electrical Engineering and Computer Science

August 15, 2014

Certified by

Lalana Kagal

Principal Research Scientist

Department of Electrical Engineering and Computer Science

Thesis Supervisor

Accepted by

Dava J. Newman

Professor of Aeronautics and Astronautics and Engineering Systems

Director, Technology and Policy Program

Accepted by

Leslie A. Kolodziejcki

Professor of Electrical Engineering and Computer Science

Chair, Department Committee on Graduate Students

PrivacyInformer: An Automated Privacy Description Generator for the MIT App Inventor

by

Daniela Yidan Miao

Submitted to the Engineering Systems Division
and
Department of Electrical Engineering and Computer Science
on August 15, 2014, in partial fulfillment of the
requirements for the degrees of
Master of Science in Technology and Policy
and
Master of Science in Electrical Engineering and Computer Science

Abstract

With the advent of “smart” mobile phones and ubiquitous mobile applications, the pace at which people generate, access, and acquire data has accelerated significantly. In this thesis, we first examine how privacy issues in the mobile apps market compromise the well-being of both app consumers and developers, noting that one important problem is the lack of usable privacy policies. Subsequently, we propose a technical solution named PrivacyInformer that automatically generates mobile app privacy descriptions, thereby relieving developers the burden of manually creating them. This tool is implemented as an extension to the MIT App Inventor, a do-it-yourself mobile app building platform that has a vast international user base, as well as a growing impact on the democratizing of mobile app building. We show that by analyzing source code of mobile apps directly in App Inventor, PrivacyInformer can produce simple and useful privacy descriptions in both human-readable and machine-readable format. Specifically, these generated documents describe how mobile apps use private information, rather than simply enumerating a list of data access as done in the permissions system. Finally, we conduct an exploratory user study to evaluate the effectiveness of PrivacyInformer from the app developer’s perspective, as well as discuss the policy impact of such a tool in the mobile app development community.

Thesis Supervisor: Lalana Kagal
Title: Principal Research Scientist
Department of Electrical Engineering and Computer Science

Acknowledgments

I would like to thank my advisor, Lalana Kagal, for her immense patience and understanding in helping me complete this thesis. She has tirelessly motivated me throughout my two years here at MIT. She listened to my research interests and encouraged me to explore on my own, I could not have done this without her help.

This thesis would not have been possible without the contribution of Fuming Shih and Ilaria Liccardi, both of the Decentralized Information Group (DIG) at MIT; They provided me with insight into the fascinating world of mobile privacy. Danny Weitzner, Wendy Seltzer and K Waterman all found time in their busy schedules to help me with the policy aspects of this thesis, and for that I am extremely grateful. Hal Abelson has encouraged and supported me from the beginning, and I am very fortunate to have such an education enthusiast review my thesis. I thank Evan Patton of Rensselaer Polytechnic Institute and Oshani Seneviratne, another DIG member, for their advice on Linked Data in App Inventor. I am indebted to Andrew McKinney, Jos Flores and Weihua Li of the MIT App Inventor team for their patient explanation of the App Inventor system. Marisol Diaz has been an exceptional administrator, promptly sorting out all logistics related to my thesis. Last but not least, I want to thank everyone I have met during my time here, including members of the DIG group, App Inventor team, and the Technology and Policy program, everyone here has made a truly enjoyable environment to study and work in.

Contents

1	Introduction	11
2	Background and Motivation	15
2.1	Background of MIT App Inventor	15
2.1.1	Designer	16
2.1.2	Blocks Editor	16
2.1.3	Compiler	17
2.2	Current Mobile Apps Market	18
2.3	Motivation	20
3	Related Work	23
3.1	Android Permissions	23
3.2	Statically Analyzing Source Code	24
3.3	Machine-readable Privacy Descriptions	25
3.4	Human-readable Privacy Descriptions	26
4	Overview of PrivacyInformer	29
4.1	Sample Scenario Walk-through	31
4.2	Design & Implementation	32
4.2.1	Extract	35
4.2.2	Analyze	35
4.2.3	Build	40

5	Evaluation	43
5.1	Exploratory User Study	43
5.1.1	Methodology	44
5.1.2	User Background	45
5.1.3	Feedback on PrivacyInformer	46
5.2	In-person Interviews	48
6	Discussion	51
6.1	Impact of PrivacyInformer in the Mobile Apps Market	52
6.2	Privacy Description as a Legal Document	53
6.3	PrivacyInformer as a Regulatory Mechanism	56
7	Conclusion	59
A	Sample Ontology for Android and App Inventor	61
B	Questionnaire for PrivacyInformer	63

List of Figures

2-1	Sample Designer Interface for the MIT App Inventor	17
2-2	Build Dropdown Menu in the App Inventor Toolbar	18
4-1	Location of privacy description button to access PrivacyInformer . . .	32
4-2	Sample view of privacy description editor for Andrew’s application . .	33
4-3	Sample workflow of user accessing PrivacyInformer	34
4-4	General structure of PrivacyInformer	34
4-5	Example of a source JSON file containing Designer view information .	36
4-6	Example of a source XML file containing Blocks view information . .	36
4-7	A list of App Inventor components that are considered privacy-sensitive	37
4-8	Example of a privacy template for the location sensor	37
4-9	Sample conversion from Linked Data format to HTML format	39
4-10	Mobile view of the generated privacy description, with potentially data leaking actions highlighted in red	41
5-1	Correlation of App Inventor experience with perception of privacy de- scriptions	46
5-2	Correlation of perception of privacy descriptions with PrivacyInformer reception	47
5-3	Amount of potential time spent on devising privacy descriptions with and without PrivacyInformer	49
A-1	Snippet of App Inventor web ontology	61
A-2	Snippet of Android web ontology	62

Chapter 1

Introduction

In this age of technology, users leave a staggering number of sensitive, personally identifiable information on their phones – a reality that results from vastly improved communications in this networking era. Consumers are amassing mobile applications (apps) on their cell phones at an unprecedented rate – the mobile apps market generated an estimated \$20 billion in revenue in 2011 [16]. This may come as no surprise to many since mobile apps present the most common form of interaction with our cell phones. However, recent events surrounding privacy breaches have prompted many users to grow wary over potential privacy risks imposed by their mobile apps [25]. A recent study by Pew Research Center shows that 54% of mobile users have uninstalled or avoided an app due to privacy concerns [4].

Usually users are presented with little information on the privacy implication of apps, but the mobile operation system provides a mechanism by which more concerned users could view detailed privacy policies and permission settings of apps. However, privacy policies are often long and fraught with legal jargon which render the document almost incomprehensible to the average user. For example, the Angry Birds game app displays a 3358-word policy at the time of installation [26]. Works by both McDonald [23] and Kelley [19] have attempted to study better ways of presenting users with an easy-to-understand view of privacy-sensitive behaviors of apps. As will be discussed in more detail in Section 3, McDonald and Kelley both analyzed comprehensibility of different formats of privacy policie, ranging from

natural-language paragraphs to information presented in a “privacy nutrition label”. Both pieces of work identify ideal ways to produce privacy documents, in order to help smart phone users to determine which mobile apps are more privacy-friendly. Nonetheless, producing such as privacy documents is still a non-trivial and onerous task.

For the average mobile app developer, writing privacy documentation is a time-consuming and mundane process. This is especially true for independent and small-company developers, who make up majority of the mobile development community. They are often under tight timelines and lack in resources that allow them to constantly update privacy documentation according to changing privacy regulation. Given the trade-off between generating accurate privacy descriptions and maintaining a fast release cycle, such privacy documentation is often neglected.

This thesis introduces a privacy description generation tool, PrivacyInformer, which alleviates mobile app privacy concerns by helping app developers generate and maintain privacy descriptions automatically, via a platform named the MIT App Inventor[30]. App Inventor is an open-source online software application originally created by Google, and currently maintained by MIT. It allows newcomers to computer programming to create mobile apps for the Android operating system. App Inventor offers a controlled environment where all the components and functions of the application are known prior to compiling the APK package, setting the perfect ground for PrivacyInformer to create privacy documents based on full knowledge of the entire development system. In addition, we chose to work with this platform because of its international popularity – with over 1.5 million users around the world[7]. Integrating privacy enhancements into a widely used system will maximize the social impact it has on the mobile app community. Furthermore, while many mobile development tools are used and tested by limited groups of proficient computer programmers, App Inventor acts as a technological faucet to the public at large. App Inventor tools are exposed to everyone regardless of technical background, meaning privacy enhancements provided by PrivacyInformer can help raise privacy awareness on a global scale.

The contributions of this thesis can be summarized as follows:

- Presented an analysis of privacy issues in the current mobile apps market, specifically pointing out the issue with the lack of usable privacy documents.
- Created a tool named PrivacyInformer that can automatically analyze App Inventor app source code and extract privacy-relevant information into machine-readable format. Privacy information generated is particularly useful because it describes on how data is collected and disseminated, rather than simply itemizing the list of data accessed without clear explanations.
- Demonstrated flexibility of the above mentioned approach by converting the capture privacy information from machine-readable format into concise HTML text.
- Exposed both machine-readable and human-readable privacy descriptions by packaging them with the Android application package file (APK) for the mobile app. This serves as an enabling mechanism for potentially better visual representation of privacy-related information and smart matching of users' privacy preferences with the policy of mobile apps.
- Provided future outlook of the mobile apps market with specific contributions from PrivacyInformer, and suggested possible privacy regulations that could utilize this tool.

This thesis is as organized as follows:

Section 2 gives a brief introduction to the MIT App Inventor, then transitions into a general analysis of privacy issues that exist in the current mobile apps market, recognizing the lack of usable privacy policies as an especially concerning one. Section 3 studies existing work in the field of mobile privacy, the most relevant being the current Android permission system, which also seeks to provide privacy information to end users. Section 4 outlines the design and implementation of the PrivacyInformer tool, including how the App Inventor project source code is extracted, how analysis is performed in order to generate the privacy description, and how the privacy

descriptions are packaged into the final Android mobile app. Section 5 presents an exploratory user study that was conducted in order to evaluate the effectiveness of the system. Section 6 offers a discussion of this tool in the context of the mobile apps market, legal field, and regulatory regime. Finally, Section 7 provides a summary of our achievements so far as well as future outlook for PrivacyInformer in the smart phone mobile apps market.

Chapter 2

Background and Motivation

In order to understand the tool presented in this thesis, it is important for readers to first gain a basic picture of the MIT App Inventor platform, and how it helps mobile app users become mobile app creators. This section begins by providing this background, then transitions into more general privacy issues that currently exist in the mobile apps market. These issues gave rise to the conception of this thesis, as the PrivacyInformer tool attempts to address most of them. Lastly, an outline of specific motivations to use the tool is given, delineating how the tool could be viewed from both developer and user perspectives.

2.1 Background of MIT App Inventor

Given that PrivacyInformer is tightly integrated with App Inventor, it is important to gain a basic understanding of the capabilities of App Inventor first. App Inventor primarily uses a graphical interface that allows users to drag-and-drop visual objects to create an application that can run on Android devices. It is composed of 3 major components:

1. **Designer** - The Designer is an interface where a mobile application's components are displayed. This includes visible components, such as buttons and images, and non-visible components, such as sensors and web connections.

2. **Blocks Editor** - The Blocks Editor provides an interface where the application's programming logic can be created. It contains methods and variables associated with each component as seen in the Designer.
3. **Compiler** - The Compiler is a tool that builds and packages the finished App Inventor project into an APK file that can then be installed on any Android device, or distributed publicly via the Google Play Store.

In the PrivacyInformer tool, we are primarily interested in the contents of Designer and Blocks Editor. However, the Compiler is crucial in ensuring the generated privacy document is properly included with the downloaded mobile application package.

2.1.1 Designer

The Designer provides the entry point for new App Inventor projects. Users can find a palette on the left side that contains a list of available components to be used in the mobile app. Components are either visible or non-visible, where visible components can be dragged onto the simulated screen and non-visible components are arranged below the screen. The screen is simulated to resemble a real Android application interface, and users are free to customize each component via the Property Editor at the right side. For example, users can modify the color, font, position, alignment or hint text for a TextBox component. In addition, all components currently contained in the mobile application can be viewed in the "Components" Tree View located between the simulated screen and the Property Editor. An overview of the Designer Interface can be found in Figure 2-1.

2.1.2 Blocks Editor

The Blocks Editor (or shortened to just "Blocks") presents an interface where users can program the mobile application's logic by dragging and dropping blocks together. On the left side of the window, there is a Blocks Palette that lists all blocks available to the App Inventor project. This includes a list of built-in blocks, such as Text, Math,

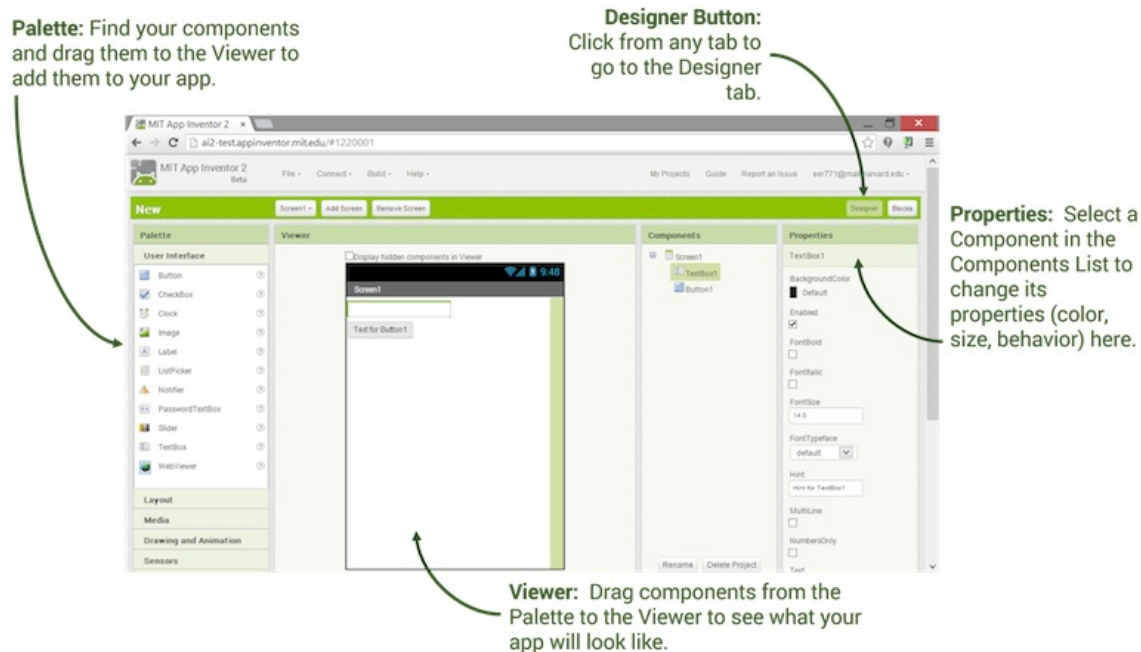


Figure 2-1: Sample Designer Interface for the MIT App Inventor

Lists, Colors, as well as component-specific blocks, which depend on components that have been added to the project in the Designer view. Clicking on any component invokes a list of methods corresponding to that component. For instance, for the Button component, a relevant method is “when Button1.Click”, which specifies the event that the said button has been clicked. At this point, users are free to specify subsequent actions that occur after the button click.

2.1.3 Compiler

The Compiler has no visible interface since it is a service that runs in the background when the user decides to build their project into an installable APK package. However, it is responsible for reading and understanding the contents of both Designer and Blocks in order to create the final compiled mobile app. This action is invoked by going to the “Build” menu on the App Inventor toolbar as shown in Figure 2-2. Selecting “App (save .APK to my computer)” will invoke a pop-up box that informs the progress of the compilation process. Upon completion, the APK file will be downloaded to the user’s computer, at which point it can be installed to any Android device

or uploaded to the Google Play Store for public distribution.

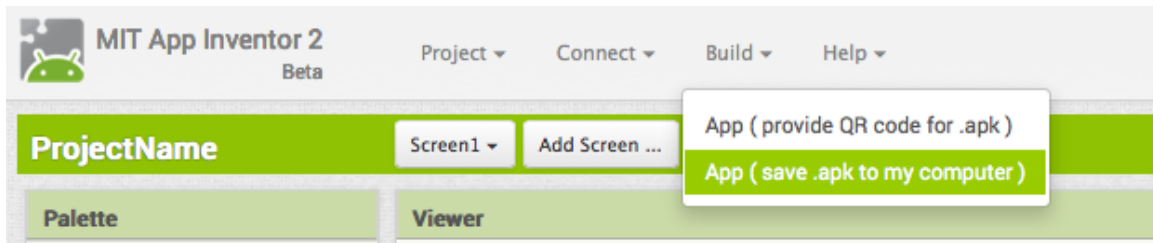


Figure 2-2: Build Dropdown Menu in the App Inventor Toolbar

2.2 Current Mobile Apps Market

Mobile applications present the most common form of interaction with our cell phones, and many of them require user information to function correctly: GPS navigation applications need access to location services; social messaging applications need access to phone numbers and contact lists; fitness apps need age, gender and weight information to accurately calculate the amount of calories burnt. Due to complexity of most software algorithms and the limited computing power of mobile devices, data is often transmitted back to the service provider, processed remotely on centralized servers, and then converted into useful results that are eventually sent back to the user. Unfortunately, this process isn't perfect and can lead to privacy issues when data is collected, disseminated and used. In this section, several different types of privacy-related problems are discussed: the trade-offs between application utility and privacy; lack of privacy policies and control over market entry; and the irreversibility of privacy invasion effects.

Many users tolerate known privacy risks by granting personal data access to mobile apps because the cost of denying access is too high. A typical method of permission request involves a pop-up alert upon opening the app, offering a simple binary choice of granting permissions. Denying access at this stage may result in a non-functional app. This "agree or expect no service" technique is often strong enough of a deterrent to prompt most users to grant access permission. For instance, Whatsapp, an extremely popular social networking tool that enables users to send free picture and

text messages, is very intrusive – it frequently scans the entire user contact list and uploads the data to their own servers. The process persists even when the app is idle or not in the foreground [5]. Yet, this app is extremely popular, and many users knowingly choose to tolerate intrusive behaviors, giving up their privacy for app utility value. Even if a more privacy-friendly alternative existed, there is currently no way of searching for mobile apps based on privacy preferences.

One way to deter excessive user data collection in the first place, is to enforce the necessity of mobile app privacy policies, thereby assigning legal liabilities to mobile app developers. Although privacy policies should not be the only way companies communicate to users about data usage, posting a privacy policy is an essential step for them to become accountable for practices of collecting and disseminating consumer data. However, there are currently no existing laws or regulations requiring mobile app developers to include privacy policies with their apps. In fact, a study conducted by the Future of Privacy Forum found that less than half of mobile apps (across various app stores) offer in-app access to privacy policies [27]. The lack of regulations in this area isn't without reason – imposing rigid privacy rules on the mobile apps industry could lead to stagnation and loss of innovative dynamism. Indeed, flexibility has been the key to success of many small mobile app businesses. These small businesses are not able to properly cope with complex regulations or standards [16]. They do not have the resources for legal departments to deal with evolving regulatory changes. The addition of legal departments brings burden of increased compliance costs and financial penalties that could prove detrimental to the industry - an undesirable outcome for the U.S. economy. Hence, currently it appears challenging to create regulations that can alleviate privacy issues in the mobile apps market, without crippling the market itself.

Finally, even after a user grows wary of the privacy risks involving data collection and dissemination, and tries to erase or dissociate the phone from further such risks, it is difficult to do so due to permanency of certain data properties, such as the phone ID. Effects of privacy invasion on a mobile device are irreversible and impossible to eradicate [34]. As reported by Chris Soghoian, a policy analyst at the American Civil

Liberties Union (ACLU), unlike cookies on the computer that can be deleted once users become suspicious or concerned about protecting their privacy, phone ID is a permanent serial number that is attached to the hardware and cannot be deleted or altered. Once a third-party company obtains any information to establish a profile based on the phone ID, any information coming from that phone can always be mapped back to the same profile. Advertising companies have embraced the use of these permanent IDs in order to facilitate targeted mobile advertising. The only way for a user to start over clean completely, profile-less, is to purchase a new phone, which is obviously a huge inconvenience and unrealistic.

2.3 Motivation

As mentioned earlier, application developers currently lack the incentive to include a privacy policy in their app. Since PrivacyInformer is tightly integrated with App Inventor and produces a privacy description that is similar in nature to a privacy policy, it is important that App Inventor users are properly motivated to use this tool. Other than making the tool itself intuitive and easy to use, there are two major incentives for mobile app developers to use PrivacyInformer:

- **More Information:** Given that App Inventor is an educational software aimed to empower the average user with mobile programming abilities, PrivacyInformer further enhances these abilities by providing application developers with an informative document outlining behaviors of their app. This is helpful because many App Inventor developers do not have a programming background and can benefit significantly from additional documentation on their application. Creating functional and useful apps could be especially difficult for developers that are just warming up to the mobile developing environment, here PrivacyInformer can act as an educational tool that clearly delineates privacy-related behaviors of the application. It is likely that PrivacyInformer reveals behaviors or privacy implications that were not obvious to the application developer before.

- **Higher App Exposure:** As discussed in Section 1, one major privacy concern with the current mobile applications market is the lack of mechanisms for users to select privacy-friendly applications, or gain access to privacy-related information. This could lead to lowered utility for legitimate and privacy-friendly mobile apps because there is no way for application developers to distinguish their privacy-aware apps from the more intrusive ones. PrivacyInformer aims to kick-start this information channel from application developers to application users by offering an easy way for developers to generate a privacy description that advertises privacy-related benefits of their app to users. That being said, there remains a need for a mechanism that allows users to quickly find apps that meet their privacy requirements, this will be discussed later in the future works section.

All the reasons given above certainly contribute to incentivizing App Inventor developers to use PrivacyInformer to increase transparency of their mobile apps. However, an obvious function of PrivacyInformer is to encourage developers to generate privacy documents simply based on good faith. It is possible that developers that had intentions to do so before found the process too burdensome, PrivacyInformer seeks to alleviate that burden significantly.

PrivacyInformer ultimately aims to serve both developers and users by enabling a channel for users to find apps that match their privacy preferences. As stated in the introduction chapter, users have demonstrated significant interest in learning more about mobile application behaviors. Rosen’s work on mobile app behavior was distributed as an Android application in the Google Play Store and had gained 1500 users at the time of publication. Not only are users searching for a better way to understand what their apps are doing, but they are also looking for ways to preserve their privacy without sacrificing app utility value. For instance, a concerned user may not want to avoid instant messaging apps altogether, but use one that does not require sharing his or her contact list. Currently, there is no way for the user to search for apps based on such privacy preference. With PrivacyInformer-enabled apps, users can manually browse and compare privacy descriptions of different instant messaging

apps, and select one which matches his or her preferences. In the future, this process could be automated and performed at the App Store level instead.

An additional benefit of PrivacyInformer is increased level of privacy awareness in users. Since App Inventor is an ubiquitous online platform, any mobile app user can easily become an app developer as well. By presenting more mobile apps with privacy descriptions, users grow increasingly sensitive to privacy issues and potential implications of app behaviors. This leads to more well-informed users who become better app developers, for they will take privacy into account when designing their own mobile apps as well.

Chapter 3

Related Work

PrivacyInformer is not the first tool to attempt to alleviate privacy issues in the mobile apps market. Much concern has risen in the past, and various industrial leaders and academic researchers have proposed promising solutions. Most prominently, mobile app distributors have imposed their own privacy rules on their respective app platforms. The Android operating system specifically presents a permissions screen, which the user must accept before a mobile app can be installed. This section addresses the relevance of this permission system in informing users about privacy, and how PrivacyInformer differs from this existing mechanism. Furthermore, other tools have similarly employed the static source code analysis technique in producing privacy documents, we will also compare these with PrivacyInformer. Lastly, there is also related work in creating privacy documents in both machine-readable and human-readable formats, some of which are used to improve PrivacyInformer – details are discussed in this section.

3.1 Android Permissions

The Android permission system is an important step towards addressing the problem with user notification of privacy implications. However, as discussed in [31], the classes of functionality it covers are excessively broad and offer little meaningful information to users. This system serves as an enforcement mechanism for developers to

declare app capabilities, so most of the permissions are useful at the development level only [12]. As an example, the “Read Phone State” permission allows access to data ranging from phone numbers and phone ID to operation system version. Without providing the specific usage context and detailed behavior, users are unable to make meaningful decisions by simply reading a list of Android permissions [13]. In addition, App Inventor operates on the basis of individual components that in turn generate Android permissions, meaning many App Inventor apps cannot be distinguished from each other at the permissions level. For instance, the “Internet” permission is required by components ranging from Image, Audio to Twitter and WebView. When such prevalent permissions appear on the list, users are likely to ignore them altogether [14].

Finally, even when permissions are fine-grained and understandable by the end user. Many apps request permissions that they do not end up needing or using [12], causing the app to appear more invasive than it actually is. In other cases, the permissions system fails to capture certain privacy-sensitive behaviors [15]. As a result, we argue Android permissions form an incomplete set of abstractions – in order for end users to understand application behavior, they need to be properly informed of *what* data is collected, *how* the data is used and what data *interactions* occur. Hence, PrivacyInformer fills a compelling need for enabling developers to easily expose such information to end users.

3.2 Statically Analyzing Source Code

The prevalence of Android devices gave rise to many tools that perform static analyses of mobile app source code in order to derive application behavior. Rosen et al. provide high-level application behavior profiles by statically analyzing Android API calls and mapping them to fine-grained privacy-sensitive behaviors [31]. Flow Permissions, as presented in [33], provides semantic information based on information flows from one source to another. Both of these works rely on decompiling mobile apps back to Java source so that relevant API calls can be accurately identified. This process is not perfect and could often lead to errors in analyses [11]. In contrast,

PrivacyInformer has direct access to original source code of the mobile app since it is directly integrated into App Inventor’s app-building environment. This is also differentiable from previous works cited because all components and methods used in App Inventor are known to PrivacyInformer a priori, while the tools mentioned above may not have full knowledge of the complete set of Android API methods and third-party library methods.

There has also been a slew of work aimed at malicious app detection. Enck et al. demonstrates possibility of using existing static analysis tools to detect malicious behavior in Android apps [10]. Similarly, Chan et al. uses static analysis to correlate user actions with privacy data leaks, identifying apps that contain user-driven security vulnerabilities [6]. More recent work by Book et al. shows privacy leakages from API methods in specific ad libraries [3]. Unlike these cited works, PrivacyInformer aims to only produce a descriptive document that delineates privacy-related behaviors of mobile apps, rather than noting whether the captured behavior is acceptable – we leave that judgment to the users, based on their own tolerance.

3.3 Machine-readable Privacy Descriptions

There have been past efforts by the World Wide Web Consortium (W3C) to standardize and popularize computer-readable privacy policies for web sites. The Platform for Privacy Preferences (P3P) is one such privacy standard that allows websites to post their privacy policies in P3P format and web browsers would download them automatically and compare them with each user’s privacy settings [8]. P3P also offers a rich vocabulary with which websites can describe their privacy practices. While the idea behind P3P is very similar to the framework created by PrivacyInformer, its execution was not properly motivated. Companies that make web sites feel no pressure to create and publish such machine-readable policies, and no regulator has stepped in to encourage otherwise. On the other hand, while mobile privacy policies are also optional for the most part, recently passed laws in California suggest that regulators are starting to crack down on mobile privacy [16]. More and more inde-

pendent mobile developers are now seeking help in generating privacy documents for their applications [28]. As a result, PrivacyInformer’s automatic privacy description generation feature, which does not require any manual input from app developers, fills a proper need in the mobile apps development community.

Keeping in mind the need to produce mobile privacy documents in machine-readable format, it is recommended by W3C that web ontologies be re-used as often as possible. However, in this case the P3P ontology as published is does not sufficiently encompass many common concepts encountered by PrivacyInformer, especially in the mobile context. For instance, the mobile privacy document will frequently need to mention smart phone sensors and related data, such as location, acceleration, phone contacts and camera access. These terms are not defined by the P3P ontology, and hence a new ontology must be created to represent mobile privacy concepts properly. As such, the author has created and published web ontologies for both the Android system and App Inventor, snippets of these can be found in Appendix A.

3.4 Human-readable Privacy Descriptions

A slew of related work has appeared recently with regards to presentation of privacy information to end users. The National Telecommunications & Information Administration (NTIA) released a Code of Conduct for Mobile Privacy Notices in July 2012, detailing what information developers should include in a privacy notice, and in what format [36]. The Federal Trade Commission (FTC) released similar guideline for mobile app developers in January 2013. Many online, questionnaire-based privacy policy generation tools have picked up on these guidelines and updated their systems accordingly [28]. A study by McDonald et al. found two such privacy policy generators, TRUSTe and Privacy Choice, to be more successful than traditional natural language policies in conveying privacy concepts to end users [23]. Works by Kelley et al. have also indicated effectiveness of privacy notices that are succinct and presented in a tabular manner [19]. However, all these works focus on improving visual presentation of privacy information, which is orthogonal to the purpose of PrivacyInformer, which is

to automatically extract privacy-related behavior from application source code. Since PrivacyInformer captures such behaviors in machine-readable format first, one can then easily convert this into various types of human-readable formats. In this paper, we exemplify this by converting the privacy description from Linked Data format to short excerpts of HTML text, following guidelines for “standardized short text” descriptions as described in Kelley’s study [20]. For future enhancements, one can further annotate the human-readable privacy description by inserting icons, graphics, and animations, independent of the PrivacyInformer itself.

Chapter 4

Overview of PrivacyInformer

PrivacyInformer aims to alleviate current privacy issues by offering an automatic privacy description generator for App Inventor. Currently, there exists many on-line tools that attempt to help application developers quickly produce a notice or document that describes their app’s privacy-related behaviors [28]. Despite providing user-friendly graphical interfaces, such sites still require the developer to answer pages and pages of questions regarding their application. This time commitment alone can become a significant barrier as developers simply do not find enough incentive to use such tools. The PrivacyInformer is able to circumvent this problem altogether by automating the entire privacy notice generation process. This is achievable since the App Inventor offers a controlled environment where all the components and functions of the application are known prior to compiling the APK package.

Upon starting an App Inventor project, every developer will have the option of “opting into” services provided by PrivacyInformer, which will automatically generate and package a privacy description with the application APK package each time the developer decides to “save APK” to their computer. The system is designed such that, at the time of compilation, PrivacyInformer statically analyzes the source code of the App Inventor project, and gains an understanding of how components are used within the application. PrivacyInformer begins by identifying privacy-sensitive components in the application, subsequently imports pre-generated privacy templates corresponding to these components. For instance, if a mobile app uses the Web component, its

corresponding privacy template states the application’s ability to transfer data to the Internet. Upon more detailed analyses on the methods used by each component, we can enrich this privacy template further with additional information.

All privacy templates store information regarding the components in machine-readable, Linked Data format because this allows us more flexibility for data manipulation later. Indeed, PrivacyInformer uses an intermediate data structure to combine, process and annotate these templates before producing the application-specific privacy description, also in Linked Data format. At the time of application APK compilation, a human-readable version of the privacy description is generated and attached, along with the Linked Data version, to the meta-data of the application. This way, mobile application users can access the human-readable version via the regular Android menu options. In the future, corresponding rules can be constructed based on the Linked Data version to match user’s own privacy preferences. More future work is discussed in Section 6.

It is important to distinguish between the privacy document PrivacyInformer produces and a legally binding document such the privacy policy of a mobile application. Privacy policies have to follow strict legal guidelines and restrictions, which means they are often prepared by lawyers or companies with specialized legal expertise. Unfortunately, such legal requirements lead to inevitably long, vague and incomprehensible privacy policies. In addition, application developers are held liable to the contents of privacy policies, which further lowers incentives to create them in the first place. The aim of PrivacyInformer is to generate a document that is easily understandable by mobile app users, that simply describes the behavior of the application without the contents being legally binding or burdening to the application developer. The goal is to produce an informative document that will be useful to both the application developer and the application user. Perhaps, if an application developer is interested in formulating a legally binding privacy policy, the generated privacy document can serve as a starting point for this policy draft, but that is completely outside of the scope of PrivacyInformer’s work.

4.1 Sample Scenario Walk-through

As just described, PrivacyInformer is developed as a “plug-in” that automatically generates a tailored privacy description by analyzing components that App Inventor users include in their applications. By studying both the “Designer” and “Blocks” view, the PrivacyInformer can learn a lot of information regarding what type of data is being collected, when it is collected, and where it is sent to. A hypothetical workflow for an App Inventor user, Andrew, is described below:

1. Andrew first designs the user interface (UI) of his application by using basic components such as TextBoxes, Buttons and Labels. More importantly, he also drags a few sensors into the Designer View: LocationSensor and AccelerometerSensor. Finally, he includes the Web component to enable transferring of data to his server.
2. Andrew then completes the logic of his application by using the Blocks Editor: he sets up the user interface behavior, streamlines the data collection process, and inputs his server’s URL for data to be transmitted.
3. After Andrew finishes his application, he is ready to build the project and download the compiled mobile app. However, before that, he can click on the “Privacy Description” button located next to the “Designer/Blocks” buttons to create a privacy document for his application, as shown in Figure 4-1.
4. When User clicks on the “Privacy Description” button, a new window pops up that shows an editor containing an automatically generated privacy description for his application. The content is tailored specifically to his application, based on the components, as well as the programming logic of the application. For example, since he is using the Accelerometer, there will be a paragraph in the privacy description informing the user that the mobile app has the ability to measure acceleration of the phone, hence detecting phone movements. A sample of this is illustrated in Figure 4-2.

5. If Andrew is satisfied with the privacy description, he can check the “opt-in” option to indicate that he wishes to include a privacy description in his application. From this point on, every time Andrew builds his application’s APK package, the privacy description will be included in the meta-data of the application.
6. Note that the privacy description is generated at compile time every time an APK package is built, which means if Andrew changes his application after viewing the preview, the privacy description included in the downloaded mobile APK package will be updated to reflect his changes, therefore contents may differ from the preview he saw originally.
7. When a user installs Andrew’s application, the privacy description can be accessed via the regular Android menu option, as shown in Figure 4-10.

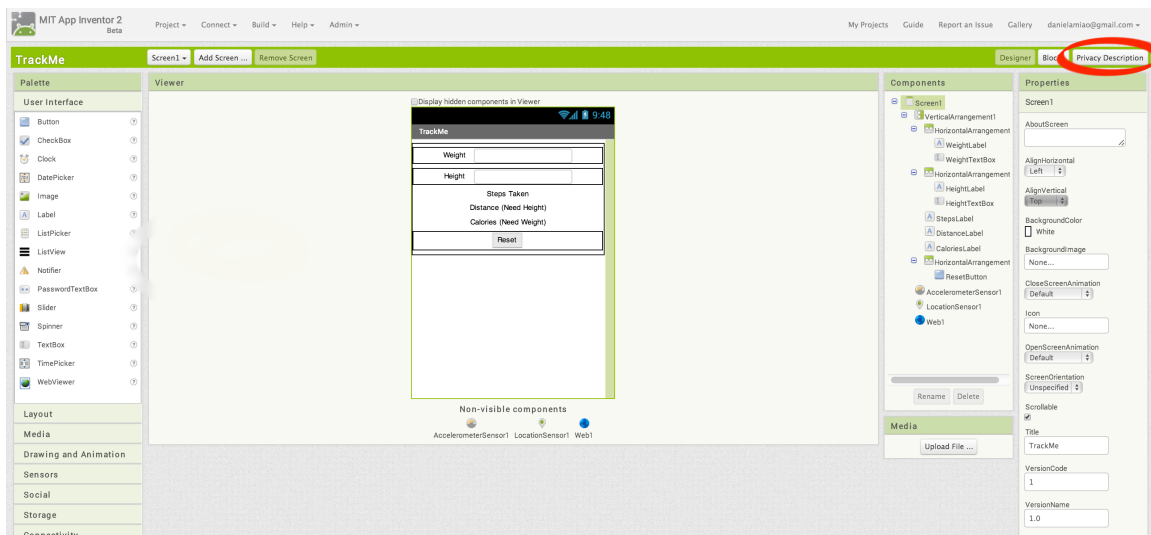


Figure 4-1: Location of privacy description button to access PrivacyInformer

4.2 Design & Implementation

In this section, we present the overall system design of PrivacyInformer, as well as stages of implementation that enable automatic production of the privacy description for App Inventor projects. All actions are initially triggered by a user option

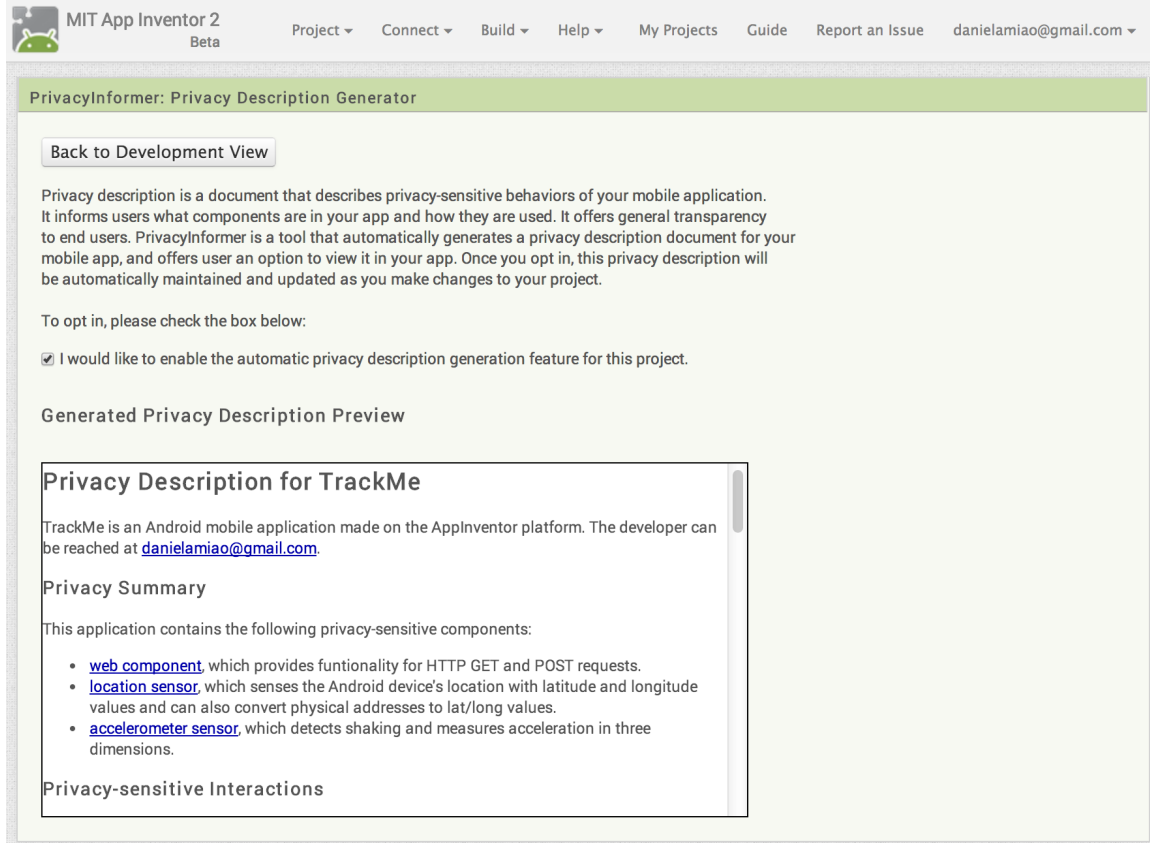


Figure 4-2: Sample view of privacy description editor for Andrew’s application

presented through the App Inventor interface, as indicated in Figure 4-3. For each App Inventor project, the user may access the PrivacyInformer tool by clicking on a button, then enabling the PrivacyInformer functionality via the subsequent screen. Upon opting into the automatic privacy description generation feature provided by PrivacyInformer, it ensures that a privacy description is always generated and packaged into the mobile app. Whenever PrivacyInformer wishes to produce a privacy description, it goes through 3 main stages of operation: extracting project source files, analyzing source files and packaging generated descriptions in the Android application. The interactions and data flows contained in these stages are summarized in Figure 4-4.

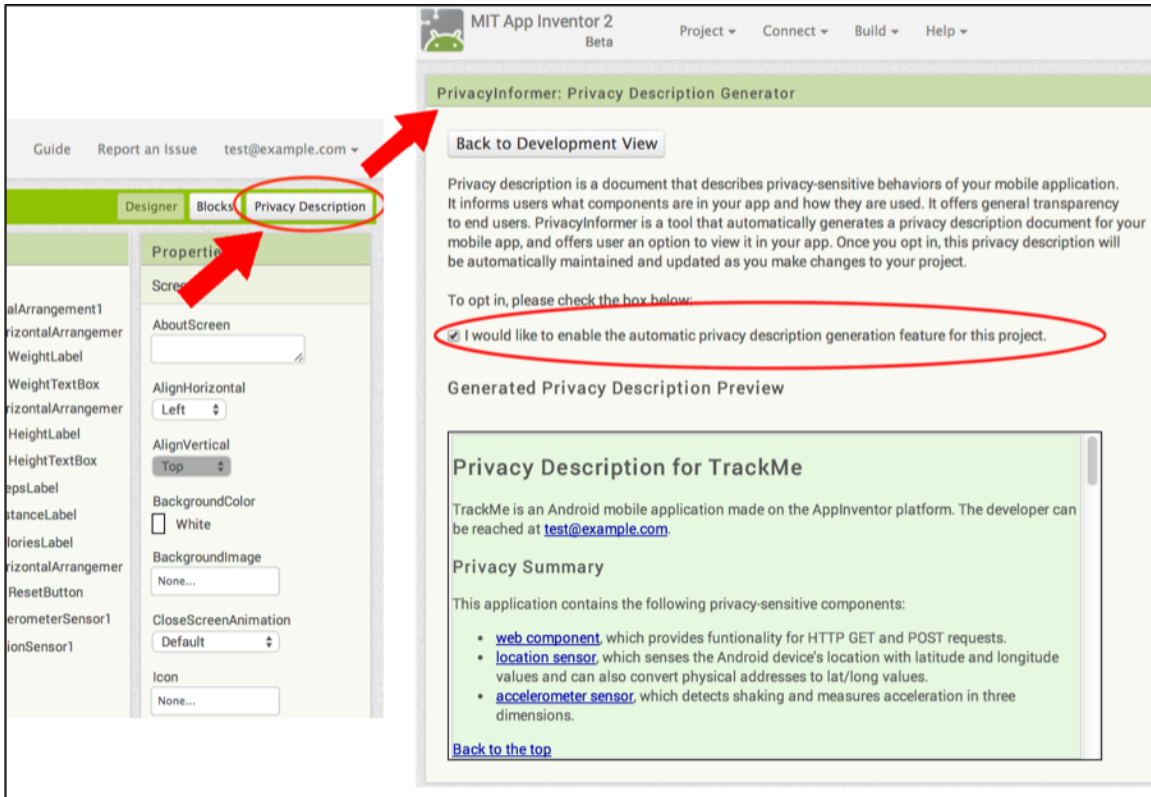


Figure 4-3: Sample workflow of user accessing PrivacyInformer

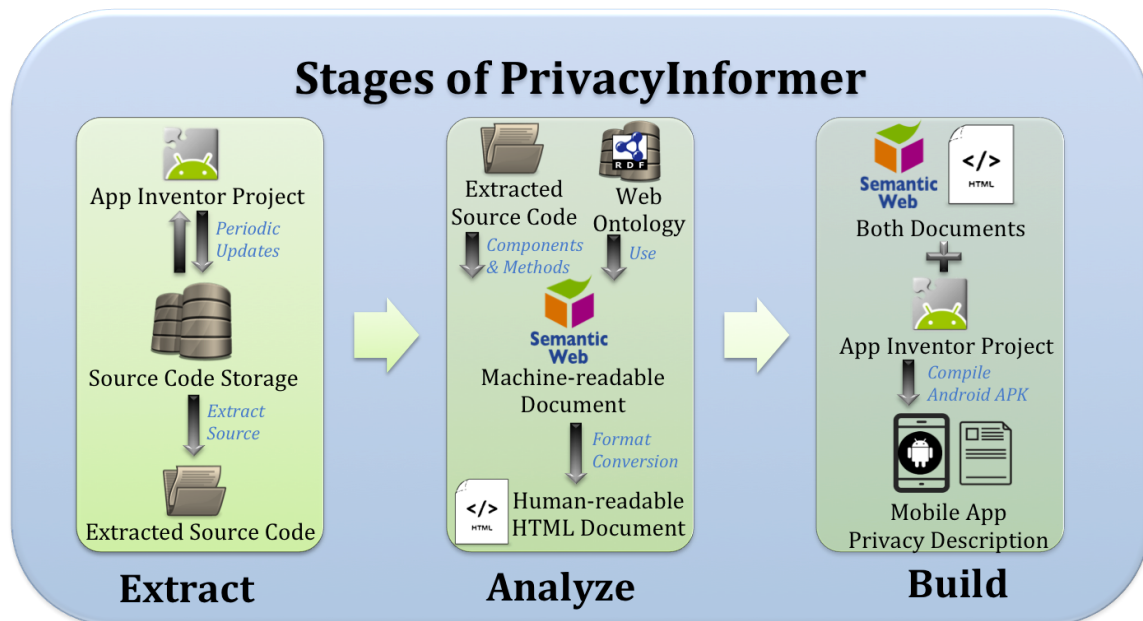


Figure 4-4: General structure of PrivacyInformer

4.2.1 Extract

Upon enabling the PrivacyInformer functionality, a corresponding configuration parameter is marked as enabled, which triggers privacy description generation whenever the App Inventor user wishes to preview the description, or whenever the mobile app is built. PrivacyInformer's first performs a project source code extraction, which is illustrated on the left in Figure 4-4. App Inventor projects are supported by Google App Engine as the storage backend for all project-related information. As users make changes to their project in the frontend, periodic updates are sent to the backend to ensure the most recent changes are backed up. Once PrivacyInformer is enabled, it performs such an update from the frontend to the backend, to ensure the storage contains the most accurate source code regarding the project. Then, PrivacyInformer extracts 2 relevant project files: a JSON (Javascript Object Notation) file containing a list of components used, corresponding to the Designer View of App Inventor, as shown in Figure 4-5, as well as an XML (Extensible Markup Language) file containing methods used by these components, which corresponds to the Blocks View in App Inventor (illustrated in Figure 4-6). These extracted source files are then parsed by PrivacyInformer using standard JSON and XML parsers, preparing for the next stage of processing.

4.2.2 Analyze

Given the 2 extracted source files from the "Extract" stage, PrivacyInformer continues on to the analysis stage. First, it studies the list of components used in the current App Inventor project, as delineated in the JSON source file. For each component that is considered "privacy-sensitive", PrivacyInformer includes a short list of component functionality in the generated privacy document for the mobile app. For purposes of PrivacyInformer, we have categorized App Inventor components to be "privacy-sensitive" if the component is capable of gathering or distributing user data. A list of such components can be found in Figure reffig:privacysensitive. For each privacy-sensitive component, "privacy templates" are pre-generated in Linked Data

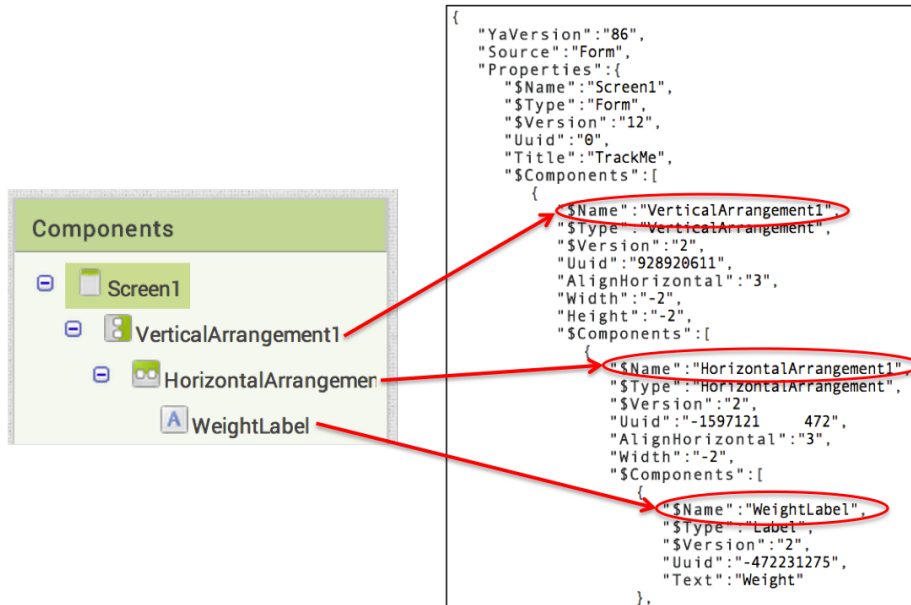


Figure 4-5: Example of a source JSON file containing Designer view information

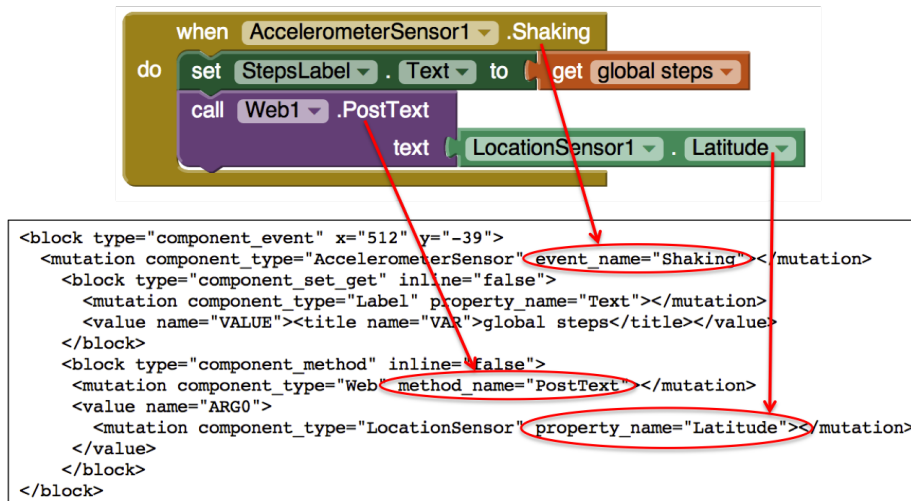


Figure 4-6: Example of a source XML file containing Blocks view information

format, containing descriptions of the component’s functionality. An example of such a privacy template can be seen in Figure 4-8, describing basic functionality of the Location Sensor component in App Inventor. In order to support terminologies used in App Inventor component descriptions, as well as general Android-related features, we created new web ontologies for App Inventor and Android, respectively. Snippets of these ontologies can be found in Appendix A.

All web ontology files and privacy templates for privacy-sensitive App Inventor

Media	Sensors	Social
Camcorder ?	AccelerometerSensor ?	ContactPicker ?
Camera ?	BarcodeScanner ?	EmailPicker ?
ImagePicker ?	LocationSensor ?	PhoneCall ?
SoundRecorder ?	NearField ?	PhoneNumberPicker ?
Storage	Connectivity	Sharing ?
File ?	ActivityStarter ?	Texting ?
FusiontablesControl ?	BluetoothClient ?	Twitter ?
TinyDB ?	BluetoothServer ?	
TinyWebDB ?	Web ?	

Figure 4-7: A list of App Inventor components that are considered privacy-sensitive

```
# Component Description
location:LocationSensorComponent
  rdfs:label "location sensor";
  rdfs:subClassOf ai:Component;
  ai:description "senses the device's location with latitude and longitude";
  ai:category ai:Sensors;

# Privacy Implication
  ai:dataUsage ai:DetectLocation;
  ai:purpose ai:DataGathering;
  ai:requires android:CoarseLocation, android:FineLocation .
```

Figure 4-8: Example of a privacy template for the location sensor

components are stored on a public web server, so PrivacyInformer can access them dynamically during the analysis stage. This provides additional flexibility for future component updates, as changes to the templates can be managed independent of the core analysis function of PrivacyInformer.

Upon accessing the privacy templates, PrivacyInformer imports relevant descriptions into the generated privacy document for the mobile app. For instance, if the Accelerometer Sensor component is used in an App Inventor project named “TestProject”, then contents of its corresponding privacy template will be copied and included in the privacy description for “TestProject”. This is repeated for all privacy-sensitive components used in “TestProject”, hence at this stage, the privacy description of “TestProject” is similar to a concatenation of privacy templates.

Next, PrivacyInformer begins analyzing the XML source file containing the list of all component methods used in the project, which forms the complete programming

logic for mobile app behavior. Specifically, we are interested in *interactions* between privacy-sensitive components, since such interactions are likely to leak user data externally. As seen in Figure 4-6, data is captured in nested “block” tags, hence all interactions fall under a parent-child relationship in XML format. In the figure, the Accelerometer Sensor component event “Shaking” triggers an action by the Web component method “PostText”, which performs an HTTP POST operation in the mobile app. Such an interaction is captured by a pair-wise relationship in the XML source file, where the Accelerometer Sensor event “Shaking” is a parent of the Web method “PostText”, as already illustrated in Figure 4-6.

By using the same line of reasoning, PrivacyInformer iteratively parses all parent-child relationships and records all pair-wise interactions that occur between privacy-sensitive components. In recording such interactions, a new ontological term is introduced name “`ai:connectsTo`”. It is a simple term that preserves the directional order of the interaction between components, so “`Accelerometer:Shaking ai:connectsTo Web:PostText`” means the Accelerometer Sensor shaking event is a parent of Web component method PostText. Next, for each recorded interaction, PrivacyInformer further classifies it as either “potentially privacy leaking” or otherwise. A particular interaction is “potentially privacy leaking” if there is a component method involved that is capable of distributing data externally. For instance, any interaction involving the Web component’s PostText method, which is in effect an HTTP POST, will be considered potentially privacy leaking because it is communicating with external web entities. While not all interactions captured are potentially privacy leaking, we included all of them in the mobile app’s privacy description because of their privacy-sensitive nature. For example, users may be interested in finding out that their mobile app is collecting their location every time the phone shakes, despite the data not being immediately transmitted off the phone.

Finally, after the complete privacy description has been generated in Linked Data format, PrivacyInformer converts the document into human-readable, HTML format as well, shown in the middle of Figure 4-4. Since Linked Data uses the Resource Description Framework (RDF) structure of “triples”, all statements in the privacy

document are in the form of subject-predicate-object. During the format conversion, each RDF triple is translated into 1 HTML bullet point such that all the information are expressed succinctly. To accomplish this, the ontological term “`rdfs:label`” is used, where it indicates the human-friendly version of the subject. For instance, an excerpt of Accelerometer Sensor’s privacy template, may look as follows:

`Accelerometer:Shaking`

```
rdfs:subClassOf ai:ComponentEvent;
rdfs:label ''accelerometer detects device shaking'' .
```

PrivacyInformer reasons through the above privacy template and extracts the label for Accelerometer Sensor’s shaking event, and in all triples where “`Accelerometer:Shaking`” is found, PrivacyInformer will automatically translate it to “accelerometer detects device shaking”. This process is repeated for all terms contained in the triple, as illustrated in Figure 4-9. Note that in this figure, the term “`ai:ConnectsTo`” translates to different HTML text, depending on the context in which it is used. This is an additional layer of reasoning in the format conversion process, where the label for `ai:ConnectsTo` differs depending on whether what term precedes it. Once PrivacyInformer iteratively translates all triples in the machine-readable privacy description, it produces the human-readable HTML version of the privacy description for the App Inventor user to preview, as already seen in Figure 4-3.

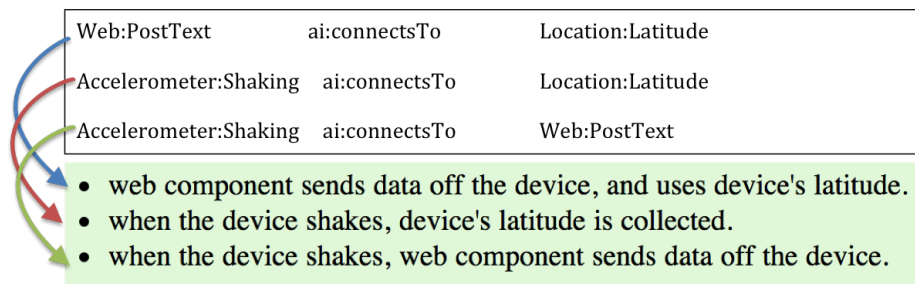


Figure 4-9: Sample conversion from Linked Data format to HTML format

4.2.3 Build

As long as the App Inventor user has opted into the PrivacyInformer feature, both machine-readable and human-readable versions of the privacy description will be automatically generated for the mobile app. Upon building the application, PrivacyInformer proceeds to package both privacy descriptions into the downloaded Android application package (APK) file, which allows the app to be installed on a physical device. In order to do so, however, the files must be first sent to the build server handling compilation of Android applications. The App Inventor build server is usually hosted on a separate machine from the one that hosts the App Inventor project front-end, hence communication between the 2 are completed via simple HTTP requests. Modifications to the existing implementation of App Inventor were done in order to include the privacy description files into the zip file object that is sent to the build server. Additional modifications are then needed on the build server in order to compile those files into the Android application.

The build server is initially responsible for receiving zipped project files from the App Inventor front end, then compiling an installable Android APK from the source files received. In order to enable PrivacyInformer, modifications were completed in Java to allow the compiler to recognize and accept the privacy description files as a part of the project source as well. Upon unzipping the source files, both versions of the privacy description file are packaged into the Android APK under the “assets” directory. In particular, the machine-readable version is stored under file name “privacy.ttl” and human-readable version stored as “privacy.html”. Finally, an additional “View Privacy Description” option is inserted into the mobile app’s Android menu, so users can read contents of the attached “privacy.html” file from inside the application. An example of this final addition to the user interface can be seen in Figure 4-10.

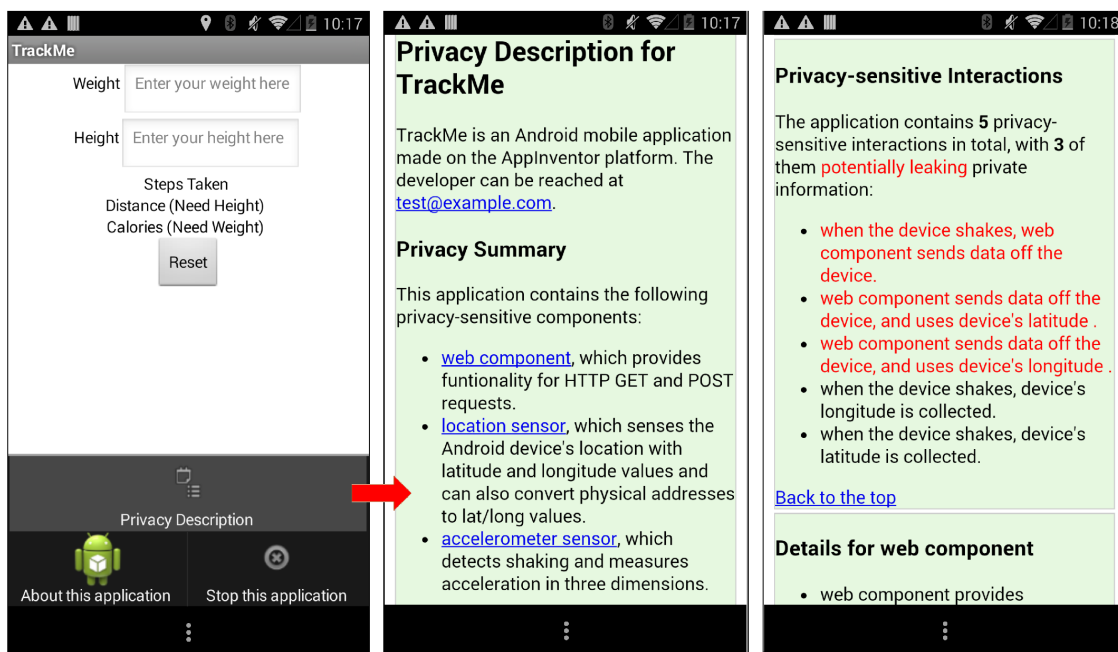


Figure 4-10: Mobile view of the generated privacy description, with potentially data leaking actions highlighted in red

Chapter 5

Evaluation

PrivacyInformer is a tool primarily designed for App Inventor users, who are mobile app developers. Given that the target audience is specifically App Inventor users, initial evaluation of the tool involves soliciting feedback from such users that are familiar with the App Inventor interface. Two main stages of evaluation are presented in this section, first an exploratory user study that was conducted via an online questionnaire, specifically used to gauge overall reception of the tool; next, a few in-person interviews were conducted at the annual App Inventor Summit, where users are given a chance to test PrivacyInformer hands-on and provide more in-depth feedback afterwards.

5.1 Exploratory User Study

Upon completing design and development for PrivacyInformer, an exploratory user study was conducted in order to gain some initial feedback and evaluation on the system. In order to do so, we targeted existing App Inventor users that already familiar with the online platform. Feedback was collected via an online questionnaire, and for feedback discussion in this paper, identities will be kept anonymous for confidentiality purposes.

5.1.1 Methodology

The App Inventor platform was originally created to help educators popularize mobile application building, so many existing users remain novice app developers. We wanted to focus our user study on more advanced developers that were making publicly available apps, and subsequently could have an impact on mobile app privacy. In order to seek out these users, a web crawl over Google Play Store was performed to obtain a list of App Inventor apps with the most downloads, along with publicly available developer emails. To further narrow our user study to a manageable size, we filtered the list of apps to retain only those that had at least 10,000 downloads, postulating such apps have non-trivial functionality given the wide reception. This filter resulted in a total of 323 unique app developer email addresses. Subsequently, an email with an online questionnaire link was sent to each app developer, explaining the purpose of the study and soliciting their feedback. The questionnaire contained 7 questions in total, split into 2 major sections: (1) User Background, which asks about the developer’s experience with App Inventor and mobile privacy in general; and (2) PrivacyInformer Feedback, which introduces the tool by presenting screen-shots and asks for specific feedback on usefulness of the tool. A copy of the questionnaire can be found in Appendix B.

The initial email was sent out to app developers on June 2nd, 2014, with subsequent weekly reminders to complete the questionnaire throughout the month of June. Out of the 323 developer emails, 79 were automatically rejected by mail servers, and a total of 31 questionnaire responses were collected as of June 31st, 2014, at which time the questionnaire was closed. We postulate the main reason for the low response rate is lack of incentives, since no compensation was provided for completing the questionnaire. Other potential reasons include non-monitored emails, lack of interest, and expected low response rate given a study that cites typical online survey response rate to be around 11% [32]. Nevertheless, we find the collected responses to be valuable in evaluating PrivacyInformer at an exploratory stage and 31 responses keep the manual analysis of comments/suggestions at a manageable level for the authors.

5.1.2 User Background

Of the 31 respondents, over half (52%) have at least 2 years of experience with App Inventor, which was only launched just over 3 years ago. 23% have between 1 and 2 years of experience, 6% have between 6 months and 1 year of experience, and 19% have less than 6 months of experience. We also asked these App Inventor developers how important they thought privacy descriptions are for mobile apps, given a Likert scale with choices “Very Important”, “Important”, “Moderately Important”, “Of little importance” and “Not important”. An overwhelming portion (94%) thought it was at least moderately important to have a privacy description for their apps. However, we want a more in-depth understanding of how their experience with mobile app development relates to their perception of privacy in the mobile apps market. In Figure 5-1, we plotted a radar graph that correlates the amount of App Inventor experience with how important developers perceive having a privacy description. Each experience category is illustrated in a different color and shape, whereas each vertex of the pentagon represents an “importance” level for mobile app privacy descriptions. The concentric pentagons indicate percentage of respondents that responded with that certain “importance” level. For instance, for respondents with less than 6 months of App Inventor experience (marked in blue squares), 50% thought privacy descriptions are moderately important in mobile apps, and the other 50% thought they were important.

From Figure 5-1, we can see that there is a weak correlation between amount of experience with App Inventor and perception of privacy description importance for mobile apps. Most notably, a small percentage of developers with 2+ years of App Inventor experience (marked in purple circles) thought privacy descriptions are “not important”, or “of little importance”. In addition, the group with 6 months to 1 year of experience had the strongest support for privacy descriptions, marking them as either “important” or “very important”. These results are interesting to us, and suggest that newcomers to the mobile app development community may not understand the importance of privacy at first, but very quickly realize its value and

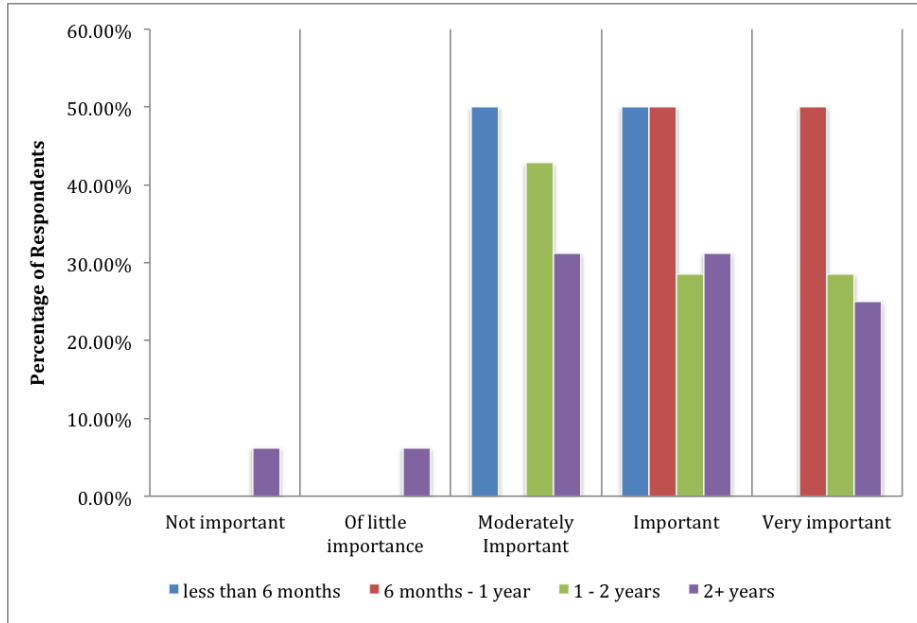


Figure 5-1: Correlation of App Inventor experience with perception of privacy descriptions

view privacy descriptions as more important than veterans. This could be attributed to the fact that new developers find themselves more relatable to end users, and think in their shoes a lot more than seasoned mobile developers who may have forgotten concerns of end users. Of course, we recognize that the sample size in this study is not large enough to draw any conclusive relationships between app development experience and privacy description perception. We offer the results here simply as an observation with our own conjectures, with the hope that more rigorous user studies can be conducted in future research.

5.1.3 Feedback on PrivacyInformer

After presenting PrivacyInformer as an added feature of App Inventor in the form of screen-shots, respondents were asked whether they would use PrivacyInformer when using App Inventor to develop mobile apps. Of the 31 responses, a majority (74%) said “yes”, and the rest answered “no”. When asked to provide reasons for not using PrivacyInformer, a few users mentioned that App Inventor is not longer their primary choice of tool for developing mobile apps, which detracts from the relevance of their

responses given PrivacyInformer is made specifically for the App Inventor platform. Others thought the Google Play Store already provides the information in the form of app permissions, which is a fair explanation but the disadvantages of this approach were discussed previously in Section 3. Finally, a few respondents thought privacy description is not a concern currently, and other components are more important to the App Inventor platform. This was an interesting comment, and prompted us to attempt correlating perception of the importance of privacy descriptions with reception of the PrivacyInformer tool. In Figure 5-2, the percentage of respondents that picked a certain importance level for mobile app privacy descriptions are plotted, split into 2 separate categories “would use PrivacyInformer” and “would NOT use PrivacyInformer”. The illustration suggests that there is indeed a correlation between privacy

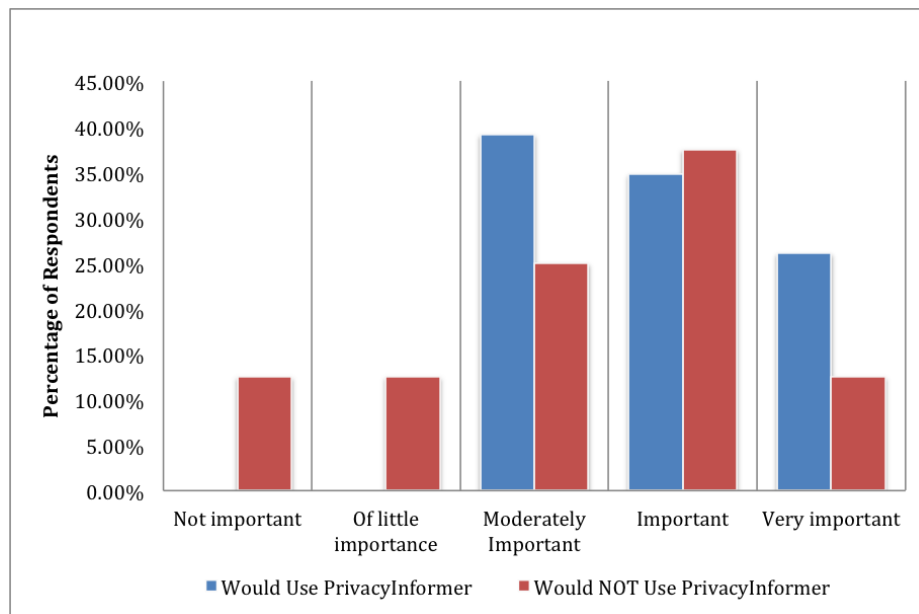


Figure 5-2: Correlation of perception of privacy descriptions with PrivacyInformer reception

description perception and PrivacyInformer reception. Around 20% of developers who would not use PrivacyInformer thought privacy descriptions are “not important” or “of little importance”, while no developers who would use PrivacyInformer answered either of those categories. Moreover, a much larger percentage of developers willing to use PrivacyInformer marked privacy descriptions as “very important”, and a larger

percentage of them also answered “moderately important”. It is interesting, however, that in the category “important” for privacy descriptions, there is about the same number of developers who would use PrivacyInformer as those that would not. This is consistent with our survey responses, which suggest there are various reasons for choosing to not use PrivacyInformer, and perception of privacy descriptions is only one such factor, but an obvious cause nevertheless.

Finally, we asked developers why they would choose to use PrivacyInformer, and many commented on its convenience and time-effectiveness. Specifically, User 25 noted that “an automatic tool would reduce the time to create the privacy description”. User 8 suggested that “it would help me write privacy description way faster”. To capture this particular benefit of PrivacyInformer quantitatively, we also asked the developers to estimate how long it would take them to come up with privacy descriptions for their mobile apps, with and without PrivacyInformer. A sample privacy description is provided in the questionnaire in order to clarify the contents of a typical privacy description. The results are presented in Figure 5-3. Results are generally positive, where most people thought writing a privacy description would take them between 1 to 2 hours without PrivacyInformer, and only less than 15 minutes with PrivacyInformer. The graph clearly illustrates a significant drop in amount of time spent devising the privacy description with the help of PrivacyInformer. In fact, no developer thought using the PrivacyInformer would increase the amount of time they spend on the privacy description.

5.2 In-person Interviews

At the annual App Inventor Summit held at MIT, App Inventor users are invited to attend talks on the future of App Inventor from all over the world. PrivacyInformer was presented in one of the talks, as well as in the poster session. Users were asked to try out PrivacyInformer on the spot and give in-depth comments or suggestions on the tool. Overall, reception of PrivacyInformer is positive, with majority of the surveyed users willing to try it out once it is released to the official App Inventor

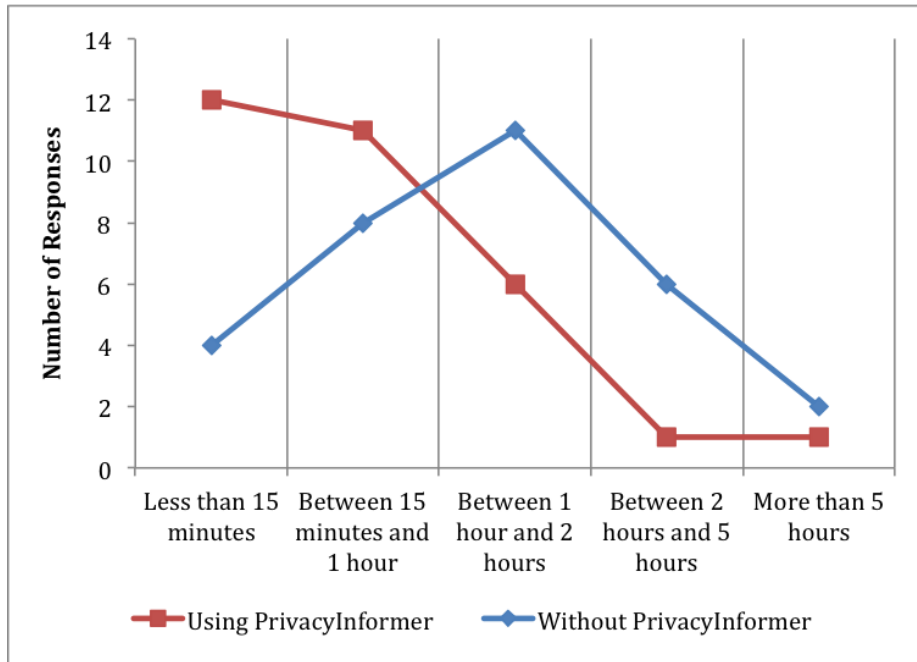


Figure 5-3: Amount of potential time spent on devising privacy descriptions with and without PrivacyInformer

platform. More importantly, we have some evidence that it would indeed save developers time from writing their own privacy descriptions manually. App Inventor users noted that in most cases manually writing privacy documents could take up to hours, whereas PrivacyInformer produced the document in seconds. One user working in education noted, in particular, that his students often have trouble understanding the exact behaviors of App Inventor blocks, and that PrivacyInformer could help inform them of such. In addition, some suggestions for improvement are received, including modifications to aesthetics and wording of the description itself. All in all, these preliminary interviews serve as an initial acknowledgement of the usefulness and effectiveness of PrivacyInformer.

Chapter 6

Discussion

Given that PrivacyInformer is still in its first phase of development, many improvements can be made, with help of experts from various fields. This work is only the beginning to many more interesting research projects in the domain of mobile privacy. As a tool that bridges the gap between technical development (analyzing source code) and policy construction (producing privacy documents), PrivacyInformer has the potential to impact both the mobile app development community and the regulatory realm.

In this section, the author will provide discussions with regard to the general impact of PrivacyInformer to the mobile apps community; how it contributes to the development process of mobile apps by providing privacy information in real-time to developers, how it helps shape the way privacy information is presented to end users, and how it could jump-start a new web standard for machine-readable mobile privacy documents. Moreover, the privacy document provided by PrivacyInformer serves as a basis for creating privacy policies for mobile apps and this is discussed as well. Finally, it is possible that regulatory agencies such as the Federal Trade Commission (FTC) could benefit from the use of technical tools such as PrivacyInformer in order to explicitly govern usage of privacy policies in the mobile apps market.

6.1 Impact of PrivacyInformer in the Mobile Apps Market

The design of PrivacyInformer generated many interesting research questions in the general areas of mobile privacy and human-computer interactions. As with all tools that statically analyze code, PrivacyInformer is not able to infer all information that users may be interested in. The most prominent example is *purpose* of data collection, which is a parameter difficult to extract accurately from studying the application’s source code alone. In this case, it seems sensible to allow human intervention in the privacy description generation process, allowing developers to customize the generated document before packaging it with the APK. However, the main goal of PrivacyInformer is to provide a quick, easy and automatic way to produce privacy documents with no human input requirement. Hence, in this first iteration of our implementation, we have decided to opt for a “one-stop” design, where developers opt-in once manually, and the privacy description generation process is automatic during all subsequent compilations. In addition, data usage purpose is fickle – application developers may collect data at different times with changing purposes. As a result, it is likely that developers will put down generic data purposes to account for changes in the future, and we have incorporated such data purposes as defaults in PrivacyInformer templates. That being said, it may still be helpful to provide a fixed set of options for developers to choose from at the time of generating the privacy description, and this work will be considered in future iterations of PrivacyInformer.

A second interesting area of research PrivacyInformer touches upon is presentation of privacy information to mobile application users. It is known that most users do not read long privacy documents, and even if they do, little useful information can be extracted from long tracts of text [24]. Much research has been conducted on effective ways of communicating privacy-related information and visualization techniques that make the information comprehensible to end users [23]. As already mentioned in Section 3, PrivacyInformer draws upon such pieces of work and produces a human-readable version of the privacy description that is succinct and clear. However, this is

not the primary focus of our work. Instead, we build flexibility within PrivacyInformer for the privacy description to be rendered into many different formats. Since we use an intermediate, machine-readable data structure to capture important privacy-related information and publish it as Linked Data in the application’s meta-data section, one can easily use this to further process, as well as visualize the privacy description in creative ways. Many companies have already provided pre-generated graphics related to privacy such as the Mozilla Privacy Icons ¹, hence these can be imported to enhance aesthetics of the human-readable privacy description.

Finally, an additional benefit of including the privacy description in Linked Data format is standardization within the mobile applications market. Since Open Linked Data is a completely open and public standard, mobile apps created via the standard Android Software Development Kit (SDK) can follow the same paradigm as applications created using App Inventor. In fact, all web ontologies created to represent concepts of App Inventor and Android are published online and hosted locally at MIT². Developers not using the App Inventor but still wanting to provide privacy-related information in their mobile apps, can simply create privacy descriptions in Linked Data format with the help of these open web ontologies, possibly using existing privacy descriptions of App Inventor apps as templates.

6.2 Privacy Description as a Legal Document

As mentioned previously, the document reproduced by PrivacyInformer is named a “privacy description” because it simply describes the privacy behaviors of the mobile app by studying the source code only. A distinction is specifically made with the term “privacy policy” because privacy policies have gained a somewhat notorious reputation for being long, onerous and incomprehensible. According to researchers at Carnegie Mellon University, it would take an average person 250 working hours, or 30 working days to fully read privacy policies of the websites they visit in a year [24]. The reason

¹https://wiki.mozilla.org/Privacy_Icons

²Ontologies can be found at <http://dig.csail.mit.edu/2014/PrivacyInformer>

for the length of privacy policies can be attributed to legal experts trying to lower the risk of litigation for companies and developers, hence often using legal jargon to fully qualify statements, obscure true meanings and allow plenty of room for exceptional circumstances. This is an unfortunate phenomenon that will be addressed later in this section. On the contrary, the privacy description produced by PrivacyInformer has not been advised by such legal experts, mainly due to timing constraints and shifted focus instead on the effectiveness of the document. Indeed, it was the author's primary goal to make an easy-to-understand document for users to quickly gain a grasp of privacy implications of mobile apps. Filling the document with qualified legal terms will certainly decrease its comprehensibility, as illustrated in the study by McDonald et al. [23].

Despite of the differences between privacy policies and privacy descriptions, one cannot say privacy descriptions carry no legal liability. Any piece of public privacy statement included with a mobile app can become the basis for litigation, if developers are found to be violating its terms. After all, users want to see that the application is behaving faithfully to its privacy description. This extra layer of accountability is what makes privacy policies and by extension, privacy descriptions, useful in the first place [21]. Indeed, their importance can be evidenced in the increased popularity of online privacy generators, including iubenda³, the TRUSTe generator⁴, and the Docracy Blog which introduced an open source privacy policy template for web and mobile developers⁵. At the same time, regulatory entities have also been keen on increasing usage of privacy policies in mobile apps. The Attorney General of California made it clear that its Online Privacy Protection Act would be enforced on mobile apps (CalOPPA). In fact, California's Department of Justice set up a Privacy Enforcement and Protection Unit in July of 2012 to ensure its laws were being followed [2]. While this may be only applicable in California currently, it's actually a call for compliance for anyone possibly targeting Californians, leading to potential changes in laws of other states. Both the FTC and NTIA have released developer

³<http://www.iubenda.com/en>

⁴<http://www.truste.com/free-mobile-privacy-policy/>

⁵<http://blog.docracy.com/post/27931026976/an-open-source-privacy-policy-for-mobile-apps>

guidelines on creating privacy policies for mobile apps [17]. The White House released a new Consumer Privacy Bill of Rights in February 2012, urging the Congress to grant the FTC more authority to enforce privacy rights of consumers, including in the mobile apps market [29]. Companies like Path and Delta have already been charged or fined because of non-compliance with privacy laws [18]. Such penalties act as a warning for the mobile development community as a whole, urging developers to be more privacy-friendly, discouraging overly intrusive apps from appearing and fostering a more privacy-aware market overall.

However, while having a system of privacy policy and privacy law enforcement bring about benefits described above, developers still lack motivation to create one. After all, there are no clear regulations dictating the necessity of having a privacy statement of any form, nor are there policies specifying the exact content, style and format of such privacy policies. There is no reason to include a privacy statement, which could become the basis for litigation, when one is not required. For developers that do want to publish privacy policies, they often have to be written to be lengthy, burdensome, and filled with carefully chosen legal terminologies. While unfortunate, this phenomenon is not surprising - developers are risk averse and want to avoid being penalized by exceptional cases. For instance, if PrivacyInformer produces a privacy description that reports user location to be collected and used locally only, but the location is accidentally leaked in a debug report to the developer, when the mobile app crashes, the developer may be penalized for violating the terms of the privacy description. In this case, the developer may have to suffer both financially and emotionally for an exceptional case that was originally overlooked. Here, it seems appropriate for the law to create an area of safe harbor for mobile developers to safely present privacy statements in good faith, and be exempt from litigation when accidental violation occurs that was not of malicious intentions. In this way, developers may feel more motivated to use a tool like PrivacyInformer to attach privacy descriptions to their apps, without fearing unpredictable penalties. Of course, it is difficult to suggest specific clauses that could create such a safe harbor without extensive legal knowledge, but given that the FTC currently investigates deviations

from privacy policies, it is also in a position to increase burden of proof, such that only deliberate and malicious attempts of privacy violations are penalized. There is hope that a properly formulated regulation can help jump start a healthy, privacy-friendly mobile market, where developers have no qualms making privacy claims about their apps.

6.3 PrivacyInformer as a Regulatory Mechanism

In recent years, as the discussion on mobile app privacy issues has become more and more heated, multiple legislation were passed in order to resolve the situation. However, most existing regulatory actions have done little in mitigating privacy concerns. As described in Chapter 2, overly rigid regulations could have a detrimental effect on this highly dynamic and innovative market. As a result, regulatory agencies have taken a softer approach by releasing guidelines instead of inflexible laws. On September 5th, 2012, the Federal Trade Commission (FTC) published its first guide containing a set of non-binding guidelines to mobile app developers [29]. This is the first promising step towards setting proper mobile application standards, but it is far from enough. The content of this published guide is generic, non-compelling and lacks novelty. The guidelines are filled with ambiguous wording and concepts that require specific clarifications before it can facilitate the adoption of guidelines by mobile app developers. For instance, the FTC urges developers to “collect information only as necessary ...limit access to information on a need-to-know basis ...” Yet, whether or not collection of data is necessary is subjective and depends on different points of view. To a user, access to personal calendar does not seem necessary for a flight-tracking application, while app developers may deem the same information “necessary” for financial viability of the app, since user data has advertising revenue value. Overall, subjective and ambiguous guidelines are as ineffective as having no guidelines at all. If privacy issues are to be rectified, more detailed and actionable regulations are needed.

One influential party in the mobile apps market is the app platform owner, in-

cluding juggernauts such as Apple, and Google, and to a lesser extent Amazon and Microsoft. Currently, most platforms have active awareness of privacy issues that exist in the market, and have revealed different methods of dealing with the problem. In December 2013, Google accidentally released a fine-grained privacy access control menu “App Ops” originally only intended for internal development use [22]. This menu was quickly taken out in subsequent Android updates, but its existence had already been lauded by many users as well as the Electronic Frontier Foundation [9]. More recently, Apple announced new privacy features in the upcoming iOS 8 release, citing abilities to give context to privacy by notifying users of personal data access while app is not in use (background processes) [35]. Even though these are small steps towards addressing privacy concerns, it shows the increasing interest app platforms have in protecting their users’ privacy from intrusive apps. This creates an opportunity for regulatory agencies, such as the FTC, to focus on managing a small number of app platforms, rather than targeting the mobile app development community as a whole. Next, we examine how PrivacyInformer can play an important role in facilitating actionable guidelines or regulations for these app platforms.

PrivacyInformer opens the door for much research in the area of increased privacy accountability of mobile apps at the app distribution level. If regulation specifies privacy description processing at the app platform level, for example Google Play Store in Android, PrivacyInformer can essentially act as an enabling mechanism for users to select apps based on their customized privacy preferences. Given the machine-readable privacy description attached with each App Inventor app, one can easily reason over it and produce a matching algorithm to present users apps that best match their privacy preferences. In the future, the concept of PrivacyInformer could be extended beyond the App Inventor framework, and enable mobile app filtering across different platforms as well. Moreover, the same privacy description could be extracted and processed to generate corresponding rules that govern data access at the operating system level. If enforced by regulatory agencies, app platforms could be required to host systems like the Open Mustard Seed (OMS) [1] trust framework, where all third-party applications must explicitly request data from the user’s personal

data store. All this requires integration of privacy description processing at the app platform level, hence compliance from companies that own these platforms is critical. By referring to a specific tool such as PrivacyInformer, the FTC is able to propose unambiguous and directly actionable regulations to govern how app platform owners are mitigating mobile privacy issues.

Chapter 7

Conclusion

In this thesis we attempted to alleviate the current privacy issues in the mobile applications market. We introduced a tool as a part of the MIT App Inventor named “PrivacyInformer”. It provided a quick and automatic way to help developers produce privacy descriptions for their mobile apps. By analyzing the source code of the application at compilation time, we were able to produce a short and useful description of privacy-related behaviors of the application to mobile app users. We did so in both human-readable and machine-readable formats, hence opening up many future research opportunities to improve the outlook of mobile privacy, both in areas of enhancing visualization of privacy information and matching user preferences against appropriate mobile apps.

We also conducted an exploratory user study to gauge interest from the mobile app development community, and received positive feedback overall. Many developers have expressed interest in increasing transparency of their apps by presenting such privacy documents as produced by PrivacyInformer. Others have found the tool to be useful in keeping them fully informed of their apps’ privacy implications, which may not have been immediately obvious. Most importantly, developers liked using the tool to promote their apps as privacy-friendly, given concern for privacy has been rising in the mobile apps market.

Lastly, we presented discussions on policy impacts of PrivacyInformer, as it may open the door for new regulations in governing inclusion of privacy policies in mobile

apps. An automatic privacy document generation tool can serve as a starting point for legal experts to quickly obtain information about a mobile app when devising privacy policies. In the future, regulatory agencies could look to leverage the tool when enforcing privacy laws, as well as investigating deviations from privacy policies.

Appendix A

Sample Ontology for Android and App Inventor

```
#  
# Top-level classes for AppInventor components and component functions  
#  
:Component  
  rdfs:comment "AppInventor are composed of components";  
  rdfs:label "An Appinventor component ";  
  a rdfs:Class .  
  
:PrivacyDescription  
  rdfs:label "Privacy description for mobile application ";  
  a rdfs:Class .  
  
#  
# Properties for the :PrivacyDescription class  
#  
:contains  
  a rdf:Property;  
  rdfs:comment "what the privacy description contains information on";  
  rdfs:label "contains privacy information on ";  
  rdfs:range :Component;  
  rdfs:domain :PrivacyDescription .
```

Figure A-1: Snippet of App Inventor web ontology

```
#  
# Top-level class for all Android components  
#  
:Component  
  rdfs:label "Android component";  
  a rdfs:Class .  
  
#  
# Instances for Android components  
#  
:AccelerometerSensor  
  rdfs:label "accelerometer sensor on the phone";  
  a :Component .  
  
:Calendar  
  rdfs:label "calendar content";  
  a :Component .  
  
:CallLog  
  rdfs:label "call log/history";  
  a :Component .  
  
:Camera  
  rdfs:label "camera component on the phone used to take pictures";  
  a :Component .
```

Figure A-2: Snippet of Android web ontology

Appendix B

Questionnaire for PrivacyInformer

PrivacyInformer: Privacy Description Generator for App Inventor

* Required

PrivacyInformer - Background Survey

Before we begin introducing the PrivacyInformer tool to you, we would like some background on your experiences with App Inventor.

1. How many years of experience do you have with mobile application building on the App Inventor platform? *

- less than 6 months
- 6 months - 1 year
- 1 - 2 years
- 2+ years

Please refer to this list in Question 2.

Media	Sensors	Social
Camcorder ?	AccelerometerSensor ?	ContactPicker ?
Camera ?	BarcodeScanner ?	EmailPicker ?
ImagePicker ?	LocationSensor ?	PhoneCall ?
SoundRecorder ?	NearField ?	PhoneNumberPicker ?
	OrientationSensor ?	Sharing ?
Storage	Connectivity	Texting ?
File ?	ActivityStarter ?	Twitter ?
FusiontablesControl ?	BluetoothClient ?	
TinyDB ?	BluetoothServer ?	
TinyWebDB ?	Web ?	

2. When building mobile apps, how often do you use the above categories of components? *

	Never	Rarely	Sometimes	Often	Always
Media	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Storage	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sensors	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Connectivity	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Social	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. When using App Inventor, were you aware that the above components (in Question 2) either collect data from users, or disseminate such data? *

- Yes
- No

Please provide a few justifications for your above given answer.

For example, if you often collect or disseminate user data, what and why do you collect it?

Please refer to this image in Question 4.

Sample Privacy Description

Privacy Description for TrackMe

This application contains the following privacy-sensitive components:

- [web component](#), which provides functionality for HTTP GET and POST requests.
- [location sensor](#), which senses the Android device's location with latitude and longitude values and can also convert physical addresses to lat/long values.
- [accelerometer sensor](#), which detects shaking and measures acceleration in three dimensions.

Privacy-sensitive Interactions

The application contains the following privacy-sensitive interactions:

- when the device shakes, web component sends data off the device.
- web component sends data off the device, and uses device's latitude .
- web component sends data off the device, and uses device's longitude .
- when the device shakes, device's longitude is collected.
- when the device shakes, device's latitude is collected.

4. In your opinion, how important is it for mobile apps to have a privacy description for users to view? *

A privacy description is a document that describes the privacy-sensitive behaviors of a mobile application, including a list of all components used as well as their respective functions. An example is shown above.

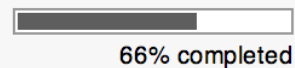
	Not important	Of little importance	Moderately important	Important	Very important
Choose One	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>


5. How long do you think it would take you to write a privacy description for an App Inventor app you have published in the Google PlayStore? *

The privacy description should be in the style of the one shown above.

	Less than 15 minutes	Between 15 minutes and 1 hour	Between 1 hour and 2 hours	Between 2 hours and 5 hours	More than 5 hours
Choose one	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

[« Back](#) [Continue »](#)



Powered by  Google Forms

This content is neither created nor endorsed by Google.
[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

PrivacyInformer: Privacy Description Generator for App Inventor

* Required

PrivacyInformer - Tool Evaluation

PrivacyInformer is a tool that automatically generates privacy descriptions for your apps by analyzing the project. Please read through the information below then complete the survey by telling us how you feel about the tool.

MIT App Inventor 2 Beta

Project Connect Build Help

PrivacyInformer: Privacy Description Generator

Back to Development View

Privacy description is a document that describes privacy-sensitive behaviors of your mobile application. It informs users what components are in your app and how they are used. It offers general transparency to end users. PrivacyInformer is a tool that automatically generates a privacy description document for your mobile app, and offers user an option to view it in your app. Once you opt in, this privacy description will be automatically maintained and updated as you make changes to your project.

To opt in, please check the box below:

I would like to enable the automatic privacy description generation feature for this project.

Generated Privacy Description Preview

Privacy Description for TrackMe

TrackMe is an Android mobile application made on the AppInventor platform. The developer can be reached at test@example.com.

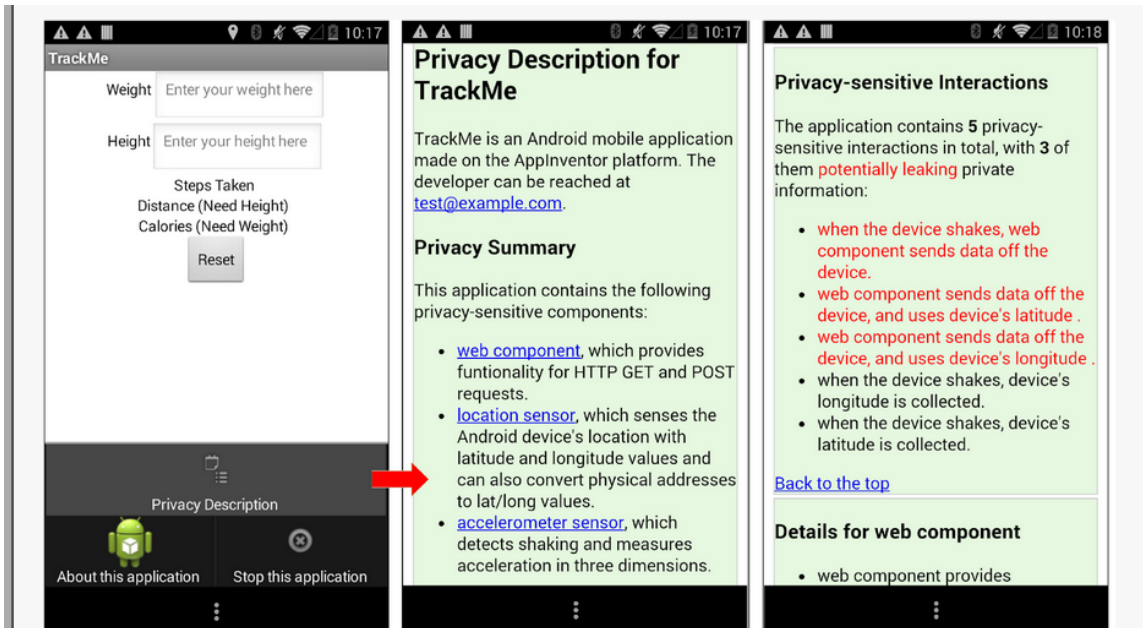
Privacy Summary

This application contains the following privacy-sensitive components:

- [web component](#), which provides functionality for HTTP GET and POST requests.
- [location sensor](#), which senses the Android device's location with latitude and longitude values and can also convert physical addresses to lat/long values.
- [accelerometer sensor](#), which detects shaking and measures acceleration in three dimensions.

[Back to the top](#)

A new "Privacy Description" button is added, leading to a "PrivacyInformer" view where you can enable the automatic privacy description generation feature.



Once enabled, PrivacyInformer will always generate a privacy description and include it in the mobile app. Users can view the privacy description by clicking on the "Privacy Description" menu option.

When building mobile apps using App Inventor, would you use the above described PrivacyInformer tool to automatically generate privacy descriptions for your apps? *

- Yes
- No

Please provide a few justifications for your above given answer. *

For example, if you would not use the PrivacyInformer tool, why not? If you would use it, what prompts you to use it?

Using this PrivacyInformer tool, how long do you think it would take for you to provide a privacy description for an App Inventor app you have published in the Google PlayStore?

*

Less than 15 minutes Between 15 minutes and 1 hour Between 1 hour and 2 hours Between 2 hours and 5 hours More than 5 hours

Choose one



Please provide any general comments you have on the PrivacyInformer tool.

For example, you could comment on its usability "the instructions are too hard to follow" or functionality "I like how easy it is to produce a privacy description with PrivacyInformer"

PrivacyInformer tool is currently available, in development form, at <http://privacyinformer-mit.appspot.com>. The test project is available at: <https://dl.dropboxusercontent.com/u/11685473/TrackMe.aia>. It is highly recommended that you use this project when testing PrivacyInformer. While it is possible to import your own projects, the privacy description may not be fully complete. PrivacyInformer is currently under testing, please report any bugs/problems you run into to dmiao@mit.edu.

Never submit passwords through Google Forms.



100%: You made it.

Powered by
 Google Forms

This content is neither created nor endorsed by Google.
[Report Abuse](#) - [Terms of Service](#) - [Additional Terms](#)

Bibliography

- [1] The open mustard seed project. <http://idhypercubed.org/>.
- [2] Attorney general kamala d. harris announces privacy enforcement and protection unit. <http://oag.ca.gov/news/press-releases/attorney-general-kamala-d-harris-announces-privacy-enforcement-and-protection>, July 2012.
- [3] Theodore Book and Dan S Wallach. A case of collusion: A study of the interface between ad libraries and their apps. In *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*, pages 79–86. ACM, 2013.
- [4] J. Boyles, A. Smith, and M. Madden. Privacy & data management on mobile devices. Technical report, Pew Internet & American Life Project, Washington D.C., 2012.
- [5] Hiawatha Bray. Smartphone apps track users even when shut down. <http://www.bostonglobe.com/business/2012/09/02/smartphone-apps-track-users-even-when-shut-down/yIh3bxEYcLuZ2PC6MyFuMO/story.html>, September 2012.
- [6] Joseph Joo Keng Chan, Kiat Wee Tan, Lingxiao Jiang, and Rajesh Krishna Balan. The case for mobile forensics of private data leaks: Towards large-scale user-oriented privacy protection. 4th Asia-Pacific Workshop on Systems (AP-SYS), 2013.
- [7] Andrew Clark. App inventor launches second iteration. December 2013.
- [8] Lorrie Faith Cranor. P3p: Making privacy policies more useful. *IEEE Security & Privacy*, 1(6):50–55, 2003.
- [9] Peter Eckersley. Awesome privacy tools in android 4.3+.
- [10] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, pages 1–6, 2010.

- [11] William Enck, Damien Ocate, Patrick McDaniel, and Swarat Chaudhuri. A study of android application security. In *USENIX security symposium*, volume 2, page 2, 2011.
- [12] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 627–638. ACM, 2011.
- [13] Adrienne Porter Felt, Kate Greenwood, and David Wagner. The effectiveness of application permissions. In *Proceedings of the 2nd USENIX conference on Web application development*, pages 7–7. USENIX Association, 2011.
- [14] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 3. ACM, 2012.
- [15] Michael C Grace, Yajin Zhou, Zhi Wang, and Xuxian Jiang. Systematic detection of capability leaks in stock android smartphones. In *NDSS*, 2012.
- [16] G. Gross. Lawmaker pushes mobile privacy legislation, September 2012.
- [17] White House. Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy. <http://www.whitehouse.gov/sites/default/files/privacy-final.pdf>, 2012.
- [18] Mike Isaac. Path ceo: ‘we thought we were doing this right’. <http://www.wired.com/2012/02/path-dave-morin-explains-data/>, February 2012.
- [19] Patrick Gage Kelley, Joanna Bresee, Lorrie Faith Cranor, and Robert W Reeder. A nutrition label for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, page 4. ACM, 2009.
- [20] Patrick Gage Kelley, Lucian Cescas, Joanna Bresee, and Lorrie Faith Cranor. Standardizing privacy notices: an online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human factors in Computing Systems*, pages 1573–1582. ACM, 2010.
- [21] Marc Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp 2002: Ubiquitous Computing*, pages 237–245. Springer, 2002.
- [22] Joe Levi. What is app ops, and why did google remove it from android?
- [23] A. M. McDonald and T. Lowenthal. Nano-notice: Privacy disclosure at a mobile scale. *Journal of Information Policy*, 3, 2013.

- [24] Aleecia M McDonald and Lorrie Faith Cranor. Cost of reading privacy policies, the. *ISJLP*, 4:543, 2008.
- [25] NewsCore. Facebook spies on phone users’ text messages, report says. <http://www.news.com.au/breaking-news/facebook-spies-on-phone-users-text-messages-report-says/story-e6frku0-1226282017490>, February 2012.
- [26] K. O’Brien. Data-gathering via apps presents a gray legal area. http://www.nytimes.com/2012/10/29/technology/mobile-apps-have-a-ravenous-ability-to-collect-personal-data.html?_r=1i&, October 2012.
- [27] Future of Privacy. Fpf mobile apps study. <http://www.scribd.com/doc/99802883/Mobile-Apps-Study-June-2012>, June 2012.
- [28] Future of Privacy Forum. Privacy policy generators. <http://www.applicationprivacy.org/do-tools/privacy-policy-generator/>.
- [29] Daren M. Orzechowski and Mariam Subjally. Ftc provides guidance to mobile app developers for compliance with privacy regulations. <http://www.whitecase.com/articles-10022012/>, October 2012.
- [30] Shaileen Crawford Pokress and José Juan Dominguez Veiga. Mit app inventor: Enabling personal mobile computing. *CoRR*, abs/1310.2830, 2013.
- [31] S. Rosen, Z. Qian, and Z. M. Mao. Appprofiler: A flexible method of exposing privacy-related behavior in android applications to end users. In *Proc. 3rd ACM conference on Data and application security and privacy*, pages 221–232. ACM, December 2013.
- [32] Barbara A Schuldt and Jeff W Totten. Electronic mail vs. mail survey response rates. *Marketing Research*, 6(1):3–7, 1994.
- [33] Feng Shen, Namita Vishnubhotla, Chirag Todarka, Mohit Arora, Babu Dhandapani, Steven Y Ko, and Lukasz Ziarek. Information flows as a permission mechanism.
- [34] Chris Soghoian. Apple’s persistent device id is a threat to privacy. <https://www.aclu.org/blog/technology-and-liberty/apples-persistent-device-id-threat-privacy>, September 2012.
- [35] Jim Tanous. ios 8 introduces new location permissions for apps.
- [36] National Telecommunications and Information Administration. Short form notice code of conduct to promote transparency in mobile app practices. <http://www.ntia.doc.gov/files/ntia/publications/july%5F25%5Fcode%5Fdraft.pdf>, 2013.