Digital Investigation 24 (2018) S60–S67

Contents lists available at ScienceDirect

Digital Investigation

journal homepage: www.elsevier.com/locate/diin

DFRWS 2018 Europe — Proceedings of the Fifth Annual DFRWS Europe

Forensics acquisition — Analysis and circumvention of samsung secure boot enforced common criteria mode

Gunnar Alendal [a], [*], Geir Olav Dyrkolbotn [a], [c], Stefan Axelsson [a], [b]

[a] Department of Information Security and Communication Technology, NTNU, Gjøvik, Norway
[b] Halmstad University, Sweden
[c] Norwegian Defence Cyber Academy (NDCA), Jørstadmoen, Norway

Keywords:
Common criteria
CC mode
Mobile security
Mobile device management
Forensic acquisition
Smart phone
Samsung secure boot

ABSTRACT

The acquisition of data from mobile phones have been a mainstay of criminal digital forensics for a number of years now. However, this forensic acquisition is getting more and more difficult with the increasing security level and complexity of mobile phones (and other embedded devices). In addition, it is often difficult or impossible to get access to design specifications, documentation and source code. As a result, the forensic acquisition methods are also increasing in complexity, requiring an ever deeper understanding of the underlying technology and its security mechanisms. Forensic acquisition techniques are turning to more offensive solutions to bypass security mechanisms, through security vulnerabilities.

Common Criteria mode is a security feature that increases the security level of Samsung devices, and thus make forensic acquisition more difficult for law enforcement.

With no access to design documents or source code, we have reverse engineered how the Common Criteria mode is actually implemented and protected by Samsung's secure bootloader. We present how this security mode is enforced, security vulnerabilities therein, and how the discovered security vulnerabilities can be used to circumvent Common Criteria mode for further forensic acquisition.

Introduction

Digital forensics is the recovery and investigation of data found in digital devices (Carrier, 2002). Garfinkel (2010) discusses the difficulties that awaits digital forensics, what challenges exist in today's tools, research and knowledge and how digital forensic research should move forward to keep digital forensics a valid method for the years to come. The prediction is that both the recovery, forensic acquisition, and investigation will become increasingly harder as complexity and security mechanisms, like encryption, grow in use. Faced with this ever increasing security of Commercial of-the-shelf (COTS) products, law enforcement faces an increasing challenge when it comes to the ability to do forensic acquisition. Where before law enforcement could bypass security mechanisms by e.g. accessing data at a lower level, like forensic de-soldering (chip-off), to read content off data storage directly, today's, often mandatory, encryption of user data on mobile devices invalidates such methods. The ability to read stored data on the device's storage is simply not enough. Reading encrypted data has little value without the corresponding encryption key(s). The addition of security features like device-tied encryption keys, supported by hardware and a TrustZone, gaining access to such encryption keys is made even harder. This might then require law enforcement to power on the device, in order to try to extract keys or decrypted data through interaction with the security mechanisms protecting the user data. This type of interaction often means installing or modifying code on the device. Even though law enforcement have legitimate cause for their “hacking”, this is activity that in other contexts would be regarded malicious and illegal, also known as an attack. Therefore, to protect against such attacks, most mobile device vendors protect code running on the devices, from the first code executed at power on and all the way through to a full operating system, like Android, is up and running. This is often referred to as a *Secure Boot*, and refers to the trust in code executed on the device. This code should only be certified and official code, made by the vendor, and properly signed to prove authenticity.

[*] Corresponding author.
E-mail addresses: gunnaale@stud.ntnu.no (G. Alendal), geir.dyrkolbotn@ntnu.no (G.O. Dyrkolbotn), stefan.axelsson@ntnu.no (S. Axelsson).

https://doi.org/10.1016/j.diin.2018.01.008

Law enforcement always strives to acquire as much data as possible to support any ongoing investigation. So bypassing such complex security schemes, if possible, forces law enforcement to invest in deeper knowledge and costly equipment to perform advanced forensic acquisition, utilising such attacks.[1] Law enforcement is then investing in the discovery and use of security vulnerabilities, to bypass security mechanisms to acquire digital evidence.

On the other hand, seen from a user and enterprise perspective, with the growing use of these devices, both end users and enterprises are demanding more secure devices to help protect sensitive data. The need to secure mobile devices, especially in an enterprise context is important, as devices moving in and out of the enterprises network, unchallenged, introduces attractive attack vectors for cyber criminals and cyber espionage.

Mobile Device Management (MDM) solutions can enable the centralised control of devices that are used in the enterprise. Enterprises can then monitor, control and administrate devices in a systematic manner, across device vendors and service providers. Samsung supports such solutions by offering a.o. a feature they refer to as *Common Criteria mode* or simply *CC mode* (Samsung, 2017a). CC mode is a security feature designed to increase the device's protection against unauthorised access and can therefore pose an additional challenge to law enforcement trying to acquire data from devices with CC mode enabled. A major challenge is that CC mode denies access to the device firmware update mechanism, a common method used by law enforcement to gain access to data.

This paper presents the reverse engineering results of CC mode and how discovered security vulnerabilities can be used to circumvent CC mode for further forensic acquisition.

The rest of the paper is organised as follows: Section “Related work and contributions” discusses related work and how our contribution relates. Section “Samsung secure boot model” introduces the Samsung secure boot model. Section “Samsung CC mode and SBOOT” describes the CC mode related parts of the Samsung secure boot and how this relates to the secure execution environment, TrustZone. Section “Unauthorised disabling of CC model” discusses attacks on the CC mode. In section “Conclusion” we discuss the implications of our findings and offer our conclusions.

Related work and contributions

Recovering data from mobile devices can be achieved by reading data from storage or from volatile memory (RAM). The two sources of data differs in both how data is stored and how data can be retrieved. Data in long term storage is often stored well structured in file systems, as it has to be able to be read by different operating systems, and other tools. Data structures in RAM are often less well documented, and the formats more volatile, as it needs only survive to the next restart of the device. RAM is repopulated each time the device is restarted.

Nathan Scrivens et al. (Scrivens and Lin, 2017) summarised many of the current options for forensic acquisition of storage on Android mobile devices. According to Scrivens et al., the main options are chip-off, de-soldering storage for off-device reading, JTAG (Joint Test Action Group) interface for in-circuit reading of storage, rooting and exploitation solutions for recovering data by breaking the security of the device, Android Debug Bridge (ADB) by utilising device debug capabilities for forensic acquisition, and finally backup solutions retrieving data through normal or rooted user access. These different methods have different requirements and weaknesses. Chip-off requires physical access to underlying storage media, and can not deal with the increasing use of encryption on storage devices. JTAG is a interface often used during development and testing of a device, and can be used to communicate directly with the underlying storage media. However, the JTAG test pins can be hard to find and access on different devices, and can also be secured against unauthorised access, and also disabled by the vendor before shipping. ADB is a powerful debug interface supported by Android, but it is not enabled by default on most Android devices, nor does it give root access. Finally, backup applications are rarely accessible to unauthenticated users and are often of limited use for forensics.

Seung Jei Yang et al. (2015) demonstrated a different approach: doing forensic acquisition of storage media through the misuse of the device firmware update protocols. This will give access to the underlying storage and the ability to dump its content. Unfortunately this method will also be insufficient if the data stored is encrypted.

Seung Jei Yang et al. (2017) recently demonstrated a different use for the device firmware update protocols. Instead of acquiring storage they have demonstrated how to acquire RAM through this update protocol. This can again be used to acquire encryption keys used to encrypt storage, in addition to save user data that resides in RAM at the time of RAM acquisition.

Guido et al. (2016) demonstrated *hawkeye*, an agent to do rapid acquisition of Android devices. Although their goal is to reduce the amount of data needed to be transferred during the acquisition process, this is an example of a forensic agent that needs to be injected into the device to function as expected. This is done by installing a custom boot image on the device to facilitate hawkeye injection. Installing this custom image is done through the device firmware update protocol and access to firmware update mechanism is a requirement.

As we can see, access to a device's firmware update protocol can be vital for successful forensic acquisition. Any functionality denying this access is therefore limiting the possibilities for law enforcement to acquire data from a given device. CC mode is preventing law enforcement access to the firmware update mode on Samsung devices. Our contribution is to analyse and circumvent CC mode to gain access to the firmware update mode. For completeness, we have also included the discussion of a MDM setting, also affecting access to the firmware update mode.

Our reverse engineering of CC mode reveals security vulnerabilities in the design and implementation of these security mechanisms, and demonstrates how such security vulnerabilities can be discovered and used in digital forensic acquisitions.

Our contribution shows that law enforcement trying to acquire data from a device can disable CC mode and get access to firmware update mode, thus removing the extra layer of security enforced by CC mode. Disabling CC mode can then enable existing methods but also increases the attack surface in general, increasing the possibility to discover new vulnerabilities and methods.

CC mode and methodology

CC mode is built on top of the phone's Android security model and hardware, to increase enterprise security. Samsung has made available several guidance documents for Common Criteria evaluation for many of their different phone models (Samsung, 2017c).

Samsung provides a wide range of management APIs to control a Samsung device (Samsung, 2017b). These APIs can be used in 3rd party MDM solutions. To further promote the use of CC mode in MDM solutions, Samsung has made available a Common Criteria mode APK (Samsung, 2017a). This Android application package

[1] With the word “attacks”, in the context of this article, we mean: exploiting vulnerabilities for forensic data acquisition purposes by law enforcement agencies.