Push Notification Testing

Masters Project Proposal by
Rodger William Byrd
29 February 2020

Department of Computer Science
at the University of Colorado at Colorado Springs
School of Engineering and Applied Science

Committee Members:
Kristen Walcott Justice, Advisor
Committee Member 1
Committee Member 2

Contents

1 Introduction

There are major security implications to wearable devices such as smartwatches and medical devices such as implantable pacemakers, implantable defibrillators and insulin pumps.[4][7][8][5][6][3] In addition to the security implications of these devices, these also have the potential to cause physical harm, in the case of the medical devices. As a first step in researching the security of medical devices, this project will focus on wearable devices with the idea that future research may be conducted on medical devices. Our previous research has identified the push-notification process[9] as a potential point of instability in the communication between the Android OS and wearable devices. This project will focus on Andriod OS and attempt to build an automated testing tool to simulate the communication and notification process between the OS and wearable devices. The hypothesis tested will include the following impacts on the communication between the OS and wearable devices:

1. Available device storage is low
2. Missing patches and updates
3. Device carriers result in performance deltas

Previous research was manual and potential error could have been introduced. This project will attempt to provide more precise findings in support of this research by creating a simulation and testing platform that will allow varying hypothesis to be tested.

2 Background and Related Work

2.1 Background

Previous research by Sultana[9] showed that some Android devices had delayed notifications on paired wearable devices. Her research was conducted by performing a small manual study measuring the time it took from calling a phone to the time the notification showed up on the wearable device. That research found that there were significant delays in some of the testing scenarios and they varied by device and operating system. Some of the potential causes noted in that research were missing patches and updates and limited available storage on devices. The results from the findings showed that the performance variations were due to Andriod OS and not the wearable devices themselves.

2.2 Related Work

Wearable and medical devices are more of a security concern then other types of devices. They are generally more personal then other devices such as a pc or smartphone. They contain health and medical information

and a lot of the same sensitive information that is on a smartphone. Most users think of them as connected specifically to their phone, but the devices have many other connections such as wifi, bluetooth, usb, and SMS capabilities to exfiltrate data. Do et al. showed that they could get root access to samsung gear devices using a custom bootloader and were able to access sensitive information, such as SMS information, contact information and biomedical data.[4] In a related study, Al-Sharrah et al. showed that Apple watches store contact details, text messages, calendar details, Emails, pictures, and wallet data including stored payment cards.[3]

3 Proposed Work

The proposal for this project is to simulate and automate testing on Andriod OS and Android Wearable devices. The initial focus for the proposed work is to test push notificaitons in Android devices. After the initial emulation is complete additional simulation and automated testing will be developed.

3.1 Hypothesis

Previous work[9] showed potential connections between poor push notification performance and the following factors:

1. Low available device storage
2. Missing patches and updates
3. Device carriers result in performance deltas
4. Performance varying by version of Android OS

The scope of this project will be to build and perform automated testing for these factors and potentially others

3.2 Initial Phase

Emulation Emulation will involve using Android Studio to emulate and test notifications to determine what can cause delays. Research will be performed on the built in debugging tools to determine the capacity of those tools to emulate test scenarios. hat can cause delays in notification in andriod devices and wearable devices.

Simulation Simulates criteria that may lead to delays, such has low storage, high memory usage, high processor usage to attempt to determine factors that can lead to delays. Review Android OS patching history to determine what fixes have been put in place related to notifications and wearable devices.

Automated Testing After the push notification process has been sufficiently modeled, automated testing will be created to test multiple results at one time. This project will use the Android UI frameworks from Google to perform testing, as well as custom software. These frameworks include Espresso[1] and UI Automator[2]. Espresso is a "white box" tool for developers and can focus on testing one app at a time. UI Automator is a "black box" tool and can perform cross-app functional testing including installed and system applications.

3.3 Additional Phase - Future Work

Test findings from initial phase on actual hardware. Future work would be to take the findings from the initial phase which focuses on emulation and simulation and advance to testing on actual device hardware.

4 Tasks and Timeline

This section is a work in progress, I need to add date goals for the milestones.

- Emulation of Android OS
  - Install Android Studio
  - Setup Andriod Studio environemnt
  - Choose operating systems to test
  - Build app to emulate push notifications?
  - Research for existing software for testing android apps
- Emulation of Wear OS
  - Test capability of Android Studio to simulate interaction between mobile and wear environments
- Create automated test to test multiple scenarios quickly
  - Integrate test environment with Andriod Studio Emulator
  - Integrate UI Automator and Espresso automated UI frameworks
- Possible downloadable app for real world testing
  - TBD
- Test bluetooth interference on notification delays

Bibliography

[1] Espresso Automated UI Testing Framework.

[2] UI Automator Automated UI Testing Framework.

[3] M. Al-Sharrah, A. Salman, and I. Ahmad. Watch Your Smartwatch. In *2018 International Conference on Computing Sciences and Engineering (ICCSE)*, pages 1–5, March 2018.

[4] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. Is the data on your wearable device secure? An Android Wear smartwatch case study. *Software: Practice and Experience*, 47(3):391–403, 2017.

[5] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 129–142, May 2008.

[6] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In *2008 IEEE Symposium on Security and Privacy (sp 2008)*, pages 129–142, May 2008.

[7] Adam J. Mills, Richard T. Watson, Leyland Pitt, and Jan Kietzmann. Wearing safe: Physical and informational security in the age of the wearable device. *Business Horizons*, 59(6):615 – 622, 2016.

[8] Youngseok Park, Yunmok Son, Hocheol Shin, Dohyun Kim, and Yongdae Kim. This Aint Your Dose: Sensor Spoofing Attack on Medical Infusion Pump. In *10th USENIX Workshop on Offensive Technologies (WOOT 16)*, Austin, TX, August 2016. USENIX Association.

[9] Taniza Sultana. Wearable Devices: Smartwatch, Fitness Tracker and Call Notifications Delay.