Survey on Android Forensic Tools and Methodologies

Venkateswara Rao V.
CSE, UCEK, JNTUK
Kakinada, India

A. S. N. Chakravarthy
CSE, UCEK, JNTUK
Kakinada, India

ABSTRACT

In recent days, Android Operating System has gained top most position in mobile Operating System (OS) market share. Usage of Smartphone and tablet devices is massively increased and major portion of these devices are come up with android OS. There is a big chance that these devices may be used in committing crimes also. While doing forensic investigation of the digital devices which are involved in the crime needs special tools and techniques to seize, acquire and analysis of the android devices. This paper highlights various techniques available in the market in terms of logical acquisition, physical acquisition and analysis.

Keywords

Cyber forensics; android forensics; mobile forensics; forensics techniques; Smartphone’s

1. INTRODUCTION

Many of the portable devices like Smartphone, tables, PDAs and many other electronic handheld gadgets are running using software program called operating system to manage the hardware and applications. The mobile operating system is the software platform on top of which other programs, called application programs, can run on mobile devices. Many mobile operating systems are available in the market such as Apple iOS, Google Android, BlackBerry OS, Nokia's Symbian, Hewlett-Packard's webOS (formerly Palm OS) and Microsoft's Windows Phone OS. Out of these Google Android is the popular and mostly used. In the below sections android architecture and various forensics methods available and various tools features are discussed. This paper discussed about Android OS, architecture, Linux kernel and file systems in Section I, presented literature review of existing work carried out in the area of android forensics in Section II, Various methods and tools existing for conducting forensic activity on android OS are discussed in Section III and study and comparison of various features available in tools with respect to required functions to carry forensic investigation are presented in Section IV.

1.1 Android Operating System

Android is an operating system (OS) developed by the Open Handset Alliance (OHA).The basic architecture of Android is shown in Figure 1. At its core, Android OS builds are based on the Linux 2.6kernel. When running on a hard drive, the Linux system device defaults to the first physical hard drive, or /dev/hd0. In addition, Linux only understands character and block devices, such as keyboards and disk drives, respectively. With Linux on flash, however, a Flash Transition layer provides the system device functionality. A Memory Technology Device (MTD) is needed to provide an interface between the Linux OS and the physical flash device because flash memory devices are not seen as character or block devices.

The Android Runtime System utilizes the Dalvik virtual machine (VM), which allows multiple applications to be run concurrently as each application is its own separate VM. Android applications (the apps of today's common parlance) are compiled into Dalvik executable (.dex) files. During a forensic examination one will be mainly concerned with the Libraries and, in particular, the SQLite databases. This is where one will find the majority of data that could be of interest in an investigation. Files can be stored on either the device's storage or on the removable secure digital (SD) memory card.

Unlike the typical desktop operating system, data or other files created by one Android app cannot automatically be viewed by other applications by default. The VM nature of Android allows each application to run its own process. Security is permissions-based and attached at the process level by assigning user and group identifiers to the applications. Application cannot interfere with each other without being given the explicit permissions to do so. The security mechanisms of the Android OS could impede a forensic examination although some of the basic tools and techniques could allow investigators to recover data from the device.

1.2 Android Architecture

Android is architected in the form of a software stack comprising applications, an operating system, run-time environment, middleware, services and libraries. Each layer of the stack, and the corresponding elements within each layer, are tightly integrated and carefully tuned to provide the optimal application development and execution environment for mobile devices as in Figure 1.

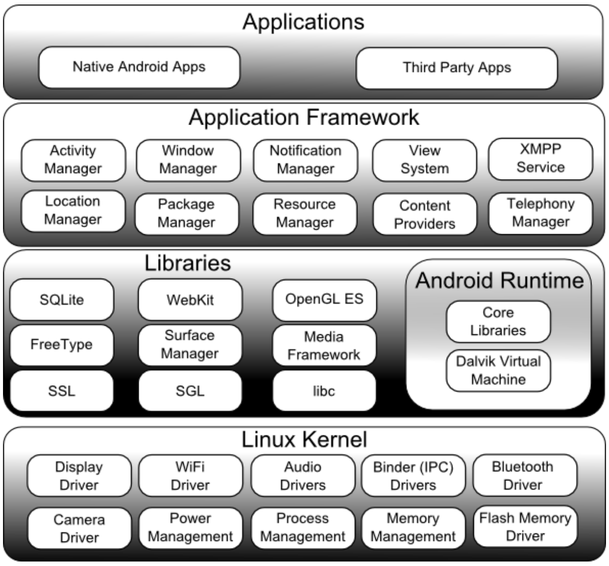


Fig 1: Android Operating System Architecture

1.3 The Linux Kernel

Positioned at the bottom of the Android software stack, the Linux Kernel provides a level of abstraction between the device hardware and the upper layers of the Android software stack. Based on Linux version 2.6, the kernel provides pre-emptive multitasking, low-level core system services such as memory, process and power management in addition to providing a network stack and device drivers for hardware such as the device display, Wi-Fi and audio.

It is important to note, however, that Android only uses the Linux kernel. I.e., it is worth noting that the Linux kernel was originally developed for use in traditional computers in the form of desktops and servers. In fact, Linux is now most widely deployed in mission critical enterprise server environments. It is a testament to both the power of today's mobile devices and the efficiency and performance of the Linux kernel which is the software at the heart of the Android software stack.

1.4 Android File system

YAFFS2 is similar in concept to YAFFS1, and shares much of the same code. The YAFFS2 code base supports YAFFS1 data formats through backward compatibility. The main difference is that YAFFS2 needs to jump through significant hoops to meet the "write once" requirement of modern NAND flash. YAFFS2 marks every newly written block with a sequence number that is monotonically increasing. The sequence of the chunks can be inferred from the block sequence number and the chunk offset within the block. Thereby when YAFFS2 scans the flash and detects multiple chunks that have identical Object IDs and Chunk Numbers, it can choose which to use by taking the greatest sequence number.

For efficiency reasons YAFFS2 also introduces the concept of shrink headers. For example when a file is resized to a smaller size, YAFFS1 will mark all of the affected chunks as dirty - YAFFS2 cannot do this due to the "write once" rule. YAFFS2 instead writes a "shrink header", which indicates that a certain number of pages before that point are invalid. This lets YAFFS2 reconstruct the final state of the file system when the system reboots.

YAFFS2 uses a more abstract definition of the NAND flash allowing it to be used with a wider variety of flash parts with different geometries, bad block handling rules etc. YAFFS2 later added support for check pointing, which bypasses normal mount scanning, allowing very fast mount times. Performance will vary, but mount times of 3 seconds for 2 GB have been reported.

2. LITERATURE REVIEW

Several works have been published for carrying out forensic activity on android OS in terms of file system analysis, study of OS architectures, processes, kernel module, rooting mechanisms, analysis of various applications, analysis of instant messaging applications, android malicious application detection and analysis and many more. Here, this paper presents some of the works carried out in this area.

In the initial study of existing android forensics methods, it is observed that special rooting scripts and methods have been used for gaining root user privileges on the device. With the gained super user privilege, imaging of required partitions and disks of the device using *dd* command through Android Debug Bridge (ADB). The acquired images are analysed using traditional forensic tools or other commercial tools [1].

A general method for digital forensics collection on android devices has been discussed through special boot methods enabling the use of custom recovery booting, data on Android devices can be collected with very little probability of corrupting user data. Use of the recovery booting provides a consistent, repeatable method of collecting numerous Android devices without "rooting" the device in normal operating mode [2].

In another method author discussed fast imaging and analysis data partition of an android device based o the yaffs2 file system. In this method a wealth of information has been recovered in a forensically sound manner in few minutes without the need of any specific tool or system [3].

Another method was proposed to address data acquisition of devices that use the Android Platform, taking into account operational system characteristics, its most popular applications and devices hardware features. In this method author discussed various tools, techniques to unlock the device, bypassing access control or mirroring partitions [4].

In another method author presented a technique to obtain complete captures of volatile memory from Android devices along with subsequent analysis of that data in both userland and the kernel. In this method author discussed about accurate physical memory acquisition and deep memory analysis of the Android kernel's structures [5].

In another method android application specific data acquisition and analysis has been discussed. In this method author presented experimental virtual setup using youwave and acquired the artefacts left by the Whatsapp instant messaging application and presented analysis of these image with respect to Whatsapp databases, logs and many more [6].

In another method of forensic acquisition and analysis of instant messaging applications (whatsapp, viber) in android platform, authors discussed on how acquire forensics image of the android device using UFED (Universal Forensic Extraction Device) and presented analysis of the evidence with UFED Physical Analyser and various manual analysis techniques [7].

Several more works have been published with respect to various social networking applications, instant messaging applications, web browsers and many other categories of applications [8]. In all of these methods customized rooting applications have been used for obtaining forensic image and presented analysis with open source tools, commercial tools and manual methods [9].

3. ANDROID FORENSIC METHODS

Android forensics is new area where various tools are emerging in this field. The tools available for android forensics are available in the form of hardware tools and software based tools [10]. Most of the tools are commercial and a few numbers of tools are open source tools. In android smart phone there are three types of memories are available. First one is phone memory where the actual operating systems partitions and OS related files. All these partitions use YAFFS2 file system.

Second internal and external memory, in internal memory all installed applications, its data, gallery and others important things. External memory is an extension to internal memory and it is used to store app related information, backup data, videos, photos and other information on user interest. Third is RAM, this memory area contains details about currently running processes, data structures and information related to communication between various running processes. This is a volatile memory and very crucial area for forensic investigation. The forensic analysis of this volatile memory is also called Live forensic analysis. Currently, this paper discusses about two major types of acquisition methods namely logical acquisition and physical acquisition.

3.1 Logical Acquisition methods and tools

In logical data acquisition data stored in the memory are acquired by using the file system or the protocol of a chip provider. Logical acquisition is a process of bit-by-bit copying of logical storage objects such as file system, directories and files.

In logical acquisition technique it extracts allocated data and is typically achieved by accessing the file system. Allocated data means that the data that is not deleted and still accessible in the file system. It is possible to recover deleted data from a logical acquisition using specialised techniques and tools but in the case of SQLite databases it contains deleted records also. There are various tools available in the market both open source and commercial for performing logical acquisition and analysis of android devices. This paper discusses some of the tools and its features.

3.1.1 SAFT

SAFT is easy-to-use mobile forensic tool used to extract valuable information from the device such as Call logs, SMS/MMS logs, contacts list, file browser logs, browser history, bookmarks, facebook and twitter logs, Youtube and instagram logs, Viber, Whatsapp and Skype logs, Email messages, location history and calendar. This forensic tool having report generation facility in Microsoft word format.

3.1.2 AFLogical

AFLogical is a logical memory acquisition tool for Android devices. The agent installed in the device will acquire the contact list, call logs, SMS, MMS and MMSParts and info.xml file details are send to the forensics workstation. In this acquisition USB debugging mode should be enabled and connect the device to the forensics workstation where AFLogical is installed. This tool acquires logical memory details of the device.

3.1.3 LiME Module

LiME or formerly called as DMD module used to dump RAM contents of the phone which will give details about recent user activities, process details, memory structures and many more volatile content. Linux Memory Extractro (LiME) is a loadable kernel module which dumps memory directory to the SD card of over network. This is an excellent tool for performing live memory analysis of android devices which gives insight into RAM dump both structured and unstructured data, encryption keys, application data, fragments of communication, open files, processes, and application, kernel and network structures.

3.1.4 Nandroid Backups

Nandroid Backups is another method for extracting the total file system of the device using NANDroid backup. These backups can be created by booting the device into a custom recovery and choosing the corresponding item in the menu. Nandroid backup can be created only if Bootloader is unlocked and custom recovery is installed or the device should be rooted, Busybox package and a Nandroid backup app should be used.

3.1.5 OSAF-TK (Open Source Android Forensics – Tool Kit)

OSAF-TK is an open source Android forensics tool kit build from Ubuntu 11.10 specifically for Android Malware Analysis. This tool kit is having all the required pre-compiled tools for code review and application analysis. With this tool kit malicious application can detected and activities of the application can be analysed.

3.1.6 Santoku Linux

Santoku Linux is a free open source software tool kit built on Ubuntu serves majorly three purposes mobile security, forensics and mobile malware analysis. SANTOKU contains set of tools for Firmware flashing tools supporting multiple manufacturers, NAND Imaging, media cards, and RAM. This bundle package contains several useful utilities and scripts specifically for mobile forensics and also contains free versions of some commercial forensics tools. The package contains emulators for mobile devices, repository of malware databases and various utilities to simulate network services in dynamic analysis. This package also contains scripts enumerating app details, automate decrypting binaries, deploy in apps, detect common security issues in apps, disassembly and recompilation tools.

3.1.7 WhatsappXtract

WhatsappXtract is a WhatsApp Backup Messages Extractor for Android and iPhone devices. These tools are able to read whatsapp chats using a backup file. In this it will read older messages, chats and other information in whatsapp backup file and can display on the computer. This tool is specifically used for Whatsapp application.

3.1.8 Andriller

It is a utility which consists of various tools for serving various purposes which includes cracking of screen lock pattern, PIN and passwords, decoding of encrypted databases and files, data extraction automatically and unpacking of android backups. This tool kit solves many of mobile forensics needs for the Android OS. In this data recovery facility is not available and data carving feature is also not available. Correlation of malicious events occurred in the conversations. The excellent feature of this tool is extraction of data from android backups without any root privileges.

3.2 Physical Acquisition and tools

A physical data acquisition from a mobile device means that a bit-for-bit copy of physical storage is extracted. This would give a forensic examiner a bit-for-bit copy of the mobile device’s flash memory; this is similar to the way data is acquired in traditional computer forensics [11]. A physical data extraction extracts the data directly from the mobile device’s flash memory. After the data is extracted, the memory dump is then decoded. This type of extraction enables the maximum amount of deleted data to be recovered. Physical data acquisition is usually the most difficult extraction type to achieve, as the manufacturers of mobile devices secure against arbitrary reading of the device’s memory. Mobile device forensic tool manufacturers often develop custom boot loaders, allowing the forensic tool to access the mobile device’s memory and, in many cases bypass pattern locks or pass codes.

3.2.1 JTAG Forensics

JTAG is a low level and advanced acquisition for acquiring raw content of the flash memory or memory chips. If the manufacturers providing whole-disk encryption, then the JTAG method will produce encrypted image or the raw content. To decrypt the image, relevant pass code needed to supply which need access to the phone's higher-level API. Once after completion of the acquisition an investigator can use any of compatible JTAG analyzers (Ex. Belkasoft Evidence Center) for retrieving call logs, SMS/MMSses, contacts, browsing history and geo-location data messenger chat histories etc. This JTAG forensics acquisition method uses standard compatible JTAG port on the devices, so this procedure can be used only on JTAG compatible devices without any whole-disk encryption.

3.2.2 Chip-Off Acquisition

Chip-off acquisition is used as a last move in the mobile forensic investigations. This method of acquisition will produce the complete binary image of the device including unallocated space of the memory. It is a lowest-level destructive and highly advanced acquisition method which needs physical de-soldering of memory chips and for reading of the memory content specialized software is required. In this method the investigator should have clear understanding of block address remapping, fragmentation, and encryption [12]. The limitation of chip-off acquisition method is that the devices with no encryption or devices using encryption algorithms with known weaknesses.

3.2.3 Cellebrite UFED Physical analyser

Unified Forensic Examination Device is hardware forensic examination device for extraction of evidences from mobile devices. CelleBrite (UFED) Communicates with a cell phone via a data cable, infrared (IR), or BlueTooth (BT). UFED can acquire data (logically and physically). UFED physical analyzer which analyzes every segment of a device's memory using advanced logical, file system and physical extractions. Using simple stand-alone method with UFED, an examiner can recover MMS/SMS messages, call logs, photos, video, and contact information [13]. UFED mainly focuses on logical extraction only. It does not recover emails, browser or search history.

3.2.4 Oxygen Forensics Suite

Oxygen mobile forensic Suite is used for logical acquisition and analysis of cell phones, PDAs and Smartphone's. This software kit is a proprietary and having registration key for forensics workstation. This mobile forensics suite can be used to extract device information, SMS messages, contacts, event logs, calendar events and files along with metadata. This tool kit supports major leading mobile platforms.

3.2.5 Paraben

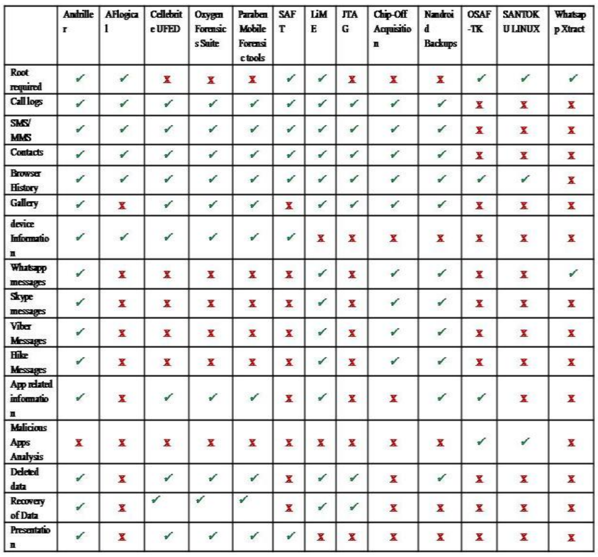
Paraben's DS7 mobile forensics tools is used for logical and physical acquisitions, password bypassing and file system extractions. This mobile forensic suite is having advanced analysis capabilities, data and application parsing, automated deleted data recovery and comprehensive reporting capabilities. This forensic suite is a commercial software toolkit. Paraben's JTAG Analysis Tool is another tool to examine wide variety of data extracted from mobile devices using a JTAG extraction. This tool is compatible with JTAG memory dumps that are created using the RIFF Box JTAG hardware tool.

4. COMPARISON OF ANDROID FORENSICS TOOLS

This paper discussed about many of the tools for performing forensics activity on android devices in terms of logical and physical acquisition of device memory. Some of the tools perform only logical acquisition and other tools perform physical acquisition [14]. In mobile forensics and especially in android the main focus is on call logs, SMS/MMS, contacts, Browser history, gallery, device information, social messaging and application related data [15]. All the available tools are compared against all mostly focused areas. Below Table 1 represents the comparison of various Android forensics tools based on the functions available in the tools.

Table 1 describes the list of forensic tools with required features in any forensics investigation. With the commercial tools like Cellebrite UFED, Oxygen Forensics Suite and Paraben mobile forensics tools only limited basic observation is done. Detailed study and analysis of these tools are not considered in this paper. Other tools and toolkits are open source. Some of the tools providing common forensic investigation features but many are specific to limited features. Andriller is an open source tool for both Android and iOS covering many investigation areas and having reporting facility also. Open source tool kits like OSAF-TK and SANTOKU LINUX are mainly used for the purpose of android malware analysis.

Table 1: Comparison of tools with respect to various features



	Andriller	AFlogical	Cellebrite UFED	Oxygen Forensics Suite	Paraben Mobile Forensic tools	SAFT	LiME	JTAG	Chip-Off Acquisition	Nandroid Backups	OSAF-TK	SANTOKU LINUX	Whatsapp Xtract
Root required	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	✓	✓	✓
Call logs	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
SMS/ MMS	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Contacts	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗
Browser History	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Gallery	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✗	✗
device Information	✓	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗
Whatsapp messages	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✓
Skype messages	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗
Viber Messages	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗
Hike Messages	✓	✗	✗	✗	✗	✗	✓	✗	✓	✓	✗	✗	✗
App related information	✓	✗	✓	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗
Malicious Apps Analysis	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗
Deleted data	✓	✗	✓	✓	✓	✗	✓	✓	✗	✓	✗	✗	✗
Recovery of Data	✓	✗	✓	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗
Presentation	✓	✗	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗

5. DISCUSSION

This paper deals with the survey of various Android forensics techniques and tools. In this forensics methods are discussed with respect to logical and physical acquisition process. Authors discussed about various tools in both the categories by studying the functionalities existing in the tools and drawbacks. Major tools are capable to provide required results for cyber crime investigation and the evidence and analysis results are acceptable in the court of law. Using these tools the tests are repeatable until unless the evidences are not tampered.

6. CONCLUSION AND FUTURE WORK

This paper also studied various free and commercial mobile forensics tools concentrating on Android devices specifically. Due to availability of vast number of models and manufacturer specific customizations many of existing tools does not support all the devices and does not have the same steps to carry out digital investigation process.

Each tool is having its own procedure to acquire and analyse the data in forensically sound manner. A forensic investigator should have knowledge of hardware and software and detailed understating of the architecture to select appropriate tool for the required task. The fastest growing and increased use of Android platform urging the digital forensics community to develop a standard framework to facilitate digital investigation of mobile device, PDAs, tablets and other Android enable devices.

7. REFERENCES

[1] Jeff Lessard and Gary Kessler, “Android Forensics: Simplifying Cell Phone Examinations”, Small Scale Digital Device Forensics Journal Vol. 4, No.1, ISSN# 1941-6164, September 2010.

[2] Timothy Vidas, Chengye Zhang, Nicolas Christin, “Toward a general collection methodology for Android devices”, 2011.

[3] Andre Morum de L. Simao, Fabio CausSicoli, LaertePeotta de Melo, “Acquisition of Digital Evidence in Android Smartphone”, 2011.

[4] Joe Sylve a, Andrew Case b, Lodovico Marziale b and Golden G. Richard, “Acquisition and analysis of volatile memory from android devices”, Digital Investigation 8, 175–184, 2012.

[5] Cosimo Anglano,”Forensic analysis of WhatsApp Messenger on Android Smartphones”, Digital Investigation, 1–13, 2014.

[6] Aditya Mahajan, Dahiya and Sanghvi, “Forensic Analysis of Instant Messenger Applications on Android Devices”, International Journal of Computer Applications (0975 – 8887), Volume 68– No.8, April 2013.

[7] Mohammad Iftekhar Husain, Ramalingam Sridhar, “Forensic Analysis of Instant Messaging on Smart Phones”, 2010.

[8] Noora Al Mutawa, Ibrahim Baggili and Andrew Marrington, “Forensic analysis of social networking applications on mobile devices”, Digital Investigation 9, S24–S33, 2012.

[9] Alfred Kobsa, Sameer Patil, Bertolt Meyer, “Privacy in Instant Messaging: An Impression Management Model”, 2012.

[10] Andri P Heriyanto,”Procedures and Tools for Acquisition and Analysis of Volatile Memory on Android smartphones”.

[11] Nedaa Al Barghouthy, Andrew Marrington and Ibrahim Baggili, “The Forensic Investigation of Android Private Browsing Sessions using Orweb”, 5th International conference on Computer Science and Information Technology(CSIT), 2013.

[12] Juanru Li, DawuGu and Yuhao Luo, “Android Malware Forensics: Reconstruction of Malicious Events”, 32nd International Conference on Distributed Computing Systems Workshops, 2012.

[13] Howard Chivers, “Private browsing: A window of forensic opportunity”, Digital Investigation 11, 20–29, 2014.

[14] KiavashSatvat, Matthew Forshaw, Feng Hao and Ehsan Toreini, “On the privacy of private browsing e A forensic approach”, Journal of information security and applications, 1-1 3, 2014.

[15] Vaibhav Rastogi, Yan Chen, and XuxianJiang,”Catch Me If You Can: Evaluating Android Anti-Malware Against Transformation Attacks”, IEEE Transactions On Information Forensics And Security, VOL. 9, NO. 1, 99, January 2014.