Project Proposal, CS 97

Samantha Goldstein, Leah Foster, Madison Garcia

October 15, 2013

Abstract

Users operate with the understanding that the information put into phones and mobile devices is their own. However, many app creators will take sensitive data and send it to advertisers for a fee, often without users' consent. This data can be contacts, location, phone numbers to name a few. Currently TaintDroid can be used to inform an individual sensitive data from her/his phone has been recorded and sent away. FeedBack aims to use this existing software to glean when data maliciously sent away. If FeedBack can detect devious intent, it will then and replace that data with random falsified information, to maintain user privacy and security without advertisers even realizing the switch.

1 Motivation

As our phones become more integrated into daily life the gap between accessibility and transparency widens. Though users rely on helpful apps, these applications often take liberties with personal data and information. Sensitive data including location, phone contacts, and information about the device, are transmitted with or without consent of the user. Often a user permits access to sensitive data for a specific reason, such as allowing location information to an application that gives directions. However, even without explicit permission for further use, applications often send this data to advertisers. A user will not necessarily know if they are using, or buying such an application, as the application probably has no warning, or it is hidden behind legal jargon in Terms and Agreements. The ambition of FeedBack is to send random falsified information to these advertisers, in lieu of sensitive data. This maintains privacy with minimal effort on the part of the user. Furthermore, by sending false information, advertisers will not have a way to decipher whether or not the information is false, allowing increased privacy for the user.

2 Background

TaintDroid is a monitoring system for the Android operating system. Applications may request data from sources they do not have permission to access, or may use allowed information in underhanded ways. TaintDroid tracks the flow of data from privacy-sensitive sources to possibly malicious sinks, by tagging data at the source [2].

However, TaintDroid only tracks the data. It provides notifications and detailed reports on what data was sent, where it was sent, and what the data's taint was. While this is helpful, the user's data have still gone off to advertisers or various malicious parties.

Two apps in particular also target private information flow in hopes of not just tracking but blocking inappropriate data use. The first, MockDroid, is a modified version of the Android OS. If an application should not have access to a particular resource–and this can be determined at runtime–MockDroid will report to the application that the resource is empty or unavailable [1]. This does impede the applications' normal functioning, perhaps sending them into a much more limited offline mode.

AppFence is a much more sensible and interesting solution. The app uses shadow data, just as TaintDroid does. It sends either faked shadow data or genuine but not particularly private data to the requester, instead of the dangerous data. If the user denied the app access to that data, but the app is trying anyway, AppFence blocks network access. [3]

As a final note about TaintDroid's security, TaintDroid itself may be not difficult to circumvent if a clever and malicious person decides to target an Android device's private information flow. The application ScrubDroid, for instance, is able to fool or go past TaintDroid's defenses using several tools [4]. In the case of more normal data misuse, as by an advertiser sending off technically allowed data, TaintDroid would still be adequate. In our project, we would concentrate more on interaction with advertising over fending off attackers.

3 Our idea

We want to build off the existing TaintDroid research to prevent the transmission of sensitive information (rather than just tracking when that information is sent). We also want to transmit falsified information, particularly to advertisers, to allow the user more functionality than stopping all transmissions while still offering a high degree of anonymity. For example, we will send randomized location data to advertising servers, which will not alter the user experience but will protect users from unwillingly or unknowingly sharing their location with advertisers. We also hope to use TaintDroid's distinction between legitimate use and misuse of data to let users send correct information when appropriate, but falsified information when it is not crucial to the app's core functionality. For example, a user will be able to send their true location to a Google Maps server when looking up directions, but send a false location to an advertiser's server in the same app.

4 Milestones

- Milestone 1
  - Research existing projects further and write them up for the Background section of our final paper. In preparation for building on TaintDroid and in anticipation of the final paper, we will write up what we discover in our background research.
  - Review source code for TaintDroid. The authors have made their source code publicly available, and this is what we will build our

work off of. However, we need to familiarize ourselves with the implementation before beginning to build onto it.

- Milestone 2
  - Stop transmission of location. Stopping data transmission is the biggest component of our project. We will first work with stopping the transmission of location, and expand to other kinds of information as we are able. We want to work primarily with location because it is more tangible to the user: geographic location is easily understood as a privacy risk, whereas the risks of sending information like IMSI or IMEI are not readily apparent to the average user.
  - Try to send out a single piece of false data. This is the second biggest component of our project. The majority of the work on sending false data will be for Milestone 3, but we aim to get the first step into this segment started for Milestone 2, even if it's just a single piece of data.
  - Determine experimental tests. To be prepared for the testing phase in Milestone 3, we will determine what kinds of tests will be appropriate and begin planning our experimentation.
- Milestone 3
  - Send back randomized falsified location data. Building off the false data sending that we will have begun in Milestone 2, we will send out false location data and try to distinguish between transmissions that should have correct data (e.g. when querying a maps database for directions) and transmissions that should have false data (primarily advertisers). As possible, we will expand to sending other kinds of false information.
  - Experimentation and testing. The later half of Milestone 3 will be mostly dedicated to testing our product. The exact tests will depend on what we determine to be necessary during Milestone 2, but in general we want to test that sensitive information is not being transmitted unless we explicitly allow it, that false or randomized information can be sent, and that legitimate and non-legitimate requests will receive true and false information, respectively.
    - Try testing AntiTaintDroid on our app. AntiTaintDroid claims to subvert the taint tracking techniques of TaintDroid, so we would like to see how our app performs with the same attacks.
  - Maintain a log of TaintDroid info. Throughout testing, we will maintain a log of what information is transmitted (and whether it is correct or not), where it is sent, and whether it is a legitimate transmission.

References

[1] Alastair R. Beresford, Andrew Rice, Nicholas Skehin, and Ripduman Sohan. Mockdroid: trading privacy for application functionality on smartphones.

In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, pages 49–54, New York, NY, USA, 2011. ACM.

[2] William Enck, Peter Gilbert, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol Sheth. Taintdroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *OSDI*, volume 10, pages 255–270, 2010.

[3] Peter Hornyack, Seungyeop Han, Jaeyeon Jung, Stuart Schechter, and David Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM, 2011.

[4] Golam Sarwar, Olivier Mehani, Roksana Boreli, and Dali Kaafar. On the effectiveness of dynamic taint analysis for protecting against private information leaks on android-based devices. Technical report, Nicta, Eveleigh, Sydney, NSW, Australia, May 2013.