2015 International Conference on Advances in ICT for Emerging Regions (ICTer): 250-258

ANDROPHSY – Forensic Framework for Android

Indeewari U. Akarawita[#1], Amila B. Perera[*2], Ajantha Atukorale[#3]

[#]*University of Colombo School of Computing*
No 35, Reid Avenue, Colombo 00700, Sri Lanka
[1]indeewariua@gmail.com
[3]aja@ucsc.cmb.ac.lk

[*]*TechCERT, A Division of LK Domain Registry*
545/4, De Soysa Rd, Molpe, Moratuwa, Sri Lanka
[2]amilads@gmail.com

***Abstract*— There have been many studies addressed towards various sections in the arena of Android platform. However conducting a digital forensics investigation using Open Source tools in Android platform is still a challenging task. As of the date of this study, there is no standard Open Source or non-commercial tool available to support the entire life cycle of the digital forensics investigation methodology. In order to bridge this gap in open source forensics tool for Android platform, the work implemented a user friendly, feature rich, open source, mobile forensic framework for Android platform that supports all phases of a full digital forensic investigation process.**

***Keywords*— Android forensics, data acquisition, evidence analysis**

I. INTRODUCTION

Nowadays smartphones are like handheld computers. They contain a diverse set of data; from personal to professional life. In an incident, smartphones can provide a wealth of evidence to prove or disprove facts of the incident. However evidence needs to be collected in a way that they cannot be challenged in a court of law. The process of gathering digital evidence from a mobile phone, under forensically sound conditions, using well accepted methods is defined as mobile forensics [1]. A digital forensic investigation process comprises of four main steps. Firstly preservation involves identifying, documenting and seizure of mobile device. Data collection is the process of obtaining raw data from mobile device. Analysis phase is where meaningful digital evidence relevant to the case will be extracted from raw data using scientific methods. The reporting phase presents examined and analysed evidence conducted in the previous step in a meaningful way [1].

Conducting a mobile forensic investigation on smartphone is a daunting task without proper supportive tools. Forensic investigation processes and methods involved vary across mobile platforms, device model and manufacturers. There is a lack of knowledge and supportive forensic tools for Android based smartphones in the forensics field. Continuous rapid growth of Android smartphone market share necessitates the existence of an Android forensic framework that supports all phases of the forensic investigation process.

Although there are few commercial tools [2] – [4] supporting the life cycle of forensic investigation, for Android smartphones, investigators especially from developing countries cannot afford the exorbitant cost of these tools. On the other hand free tools [5], [6] do not provide a complete solution and research [6] - [14] in the field address very specific tasks, such as particular application analysis or data collection, out of the complete digital forensic investigation process.

Though there is an apparent need, there has been no complete open source forensic framework for Android smartphones, which will support investigators in all four phases of forensic investigation, till this study. This paper addressed this problem and has provided a solution "ANDROPHSY" a forensic framework for Android. It provides forensic case management and supports forensic investigators in all four phases of mobile forensic investigation, through maximized data acquisition to evidence presentation, while maintaining high forensic validity of evidence.

II. RELATED WORK

Data acquisition has been a hot topic in Android forensic field. Vidas et al. have proposed a general collection methodology based on recovery image [7]. Although use of recovery image supports both logical and physical acquisition, it is controversial in terms of a general solution as recovery image depends highly on device vendor, Android version, and kernel version. The study [8] has implemented a cloud based Android application for evidence acquisition. The application is downloaded on demand on to the target device from a service cloud and the collected evidence report is stored on the cloud. Though use of cloud provides an

978-1-4673-8270-0/15/$31.00 ©2015 IEEE

ubiquitous solution, it breaches forensic validity of evidence. Use of a network overrides network buffers of the target device and there is a high possibility of downloading malwares onto the device which might destroy evidence needed for the major phases [1].

Use of cloud as storage for collected evidence violates privacy and integrity of the evidence, because any person who can gain access to the cloud can view, alter or destroy the evidence. File copying is another acquisition technique. Ahamed et al. have implemented an Android application, copying files from device internal memory to SD card preserving the same directory structure [9]. Aouad and Kechadi have performed a physical acquisition of Android smartphone using *nanddump* utility [10]. According to their approach, they have used the SD card to store the acquired memory dump. Nandroid [6] is custom recovery based Android device backup method which backups onto the device or SD card. AFLogical [5] is a well-known open source logical acquisition Android application which uses content provider extraction. It acquires data onto the SD card. It accesses shared data from a predefined set of applications and the installation process prompts user to grant access to the data. Use of SD card does not fit in most incidents as many smartphones users use a mounted SD card on their devices to expand storage capacity of devices. Replacing mounted SD card needs removing device battery.

Most of the work in examination and analysis has focused on Android social networking applications such as Facebook [11], Whatsapp [12], [13] and Viber [12]. In application analysis SQLite database files have been the main source of evidence. Chang et al. has proposed a novel algorithm for data recovery on NAND storage [14]. However they have not discussed about a practical implementation and evaluation of the algorithm.

Each study uses a diverse set of tools (e.g. Cellebrite Universal Forensic Extraction Device-UFED and Analyzer [2], SQLite browser), command line utilities (e.g. *dd* utility) to accomplish data collection and analysis. Though rooting is a prerequisite in many of the studies, it has not been discussed broadly. They [11] have used root kits such as SuperOneClick [15], SRSRoot [16], and CF-AutoRoot [17] all of which are available on web, whose provenance and impact on collected evidence is unknown to the public.

UFED [2], Oxygen forensic suit [3] and Micro System XRY [4] provide complete commercial hardware – software solutions. They support wider range of device including rooting functionality. These tools are expensive and are not affordable to many forensic authorities.

III. Android Architecture

The Android operating system is based on the Linux kernel. It organizes internal memory in partitions. Partitions are typically mapped to logical devices that are mounted in a specific path on the file system and associated with a specific file system type [18]. Partition and device mappings depend on vendor implementation. Recovery, boot, system, cache and user data partitions are common in every implementation. Recovery and boot partitions hold recovery and boot images and routines. User data partition contains user application specific information. Android maintains a strict directory structure to store application data. Application data resides inside '/data' folder. Android uses SQLite databases files to share application data with other applications. These are called content providers. Application SQLite files resides in 'data/data/<application>/database' folder [18]. Basic Android native application names such as 'com.android. provider.telephony' and folder paths are independent of Android versions and vendor implementation.

IV. Androphsy Forensic Framework

ANDROPHSY manages forensic incidents inside cases. The first step of ANDROPHSY forensic investigation system is to create a case for the incident. A case can have one or more Android devices. The framework also facilitates backup and archiving of cases.

The framework maintains personal user accounts for each user to maintain accountability and privacy. Every user is assigned a role. The user roles defined in the framework are analyst, investigator and administrator. The Analyst is only allowed evidence manipulation and report generation for permitted cases. Investigator role is granted to execute case management functionalities, data collection, evidence extraction, analysis and reporting functions. Administrator role has investigator privileges and in addition the role is responsible for user management activities such as create user accounts, assign roles, revoke or delete

user accounts and case authority transfer between users.

Every transaction in the framework is recorded in a log. The log provides audit trails of the whole investigation process.

V. Data Collection

There is a deep division among mobile forensic examiners where one group thoroughly relies on a traditional approach in which no alterations made to the device in examination and consequently retrieve only a nominal amount of data from the device while the other group takes a radical step of making minimal possible changes to the target device with the intension of collecting maximum amount of valuable data from the target device [18]. Although the later approach is debatable due to the forensic soundness of the collection method. The data collection limitations in the traditional approach and necessity of collecting invaluable evidence from modern smartphones negate opposing arguments. Such changes are permissible as long as the explanation of relevance and implication of changes are made [19].

Android forensics field experts argue data collection through device recovery image is more forensically sound than other methods [18]. However the process has limitations and is barely adapted to a context of limited resources. There is no universal recovery image that can be flashed to every Android device as it is device model, hardware profile, Android version, kernel version, build version and firmware version specific. Creating a recovery image with built in acquisition capabilities requires deep technical knowledge and experience in Android internals and related subject matter [7]. The recovery flashing process is inherently risky and therefore created recovery images should be thoroughly tested on a device with exact device profile before deploying on target device. It is impractical due to the huge permutation of Android smartphones in the market. For these reasons we pursue a modern approach in ANDROPHSY where we use low level Linux and Android built-in forensic functionalities such as *dd* and *adb* commands, which fits any situation.

The framework uses Android Debug Bridge (adb) [20] to communicate with an Android smartphone connected to the workstation over Universal Serial Bus (USB). Therefore USB debugging and 'allow unknown source' options in device settings properties have to be checked as a prerequisite.

A. Device Rooting

Android security model restricts user level processes from accessing some parts of Android file system, and installing and running binaries due to safety and commercial reasons [18]. Rooting circumvents platform restrictions and allows users to fully leverage the capabilities of the Android platform. Rooting is a mandatory requirement in Android forensics as it allows access to the root folder of the Android file system where all the evidence is, installing external binaries such as *netcat* which serves file transferring purposes in acquisition process and execution of low level kernel functionalities such as *dd*.

Rooting in forensics is a challenging task. It changes the device state and therefore it is unable to employee widely available proprietary rooting methods [15] – [17] on the Internet for Android forensics. Impact of rooting on the user data partition should be minimized, the method should not be associated with a risk of breaching target device and it should be possible to revert the rooting so that the device is brought back to its original state and be made operational after examination. Many security vulnerabilities leading to privilege escalation are present in almost all implementations of the Android application framework, libraries [21] and beneath in the Linux kernel [22].

ANDROPHSY implemented scripts for common exploits CVE–2013-6282 (i.e. safe root) [21], Argon, Gandalf, zergRush and psneuter to gain root access on target device. Scripts install *su* and *netcat* binaries on device system partition. Use of common exploits with known provenance guarantees no malicious intent and no harmful impact to user data partition. When examiner decides the exploit the framework executes it on the device over adb shell. If the exploit is not successful the script will undo installed binaries and bring device to its original state. Once superuser access is granted the script enables build properties ro.debuggable and persist.service.adb.enable to enable adb daemon in root mode which has no impact on user data partition thus preserving the user data integrity requirement. Use of multiple exploits ensures framework support for a wide array of Android smartphones.

Successful rooting turns shell prompt from $ sign to # sign upon execution of *su* command on adb shell [18]. The implementation uses this

property in root access verification together with existence of *su* binary in /system/xbin folder of device file system.

A. Physical and Logical Acquisition

Though early Android smartphones used Yet Another Flash File System (YAFFS) on a NAND storage, with the release of Gingerbread version Android moved to Ext4 file system in early December 2010 [23]. As a result Android storage technology changed to embedded Multi-Media Cards (eMMC). According to Google's statistics of Android version distribution [24], Gingerbread and later Android versions – Jelly Bean and Ice Cream Sandwich have a very high density in the smartphone market. Therefore ANDROPHSY forensic framework caters to the latter file system and storage technology.

Android forensics has two types of acquisitions. Physical acquisition creates a close to exact copy of device internal memory [18]. Typically this copy contains unallocated space hence deleted data which is not accessible through device file system. Logical acquisition is known as file system partition acquisition and carries out acquisition only on existing data in the device file system. The framework provides physical and logical acquisitions of user data, system and cache partitions.

The *dd* command is a built in command line utility used to obtain the raw image of physical drives. Typical command line use of *dd* takes if: input file, of: output file and bs: block size as parameters. Output location is usually a location on the device file system such as the SD card. The framework implementation uses *dd* to read blocks from the specified partition and adb port forwarding technique with *netcat* utility to write blocks on to the case folder of forensic workstation over the network. This implementation strategy avoids the disadvantages of using SD cards as a collection target discussed in section II. We implemented a socket program using port 31337 that takes partition logical device as the input file and block size as 1024 bytes. The implementation dynamically reads the logical device and mount point mappings of source partition from mount information. The output is in universal .dd format as opposed to proprietary formats. Therefore it can be imported into external tools for further analysis if required.

adb pull is an Android built in command for copying folder structure from device file system to forensic workstation. The framework uses *adb pull* in its logical acquisition implementation. ANDROPHSY reads mount point information at runtime to identify copy from location and copy complete directory structure to case folder of forensic workstation.

B. Log Collection

Android kernel and system logs are another source of evidence. It contains kernel events and timestamp information, application memory utilization and their process ids, account information including device sync history, accounts configured in the device and cellular radio information such as tower (location) data, encrypted SMS and low level AT commands and wireless carrier network information. The framework supports collection of kernel log, memory information log, user activity log, account and sync log and location log from the device in the form of a text file. A study on the detail of log formats in Android platform was done, and the implementation used Linux kernel commands *logcat*, *demsg* and *dumpsys* for collecting logs.

C. Data Collection Integrity

Evidence integrity is a significant dimension in forensics as it describes the need of evidence to be intact and not tampered during acquisition and analysis. Integrity is verified through hash value checksum. Though computer forensics calculates hash value of evidence during acquisition, mobile forensic deviates from the practice. Experiments showed hash values calculation at two successive acquisitions under same conditions generated two different hash values. Further hex value analysis of collected raw images explained this fact; smartphones have embedded storage medium and device internal processes such as clock which keep running during acquisition causing generation of different hash values. Therefore the framework maintains integrity on acquired raw data during analysis. Implementation calculated MD5 hashes of each of the collected raw evidence and it is verified in subsequent analysis. MD5 hash is 128 bit length and faster than other hash functions SHA256 [26]. Yet it is computationally hard to find a collision given the hash value.

VI. Examination and Analysis

Device profile is the preliminary evidence in mobile device forensics. Device profile describes device properties such as android version, kernel

version, device name, device configurations MAC address and installed application details. Device properties are stored in a property file as key value pairs in system partition. The framework implementation reads the property file using *adb getprop* command and decodes the raw output. Installed applications information including application name and path, enabled apps, disabled app, system apps, third party apps and uninstalled apps are retrieved through framework calls to adb package commands.

ANDROPHSY evaluates Android application files (.apk) collected through logical acquisition for malicious intents. National Software Reference Library (NSRL) has published hash values of computer and mobile device applications. Collection of Android applications hashes is trivial and outdated [27]. Therefore the system built a sample MD5 hash repository of up-to-date Android applications files. Repository maintains two files where one file contains hashes of known good Android applications while other file stores hashes of known malicious files. On demand ANDROPHSY compares .apk file MD5 hashes against repository to identify good files, malicious files and unknown files. Therefore the forensic examiner can restrict their investigation to identified malicious or unknown .apk files in certain situations.

ANDROPHSY decodes user account credentials, contacts, messages, calendar events, YouTube, Gmail, event logs, browser applications as default browser, Google Chrome and Firefox, social networking applications as Facebook, Linkedin, Skype, Bluetooth information and Wi-Fi information including SSID and passwords. Android stores passwords in plaintext except the device's primary Google account password. In general, application analysis is mostly based on SQLite database files. The framework processes SQLite files through JDBC connection. Bluetooth and Wi-Fi information are stored in configurations files.

The implementation facilitates file carving, feature extraction, keyword and feature search. ANDROPHSY file carving is built upon Scalpel [28]. File carving matches header and footer byte patterns defined in a configuration file. Configuration file is generated at runtime with the examiner choosing file types. The framework plots a frequency distribution bar chart for credit card numbers, phone numbers, URLs, search text, IP address, Facebook name and id, email domains from the extracted raw image. The X-axis holds feature data while Y-axis contains frequency components. The Graphs give instant knowledge about possible evidence from the device. ANDROPHSY searches given keywords in the raw image. It generates text files extracting all strings from raw image with their byte offset in image at the first search and use of the file in subsequent search improving search performance. Besides keywords it facilitates contact number, name, email and phonographic text search. ANDROPHSY has built-in SQLite decoder to facilitate analysis of custom applications. Activity time line, file browser and hex view are built-in features.

VII. REPORTING

The results of forensic investigation are detail in reports. Report is the ultimate outcome of a forensic investigation which supports verdict. Confidentiality, accountability, integrity and reliability are key features of a forensic investigation reports. ANDROPHSY provides all of these features through its automated and flexible report generation process.

ANDROPHSY stores all extracted evidence in a MYSQL database. The reporting module reads evidence from the database and generates reports in .pdf formats. The examiner is allowed to mark evidence as favourites during analysis and generate reports from favourite evidence. The implementation use iText for processing reports.

Each and every page of a report consists of header information with confidentiality information. Report includes system logged in user, report generation date, case detail and device detail proving report accountability and integrity. Significant of ANDROPHSY report generation mechanism is it enforces providing relevant evidence that support verdict rather than providing forensic examiner opinions through automated report generation techniques.

VIII. ANDROPHSY LOGICAL ARCHITECTURE

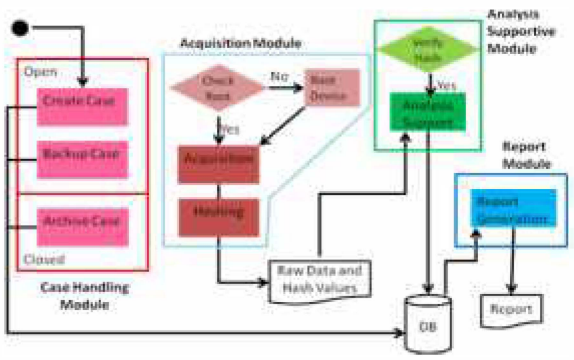

Fig. 1 ANDROPHSY architecture

Figure 1 illustrates ANDROPHSY architecture. It consists of four major modules. They are case handling, acquisition, analysis support and reporting module. Case handling module provides case specific functionalities such as case creation, backup archive. Acquisition module implements data collection functionalities as described in section V and analysis module consists of all data examination and analysis routines described in section VI. Reporting module provide report generation functionalities. ANDROPHSY modularized architecture provides rooms for future extensibility.

IX. Evaluation

ANDROPHSY was evaluated for quantitative and qualitative factors taking free versions of reputed Android forensic tools as references. Reference tools used were:

- Oxygen Forensic Suit 2014 standard 6.2.1.103 version
- ViaExtract Community Edition (CE) version 2.5 from Via Forensic, now known as NowSecure.

Evaluation criteria were

- Number of features supported by each tool and amount of data extracted by each tool
- Forensic validity of each tool
- User friendliness of each tool
- Performance

ANDROPHSY was evaluated for Android versions 4.0.3 Ice Cream Sandwich (ICS) and 4.1.2 Jelly Beans (JB). According to Google's dashboard of Android platform version distribution, Jelly Bean had reached close to 60% while ICS was 16.9% as of January 2014 [24].

Samsung leads the smartphone market share from a manufacturing perspective and HTC is at a median position in the distribution [25]. Therefore Samsung and HTC smartphones were chosen as the test bed. Table 1 gives the profile detail of Android smartphones that were employed in the evaluation process.

TABLE I
Test Device Profile

Property	**Samsung GT-I8262**	**HTC EVO Design 4G**
Device Name	Samsung Duos Core	HTC EVO Design 4G
Model	GT-I8262	PH44100
Android Version	4.1.2	4.0.3
Build Number	JZO54K.18262XXA MG2	1.05.653.3 CL367264 release-keys

None of the devices had been rooted and test data was created explicitly for evaluation. Environmental factors such as device, connected cables, device root state were kept constant at each tool evaluation. However device states were changed during each tool evaluation as devices reboot few times during rooting and un-rooting processes.

A. Evaluation Result

1) Feature Based Evaluation

We studied reputed forensic tools and identified thirty three (33) features that would help in mobile device forensics. ANDROPHSY supported thirty one (31) features (94%), Oxygen Forensic Suit supported ten (10) features (30%) and ViaExtract CE supported twelve (12) features (36%). Results remained the same for both Android version and manufacturer. Table II shows details of feature based evaluation.

The amount of data extracted deviated greatly for browser artefacts, messages and event log. ANDROPHSY extracted received and sent messages while reference tools supported only received messages. Both of the reference tools supported only default browser where ANDROPHSY also supported Chrome and Firefox. ANDROPHSY decoded in and out SMS and MMS message logs, missed, outgoing, incoming and cancelled calls log. Oxygen failed to extract cancelled calls. ANDROPHSY extracted and decoded JSON represented data from Facebook messages, contacts and notification. ViaExtract CE extracted only Facebook messages and result had not been decoded.

TABLE II
Details of Feature Based Evaluation

Feature	ANDROPHSY	Oxygen Forensic Suit	ViaExtract CE
Case Handling	Yes	No	No
User Access Control	Yes	No	No
Rooting	Yes	No	Yes
Physical Extraction	Yes	No	No
File System Partition Extraction	Yes	No	Yes
Android Content provider Extraction (No rooting)	No	Yes Limited	Yes Limited
Evidence Hash	Yes	Yes	No
Keyword Search	Yes	Yes	No
File Carving	Yes	No	No
String Extraction	Yes	No	No
Device Detail	Yes	Yes	Yes
Application Detail	Yes	No	No
User Credential	Yes	No	No
Phone Book	Yes	Yes	Yes
Messaging	Yes	Yes	Yes
Calendar Events	Yes	Yes	Yes
Gmail	Yes	No	No
Google plus	No	No	Yes

2) Forensic validity

Forensic validity describes evidence credibility. Forensic validity is measured in terms of evidence integrity, authenticity and privacy. Oxygen and ANDROPHSY provided evidence integrity check via hash checksum verification. Hash verification function could not be observed in ViaExtract CE. Oxygen installed an agent application on the device to collect data which modifies the user data partition. ViaExtract CE and ANDROPHSY used exploits in kernel to gain root access which minimized changes to user data partition. Neither Oxygen nor ViaExtract CE provided case handling and user access control. Consequently they did not support evidence privacy and authentication. ANDROPHSY provided both authenticity and privacy through user access controls and case management. Therefore ANDROPHSY preserves forensic validity more than the reference tools.

3) User friendliness.

Ease of installation, use and measures to mitigate human errors were used to evaluate framework user friendliness. Though Oxygen is easy to install it is built into two separate executables for extraction and analysis. ANDROPHSY is a single .jar file with configuration scripts which need to be installed seperately. ViaExtract CE is a Virtual Machine which needs third party virtual environment software. Oxygen's representation of evidence is more precise than the other two tools. ViaExtract CE did not decode well and presented evidence such as Facebook messages while ANDROPHSY decoded and represented extracted evidence in a comprehensive manner. Though Oxygen is easier to install and use than ANDROPHSY, ANDROPHSY's backup and archive functionalities add more value to the framework.

4) Performance

Application start-up time, time taken for logical data collection and time taken for report generation were considered in performance evaluation. ANDROPHSY start-up time was on average 11 seconds, Oxygen took on average 35 seconds and ViaExtract CE Virtual Machine start-up time was on average 62 seconds. ANDROPHSY logical data collection took on average 77 minutes, ViaExtract CE took on average 48 minutes and Oxygen took on average 28 minutes. Reports were generated in .pdf format for same set of data. ANDROPHSY spent an average 35 seconds and Oxygen spent an average 19 second for report generation. Start-up time depends highly on underlying workstation hardware. Therefore based on the statistics ANDROPHSY has less performance than the other two tools.

Feature	ANDROPHSY	Oxygen Forensic Suit	ViaExtract CE
YouTube	Yes	No	No
Facebook	Yes	No	Yes
LinkedIn	Yes	No	Yes
Browser Artifacts	Yes	No	Yes
Bluetooth Information	Yes	No	No
Wi-Fi Access Points	Yes	No	No
Event Log	Yes	Yes	No
System Log	Yes	No	No
TimeLine	Yes	No	Yes
File Browser	Yes	Yes	Yes
SQLite Viewer	Yes	No	No
File Filter	Yes	No	No
Reporting	Yes (Basic)	Yes (Advance)	No
Audit Log	Yes	No	Yes

It is very hard to evaluate tool performance as their capabilities varied a lot. Data collection technique is the heart of any forensic tool as it

boosts the amount and validity of acquired evidence. ANDROPHSY performed physical and complete logical partition acquisition while Oxygen collected data from applications content providers. Therefore Oxygen did not support much evidence that were supported by ANDROPHSY as shown in table II.

B. Evaluation Outcome

Based on the above evaluation, the tools were ranked for each of the criteria. Ranks were categorised as best, moderate and poor. Table III illustrates the summary of the evaluation results.

TABLE IIII
EVALUATION RESULT SUMMARY

Property	ANDROPHSY	Oxygen	ViaExtract CE
Feature based evaluation	Best	Moderate	Poor
Forensics validity	Best	Moderate	Poor
User friendliness	Moderate	Best	Poor
Performance	Poor	Best	Moderate

According to the analysis of evaluation result ANDROPHSY has a better capacity in feature based evaluation and forensic validity of the investigation processes. Though, the framework should be fine tune for better performance which is a critical factor in a time constraint forensic investigation. The evaluation concluded that having graphical representation of data than a textual representation would be gain more user friendliness to the final outcome of the framework.

X. CASE STUDIES.

Forensic examinations were carried out on two anonymous cases using the three tools to evaluate their practical capacities.

Case 1: BYOD policy of ADS Ltd. prohibits connecting personal smartphones to company Wi-Fi. Company policy does not allow social networking activities, and adult content web surfing during office hours. After network security system alarmed for unauthorized Wi-Fi usage, a device had been taken into custody. The goal was to find evidence of company policy violation.

Oxygen Forensic Suit and ViaExtract CE failed to prove the fact that the device had connected to company Wi-Fi as they did not support extraction of device Wi-Fi information. ANDROPHSY extracts wireless artefacts and it had extracted company Wi-Fi SSID and provided password. ViaExtract found browser artefacts for default browser. ANDROPHSY extracted more evidence as it supports many browsers.

Case 2: A man arrested after a girl made a complaint that he has used his smartphone to take photographs of her without permission at a public place. The goal was to find evidence.

Image folders were searched for evidence of pictures but there was no such image in device file system. Therefore we suspected that the images might have been deleted. As Oxygen Forensic Suit and ViaExtract CE tools do not support file carving, it was unable to find any evidence. ANDROPHSY executed file carving functionality for image files including jpeg, png, gif. It carved several thousands of images and there was evidence for the reported case.

XI. CONCLUSION

Forensic investigators who do not have access to expensive commercial Android forensic tools need lot of efforts to accomplish their job. Though there are impressive research findings, the Android forensic field lacks collaboration among findings, and has reduced their worth. In this study we implemented an open source forensic framework for Android smartphones – ANDROPHSY to serve the open source Android forensic community with the powerful features that are not available on any free solutions. ANDROPHSY supports forensic investigator in all four phases of a mobile forensic investigation. Study was based on a modern approach where we made minimum changes to the device to extract comprehensive set of evidence. Traditional forensics avoids changes to the device. However due to the Android architecture, valuable evidence cannot be collected in the traditional approach. Therefore this study established the fact that 100% forensic soundness is not possible in Android forensics. Evaluation results and Table II and III show us ANDROPHSY is a comprehensive open source forensic framework for Android and has succeeded in addressing the research problem.

The first version of ANDROPHSY including source code is available at Github [29] under GNU General Public License (GPL) license. Feedback is welcome and we invite community to contribute to further development of ANDROPHSY.

XII. FUTURE WORK

In successive versions we wish to:

- Conduct a comprehensive performance evaluation

Performance is an important aspect in any software solution. ANDROPHSY's maximized data collection resulted in degraded performance in data collection comparatively with reference tools which support limited data collection. There is a need for standard comprehensive performance evaluation, in which ratio of data collection speed is a criteria of measurements.

- Improve device support through more root exploits

Rooting maximise data collection. As there is no general method for Android smartphone rooting, implementing more root exploits converges ANDROPHSY towards more general solution.

- Provide content provider extraction for devices framework where rooting is not supported

Rooting is not a prerequisite in content provider extraction. Though it extracts only limited set of data having such feature enhances usability of ANDROPHSY.

- Implement unknown binary and malware analysis
- Advanced phonographic content search based on URL lookup

ACKNOWLEDGMENT

Authors wish to thank the University of Colombo School of Computing for providing infrastructure for the study and the digital forensic community and especially Dr. Kasun De Zoysa for various contributions to the study.

REFERENCES

[1] R. Ayers, S. Brothers, W. Jansen, "Guidelines on Mobile Device Forensics", NIST SP 800-101 Revision 1, National Institute of Standards and Technology, USA, May. 2014.

[2] (2014) Cellebrite website. [Online]. Available: http://www.cellebrite.com/mobile-forensics

[3] (2014) Oxygen Forensics website. [Online]. Available: http://www.oxygen-forensic.com/en/

[4] (2014) Micro System XRY [Online]. Available: http://www.msab.com/.

[5] (2014) ViaForensics website [Online]. Available: https://viaforensics.com/resources/reports/androidforensics/

[6] (2014) NANDroid. [Online]. Available: http://forum.xda-developers.com/wiki/NANDroid

[7] T. Vidas, C. Zhang and N. Chrisitin, "Towards a General Collection Methodology for Android Devices", *The International Journal of Digital Forensics & Incident Response,* vol. 8, pp S14-S24, Aug. 2011.

[8] C. Yang and Y. Lai, "Design and Implementation of Forensic Systems for Android Devices based on Cloud Computing", *Applied Mathematics and Information Science,* vol. 6, pp 243S-247S, Jan 2012.

[9] R. Ahamed, R. V. Dharaskar, V. M. Thakare, "Degital evidence extraction and documentation from mobile devices", *Internation Journal of Advanced Research in Computer and Communication Engineering,* vol 2, Issue 1, pp 1019 -1024, Jan 2013.

[10] L. M. Aouad and T. M. Kechadi, "Android Forensics: a Physical Approach", in *Proc. SAM'12*, 2012, paper, p. 311 -315.

[11] N. A Matuwa, I. Baggili and Al Marrington, "Forensic analysis of Scocial networking applications on mobile devices", *Digital Investigation,* vol. 9, pp S24-S3, 2012

[12] A. Mahajan, M.S. Dahiya and S.P. Sanghvi, "Forensics Analysis on Instant Messanger Applications on Android Devices", *International Journal of Computer Applications*, vol. 68, pp 38-44, April. 2013.

[13] N. S. Thakur, "Forensic Analysis of WhatsApp on Android Smartphones," M.Sc. thesis, University of New Oeleans, New Orleans, LA, Aug. 2013.

[14] X. Chang, X. Tang and J. Wu, "Forensic research on data recovery on android smartphone" in *Proc. 2nd ICCSE , 2013,* paper, p. 1188.

[15] (2014) One Click Root. [Online]. Available: http://www.oneclickroot.com/.

[16] (2014) SRS One Click Root for Android. [Online]. Available: http://www.srsroot.com/

[17] (2014) CF-Auto Root Repositary. [Online]. Available: http://autoroot.chainfire.eu/

[18] A. Hoog, *Android Forensics, Investigation, Analysis and Mobile Security for Google Android*, 1st ed., USA: Elsevier, 2011.

[19] J. Williams, "ACPO Good practice Guide for Digital Evidence", Metropolitan Police Service, Association of chief police officers, GB, version 5, 2011.

[20] (2014) Android Debugging. [Online]. Available: http://developer.android.com/tools/debugging/index.html.

[21] (2014) Android Security Vulnerabilities (Gain Privilege). [Online]. Available: http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/opgpriv-1/Google-Android.html

[22] (2014) Linux Kernel Security Vulnerabilities (Gain Privilege). [Online]. Available: http://www.cvedetails.com/vulnerability-list/vendor_id-33/product_id-47/opgpriv-1/Linux-Linux-Kernel.html

[23] (2010) T. Theodor, Android will be using Ext4 starting with Gingerbread, [Online]. Available: http://www.cvedetails.com/vulnerability-list/vendor_id-1224/product_id-19997/Google-Android.html

[24] (2014) Android distribution numbers update for January 2014. [Online]. Available: http://www.droid-life.com/2014/01/11/android-distribution-numbers-update-for-january-2014-jelly-bean-closing-in-on-60/

[25] (2014) Smartphone vendor market share. [Online]. Available: http://www.droid-life.com/2014/01/11/android-distribution-numbers-update-for-january-2014-jelly-bean-closing-in-on-60/

[26] (2011) 5.6.0 Benchmarks [Online]. Available: http://www.cryptopp.com/benchmarks.html

[27] (2014) National Software Reference Library. [Online]. Available: http://www.nsrl.nist.gov/

[28] G. G. Richard and V. Roussev "Scalpel: A Frugal, High Performance File Carver", in *Proc. 5nd DFRWS, 2005,* paper.

[29] (2015) ANDROPHSY. [Online]. Available: https://github.com/scorelab/ANDROPHSY