Manifest Destiny: A Study for the Fate of Android Permissions - Project Proposal

Fallah Gibson
University of Minnesota
gibso564@umn.edu

Muse Jama
University of Minnesota
jamax165@umn.edu

Kah Hin Lai
University of Minnesota
laixx330@umn.edu

Paul Rheinberger
University of Minnesota
rhein055@umn.edu

1 MOTIVATION

The advent of smartphones has turned our mobile devices into general-purpose computers capable of containing private user data, system sensors, and device identifiers. This one-stop shop has always been an attractive target for malicious actors, but now, well-known and reputable organizations are beginning to go after this private data through their third-party applications [5]. One motive for this behavior is to serve targeted advertisements in an effort for better marketing and ultimately profit. A web advertisement provider can serve location specific ads by polling a device for GPS coordinates, scraping photo metadata, reading a phone's area code, or by exploiting a number of other signatures found on the phone. As most of this occurs without the consent or knowledge of the user, this is a clear violation of privacy, and as such, modern mobile operating systems attempt to restrict the access these applications have to this sensitive data.

Worldwide, Android has the largest market share for mobile operating systems by a staggering lead [1]. To protect sensitive resources and information from third-party applications, Android's architecture uses a permission-based access model, such that applications must explicitly and publicly request access to resources in their manifest file [2]. When installing these applications, the user will be prompted to either allow or deny the application access to permissions deemed dangerous by Android. While numerous works have analyzed the efficacy of these prompts, including [3, 4], Reardon et al. revealed that these permissions can be circumvented, even when the application does not request access, via covert and side channels [5].

The research question we propose asks how easy is it to develop these covert and side channel permission circumventions in Android and what defense mechanisms can be implemented in Android to mitigate these leaks? Do these attacks require the technical expertise and resources only a nation state or well-funded criminal organization can provide, or can the average script kiddie get around these permissions? Are there simply fixes to stop these leaks, or is Android's permission based model inherently flawed? If the former, what are these fixes? If the latter, what architecture should be developed for future versions to prevent these attacks?

2 OBJECTIVES

The goal of this research project is to determine how easy it is to use covert or side channels to circumvent the permission framework used in the Android mobile operating system. One objective for this project is to implement several new side and covert channels which circumvent a variety of Android's permissions. Another objective for this project is to use the knowledge we gain from developing these channels to propose defensive solutions to help prevent these sorts of attacks from occurring in future versions of Android.

3 PROJECT DESIGN

To begin our research project, we believe it is important to do a thorough literature review of papers relevant to the topic of Android permission circumvention, particularly using covert and side channels. It will also be necessary to have a solid foundational knowledge of the Android permission model, which we can achieve through studying the guides found on Android's Developer website.

Once we feel we have the technical knowledge of Android permissions, as well as background on the current state-of-the-art methods for circumventing permissions, we can start to develop our own covert and side channel attacks. We plan to develop these channels to circumvent a variety of Android permissions on current versions of the operating system.

After we have implemented a number of covert and side channels to circumvent various permissions, we will turn our attention to proposing defensive mechanisms in order to prevent these information leaks.

If we are successful in circumventing permissions labeled as dangerous by Android by obtaining sensitive information in applications without the appropriate permissions, we plan on responsibly disclosing our work to the Android team so that they may issue a security patch.

4 TIMELINE

The following is our anticipated timeline for completing major project milestones. We believe this timeline for accomplishing these important milestones is both reasonable and obtainable.

October 4
- Turn in project proposal report
- Review literature
- Set up Android development and testing environment

October 19
- Begin developing covert and side channel circumventions

November 2
- Set up project report document

November 16
- Begin writing project report

November 30
- Complete rough draft of project report

December 7
- Finalize project report
- Begin preparing project presentation

December 14
- Turn in final project report
- Turn in project source code
- Complete peer evaluations
- Present project presentation in class

5 ANTICIPATED RESULTS

We anticipate that we will be able to develop applications that circumvent the Android permission framework using both covert and side channels. We plan on replicating some of the more noteworthy approaches revealed by Reardon et al. in [5], as well as develop new techniques for creating these channels. While [5] only analyzes a subset of dangerous permissions, we also anticipate that we will be able to find covert or side channels to circumvent other permissions. If we are able to circumvent a dangerous permission that leaks sensitive or private information, we plan to disclose our findings to Android in hopes of receiving a bug bounty.

6 BUDGET

We anticipate zero costs associated with this project. The members of this group own two different Android devices, namely a Google Pixel and a Samsung Galaxy S9, that can be used for testing our applications. In addition, Android provides developers a free Android emulator which can be used to test our applications on a wide range of devices and Android versions.

7 ASSIGNMENT OF DUTIES

Fallah Gibson. Mr. Gibson will be responsible for reviewing relevant literature, studying Android's permission system, researching potential covert and side channel attacks for the Android permission system, preparing the final presentation, presenting the final presentation, and authoring the final project report.

Muse Jama. Mr. Jama will be responsible for reviewing relevant literature, studying Android's permission system, researching potential covert and side channel attacks for the Android permission system, preparing the final presentation, presenting the final presentation, and authoring the final project report.

Kah Hin Lai. Mr. Lai will be responsible for reviewing relevant literature, developing applications that use covert and side channels for circumventing Android permissions, proposing techniques to defend against the channels, preparing the final presentation, presenting the final presentation, and authoring the final project report.

Paul Rheinberger. Mr. Rheinberger is the group leader. As the group leader, he will be responsible for submitting all group work, including this proposal report, the final presentation, the final report, and all source code. Mr. Rheinberger will be responsible for reviewing relevant literature, developing applications that use covert and side channels for circumventing Android permissions, proposing techniques to defend against the channels, preparing the final presentation, presenting the final presentation, and authoring the final project report.

REFERENCES

[1] 2019. Mobile Operating System Market Share Worldwide. (September 2019). https://gs.statcounter.com/os-market-share/mobile/worldwide [Accessed 30 September 2019].

[2] 2019. Permissions overview : Android Developers. (September 2019). https://developer.android.com/guide/topics/permissions/overview [Accessed 30 September 2019].

[3] K. Benton, L. J. Camp, and V. Garg. 2013. Studying the effectiveness of android application permissions requests. In *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*. 291–296. https://doi.org/10.1109/PerComW.2013.6529497

[4] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android Permissions: User Attention, Comprehension, and Behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security (SOUPS '12)*. ACM, New York, NY, USA, Article 3, 14 pages. https://doi.org/10.1145/2335356.2335360

[5] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 2019. 50 Ways to Leak Your Data: An Exploration of Apps' Circumvention of the Android Permissions System. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 603–620. https://www.usenix.org/conference/usenixsecurity19/presentation/reardon