Digital Investigation 25 (2018) 42–54

Contents lists available at ScienceDirect

Digital Investigation

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

New flash memory acquisition methods based on firmware update protocols for LG Android smartphones

Juhyun Park [a], Yun-Hwan Jang [b], Yongsu Park [a, *]

[a] Department of Computer Science, Hanyang University, Seongdongku, Seoul, 04763, South Korea
[b] Department of Information Security, Hanyang University, Seongdongku, Seoul, 04763, South Korea

ARTICLE INFO

Article history:
Received 22 December 2017
Received in revised form
10 April 2018
Accepted 10 April 2018

Keywords:
Digital forensics
Android forensics
Smartphone forensics
Android physical acquisition
Firmware update protocol

ABSTRACT

Since criminals are committing offenses using smartphones in many cases, smartphone forensic techniques are being considered important by criminal investigators. In this paper, we present a new methodology to analyze firmware update protocols using fuzz testing. This approach has enabled us to find five new methods to acquire data evidence from LG Android smartphones, released from 2011 to 2016. We conducted extensive experiments to obtain data evidence for 48 LG smartphone models (100 devices in total). Furthermore, we deal with diverse digital forensic issues such as integrity of evidence or decompression/decryption of data using our methods.

Introduction

People utilize smartphones for a variety of tasks such as calling, SMS, social networking services (SNS), Internet commerce, and mobile games. In 2016, the distribution rate of smartphones in 50 major countries in the world was 69.5% (Pew Research Center, 2016). Since the smartphone is closely related to daily life, smartphone forensics is of increasing importance in solving digital crimes. Significant research work (Son et al., 2013; Yang et al., 2015; Sylve et al., 2012; and Park et al., 2012) has been done on acquiring data from flash memory in mobile devices for use in digital forensics.

Generally, S/W-based acquisition methods for smartphones are classified into four types. The first method relies on a custom recovery image (Son et al., 2013). Using the recovery mode provided by Android, a custom recovery image can be used to get the root privilege. After flashing this image, the recovery mode can be entered as the root and then the entire flash memory can be dumped. Although this method guarantees the consistency of user data, the integrity of the entire flash memory is not guaranteed because the recovery area in the flash memory is changed (Yang et al., 2015). Also, certain manufacturers do not allow flashing of the custom recovery area (LG Electronics, 2016).

In the second method, the ADB (Android Debug Bridge) (Android Debug Bridge (ADB), 2017) protocol is used. The ADB protocol supports diverse operations on the smartphone, e.g., retrieving files and installing apps. By using the vulnerability of the smartphone, we can obtain the root privilege and dump the entire flash memory. Otherwise, we are only able to obtain the files that can be accessed as a normal user. In both cases, some data is modified in the access process such as activation of the debugging mode or application installation. There are also disadvantages in that we cannot access the data without unlocking the pattern lock or a password.

The third method uses the Android kernel vulnerability to obtain the root privilege and acquire data. However, finding new vulnerabilities is extremely hard in an environment where the Android system is frequently patched, especially for the latest system. This is a problem because the rooting work should be done in the state when the device is running, so the user data may be inconsistent.

Lastly, there are acquisition methods using the firmware update protocol (Yang et al., 2015), which is the method used in our work. Each phone manufacturer has its own private firmware update protocol for updating or upgrading firmware. Generally, this protocol is private and major manufacturers restrict its functionality, e.g., Samsung's mobile phone accepts the command to flash the

* Corresponding author.
E-mail addresses: hdhyun216@hanyang.ac.kr (J. Park), dbsghksdlwkd@hanyang.ac.kr (Y.-H. Jang), yongsu@hanyang.ac.kr (Y. Park).

https://doi.org/10.1016/j.diin.2018.04.002
1742-2876/

firmware whereas it rejects the command to get data from the flash memory. Therefore, reverse engineering must be done to analyze the protocol and to find the protocol vulnerabilities to retrieve data. However, manufacturers patch the protocol to remove bugs or use the different protocol for new models, which makes it difficult to obtain the data using this approach.

Recently, LG's share reached 20% in the US smartphone market in the first quarter of 2017 (Rutnik, 2017). We analyzed LG's firmware update protocols (for recent models) that have not yet been released or analyzed. Furthermore, we devised a method to bypass the authentication process in the protocols.

In this paper, we first present a new methodology to analyze the firmware update protocols. This methodology is generic, i.e., it can be used for analyzing diverse protocols from various manufacturers. Unlike Yang's work (Yang et al., 2015), it uses fuzz testing and reverse engineering on the client software. From this, we found five new acquisition methods that support various recent devices, i.e., LG's smartphones that were released from 2011 to 2016.

In each proposed acquisition method, we will explain the forensic issues in detail such as the data change due to mounting the file system or data conversion during transmission.

We show experimental results for 48 models of LG smartphones (100 devices in total) that were released from 2011 to 2016. The following is a summary of the contributions of this paper.

Contributions

- We developed a new methodology to analyze firmware update protocols and to find new acquisition methods. Our methodology uses a protocol analyzer, API monitor, protocol fuzzing, and reversing on client software.
- In addition to the existing acquisition method (Yang et al., 2015), five new acquisition methods are proposed that can be used for LG smartphones released from 2011 to 2016.
- In order to estimate the feasibility of the proposed methods, we conducted extensive experiments using 48 models of LG smartphones (100 devices in total).
- We analyzed the data integrity issues that may occur in the acquisition process for each method. Also, for the acquired data, we analyzed the forensic issues in converting, decompressing, or decrypting the data in order for it to be understandable for analysts.

This paper is organized as follows. Section Related work describes related work. In Section Proposed methodology to analyze firmware update protocols and to find new acquisition methods we propose a new methodology for finding acquisition methods and in Section Proposed acquisition methods for LG smartphones we describe our new acquisition methods for LG smartphones. Section Experimental results shows the experimental results and Section Conclusion concludes the paper.

Related work

Data acquisition methods for smartphones are classified into H/W-based acquisition and S/W-based acquisition (Son et al., 2013; Yang et al., 2015). Section H/W-based acquisition method deals with the H/W-based acquisition method, and Section S/W-based acquisition method explains the related work on S/W-based acquisition.

H/W-based acquisition method

H/W-based acquisition can be classified into JTAG-based acquisition methods (Kim et al., 2008) and chip-off acquisition methods (Breeuwsma et al., 2007; Jovanovic, 2012). The Joint Test Action Group (JTAG) interface connects a debugger and a chip to support diverse operations to debug a chip, which is specified in the IEEE standard 1149.1 (IEEE Std 1149.1-2001, 2001). Using this interface, the JTAG-based acquisition method can read data from the flash memory on the PCB board (Kim et al., 2008). In the JTAG-based method, there is a disadvantage that it cannot be used with the latest smartphones because it does not support the JTAG interface.

The chip-off method collects data by detaching the flash memory chip from the PCB of the smartphone and then connecting the chip to the memory reader (Breeuwsma et al., 2007). The chip-off method is used only when other methods cannot be applied due to damage such as flooding or falling/being broken. This method has a risk in that the flash memory or PCB is damaged in the process of desoldering/separating the flash memory chip.

S/W-based acquisition method

The S/W-based acquisition method can be classified into two types: logical acquisition and physical acquisition. The logical acquisition method does not extract all the data in the flash memory; it acquires only some of it at the logical level, e.g., specific files, directory structure, and user data. In the Android system, we can use ADB (Android Debug Bridge) (Android Debug Bridge (ADB), 2017) to backup the data (Android Backup Extractor, 2014). Otherwise, we can use the Content Provider (Hong, 2011) to acquire data. However, these methods have a drawback in that accessible data is limited, and we cannot ensure the integrity of the data due to the work required in entering the debugging mode and installing additional applications. Furthermore, we cannot recover the deleted data or bypass the access permission to obtain unauthorized data.

The physical acquisition method proceeds by dumping the entire flash memory of the smartphone after connecting a physical communication cable, e.g., a USB cable. This method can be classified into three types.

The first method uses the vulnerabilities in the Android kernel to acquire the root permission (Rooting (Android), 2014; Sun et al., 2015) and then dumps the entire flash memory. If we have root privilege, we can dump the block device through a disk dump tool, such as dd (diskdump). Commercial forensic tools such as (Oxygen Forensics, 2014; AccessData MPE+, 2014; MSAB XRY, 2015) support this method. However, it is now hard to use because most of the vulnerabilities have already been patched in the latest version of the firmware.

The second method uses a custom recovery image (Vidas et al., 2011; Son et al., 2013; Guido et al., 2016) that can be obtained by changing the original recovery image in the firmware or by building from the modified source code. In this method, we should first flash a custom recovery image and then enter the recovery mode to acquire the data. This method has an advantage in that it does not require the vulnerability of the operating system. However, some manufacturers set the OEM lock so as not to flash a recovery image. In this case, if we enforce flashing or try to unlock it, the warranty is lost or the entire user data is erased (LG Electronics, 2016). LG's smartphones are of this type, which is the case used in this paper.

The third method is to use the firmware update protocols provided by the manufacturers (Yang et al., 2015). This method utilizes private/proprietary firmware update protocols that were implemented by manufacturers to update the firmware. If we find read operations in the protocols, we can acquire the entire flash image. The first work was done by Yang et al. (2015), where they proposed acquisition methods for LG, Samsung, and Pantech devices. Recently, Yang et al. (2017) proposed live acquisition methods where the main memory content is acquired using firmware update protocols while the device is running normally. However, as new devices are released, firmware update protocols have been changed