Mobile Application Evaluation Using Automation and Crowdsourcing

Shahriyar Amini[1], **Jialiu Lin**[1], **Jason I. Hong**[1], **Janne Lindqvist**[2], **Joy Zhang**[1]
Carnegie Mellon University[1]
Pittsburgh, PA
shahriyar@cmu.edu

Rutgers University[2]
New Brunswick, NJ
janne@winlab.rutgers.edu

ABSTRACT

With the widespread adoption of smartphones, mobile applications have gained mainstream popularity. However, the potential privacy and security risks associated with using mobile apps are quite high, as smartphones become increasingly integrated with our lives, being able to access our email, social networking accounts, financial information, personal photos, and even our cars and homes. To address this problem, we introduce AppScanner, an automated cloud-based service based on crowdsourcing and traditional security approaches to analyze mobile applications. Considering the large and growing number of mobile applications, our envisioned service builds on crowdsourcing, virtualization, and automation to enable large-scale analysis of apps. AppScanner provides end-users with more understandable information regarding what mobile apps are really doing on their devices. Armed with transparent and descriptive information regarding app behavior, users can make better decisions when installing and running apps.

INTRODUCTION

Mobile app stores have responded to the consumer demand for mobile applications, with the number of application downloads well into the billions [4, 15]. Considering the omnipresent availability of context-aware computing resources and users' eagerness to install and use apps, mobile devices have become an attractive platform to offer desirable services. Prior work by security experts and the research community have raised concerns about misbehaving mobile apps [1, 2, 7, 8, 11, 16, 17, 20]. These works have been successful in exposing the privacy and security related risks involved with using mobile applications and have also raised awareness through media coverage. However, expert knowledge is required to set up and perform the techniques used in these prior efforts and also to digest the results obtained. Furthermore, although some of the approaches are effective in detecting the use of sensitive information [7, 8], it is not clear when the use of the sensitive information by an app is both legitimate and beneficial to the end-user.

To determine the legitimate use of sensitive information by applications and also to address the scalability problem, we envision a new cloud-based service, AppScanner, which relies on crowdsourcing, virtualization, and automation to evaluate mobile application behavior. Specifically, AppScanner relies on the use of automation and traditional security techniques to learn application behavior in fine granularity. AppScanner then takes advantage of the wisdom and scale of crowds to evaluate the application behavior. Given the scale of participants available on crowdsourcing platforms such as Amazon Mechanical Turk and the use of automated techniques, we propose that AppScanner is a scalable approach towards analyzing mobile applications and providing large coverage of application markets. Further, we conjecture that by relying on the wisdom of crowds, AppScanner can produce application behavior evaluations that would be close to expert analysis of the app behavior.

APPSCANNER

AppScanner is comprised of multiple subsystems which work together to evaluate applications and to produce understandable summaries of application behavior for end-users. For the purposes of this work, we will focus on the Android operating system. We made this decision based on Android being open source as well as the availability of tools and extensions for Android (in particular, TaintDroid [8], and TEMA [18]). Figure 1 shows a high-level architecture of our proposed approach. For more details, refer to the technical report [3].

We are developing four major subsystems. The *App Mapper*, which we have named *Squiddy*, analyzes a given mobile app and outputs a control flow graph of major screens, associated privacy-related behaviors of the app, and major tasks that can be done with the app. We are also working on an interactive visualization, Gort, which presents the application behavior information obtained by Squiddy, to enable easier analysis and better understanding of apps. The other subsystems of AppScanner have yet to be developed. The *CrowdScanner* takes the results from Squiddy, and uses crowdsourcing techniques to capture what kinds of privacy concerns and surprises people have, both at a coarse granularity (e.g. flagging that the LED flashlight app requests Internet access) [14] and a fine granularity (e.g. flagging unusual behaviors on specific screens and tasks). The *Privacy Evaluator* estimates what kinds of things a third party can infer based on the information they can get through an app. The *Privacy Summarizer* collects people's perceptions and generates a simple-to-understand summary of a mobile app's privacy-related behaviors.

IMPLEMENTATION

At present, the full implementation of AppScanner is in progress. Currently, we have prototyped the App Mapper, Squiddy, and the interactive visualization tool, Gort.

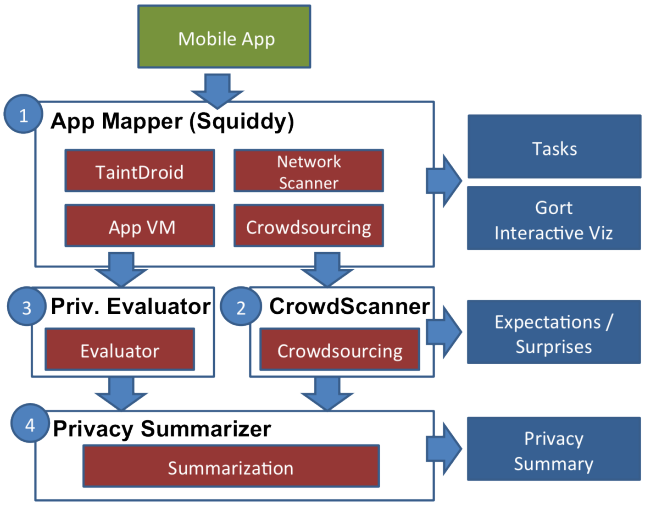

Figure 1. Android App Scanner system diagram, showing the major subsystems and their outputs. The App Mapper subsystem, Squiddy, takes an app and traverses the main screens of the app, monitors what resources are being used and what remote servers are being contacted. We envision that the App Mapper can also crowdsource what tasks can be done on various screens. As part of the App Mapper, we have developed Gort, which is an interactive visualization for the results obtained by Squiddy for a specific application. The CrowdScanner takes the tasks and the results obtained by Squiddy, and captures people's expectations and surprises regarding the application behavior. The Privacy Evaluator evaluates what can be inferred by third parties based on the app's type, frequency, and access pattern of the mobile device's resources and the user's information. The Privacy Summarizer combines the Privacy Evaluator's inferences about potential privacy leaks and people's perceptions produced by the CrowdScanner to distill a compact and understandable privacy summary for the application.

Squiddy

Squiddy is responsible for traversing mobile applications and monitoring the resources that are used. We achieve this task by building a *breadth-first search* traversal on top of TEMA [18] and monitoring the use of sensitive information by instrumenting applications using TaintDroid [8]. Squiddy currently traverses applications by clicking on interactive elements of the app UI. As the App Mapper discovers more screens of the app, it explores deeper into the application. The TaintDroid logs of sensitive information usage have timestamps which are then associated with interactions on a specific screen after the traversal is completed.

Gort Visualization

Gort presents the outputs from the Squiddy traversal and post-traversal processing in an interactive GUI. The task of identifying sensitive information use, recognizing sensitive transmissions, and mapping them to application behavior is non-trivial. As a result, having access to a GUI that summarizes the information and presents it in a manageable form is highly desirable for developers and privacy experts. The GUI allows the user to manage the amount of information presented by creating filters or by linking pieces of information using brushing and linking approaches [5]. The user can create filters based on a particular server, a specific request type (e.g., HTTP, HTTPS), or a particular resource that is involved, such that only related information corresponding to the filters are presented by the GUI. Moreover, filters can be combined in form of an intersection between categories (e.g., server and resource) or a union within categories (e.g., all requests related to servers x or y).

RELATED WORK

A number of prior works explore privacy leaks. TaintDroid [8] and PiOS [7] are two works that take advantage of dynamic and static analysis of mobile apps, respectively, to find potential leaks of sensitive information. AppScanner is similar in nature in that we attempt to detect sensitive information leaks in mobile applications, however, our main goal is to be able to present this information in an understandable fashion to end-users. Crowdsourcing has been growing quickly both as a topic of research and as a tool for research. Amazon's Mechanical Turk has been successfully used in a number of projects, including for example automating online tasks (such as image labeling) and user studies [13]. Crowdsourcing has also been used in the past to evaluate visualization design [9]. Perhaps the most sophisticated work in crowdsourcing is Soylent [6]. Soylent uses multiple ways of structuring tasks and organizing people so as to yield reasonably good results for relatively high-level tasks. This work showed how certain kinds of design patterns in organizing workers could yield reasonable results. There has been increasing research in usable privacy and security, looking at how to make tools and UIs useful and understandable for lay people [12, 19]. WebQuilt is a web logging and visualization system that helps web design teams run usability tests (both local and remote) and analyze the collected data [10]. AppScanner visualization will build on some of the WebQuilt concepts to visualize application resource usage.

SUMMARY

We presented our vision on how to enable the large scale evaluation of mobile application behaviors through crowdsourcing and automation. The number of applications on the Apple App Store and Google Play have collectively reached over a billion. Furthermore, the number of application downloads are already well into the tens of billions. As mobile apps have access to both mobile device sensors and also users' personal data, it is critical for users to know how mobile apps are using sensitive information and resources on their devices. However, considering the large number of mobile apps, mobile application behavior cannot be manually inspected for each application. Furthermore, it is necessary to transform the outputs of traditional privacy and security approaches to language that is both understandable and informative to users. To this end, we presented a cloud-based service, AppScanner, that uses crowdsourcing and automation to address the scalability problems involved in analyzing lots of applications and that also divides up the evaluation process such that its output can be presented to end-users in an understandable fashion.

ACKNOWLEDGEMENTS

This work is funded by the National Science Foundation (NSF) and a National Defense Science and Engineering Graduate (NDSEG) fellowship.

REFERENCES

1. C. Albanesius. Congress Asks Apple for Details About Address Book Privacy. *PCMAG*, Feb 2012. http://www.pcmag.com/article/print/294178.

2. A. Allan and P. Warden. Got an iPhone or 3G iPad? Apple is recording your moves. *O'Reilly Radar*, Apr 2011. http://radar.oreilly.com/2011/04/apple-location-tracking.html.

3. S. Amini, J. Lin, J. I. Hong, J. Lindqvist, and J. Zhang. Towards Scalable Evaluation of Mobile Applications through Crowdsourcing and Automation. Technical Report CMU-CyLab-12-006, CMU Cylab, 2012.

4. Apple Inc. Apples App Store Downloads Top 25 Billion. *Apple Press Info*, Mar 2012. http://www.apple.com/pr/library/2012/03/05Apples-App-Store-Downloads-Top-25-Billion.html.

5. R. A. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.

6. M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich. Soylent: a word processor with a crowd inside. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 313–322, New York, NY, USA, 2010. ACM.

7. M. Egele, C. Kruegel, E. Kirda, and G. Vigna. PiOS : Detecting privacy leaks in iOS applications. In *NDSS '11*, 2011.

8. W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, pages 1–6, Berkeley, CA, USA, 2010. USENIX Association.

9. J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 203–212, New York, NY, USA, 2010. ACM.

10. J. I. Hong and J. A. Landay. Webquilt: a framework for capturing and visualizing the web experience. In *Proceedings of the 10th international conference on World Wide Web*, WWW '01, pages 717–724, New York, NY, USA, 2001. ACM.

11. P. Hornyack, S. Han, J. Jung, S. Schechter, and D. 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*, CCS '11, pages 639–652, New York, NY, USA, 2011. ACM.

12. P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder. A "nutrition label" for privacy. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, SOUPS '09, pages 4:1–4:12, New York, NY, USA, 2009. ACM.

13. A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

14. J. Lin, S. Amini, J. I. Hong, J. Lindqvist, N. Sadeh, and J. Zhang. Expectation and purpose: understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 501–510, New York, NY, USA, 2012. ACM.

15. Scott Lowe. Google Play celebrates 25 billion downloads with 25 cent apps, discounted books, music, and movies. *The Verge*, 09 2012. http://www.theverge.com/2012/9/26/3409446/google-play-25-billion-downloads-sale.

16. T. Shields. Mobile Apps Invading Your Privacy. *Vercode Blog*, Apr 2011. http://www.veracode.com/blog/2011/04/mobile-apps-invading-your-privacy/.

17. E. Smith. iPhone Applications & Privacy Issues: An Analysis of Application Transmission of iPhone Unique Device Identifiers (UDIDs). Oct 2010. http://www.pskl.us.

18. T. Takala, M. Katara, and J. Harty. Experiences of System-Level Model-Based GUI Testing of an Android Application. In *Proceedings of the 2011 Fourth IEEE International Conference on Software Testing, Verification and Validation*, ICST '11, pages 377–386, Washington, DC, USA, 2011. IEEE Computer Society.

19. J. Tam, R. W. Reeder, and S. Schechter. I'm Allowing What? Technical Report MSR-TR-2010-54, Microsoft Research, 2010.

20. A. Thampi. Path uploads your entire iPhone address book to its servers. Feb 2012. http://mclov.in/2012/02/08/path-uploads-your-entire-address-book-to-their-servers.html.