Mobile Forensic Data Acquisition in Firefox OS

Mohd Najwadi Yusoff, Ramlan Mahmod, Mohd Taufik Abdullah, Ali Dehghantanha
Faculty of Computer Science & Information Technology,
Universiti Putra Malaysia,
Serdang, Selangor, Malaysia.
najwadi@cs.usm.my,{ramlan,taufik,alid}@upm.edu.my

***Abstract*—Mozilla Corporation has recently released a Linux-based open source operating system, namely Firefox OS. The arrival of this Firefox OS has created new challenges, concentrations and opportunities for digital investigators. Currently, Firefox OS is still not fully supported by most of the existing mobile forensic tools. Even when the phone is detected as Android, only pictures from removable card was able to be captured. Furthermore, the internal data acquisition is still not working. Therefore, there are very huge opportunities to explore the Firefox OS on every stages of mobile forensic procedures. This paper will present an approach for mobile forensic data acquisition in a forensically sound manner from a Firefox OS running device. This approach will largely use the UNIX dd command to create a forensic image from the Firefox OS running device.**

Keywords—Mobile forensic; data acquisition; Firefox OS

I. INTRODUCTION

The advancement of smartphone technology has attracted many companies in developing their own mobile operating system. Recently released Firefox OS is an open source mobile operating system which is purely based on Linux and Mozilla's Gecko technology [1]. Firefox OS boots into a Gecko-based runtime engine and thus allow users to run applications developed exclusively using HTML5, JavaScript, and other open web application APIs. According to Mozilla Developer Network, Firefox OS is free from proprietary technology, but still a powerful platform; it offers application developers an opportunity to create tremendous products [1]. Mozilla introduced WebAPI by bridging the capability gap between native frameworks and web applications. WebAPI enable developers to build applications, and run it in any standards compliant browser without the need to rewrite their application for each platform. In addition, since the software stack is entirely HTML5, a large number of developers were already established, and users can embrace the freedom of pure HTML5 [2].

To our knowledge, none of the existing mobile forensic tools are working perfectly with Firefox OS. For example, MobilEdit! is able to detect Firefox OS running phone, but listing it as an Android device. When we tried to perform data acquisition, MobilEdit! was only capable to acquiring some of the pictures from removable card, the remaining are left undetected. In addition, we have also tried using Paraben Device Seizure, Oxygen Forensic Suite, Cellebrite Mobile Forensics as well as Micro Systemation XRY; and the result were even worse. The acquisition process for Firefox OS running phone become more exciting because the phone itself was detected as Android. This may be due to the similarity of both Android and Firefox OS in their based kernel. For that reason alone, this paper will demonstrate the use of Android Debug Bridge (ADB); to connect the phone with the host machine and acquiring the phone image using UNIX dd command.

There are three types of acquisition of mobile devices; manual, logical and physical [3]. Manual acquisition is defined as the capability of acquiring data by interacting with the device itself. Logical acquisition is recovering a bitwise copy of entities that reside in a logical storage, and lastly; the physical acquisition is solely related to the physical storage medium. In most cases, manual acquisition takes place simultaneously with the other two acquisition methods. On the contrary, there are strengths and weaknesses of each types of acquisition. Grispos stated that, logical acquisition is more efficient for recovering user data, whereas physical acquisition can retrieve deleted files [4]; but this procedure can damage the device while it is being dismantled. And for that reason, this paper will only perform the combination between manual and logical types of acquisition. During this process, we will be making a bitwise copy of all partitions, and keeping the log of actions taken.

The objective of this paper is to present the detail steps on how we manage to acquire mobile forensic image from Firefox OS using UNIX dd command without making any changes to the phone during acquisition. This paper is organized as follows; Section (2) will explain about the state of the arts. Section (3) will present acquisition methodology and the detail steps. Section (4) will give a brief conclusion and the future work to be considered. Acknowledgement and references are also presented at the end of this paper.

II. STATE OF THE ART

Data acquisition is the procedure of imaging and obtaining evidence from a mobile device and its peripheral equipment [5]. In the earliest mobile forensic investigation, most of the digital evidences in mobile phone were stored in SIM cards. Research by Goode stated that, it is vital to acquire the data such as contacts and SMSs stored in SIM cards [6]. Similar work carried out by Willassen was by exploring SIM card and core network data in GSM phones [7]. According to Willassen, the contents of a SIM card are binary data that can be downloaded, provided that the user has authentication either with a PIN or a PUK code. In similar attempt, Casadei used

ISBN: 978-1-4799-3906-0 ©2014 IEEE

open source tools, both in Windows and Linux for digital extraction from SIM [8]. As the result, Casadei was able to acquire the raw data in Binary format from the SIM cards. Marturana extended the acquisition process in SIM cards by comparing data in SIM and Smartphones [9]. According to Marturana, acquisition in the smartphone is much more complicated; this is due to the possibility of evidences are also stored in many places such as internal and flash memory.

With the emergence of smartphones, focuses are more on the Windows Mobile OS due to its similarity in nature with the desktop environment. Windows Mobile OS is a simplified version of Windows OS developed by Microsoft; mainly for mobile devices. Research by Chen was able to extract SMS, phone book, call recording, scheduling, and documents from Windows Mobile OS via Bluetooth, Infrared and USB mode using Microsoft ActiveSync [10]. Microsoft ActiveSync used Remote API (RAPI) to read the data stored in the phone. Similar research continued by Irwin and Hunt by extracting evidences over wireless connections. They successfully demonstrated in mapping internal and external phone's memory and transfer all files and folder to desktop computers [11]. Casey extended the finding by describing various methods of acquiring and examining data on Windows Mobile devices. Casey was also able to capture text messages, multimedia, e-mail, Web browsing, and Registry entries [12]. Some of the captured data by Casey are locked by the OS itself, and require XACT from Micro Systemation and ItsUtils to work together with Microsoft ActiveSync. These tools will help to unlock certain files and convert the ASCII format in cemail.vol structure to a readable SMS.

The proliferation of mobile technology has made many companies to produce their own mobile OS. Forensic approaches for Windows Mobile OS might not be applicable to other mobile platforms. Therefore, Savoldi made a brief survey and comparison between mobile forensic for Windows Mobile OS and Symbian S60 [13]. In his work, Savoldi acquired the evidences using both logical and physical methods. Savoldi also illustrated the differences and identified possible common methodology for future forensic exploration. Conversely, Mohtasebi studied four mobile forensic tools; namely Paraben Device Seizure, Oxygen Forensic Suite, MIAT, and MOBILedit! to extract evidences from Nokia E5-00 Symbian phone [14]. The comparison was to check the ability to extract evidence and to examine information types such as call logs, map history, and user data files. On the contrary, Casey has proposed a methodology for acquiring and examining forensic duplicates of user and system partitions; from a device running on webOS [15]. These captured data is in .db3 format and can be analysed using SQL viewer.

Most of the mobile manufacturers provide a software package to communicate with their own products. One of the examples is Microsoft ActiveSync which is used for Windows Mobile OS. As for Apple iOS, Husain and Sridhar used iTunes to force backup the iPhone and logical copy of backup can be found in computer hard drive [12-13]. Similarly, Chun and Park used Samsung Kies to extract SMS, photo and mobile image from Samsung Galaxy S [18]. However, most of the bundle software package that used to acquire evidences, placing an agent into the mobile devices. This action may alter the stored data in mobile devices such as the last synchronization date and time; or the name of the last computer synchronize with the devices. For this reason, Rehault has proposed a method of using boot-loader concept; which is non-rewritable and is able to protect the evidences from being altered [19]. An extended work for boot-loader concept by Rehault was published by Chen for Google Android OS [20]. The concept of acquisition evidences is similar but this time it is using Secure Digital (SD) card. This method claimed can effectively perform the recovery of any deleted data. Similarly, Vidas proposed a general method of acquiring process for Android by using boot modes [21]. This technique repurposed the recovery partition and associated recovery mode of an Android for acquisition purpose. The acquired data has to be in recovery image format. Custom boot-loader method has become popular in Google Android OS because the user is able to get root permission; and able to acquire an image of the flash memory. Another research using boot-loader method was conducted by Park [22]. This research mainly focused on fragmented flash memory due to the increase of flash memory deployment in mobile phones.

Apart from using the bundled software packages, the other way to obtain mobile images are by using the SSH connection. To use SSH, the phones should have SSH installed and the data has to be transferred to a remote host via the network; in most cases are using wireless connection. However, this is a lengthy process and very time consuming; it may require up to 20 hours depending on the image size. Alternately, Gómez-Miralles and Arnedo-Moreno have presented a novel approach by using an iPad's camera connection kit attached via USB connection [23]. In order to acquire iPad's image, this approach greatly reduces the transferring time; and flasher boxes were used by forensic investigation for some acquisition process. Work by Jonkers used flasher boxes to acquire data but there are some limitation observed; such as verifying the data integrity and makes it not really practical [24]. Removable cards become popular alternative to physical acquisition process. Rossi demonstrated internal forensic acquisition in mobile devices using removable cards [25] and this work becomes a stepping stone for the boot loader concept. However, some approach does not work for volatile memory. Therefore, Sylve present the first methodology and toolset for acquisition of volatile physical memory from Android devices [26]. This method has created a new kernel module for dumping memory and Sylve has further develop a tool to acquire and analyse the data. Similar method was also proposed by Dezfouli using force backup in an isolated folder [27], but this is yet to be implemented. The newly acquisition method is the live acquisition. Thing proposed an automated system in acquiring evidences and claimed that this method consistently achieved 100% evidence acquisition rate for outgoing message and 75.6% to 100% evidence acquisition rate for incoming message [28]. Although the acquisition rate is high, this method was tested using only own developed chat bot and yet to be tested using commercial Instant Messaging. Another live acquisition research is by Lai. Lai has proposed data acquisition in Android; and deliver the data to the Google cloud server in real time [29]. This method really can deliver the intended data, but the integrity of the data is questionable.

III. ACQUISITION METHODOLOGY

The goal of this paper is to propose a methodology to acquire the mobile forensic image from Firefox OS running phone. In general, Firefox OS architecture consist of 3 layers [30]. The first layer is an application layer called Gaia and it works as the user interface for smartphones. The second layer is an open web platform interface; using Gecko engine and provide all support for HTML5, JavaScript as well as CSS. All the targeted evidence are stored in this layer. The third layer called Gonk; is an infrastructure layer and based on Linux-Kernel. There are two types of storage in Firefox OS running phone which are internal storage and additional micro SD card. Acquiring data from the micro SD card is relatively easy; the phone only need to be connected to the host machine and micro SD card can be mounted as removable drive. However, acquiring data from internal storage and other user partitions is quite a challenging tasks. Subsection below will further elaborate about experimental setup and imaging process for Firefox OS running phone.

A. Firefox OS Running Phone

For acquisition process, we will use Firefox OS running phone released by Geeksphone, model name Peak. It was release in April 2013.

Fig. 1. Geeksphone Peak.

This phone is equipped with Firefox OS version 1.1.1 as shows in Fig. 2. Mozilla updated their OS regularly and any stable build can be update via over-the-air.

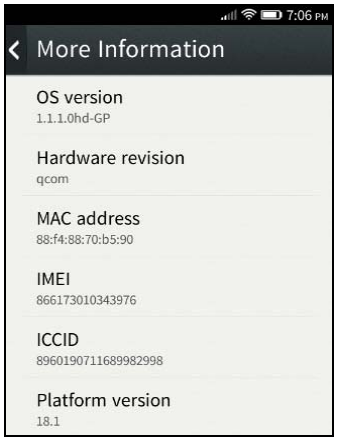

Fig. 2. Geeksphone Peak detail.

This phone powered by the dual core Qualcomm Snapdragon S4 processor and based on the ARMv7 instruction set. Table 1 below shows the specification detail for this phone.

TABLE I. GEEKSPHONE PEAK SPECIFICATION

Hardware	**Detail**
Processor	1.2 GHz Qualcomm Snapdragon S4 8225 processor (ARMv7)
Memory	512 MB Ram
Storage	-Internal 4GB -Micro SD up to 16GB
Battery	1800 mAh
Display	540 × 960 px (qHD) capacitive touchscreen, 4.3"
Sensor	-Ambient light sensor -Proximity sensor -Accelerometer
Camera	8 MP (Rear), 2 MP (Front)
Connectivity	-WLAN IEEE 802.11 a/b/g/n -Bluetooth 2.1 +EDR -micro-USB 2.0 -GPS -mini-SIM card -FM receiver
Dimension	-Width: 133.6 millimetres (5.26 in) -Height: 66 millimetres (2.6 in) -Thickness: 8.9 millimetres (0.35 in)

B. Forensic Requirement Setup

To begin with a forensic requirement setup, an additional driver for Geeksphone Peak need to be installed into the host machine. We will use Windows 8 as an operating system in the host machine. Once connected using the micro-USB 2.0 port, Windows 8 will ask for the driver. The supported USB driver can be downloaded from Geeksphone web. Once the installation finished, Geeksphone Peak will appear in the Device Manager as shows in Fig. 3 and micro SD card will be mounted into the host machine.

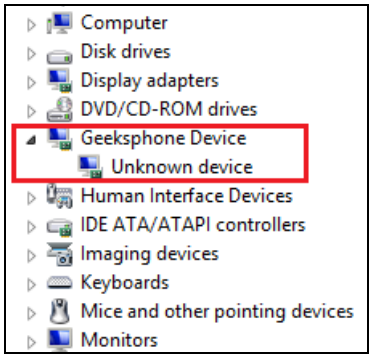

Fig. 3. Windows 8 detect as unknown device.

Subsequently, we need to make a connection between the phone and the host machine. Firefox OS is based on Linux-Kernel and the design more or less are similar with Google Android. For that reason, we can easily access the phone using Android Debug Bridge (ADB). The ADB is a toolkit integrated in the Android SDK package and consists of both client and

server-side codes. The codes are able to communicate with one another. Since Firefox OS is a Linux-based open-source mobile OS, any rooting procedure are not required. To have ADB installed in the host machine, we need to download the Android SDK from Android developer page. The file is about 480MB and unzip is necessary. After that, `SDK Manager` is launched and we need to install `Android SDK Tools`, `Android SDK Platform-tools`, and `Android SDK Build-tools` as shows in Fig. 4. These three tools are required to run ADB. Now we are ready to start with the acquisition process.

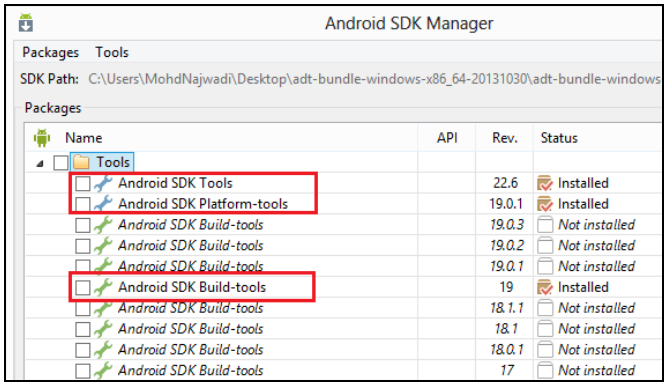

Fig. 4. ADB required tools.

C. *Device Imaging Process*

First of all, we need to know what data to be acquired and where it is stored. From a forensic standpoint, imaging the whole disk can preserve its contents from any changes. As for Firefox OS, the targeted phone image will consist of several partitions. It will cover the entire systems, user data and installed applications. Before we start, we need to unmount micro SD card from the host machine. In order to do that, go to `phone Settings > Storage` and disable phone storage as shows in Fig. 5.

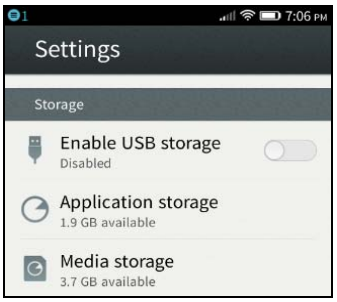

Fig. 5. Disable USB storage.

In order to create phone image, we will use UNIX dd command in ADB environment. To start ADB, we need to run command prompt (CMD) and point it to `%Android SDK%\sdk\platform-tools` folder. Next is to type the following;

```
adb shell
```

This command will establish the connection between the phone and the host machine and `root@android:/ #` access will appear in the CMD. In order to check the partition location, type the following;

```
cd dev
cd block
ls
```

These command will display all the existing partitions in the phone. The targeted partition name started with `mmcblk0p1` till `mmcblk0p21`. We have two options to create the phone image. The first option is to run UNIX dd command for each partition, one by one; started with `mmcblk0p1` till `mmcblk0p21`. The second option is to run UNIX dd command to the parent tree of the partition which is `mmcblk0;` and it will later combined all the partitions into one image. We decided to run the second option and type the following;

```
dd if=/dev/block/mmcblk0
of=/mnt/emmc/evidence.img bs=1024
```

The image is pointed into the micro SD card and we select 1024KB as the block size. The process will take up until 10 minutes and the acquired image is around 3.7GB (internal storage size). Fig. 6 shows the created log.

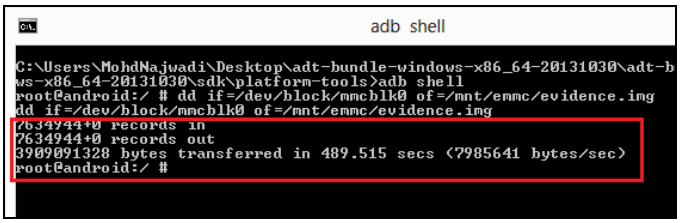

Fig. 6. Log for acquired image.

In order to transfer the acquired image into the host machine, we need to mount back the micro SD card and it will again detected in the host machine as shows in the Fig. 7

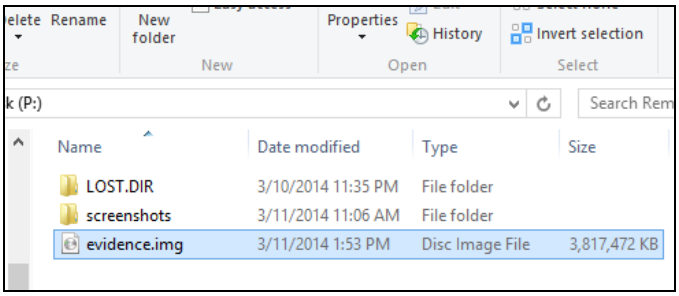

Fig. 7. Acquired image from the Firefox OS running phone.

ISBN: 978-1-4799-3906-0 ©2014 IEEE

IV. Conclusion and Future Work

The arrival of Firefox OS has created new challenges, concentrations and opportunities for digital investigators. It is a very exciting tasks to explore new version of mobile OS since all the apps is purely built on HTML5. To our concern, even though Firefox OS based on Linux-kernel mobile OS, it does not mean existing mobile forensic tools will work fine with this release. We have proved that, only certain data able to read or captured from Firefox OS running phone by using existing mobile forensic tools. This may be due to the differences of existing user data and the partition arrangement. There are many more aspects to be explored. Our next focus will be on analyzing parts and we will go deeper on the system files, user data and application logs.

Acknowledgment

Special thanks to academic staff of Universiti Putra Malaysia for providing continuous guide and support, and also to Ministry of Education Malaysia for granting the scholarship to me.

References

[1] Mozilla Developer Network, "Firefox OS," https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS. 07-May-2013.

[2] R. Goodwin, "Mozilla's Boot 2 Gecko and why it could change the world," http://www.knowyourmobile.com/products/16409/mozillas-boot-2-gecko-and-why-it-could-change-world. 07-May-2013.

[3] K. Barmpatsalou, D. Damopoulos, and G. Kambourakis, "A critical review of 7 years of Mobile Device Forensics," Digit. Investig., vol. 10, no. 4, pp. 323–349, 2013.

[4] G. Grispos, T. Storer, and W. B. Glisson, "A comparison of forensic evidence recovery techniques for a windows mobile smart phone," Digit. Investig., vol. 8, no. 1, pp. 23–36, Jul. 2011.

[5] W. Jansen and R. Ayers, Guidelines on Cell Phone Forensics - Recommendations of the National Institute of Standards and Technology. 2007.

[6] A. J. Goode, "Forensic extraction of electronic evidence from GSM mobile phones," in IEE Seminar on Secure GSM and Beyond: End to End Security for Mobile Communications, 2003, pp. 9/1–9/6.

[7] S. Y. Willassen, "Forensics and the GSM mobile telephone system," Int. J. Digit. Evid., vol. 2, no. 1, pp. 1–17, 2003.

[8] F. Casadei, A. Savoldi, and P. Gubian, "SIMbrush: an open source tool for GSM and UMTS forensics analysis," in First International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'05), 2005, pp. 105–119.

[9] F. Marturana, G. Me, R. Berte, and S. Tacconi, "A Quantitative Approach to Triaging in Mobile Forensics," in 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, 2011, pp. 582–588.

[10] S. Chen, X. Hao, and M. Luo, "Research of Mobile Forensic Software System Based on Windows Mobile," in 2009 International Conference on Wireless Networks and Information Systems, 2009, pp. 366–369.

[11] D. Irwin and R. Hunt, "Forensic information acquisition in mobile networks," in 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, 2009, pp. 163–168.

[12] E. Casey, M. Bann, and J. Doyle, "Introduction to Windows Mobile Forensics," Digit. Investig., vol. 6, no. 3–4, pp. 136–146, May 2010.

[13] A. Savoldi, P. Gubian, and I. Echizen, "A Comparison between Windows Mobile and Symbian S60 Embedded Forensics," in 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009, pp. 546–550.

[14] S. Mohtasebi, A. Dehghantanha, and H. G. Broujerdi, "Smartphone Forensics□: A Case Study with Nokia E5-00 Mobile Phone," Int. J. Digit. Inf. Wirel. Commun., vol. 1, no. 3, pp. 651–655, 2012.

[15] E. Casey, A. Cheval, J. Y. Lee, D. Oxley, and Y. J. Song, "Forensic acquisition and analysis of palm webOS on mobile devices," Digit. Investig., vol. 8, no. 1, pp. 37–47, Jul. 2011.

[16] M. I. Husain and R. Sridhar, "iForensics□: Forensic Analysis of Instant Messaging on," Lect. Notes Inst. Comput. Sci. Soc. Informatics Telecommun. Eng. - Digit. Forensics Cyber Crime, vol. 31, pp. 9–18, 2010.

[17] M. I. Husain, I. Baggili, and R. Sridhar, "A Simple Cost-Effective Framework for iPhone," Lect. Notes Inst. Comput. Sci. Soc. Informatics Telecommun. Eng. - Digit. Forensics Cyber Crime, vol. 53, pp. 27–37, 2011.

[18] W. Chun and D. Park, "A Study on the Forensic Data Extraction Method for SMS , Photo and Mobile Image of Google Android and Windows Mobile Smart Phone," Commun. Comput. Inf. Sci. - Converg. Hybrid Inf. Technol., vol. 310, pp. 654–663, 2012.

[19] F. Rehault, "Windows mobile advanced forensics: An alternative to existing tools," Digit. Investig., vol. 7, no. 1–2, pp. 38–47, Oct. 2010.

[20] S.-W. Chen, C.-H. Yang, and C.-T. Liu, "Design and Implementation of Live SD Acquisition Tool in Android Smart Phone," in 2011 Fifth International Conference on Genetic and Evolutionary Computing, 2011, pp. 157–162.

[21] T. Vidas, C. Zhang, and N. Christin, "Toward a general collection methodology for Android devices," Digit. Investig., vol. 8, pp. S14–S24, Aug. 2011.

[22] J. Park, H. Chung, and S. Lee, "Forensic analysis techniques for fragmented flash memory pages in smartphones," Digit. Investig., vol. 9, no. 2, pp. 109–118, Nov. 2012.

[23] L. Gómez-Miralles and J. Arnedo-Moreno, "Versatile iPad forensic acquisition using the Apple Camera Connection Kit," Comput. Math. with Appl., vol. 63, no. 2, pp. 544–553, Jan. 2012.

[24] K. Jonkers, "The forensic use of mobile phone flasher boxes," Digit. Investig., vol. 6, no. 3–4, pp. 168–178, May 2010.

[25] M. Rossi and G. Me, "Internal forensic acquisition for mobile equipments," in 2008 IEEE International Symposium on Parallel and Distributed Processing, 2008, pp. 1–7.

[26] J. Sylve, A. Case, L. Marziale, and G. G. Richard, "Acquisition and analysis of volatile memory from android devices," Digit. Investig., vol. 8, no. 3–4, pp. 175–184, Feb. 2012.

[27] F. N. Dezfouli, A. Dehghantanha, R. Mahmoud, N. F. Binti Mohd Sani, and S. bin Shamsuddin, "Volatile memory acquisition using backup for forensic investigation," in Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012, pp. 186–189.

[28] V. L. L. Thing, K.-Y. Ng, and E.-C. Chang, "Live memory forensics of mobile phones," Digit. Investig., vol. 7, pp. S74–S82, Aug. 2010.

[29] Y. Lai, C. Yang, C. Lin, and T. Ahn, "Design and Implementation of Mobile Forensic Tool for Android Smart Phone through Cloud Computing," Commun. Comput. Inf. Sci. - Converg. Hybrid Inf. Technol., vol. 206, pp. 196–203, 2011.

[30] Mozilla Developer Network, "Firefox OS architecture," https://developer.mozilla.org/en-US/docs/Mozilla/Firefox_OS/Platform/Architecture. 07-May-2013.