TrubiZone

Securing Critical Mobile Applications for Android Using ARM TrustZone

Tiago Brito
Supervisor: Prof. Nuno Santos

Instituto Superior Técnico,
Av. Rovisco Pais, 1049-001 Lisboa - PORTUGAL
tiago.de.olveira.brito@tecnico.ulisboa.pt

Abstract. With the ever-growing number of connected devices worldwide, and with a more conscious and sharing society, mobile devices are becoming interesting data banks. With such an impact in our lives, mobile developers must focus on developing secure and privacy aware applications to protect users and services. Today's mobile operating systems do not offer fully secure methods to support critical applications, such as e-Health, e-voting and e-money, by not taking advantage of secure hardware technology like TrustZone. This paper proposes a new model for the development of critical mobile applications and a system based on TrustZone implementing it.

1 Introduction

Mobile devices are becoming the predominant device. Actions previously performed by powerful desktop computers can now be easily replicated on mobile devices. A recent study [4] from early 2015, with key statistics for the U.S. market, even shows that mobile devices, such as smart-phones and tablets, dominate digital media time over the Personal Computer (PC), with the trend being to continue raising.

Present in our everyday personal and professional lives, mobile applications (apps) start to handle privacy and security-sensitive data. Most notably these apps are handling photos, health and banking information, location and general documentation. The growing role of mobile devices has the negative consequence of becoming an attractive target for attacks. Among the several mobile platforms, Android is the one which attracts much more malware attacks [3].

The health sector is an interesting market for attackers due to its information value. According to Reuters [1], "medical identity theft is often not immediately identified by a patient or their provider, giving criminals years to milk such credentials. That makes medical data more valuable than credit cards". This is way regulatory laws such as the Health Insurance Portability and Accountability

[1] http://www.reuters.com/article/2014/09/24/us-cybersecurity-hospitals-idUSKCN0HJ21I20140924

Act (HIPAA), which establishes standards for electronic health care transactions, are so important to follow when managing sensitive health data.

Valuable health care information is a motivation for criminals to perform critical attacks on hospital networks around the world, consequently leaking or stealing health records of millions of patients. In September 2014, a group of hackers attacked the network of University of California, Los Angeles (UCLA)'s Hospital accessing computers with sensitive records of 4.5 million people. According to Cable News Network (CNN) [2], among the stolen records were the names, medical information, Social Security numbers, Medicare numbers, health plan IDs, birthdays and physical addresses of UCLA's patients.

In the mobile context, data is even more exposed and vulnerable due to the inherent portability of these devices, the sharing of information to third-party advertisers by device manufacturers or mobile app developers, unregulated management of sensitive medical information, specially because regulatory laws such as HIPAA do not account for the mobile sector, and because of security flaws on medical or consumer device software.

Parallel to the growth of the mobile app market, the number of e-Health mobile app, also known as Mobile Health (mHealth), available is increasing rapidly. In 2013, Research2Guidance [2] reported the existence of more than 97.000 mHealth apps across 62 app stores, with the top 10 apps generating up to 4 million free and 300.000 paid downloads per day. According to a report from MarketsAndMarkets [1], this market is expected to grow even further, from $6.21 billion in revenue in 2013 to $23.49 billion by 2018.

To protect sensitive data on mobile apps developers rely on mechanisms such as Digital Rights Management (DRM) [13], access control mechanisms, permission refinement, security Application Programming Interface (API), privacy enhancement systems and access control hooks, either from native Android [3], iOS [4] and Windows [5] or from extensions. These mechanisms rely on ad-hoc Operating System (OS) and application-level methodologies, which in most cases depend upon a very complex Trusted Computing Base (TCB) code, and do not fully enjoy the potential of the hardware of most modern smartphones, by not taking advantage of technology such as ARM's TrustZone.

Objectivos

Restrições

Especificação do sistema

Arquitectura

[2] http://money.cnn.com/2015/07/17/technology/ucla-health-hack/
[3] https://www.android.com/
[4] http://www.apple.com/ios/
[5] http://www.microsoft.com/en-us/windows

Contribuições
The remainder of this document proceeds as follows. Section 2 highlights the related work on mobile health, mobile security and TrustZone technology. Section 3 highlights the architecture of TrubiZone.

2 Related Work

This section presents the related work for this project and characterizes the main contributions of past works and how these contributions helped in the development of this project. This section is organized is three main parts: mHealth, mobile security and TrustZone.

The first part focuses on describing the attack surfaces of mHealth apps, the most common threats and their seriousness, present a few publicly available unsecure mHealth apps as well as some compliance recommendations that app developers should follow to avoid unnecessary security risks when handling sensitive health information. The second part of this sections describes the state-of-the-art of mobile security with a particular focus on the Android Operating System and how developers can build secure applications with a trusting OS and security mechanisms built upon the application layer. The third part of the related work describes TrustZone, a hardware technology available in most modern ARM Holdings (ARM) processors which supports execution of two isolated worlds, a hardware-protected secure world and a normal world. Besides describing the technology, this sections also presents previous work developed using TrustZone and how these previous contributions may be helpful in achieving the main goals of this project.

2.1 mHealth

As discussed above, mobile devices are increasing in number at astonishing rates and with this growth the mobile market becomes cheap and accessible. This motivates the shift from mainframe systems, located in the facilities of healthcare providers, to apps on mobile devices as well as storage in shared cloud services. This accessibility also motivates the private sector in building more healthcare applications to support both patients and healthcare agents. Thus, the mobile health market is becoming a competitive market and one which is increasingly handling more sensitive data.

Kotz, David [16] defines a threat taxonomy for mHealth. In this paper the author categorizes threats into three main categories: *Identity Threats*, *Access Threats* and *Disclosure Threats*.

Identity threats are described as mis-use of patient identities and include cases where a patient may lose (or share) their identity credentials, allowing malicious agents to access their Personal Health Record (PHR). *Insiders* (authorized PHR users, staff of the PHR organization, or staff of other mHealth support systems) may also use patient identities for medical fraud or for medical

identity theft and *outsiders* may be able to observe patient identity or location from communications.

Access threats are described as unauthorized access to PHR and include cases where the patient, which controls the data, may allow broader-than-intended access or disclosure of information, *insiders* who may snoop or modify patient data, with intents other than improving the healthcare of the patient, and even *outsiders* which, by breaking into patient records, may leak or modify this data.

Disclosure threats include cases where an adversary captures network packets in order to obtain sensitive health data, this problem can be mitigated by using strong encryption methods, but even if the network traffic is encrypted it is possible to analyse the traffic to determine its characteristics [21]. The adversary may also use physical-layer or link-layer fingerprinting methods to identify the device type and, because the wireless medium is open, an active adversary may inject frames or may selectively interfere with (cause collisions with) wireless frames. These methods may enable the adversary to create a man-in-the-middle situation, to use link-layer fingerprinting methods, or to compromise the devices in a way that divulges their secrets.

This work by Kotz, David [16] helped in understanding what threats are inherent to mHealth systems and their development. An explicit set of rules for developing privacy-sensitive health applications was still needed to ease the development process. Avancha, Baxi and Kotz [7] filled the gap by surveying the literature, developing a privacy framework for mHealth and discussing the technologies that could support privacy-sensitive mHealth systems.

This survey considers essential accounting for privacy in the design and implementation of any mHealth system, given the sensitivity of the data collected. A developer may use the list of privacy properties provided by this article as a check-list that should be considered in any design. Furthermore, a set of questions is left open for researchers to improve the efficiency and effect of these properties. It is also mentioned that the privacy challenges identified by this article need to be addressed with urgency, because mHealth devices and systems are being deployed now, and retrofitting privacy protections is far more difficult than building them in from the start.

He, Dongjing, et al. [14] analyse several mHealth applications available in Android's app store contributing to the understanding of security and privacy risk on the Android platform.

In this paper, three studies were made with the following goals:

- **Study 1: What are the potential attack surfaces?**
- **Study 2: How widespread is the threat?**
- **Study 3: How serious is the threat?**

In the first study 160 apps are analysed to find evidence of threats. From analysing previous literature [22,18,8,9,11,3,10], seven attack surfaces are determined to be in need of protection. These seven attack surfaces are shown in table 1.

In this study the authors also document that the 160 apps studied target two different audiences: 129 (81.65%) are for patients, 32 (20.25%) are for healthcare

Table 1. Description of attack surface (taken from He et al. [14])

Attack Surface	**Description**
Internet	Sensitive information is sent over the internet with unsecure protocols (e.g. HTTP), misconfigured HTTPS, etc.
Third Party	Sensitive information is stored in third party servers
Bluetooth	Sensitive information collected by Bluetooth-enabled health devices can be sniffed or injected
Logging	Sensitive information is put into system logs where it is not secured
SD Card Storage	Sensitive information is stored as unencrypted files on SD card, publicly accessible by any other app
Exported Components	Android app components, intended to be private, are set as exported, making them accessible by other apps
Side Channel	Sensitive information can be inferred by a malicious app with side channels, e.g. network package size, sequence, timing, etc.

professionals and the remaining 3 (1.90%) are targeted for both. Most apps targated for patients (60%) are in the Life Management category followed by apps that manage and synchronize user health information (PHR Management) which occupy nearly half (46.88%) of these apps. These numbers are a good indicator of what data is handled by most commercial mHealth apps available and motivate the choice made to build a PHR Management app as demo for TrubiZone.

A few examples of vulnerable applications are also revealed during this study. With regard to sending unencrypted information over the Internet, Doctor Online [5] (patients can talk to doctors online) and Recipes by Ingredients [6] send unencrypted sensitive information, including the user's email and password, in clear text. With regard to logging sensitive information, the study pinpoints CVS/pharmacy [12], which logs the prescription refill details from user inputs, including name, email address, store number, and Rx number and also logs user login credentials in a debug log message. A malicious party can use this information to view user profiles, prescription history, which could support medical identity theft and it is even possible to do pharmacy online shopping with user's stored credit card information. Further applications are given as example for exposing sensitive information, Noom Weight Loss Coach [15] reveals user workout history by exposing its Content Providers to external apps, which means any app can access the exposed Content Providers without declaring any permission. Finally this first study concludes with the example of sleep monitoring apps, such as SnoreClock [19] and Sleep Talk Recorder [17] which store the sleep records of users as unencrypted audio files on external storage. With read permission for the SD card as well as internet permission, a malicious app can read a user's sleep recordings and even send them to remote servers.

In the second study, 27 apps, from the top 1080 free apps from the Medical and Health & Fitness categories on Google Play, were analysed according to their vulnerabilities. From this analysis three attack surfaces are identified as the most

important ones: *Internet*, *Third Party Services* and *Logging*. Only 7 of the 27 apps analysed use the Internet to effectively send medical information over to remote servers. It is important to understand if the information sent over the Internet is protected. To achieve this the authors captured the network traffic and concluded that only 57.1% (4/7) of these apps use encrypted communication and the remaining 42.9% (3/7) use unencrypted communication to send sensitive health information. Among the unencrypted contents sent by these 3 apps were emails, usernames and passwords. This study also concludes that 85.7% (6/7) of these apps are hosted and store the recorded data on third party servers. This is an economical and scalable solution for mobile application, but storing sensitive health records on third party servers can have serious implications, mostly due to app users not being aware that their data is being stored on third party servers and because these users are not able to tell if this data is stored in encrypted is such a way that hosting companies do not have access to this data.

Some health apps use Bluetooth devices to collect personal health information such as heart rate, respiration, pulse oximetry, electrocardiogram (ECG), blood pressure, body weight, body temperature, quality of sleep and exercise activities. Naveed et al. [18] show how a malicious app can stealthily collect user data from an Android device or spoof a device and inject fake data into the original device's app in what is called an external-device misbonding (DMB) attack. Ont of the 27 apps analysed in this study connects to external health sensors and uses default PIN code 0000, which makes it vulnerable to the DMB attack. Along with the logging, external storage, exported components and the other problems discussed above, which represent the explicit channels used for attacks, side channels can be exploited by a malicious party to infer sensitive information from apps, even when they are well-designed and implemented. This study mentions an example by Zhou et al. [22] where a correlation between network payload size, which is publicly accessible in Android, and the disease condition a user selects on WebMD mobile [20]. This problem is mitigated by enforcing limitations on accessing Android public resources by modifying the Android kernel.

In the third study, another 22 apps, which send information over the Internet, are randomly selected from the same top 1080 apps and used to understand the seriousness of the threat considering the sensitive information sent. The 22 apps are analysed to understand what information is effectively being sent over the Internet and the conclusion is that when used as intended these apps gather, store and transmit a variety of sensitive user data which includes at least personal profiles, health sensor data, lifestyle data, medical information browsing history and third-party app data (e.g. Facebook account information). Figure 1 shows the distribution of sensitive information in the 22 apps. The consequences of data breaches, information disclosure or tempering with sensitive health data depends on the type, sensitivity and volume of the data breached, but it is clear that profiling, medical identity theft and healthcare decision-making errors are possible consequences. This is way the authors suggest the use of encryption for

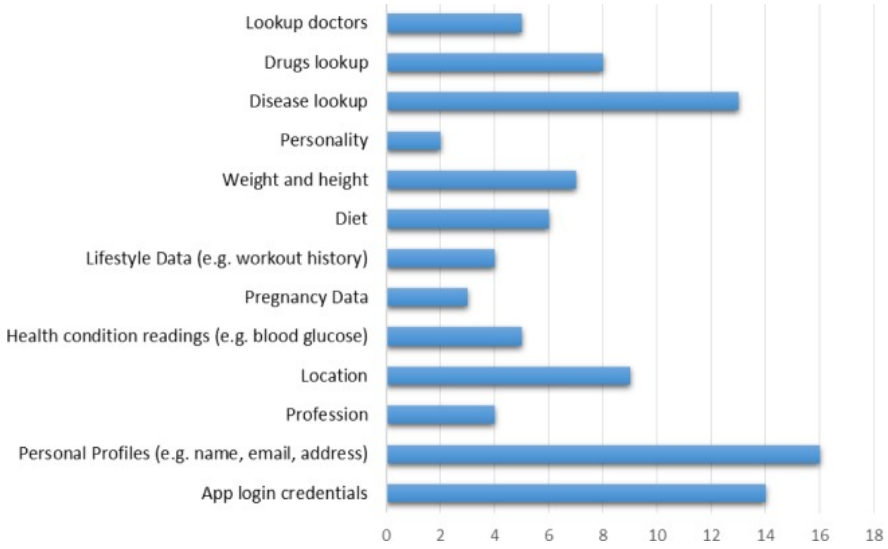

Fig. 1. Sensitive information distribution for study 3 (taken from He et al. [14])

communication and storage and encourage developers to create a set of security and privacy guidelines that offer a baseline for protection.

The work by He, Dongjing, et al. [14] helped in understanding what are the most common attack surfaces when considering mHealth applications on Android and it also helped assessing the security risks inherent to applications which handle privacy sensitive information. This paper generally describes the state-of-the-art of security in commercial mobile health apps and pinpoints what should be fixed in order to build better and more secure application for the healthcare industry. It is clear that developers focus much more on building feature-full applications rather than secure applications and this is way it is important to build a framework which allows developers to completely focus on the features they want to provide without having to focus heavily on security. We believe that building a developer framework over ARM's TrustZone technology is a good solution because with TrustZone is possible to fully enjoy the potential of the hardware already available in most mobile devices. This technology will be described in section 2.3.

2.2 Mobile Security

2.3 TrustZone

References

1. Mobile health apps & solutions market by connected devices (cardiac monitoring, diabetes management devices), health apps (exercise, weight loss, women's

health, sleep and meditation), medical apps (medical reference) – global trends & forecast to 2018. Technical report, MarketsAndMarkets, 09 2013. http://marketsandmarkets.com/.

2. Mobile Health Market Report 2013-2017. Technical report, Research2Guidance, 03 2013. http://research2guidance.com/.
3. Mobile Threat Report. Technical report, F-Secure Labs., 03 2014. https://www.f-secure.com/documents/996508/1030743/Mobile_Threat_Report_Q1_2014.pdf.
4. Digital Future in Focus. Technical report, Comscore Inc., 03 2015. http://www.comscore.com/Insights/Presentations-and-Whitepapers/2015/2015-US-Digital-Future-in-Focus.
5. Airpersons. Doctor online @ONLINE. https://play.google.com/store/apps/details?id=com.airpersons.airpersonsmobilehealth.
6. AB Mobile Apps. Recipes by ingredients @ONLINE. https://play.google.com/store/apps/details?id=com.abMobile.recipebyingredient.
7. Sasikanth Avancha, Amit Baxi, and David Kotz. Privacy in mobile technology for personal healthcare. *ACM Computing Surveys (CSUR)*, 45(1):3, 2012.
8. Adam J Aviv, Benjamin Sapp, Matt Blaze, and Jonathan M Smith. Practicality of accelerometer side channels on smartphones. In *Proceedings of the 28th Annual Computer Security Applications Conference*, pages 41–50. ACM, 2012.
9. Liang Cai and Hao Chen. *On the practicality of motion based keystroke inference attack*. Springer, 2012.
10. Chin and O'Neil. Seven ways to hang yourself with android @ONLINE. https://www.defcon.org/images/defcon-19/dc-19-presentations/O'Neil-Chin/DEFCON-19-O'Neil-Chin-Google-Android.pdf, 2011.
11. Erika Chin, Adrienne Porter Felt, Kate Greenwood, and David Wagner. Analyzing inter-application communication in android. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 239–252. ACM, 2011.
12. CVS/pharmacy. Cvs/pharmacy @ONLINE. https://play.google.com/store/apps/details?id=com.cvs.launchers.cvs.
13. OMA DRM. Open mobile alliance digital rights management.(2010). *Retrieved May*, 2, 2011.
14. Dongjing He, Muhammad Naveed, Carl A Gunter, and Klara Nahrstedt. Security concerns in android mhealth apps. In *AMIA Annual Symposium Proceedings*, volume 2014, page 645. American Medical Informatics Association, 2014.
15. Noom Inc. Noom weight loss coach @ONLINE. https://play.google.com/store/apps/details?id=com.wsl.noom.
16. David Kotz. A threat taxonomy for mhealth privacy. In *COMSNETS*, pages 1–6, 2011.
17. MadinSweden. Sleep talk recorder @ONLINE. https://play.google.com/store/apps/details?id=com.madinsweden.sleeptalk.
18. Muhammad Naveed, Xiaoyong Zhou, Soteris Demetriou, XiaoFeng Wang, and Carl A Gunter. Inside job: Understanding and mitigating the threat of external device mis-bonding on android. In *Proceedings of the 21st Annual Network and Distributed System Security Symposium (NDSS)*, pages 23–26, 2014.
19. Ralph Schiffhauer. Snoreclock @ONLINE. https://play.google.com/store/apps/details?id=de.ralphsapps.snorecontrol.
20. WebMD. Webmd for android @ONLINE. https://play.google.com/store/apps/details?id=com.webmd.android.

21. Charles V Wright, Fabian Monrose, and Gerald M Masson. On inferring application protocol behaviors in encrypted network traffic. *The Journal of Machine Learning Research*, 7:2745–2769, 2006.
22. Xiaoyong Zhou, Soteris Demetriou, Dongjing He, Muhammad Naveed, Xiaorui Pan, XiaoFeng Wang, Carl A Gunter, and Klara Nahrstedt. Identity, location, disease and more: Inferring your secrets from android public resources. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 1017–1028. ACM, 2013.