Forensic Investigation of Social Media and Instant Messaging Services in Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp and Line as Case Studies

Mohd Najwadi Yusoff  
School of Computer Science, Universiti Sains Malaysia, Penang, Malaysia.  
mohd.najwadi@gmail.com

Ali Dehghantanha  
School of Computing, Science and Engineering, University of Salford, Manchester, United Kingdom.  
A.Dehghantanha@Salford.ac.uk

Ramlan Mahmod  
Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Malaysia.  
ramlan@upm.edu.my

Abstract

Mobile devices are increasingly utilized to access social media and instant messaging services, which allow users to communicate with others easily and quickly. However, the misuse of social media and instant messaging services facilitated conducting different cybercrimes such as cyber stalking, cyber bullying, slander spreading and sexual harassment. Therefore, mobile devices are an important evidentiary piece in digital investigation. In this chapter, we report the results of our investigation and analysis of social media and instant messaging services in Firefox OS. We examined three social media services (Facebook, Twitter and Google+) as well as three instant messaging services (Telegram, OpenWapp and Line). Our analysis may pave the way for future forensic investigators to trace and examine residual remnants of forensics value in FireFox OS.

Keywords: Firefox OS forensics; mobile forensics; social media forensics; instant messaging forensics; mobile applications investigation
1. Introduction

The exponential growth of social media and instant messaging applications facilitated development of many serious cybercrime and malicious activities (Mohtasebi and Dehghantanha, 2011a). Cybercriminals are constantly changing their strategies to target rapidly growing social media and instant messaging users. The misuse of social media and instant messaging in mobile devices may allow cybercriminals to utilize these services for malicious purposes (Mohtasebi and Dehghantanha, 2011b) such as spreading malicious codes, obtaining and disseminating confidential information etc. Many social media and instant messaging providers have extended their services to mobile platforms (Dezfouli et al., 2015) which worsen the situation as users are in danger of losing even more private information (Taylor et al., 2012). Copyright infringement, cyber stalking, cyber bullying, slander spreading and sexual harassment are becoming serious threats to social media and instant messaging mobile users (Dezfouli et al., 2013). Therefore, it is common to confront with different types of mobile devices during variety of forensics investigation cases (Damshenas et al., 2014). Mobile devices are now an important source of forensic remnants relevant to users social media and instant messaging activities (Mohtasebi and Dehghantanha, 2013). However, difference between mobile devices mandate forensics investigators to develop customized methods and techniques for investigation of different phones (Mohtasebi et al., 2012).

With the emergence of smartphones, almost all parts of phones such as internal storage, flash memory and internal volatile memory contains valuable evidences (Yang et al., 2016). Chen extracted SMS, phone book, call recording, scheduling, and documents from Windows Mobile OS via Bluetooth, Infrared and USB using Microsoft ActiveSync (Chen et al., 2009). The acquired data were extracted from mobile phone internal storage, SIM card as well as removable memory. Irwin and Hunt have successfully mapped internal and external memory of Windows Mobile ver.5 running on IPAQ Pocket PC over wireless connections using their own developed forensic tools (Irwin and Hunt, 2009). Pooters has developed a forensic tool called Symbian Memory Imaging Tool (SMIT) to create bit-by-bit copies of the internal flash memory of Symbian OS phones such as Nokia E65, E70 and N78 model (Pooters, 2010). Lessard and Kessler (Lessard and Kessler, 2010) have acquired a bit-by-bit logical image of a HTC Hero memory using UNIX dd command and analyzed the resulted images using AccessData Forensic Toolkit (FTK) Imager v2.5.1 (Accessdata, 2007). Gómez-Miralles and Arnedo-Moreno have utilized a Jailbroken iPad’s camera connection kit to acquire an image of the device via USB connection (Gómez-Miralles and Arnedo-Moreno, 2012). Iqbal has enhanced Gómez-Miralles and Arnedo-Moreno method by developing a method to acquire iOS memory images without jail-breaking the device (Iqbal et al., 2012). Sylve has presented a methodology and toolset for acquisition volatile physical memory of Android devices by creating a new kernel module for dumping the memory (Sylve et al., 2012).

Beyond evidence acquisition, many researchers have shown big interest in investigating social media and instant messaging services on different mobile platforms. Husain and Sridhar have analyzed AIM, Yahoo! Messenger and Google Talk instant messaging applications in Apple iOS to detect potential application of these instant messaging services, particularly in
cyber bullying and cyber stalking (Husain and Sridhar, 2010). They have managed to detect username, password, buddy list, last login time and conversation contents together with timestamp. Jung has analyzed eight social media applications in Apple iOS namely Cyworld, Me2Day, Daum Yozm, Twitter, Facebook, NateOn UC., KakaoTalk and MyPeople (Jung et al., 2011) and managed to retrieve user info, friend list, message, contact and media information of each application. Tso has observed the diversification of the backup files for Facebook, WhatsApp Messenger, Skype, Windows Live Messenger and Viber in Apple iOS (Tso et al., 2012). Anglano has analyzed WhatsApp Messenger application remnants on an Android smartphone and reconstructed the list of contacts and the chronology of communicated messages (Anglano, 2014). Karpisek has successfully decrypted the network traffic of WhatsApp Messenger and managed to obtain forensic artifact related to call features and visualized messages that have been exchanged between users (Karpisek et al., 2015). Walnycky has examined 20 popular instant messaging applications on Android platform and has reconstructed some of the entire message content of 12 applications (Walnycky et al., 2015). Said has conducted a comparative study of Facebook and Twitter remnants on Apple iOS, Windows Mobile and RIM BlackBerry (Said et al., 2011) and extracted Facebook and Twitter remnants of Apple iOS, and Facebook. Mutawa has compared evidences of Facebook, Twitter and MySpace in three different operating systems namely Apple iOS, Google Android and RIM BlackBerry and could not recover any artifact from Blackberry while iPhones and Android contained many valuable artefacts (Al Mutawa et al., 2012). Iqbal (Iqbal et al., 2014) has compared artifacts of Samsung’s ChatON application between a Samsung Galaxy Note running Android 4.1 and an Apple iPhone running with iOS 6 and managed to detect all sent and received messages with timestamp and location of the sent files on both platforms. Dezfouli investigated Facebook, Twitter, LinkedIn and Google+ on Android and Apple iOS platforms and managed to recover many artifacts including username, contact information, location, friend list, social media post, messages, comments and IP addresses of selected social media applications on both platforms (Dezfouli et al., 2015). Table 1 is summarizing the literature on social media and instant messaging investigation forensics.

Table 1 - Summary of Social Media and Instant Messaging Investigation on Multiple Mobile Platforms

| Researcher(s)          | Application(s)                                      | Platform(s)          |
|------------------------|-----------------------------------------------------|----------------------|
| Husain and Sridhar 2010| AIM, Yahoo! Messenger, Google Talk                 | Apple iOS            |
| Jung et al. 2011       | Cyworld, Me2Day, Daum Yozm, Twitter, Facebook, NateOn UC., KakaoTalk, MyPeople | Apple iOS            |
| Tso et al. 2012        | Facebook, WhatsApp Messenger, Skype, Windows Live Messenger, Viber | Apple iOS            |
| Anglano 2014           | WhatsApp Messenger                                  | Google Android       |
| Karpisek et al. 2015   | WhatsApp Messenger                                  | Google Android       |
| Walnycky et al. 2016   | WhatsApp Messenger, Viber, Instagram, Okcupid, ooVoo | Google Android       |
Tango, Kik, Nimbuzz, eetMe, MessageMe, TextMe, Grindr, HeyWire, Hike, textPlus, Facebook Messenger, Tinder, Wickr, Snapchat, Blackberry Messenger

| Study               | Applications                                    | Platforms                                      |
|---------------------|-------------------------------------------------|------------------------------------------------|
| Said et al. 2011    | Facebook, Twitter                              | Apple iOS, Windows Mobile, RIM BlackBerry      |
| Al Mutawa et al. 2012 | Facebook, Twitter, MySpace                     | Apple iOS, Google Android, RIM BlackBerry      |
| Iqbal et al. 2014   | Samsung’s ChatON                               | Apple iOS, Google Android                      |
| Dezfooli et al. 2015 | Facebook, Twitter, LinkedIn, Google+           | Apple iOS, Google Android                      |

As can be seen from Table 1, there was no previous work investigating remnants of Facebook, Twitter, Google+, Telegram, OpenWapp and Line applications on Firefox OS (FxOS) which is the gap targeted in this paper. In this chapter, we present investigation of a Geeksphone, model name Peak (Geeksphone, 2013) running FxOS 1.1.1 as the main subject of the investigation. Two binary images were taken from the phone internal storage and internal memory and then valuable forensics remnants were examined. We have mainly interested in evidential remnants of user activities with Facebook, Twitter, Google+, Telegram, OpenWapp and Line applications.

The rest of this chapter is organized as follows. In section 2, we will explain the methodology used in our research and in section 3, we will outline the setup for our experiment. In section 4, we will present our research findings and conclude our research in section 5.

2. Methodology

This research has performed an investigation on Mozilla FxOS running on a phone released by Geeksphone, called Peak (Geeksphone, 2013). Released in April 2013, this phone is equipped with FxOS version 1.1.1. Table 2 shows the full specification of Geeksphone Peak.

| 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                                                            |
|                 | - micro-USB charging                                                  |
| Data Inputs     | Capacitive multi-touch IPS display                                     |
| Display         | 540 x 960 px (qHD) capacitive touchscreen, 4.3”                       |
| Sensor          | - Ambient light sensor                                                |
|                 | - Proximity sensor                                                    |
|                 | - Accelerometer                                                       |
| Camera          | 8 MP (Rear), 2 MP (Front)                                             |
As illustrated in the Figure 1, initially the device setting were wiped and restored to the default factory setting. The acquisition process was then executed to acquire FxOS phone image (.ffp) and memory image (.ffm). The first two binary images were then marked as base image and their MD5 hash values were preserved. Next, the investigator installed social media or instant messaging applications to the phone via Mozilla Market Place. The investigator simulate the actual use of the application by running communication activities such as posting, sending private message, received reply, upload picture, mentioning, following, and many more social media and instant messaging activities. Each of detail steps, credentials and communication activities were documented properly in forensically sound manner. Finally, second acquisition process performed to identify and investigate what artifact are likely to remain, what type of credential can be extracted, and what data remnants could be recovered. Both images were named according to the type of application installed and its MD5 hash values were taken. This step meant for comparison to the earlier data acquisition, to see if there were any differences with the other MD5 hash values.
Our method to create forensic evidences in Figure 1 were performed by using Facebook, Twitter, Telegram, OpenWapp and Line applications. We also repeat this method using Facebook, Twitter, Google+ and Telegram mobile web, in order to compare both FxOS applications artifacts and mobile web activities artifacts. However, comparison was not possible for OpenWapp and Line application artifacts, since both applications were not offered in mobile web platforms. It is vital to include the mobile web activities in this investigation since FxOS application were designed based on web application. Therefore, this research was attempting to proof that the investigation result of web-centric OS are totally different with other mobile OS investigation.

3. Experiment Setup

The research experiment was divided into five stages; (1) Preparing the host machine for acquisition process and analysis work; (2) Acquisition of phone image and memory image; (3) Installing the phone with targeted application; (4) Executed the activities and documenting all steps taken; and (5) Comparing base image with activities images.

3.1 Preparing the Host Machine for Acquisition Process and Analysis Work

The evidences were acquired using Ubuntu 14.04 LTS and then analyzed in Windows 8.1 machine. The need to use separated operating system is due to the fact that memory acquisition will only work on Linux environment; whereas most of the analysis tools were design under Windows operating system. To successfully capture the phone image using Ubuntu, the host machine need to be running under Android Debug Bridge (ADB) (Yusoff et al., 2014a, 2014b, 2014c, 2014d). Following command was used to configure ADB package in Ubuntu:

```
# sudo apt-get install android-tools-adb
```

As for the volatile memory acquisition, we configure Linux Memory Extractor (LiME) (Sylve et al., 2012) using the following command:

```
# sudo apt-get install lime-forensics-dkms
```

AccessData Forensic Toolkit (FTK) version 3.1.2 (Accessdata, 2007) and HxD Hex Editor 1.7.7.0 (Maël Hörz, 2009) were installed in Windows machine to analyze captured forensic images and QtADB 0.81 Android Manager (QtADB, 2011) was used to browse the system files in the Geeksphone Peak.

3.2 Acquisition of Phone Image and Memory Image

Two types of binary images were extracted, meant to be used as forensic evidence in our case studies. The first binary image were extracted from FxOS phone internal memory using dd command. FxOS phone was first connected to the host machine and then ADB connection was
started before we proceed to execute dd command. We have used the following command to start ADB connection between the phone and the host machine.

```
# adb shell
```

Once the connection is established, we have performed the following dd command to copy bit-by-bit of phone internal memory into SD card and we named it as FxOS phone memory (.ffp):

```
# dd if=/dev/block/mmcblk0 of=/mnt/emmc/base.ffp bs=2048
```

The whole binary image of FxOS internal memory was then copied from SD card into the host machine and we named it as base.ffp. We used block size 2MB during dd command execution as per suggested in previous studies (Yusoff et al., 2014b).

The next step was the extraction of the second binary image from FxOS phone volatile memory using LiME. LiME were design as such to allow the acquisition of volatile memory from Linux-based devices and originally tested using Android phones (Sylve et al., 2012). ADB was then used to load LiME, and the way of volatile memory acquisition at this step is similar to the dd command. We used the following command in ADB to load LiME from phone SD card:

```
# adb push lime.ko /mnt/emmc/lime.ko
```

After LiME were loaded, we run insmod to copy live bit-by-bit of volatile memory into SD card. For volatile memory, we named it as FxOS memory (.ffm)

```
# insmod /mnt/emmc/lime.ko "path=/mnt/emmc/base.ffm format=lime"
```

Both internal phone images (.ffp) and memory images (.ffm) were directed to SD card. After the acquisition process has completed, we killed the ADB connection, and then mounted the SD card to copy both images. We set the same name for both images and only the extension and file size are left as the differences. The steps were repeated for every action in our experiment.

### 3.3 Installing the Phone with Targeted Application

The objective of this experiment is to investigate both social media network and instant messaging platform in FxOS. Due to the limited number of applications offered in Mozilla Market Place, we have only managed to investigate 5 applications and 4 mobile webs as shows in Table 3

| Group       | Application | Mobile Web |
|-------------|-------------|------------|
| Social Media| 1. Facebook | 1. Facebook|
|             | 2. Twitter  | 2. Twitter |
|             |             | 3. Google+ |
The investigation were performed on both application and the mobile web. The purpose of investigating both applications and mobile web, were to list out the differences and similarities between FxOS applications with its mobile web.

3.4 Executed the Activities and Documenting All Steps Taken

Our investigation in this experiment has started the moment we installed the applications in to the phone. We have also created a few dummy accounts for the experiments. All detail steps, credentials and communication activities were properly documented. In order to facilitate the use of credential during our experiment, we created all social media account using the same email address and password which are “mohd.najwadi@gmail.com” and “najwadi87” respectively. In general, all social media and instant messaging application will have one extra step; which is installation of the application. In contrary for mobile web, we can just browse directly from the site. Table 4 shows all steps and activities taken for social media experiment.

| Social Media      | Steps and Action                                                                 | Image Filename |
|-------------------|---------------------------------------------------------------------------------|----------------|
| **Facebook**      | Installing Facebook Application                                                  | Facebook-Install |
| **Application**   | Login Facebook Account (ID : Root Wadi)                                         | Facebook-Login |
|                   | - Email: mohd.najwadi@gmail.com                                                 |                |
|                   | - Password : najwadi87                                                         |                |
| **Facebook**      | Facebook Activities                                                            | Facebook-Post  |
|                   | - Post Status : Posting with love                                              |                |
| **Application**   | - PM Mohd Najwadi : Hi. Test send msg                                           |                |
|                   | - Received Reply : received with thanks                                        |                |
| **Facebook Web**  | Login Facebook thru web www.facebook.com                                       | Facebook-Browse|
|                   | - Email : mohd.najwadi@gmail.com                                                 |                |
|                   | - Password - najwadi87                                                         |                |
| **Twitter**       | Installing Twitter Application                                                  | Twitter-Install|
| **Application**   | Login Twitter Account (ID : @wadieq)                                            | Twitter-Login  |
|                   | - Email : mohd.najwadi@gmail.com                                                 |                |
|                   | - Password : najwadi87                                                         |                |
| **Twitter**       | Twitter Activities                                                             | Twitter-Tweet  |
| **Application**   | - Tweet : My first tweet                                                        |                |
|                   | - Mention : @wadieq test mention                                                |                |
|                   | - Favourites : @wadieq Taman tepi rumah                                        |                |
| **Google+ Web**   | Login Google+ thru web plus.google.com                                           | Google-Login   |
| **Twitter Web**   | Login Twitter thru web www.twitter.com                                          | Twitter-Browse |
| **Twitter**       | - Email : mohd.najwadi@gmail.com                                                 |                |
| **Application**   | - Password : najwadi87                                                         |                |
| **Twitter**       | Twitter Activities                                                             | Twitter-xtvt   |
| **Application**   | - Tweet : My second tweet                                                        |                |
|                   | - Reply : @wadieq Reply comment                                                 |                |
|                   | - Unfollow @MunirRashid                                                        |                |
There were 12 time acquisition processes were performed for each internal phone image and memory image altogether; and the files were name according to the steps taken as shown in Table 4. We have also tested with Facebook and Twitter, both on applications and mobile web, while for Google+ we can only test it on web since there is no application supported for Google+ in Mozilla Market Place at the moment we conducted the experiment. The phones were restored back to their factory setting only after we want to start with new application or mobile web experiments. For social media investigation, we restored the phones 5 times, the number of experiment conducted. After the completion of the social media investigation, we then proceeded with instant messaging step. Instant messaging accounts were tied to different mobile numbers. In this experiment, we registered instant messaging account using mobile number “+60162444415” and we performed the communication with mobile number “+60125999159”. Table 5 shows all the steps and activities taken for instant messaging experiment.

Table 5 - Detail Steps for Instant Messaging Experiment

| Instant Messaging | Steps and Action | Image Filename |
|-------------------|------------------|----------------|
| Telegram Application | Installing Telegram Applications | Telegram-Install |
| Register Telegram account | Telegram-Register |
| - select Malaysia +60162444415 |  |
| - generating telegram code : 99246 |  |
| - received sms from +93450009276 |  |
| Telegram Activities | Telegram-chat |
| - Sent to +60125999159 : Hi |  |
| - Received from +60125999159 : Hello |  |
| - Received from +60125999159 : InstaSize_2015_4_36453.jpg |  |
| - Save picture : asking for access picture > |  |
| - Received from +60125999159 : 200507000.pdf |  |
| - Save file : asking for access memory card > |  |
| Telegram Web | Telegram-Browse |
| Login Telegram thru web web.telegram.org | Telegram-webxtvt |
| - select Malaysia +60162444415 |  |
| - generating telegram code 34303 |  |
| - received sms from +93450009276 |  |
| Telegram Activities | Telegram-webxtvt |
| - Sent to +60125999159 : Hi from web |  |
| - Received from +60125999159 : Hi back |  |
| - Received from +60125999159 : IMG_20150508_200905.jpg |  |
| - Save picture : failed |  |
| - Sent to +60125999159 : Thanks for the picture |  |
| OpenWapp | OpenWapp-install |
| Installing OpenWapp Application | OpenWapp-reg |
| Register WhatsApp |  |
| - select Malaysia +60162444415 |  |
| - generating WhatsApp code 560-103 |  |
| - received sms from 63365 |  |
| OpenWapp Activities | OpenWapp-chat |
| - Received from +60125999159 : Hello |  |
- Sent to +60125999159 : Testing openwapp from Firefox phone
- Received from +60125999159 : IMG_20150508_171904.jpg
- Save picture : agxahpqv0gm3uwkv7glgnxz6ig4k_bh6jq_bivja5q0.jpg

Line Application
- Installing Line Application
- Register Line Account
  - select Malaysia +6016244415
  - Reg code 1877
- Received sms from +601117224258

Line Activities
- najwadi added syamsaziela as friend
- najwadi sent msg : Hi syamsaziela (6.15PM)
- syamsaziela replied : send to firefox phone (6.15PM)
- syamsaziela sent : IMG_8933.jpg
- najwadi received, saved (read 6.18PM) : New folder name LINE - 2015412_185355518.jpg

Line-Install
- Line-Reg
- Line-xtvt

In instant messaging experiment, we have performed the acquisition processes 11 times for each internal phone image and memory image. The files were also named according to the step taken as shown in Table 5. The experiment were performed using OpenWapp and Line applications, and both Telegram application and mobile web. OpenWapp is the third party application for WhatsApp in FxOS because WhatsApp was not officially offered in Mozilla Market Place. Both WhatsApp and Line does not offer mobile web support. During our investigation, we restored the phone to their factory setting 4 times, the number of experiment conducted.

3.5 Comparing Base Image with Activities Images

Acquired forensic images were analyzed using AccessData Forensic Toolkit (FTK) version 3.1.2 and HxD Hex Editor 1.7.7.0. Our major analysis technique were solely rely on the actions and execution to the phone during experiment. The base image were then compared with the captured forensic images according to the detail steps and actions taken. MD5 hash value of all images were recorded as shows in Table 6.

Table 6 - MD5 Hash Value for Acquired Images

| Filename       | Phone Image (.ffp) | Memory Image (.ffm) |
|----------------|--------------------|---------------------|
| Base           | F080B51EDBCCA1DFBF85023A96C86B95D | 7D93024506D837EB85682AC6C2DAE7A9 |
| Facebook-Install | 94A4CEB5333D5A8D7E8498E531DFCF7 | 24E0273A4CB7C852E20230E823675A50 |
| Facebook-Login | 755811F684ED80A987E07C2B7D047560 | DAA9A489F04F73415BE37E26F23B4612 |
| Facebook-Post | 51ABC84775AE43268147OT450D21D23 | 84EDE342059CE45CE55C19566B4F116 |
| Facebook-Browse | FCBOC5CFEEDC292185840C89963391 | A04D089CBF82D3F8A09082F9E9FED4C3E |
| Facebook-xtvt | 4867919D871E0B71D98099E82C13DD95 | 3C3F77EE0FA59A93B4274265E4515967 |
| Twitter-Install | BF63A1A105A56D3A4CFCF68D511DD76 | E3178BA282722C70DBBB65DE9CB2F896 |
| Twitter-Login | 6B7F1035FIB2C4DECA3793F3AF36CF5 | 987C4B27C276D3823B5021121E93F |
| Twitter-Tweet | 6A18EDCAD1A2AF26CA27406712F27B34 | 51F6B1B243A47AB1256B12321BF9E6 |
| Twitter-Browse | 57F6F990C323DF171C32E8F8F81FF550A3 | 242247A35D86EDE9F642D17145FEC5 |
| Twitter-xtvt | 4CE5136D1D7F6DDE29ECA974439763A3B | 872957E4C173191C4D34CF7E96C8FE3 |
| Google-Login | C5B877501D82CFA698219FF3767FFD | E947D21413CF18A146A60B8CB6D181 |
| Google-xtvt | 5BCBBD245F29662C7127B26341952B | CDD2226A5694132194E85C9CF8760E1 |
| Telegram-Install | B15D7872C6E42244F859A7A324C25B | 07F95ABE0DB4DD920B90A8CA26E3FCB |
| Telegram-Register | A46445F6DB4A79C6FDD354CED983DF0A | A43650BD2947C17C3F6AB1117CF4070F |
MD5 hash value were taken and preserved. Any modification of forensic images will be easily detected by looking at their hash value. In our experiment, none of these forensic images shared the same hash values. This conclude that our forensic images was giving differences in value, modification and evidence in it.

4. Discussion and Analysis

In this section, we have divided our analysis into two parts. The first analysis has focused on social media investigation, while the second part has focused on instant messaging investigation. For both analysis, we started with the internal phone images (.ffp) and then proceeded with memory images (.ffm). The purpose of separating the two analysis was to differentiate between both analysis results. By doing this analysis separately, we have discovered that certain information which was not recoverable in the phone images, was able to be retrieved from memory images.

4.1 Social Media Investigation

In social media experiment, we were focusing the investigation for 2 social media applications and 3 social media mobile web. We have successfully separating the captured images based on major steps and details as shown in Table 3. There were several forensic worth of evidence that we were trying to recover and trace. First, we have explored the residual artifacts generated by application and URL involved for mobile web. Second, we were trying find any ID name that able to be captured after we login into social media. Third, we have searched for any credential involving username and password after login process. Forth, we have traced back what activities that has been captured in the images. Lastly we have checked for the data remnant and left over after complete uninstallation of the application.

4.1.1 Social Media Phone Image

Each of the phone images were opened using AccessData Forensic Toolkit (FTK) version 3.1.2 and HxD Hex Editor 1.7.7.0. The analysis were conducted manually by searching several keyword related to previous experiment in forensically sounds manner. The first binary search performed was the application keyword. This search executed once the selected application successfully installed from Mozilla Market Place. Since our factory setting base image was only comes with preinstalled application and none of it were used in our investigation, we cannot find
any keyword of selected social media from the base image. Figure 2 shows Twitter keyword appear when we executed the first search.

![Image](74x709)

Figure 2 - Application Keyword Found After Installation in Phone Image

The second search were performed to find URL for social media mobile web. Mobile web is totally different with application. We do not had to install it but only need to browse using preinstalled Firefox browser. Figure 3 shows Twitter URL was found in the images and the result has also showing the visited date and time.

![Image](74x159)

Figure 3 - URL for Social Media Mobile Web Found in Phone Image

The third search were to find our profile name or user ID once we logged into the social media. Profile name is different with username. Username used to login into the social media
while profile name were the name displayed in our social media account. Figure 4 shows our profile name together with the user ID and Facebook profile path that we managed to find.

Figure 4 - Facebook Profile Name Appear in Phone Image.

One of the most valuable information that we need to protect in social media, were the account credentials. In the event that the account credentials fall into wrong hands, they can definitely log in into our account and pretended to be us. When they commit any crime using our account, prosecution will be charged under our name. Next search were to find username and password that we used during our login process. Username rules were different between social media. Facebook and Google+ only accept email as username, while twitter can accept either email or selected string. Figure 5 shows Twitter username appear in our search.

Figure 5 - Username in Twitter Phone Image
Activities and communication in social media have contributed massive valuable information in forensic investigation. From the communication pattern, we might be able to identify if there is any cyber bullying as well as cyber stalking. We also can investigate if any sexual harassment had occurred. In the next search, we were trying to find activities that can be recovered in our experiment. We purposely used a certain communication string in Malaysian language so that the search will not redundant with common word like “hello” and “hi” in the images. Figure 6 below shows the communication string that we use earlier in Twitter experiment, appearing in the search result.

Figure 6 - Communication String That Able to Recover from Phone Image

The search process were executed by using prepared detail steps with the phone images accordingly. If the information were able to be found and recovered, we marked it as right. After all search has been executed and completed, the findings were recorded as shows in Table 7

| Evidences               | Facebook Application | Facebook Web | Twitter Application | Twitter Web | Google+ Web |
|-------------------------|----------------------|--------------|---------------------|-------------|-------------|
| Application keyword     | ✓                     | N/A          | ✓                   | N/A         | N/A         |
| Web URL                 | N/A                  | ✓            | N/A                 | ✓           | ✓           |
| Name after login        | ✓                     | ✓            | ✓                   | ✓           | x           |
| Username after login    | ✓                     | ✓            | ✓                   | ✓           | x           |
| Password after login    | x                     | x            | x                   | x           | x           |
| Activities              | x                     | x            | ✓                   | ✓           | x           |
| Uninstalled data remnant| ✓                     | N/A          | ✓                   | N/A         | N/A         |

In general, all application keyword will appear in the search once we successfully installed the application. These result show that, there were changes in application list once it was installed. The same search result happened to web URL keyword. Once we visited the web,
the keyword will be stored in the browsing history, including date and time stamp. When we login into the selected social media, all profile names were recorded in the phone except Google+ mobile web. As for social media username, only Twitter application and Twitter mobile web were able for retrieval. No password can be seen for all social media. By the same token, the communications and activities only appear in Twitter application and Twitter mobile web. No activities can be seen in Facebook and Google+. Data remnant were able to be retrieved from both Facebook and Twitter application.

4.1.2 Social Media Phone and Memory Images

Analysis of the memory images were the same with the previous analysis. We have also performed the keyword search by using a prepared detail steps. The search were repeated, but confined to only for the value that does not appear in our first search using phone images. Each of the memory images were also opened using AccessData FTK and HxD Hex Editor. For this analysis, we focused more on social media credential and communication activities, since most of this part were not recovered from the phone images analysis. Figure 7 shows Facebook credential that we used to login appearing in our first search attempt.

![Figure 7 - Facebook Credential Found Without Encrypted in Memory Image](image)

The search was continued with other social media credential and communication activities. The summary in Table 8 show the result of what evidences can be retrieved from both phone images and memory images.

Table 8 - Social Media Phone Images and Memory Images Finding Summary

| Evidences         | Facebook Application | Facebook Web | Twitter Application | Twitter Web | Google+ Web |
|-------------------|----------------------|--------------|---------------------|-------------|-------------|
| Keyword after install | ✓                    | N/A          | ✓                   | N/A         | N/A         |
| Web URL           | N/A                  | ✓            | N/A                 | ✓           | ✓           |
The finding of this analysis has shown that, all credentials especially password cannot be seen in phone images, but we were able to retrieve it in memory images except for the Google+. As for the communication activities, we still do not find any keyword matching in Facebook and Google+. We then randomly checked with Facebook and Google+ memory images to find any communication occurrence other than what we recorded, but failed to find any. The result of the study was suggesting that the communication activities in Facebook and Google+ for FxOS were either encrypted or stored in their respective server only.

### 4.2 Instant Messaging Investigation

For instant messaging experiment and investigations, we have managed to acquire 3 instant messaging applications and 1 instant messaging mobile web activities. First, we have explored the residual artifacts generated by application and URL involved for mobile web. Second, we have tried to find the user phone number used during registration. Third, we have tried to get the registration code and SMS or call verification received. Forth, we have traced back what activities that has been captured in the images. Lastly, we have checked for data remnant after we complete the uninstallation the application.

#### 4.2.1 Instant Messaging Phone Image

Like the previous experiment, each of the phone images were opened using AccessData Forensic Toolkit (FTK) version 3.1.2 and HxD Hex Editor 1.7.7.0. The analysis has also conducted by manually searching several keyword related to previous experiment in forensically sounds manner. The first binary search performed was the application keyword. This search executed once the selected application successfully installed from Mozilla Market Place. Since our factory setting base image was only comes with preinstalled application and none of it were used in our investigation, we cannot find any keyword of selected instant messaging from the base image. Figure 8 shows Telegram keyword appear when we executed the first search.
The second search were executed to find URL for instant messaging mobile web, in this experiment only Telegram we tested for mobile web. Mobile web is totally different with the application. We do not had to install, instead we only need to browse using preinstalled Firefox browser. Figure 9 shows Telegram URL was found in the images and the result was also showing the visited date and time.

The third search were to find the phone number used during registration. Most of our selected instant messaging services asked the phone number to be tied with account registration. Instant messaging services like Telegram and WhatsApp will display the phone number together with selected name when we send private message to other user, while Line will not display any number but only selected name. The confirmation during registration were either code from SMS
or received call from the providers. Figure 10 and Figure 11 shows the phone number and registration code for WhatsApp services respectively.

![Image of phone numbers and registration codes]

Figure 10 - Phone Number Used During Registration Found in Phone Memory

![Image of phone numbers and registration codes]

Figure 11 - Registration Code Received During Account Confirmation Found in Phone Memory

The search process were executed by using a prepared detail steps with the phone images accordingly. If the information were able to be found and recovered, we marked it as right. After all search has been executed and completed, the findings were recorded as shows in Table 9

Table 9 - Instant Messaging Phone Image Finding Summary

| Evidences          | Telegram Application | Telegram Web Application | OpenWapp Application | Line Application |
|--------------------|----------------------|--------------------------|----------------------|------------------|
| AS9787E0           | 00 00 00 00 00 00 00 |                          |                      |                  |
| AS9787F0           | 00 00 00 00 00 00 00 |                          |                      |                  |
| AS978800           | 7B 22 73 74 61 74 75 73 22 3A 22 6F 6B 22 2C 22 |                |                      |                  |
| AS978810           | 6C 6F 67 69 6E 22 3A 22 36 3C 32 33 34 35 36 37 38 39 3A | | | |
| AS978820           | 34 31 35 22 2C 22 70 77 22 3A 22 4E 43 69 30 66 | | | |
| AS978830           | 45 50 6F 51 43 69 37 59 55 50 65 63 53 64 30 63 | | | |
| AS978840           | 61 36 71 5A 51 41 3D 22 2C 22 74 79 70 65 22 3A a8q2Q4"=""type": | | | |
| AS978850           | 22 65 75 69 73 74 69 6E 67 22 2C 22 65 75 70 69 | "existing","exp| | |
| AS978860           | 72 61 74 69 6E 6F 6E 22 3A 31 34 36 32 36 30 37 39 | | | |
| AS978870           | 39 3E 2C 22 6E 6F 64 64 22 3A 22 66 62 65 65 22 | | | |
| AS978880           | 2C 22 70 70 69 63 65 22 3A 22 55 53 24 30 2E 39 | | | |
| AS978890           | 39 22 2C 22 63 6F 73 74 22 3A 22 30 2F 39 39 39 22 | | | |
| AS9789A0           | 2C 22 63 75 72 72 65 6E 63 79 22 3A 22 55 53 44 | | | |
| AS9789B0           | 22 2C 22 70 72 69 63 65 6F 65 78 70 69 72 61 74 | | | |
| AS9789C0           | 69 6F 6E 22 3A 31 34 33 33 37 39 38 31 39 31 7D | | | |

Table 9 - Instant Messaging Phone Image Finding Summary
In general, all application keyword will appear in the search once we have successfully installed the application. These result shows that, there were changes in application list once it was installed. The same search result happened to web URL keyword. Once we visited the web, the keyword were stored in the browsing history, including date and time stamp. We try to find the phone number we were used during registration but we only manage to find it in OpenWapp phone images. However, all registration code, SMS as well as call verification were manage to find in all instant messaging phone images. In contrary, no communication activities were manage to trace in this analysis. Data remnant were able to be retrieved from both Facebook and Twitter application.

4.2.2 Instant Messaging Phone and Memory Images

Analysis of the memory images were the same with the previous analysis. We have also performed the keyword search using a prepared detail steps. The search were repeated, but confined to only for the value that does not appear in our first search using phone images. Each of the memory images were also opened using AccessData Forensic Toolkit (FTK) version 3.1.2 and HxD Hex Editor 1.7.7.0. For this analysis, we focused more on registration number and instant messaging communication activities, since most of this part were not recovered from the phone images analysis. Figure 12 shows OpenWapp communication string that we used during this experiment.

| Keyword after install | ✓ | N/A | ✓ | ✓ |
|-----------------------|---|-----|---|---|
| Web URL               | N/A| ✓   | N/A| N/A|
| Number after register | ✘ | ✘   | ✓  | ✘ |
| Registration code     | ✓ | ✓   | ✓ | ✓ |
| SMS/Call verification | ✓ | ✓   | ✓ | ✓ |
| Activities            | ✘ | ✘   | ✘ | ✘ |
| Uninstalled data remnant | ✓ | N/A | ✓ | ✓ |

| 9CBC1690 | 76 4C 32 4E 6C 63 6E 52 70 7A 6D 6C 6A 59 58 52 | vL2N1enRpi2mljYXR |
| 9CBC16A0 | 6C 63 79 35 6E 62 32 52 68 5A 47 52 35 4C 6D 4E | lcy5nb2Rh2G5SLmN |
| 9CBC16B0 | 76 62 53 39 79 5A 58 42 76 63 32 6C 30 33 4A | vbs9yZXBve210B3J |
| 9CBC16C0 | 35 4D 54 41 77 4C 67 59 44 56 51 51 44 45 79 64 | SMTAwLgYD6Q2DEyd |
| 9CBC16D0 | 48 62 79 42 45 59 57 52 6B 65 53 42 54 5A 57 4E | HbyBEYWrkeSBT2WN |
| 9CBC16E0 | 31 63 6D 58 67 51 32 56 79 64 47 6C 6D 61 57 4E | lcmUgZ2ydG1mAWN |
| 9CBC16F0 | 68 64 54 65 73 74 65 6E 67 20 6F 70 65 6E 77 61 | hFTestiNg opeNwe |
| 9CBC1700 | 70 70 20 66 72 6F 6D 20 46 69 72 65 6E 78 20 | pp from Firefox |
| 9CBC1710 | 70 68 6F 6E 68 4D 5A 67 33 4D 42 34 54 65 4E 45 | phoNeHjg3MB1AXDE |
| 9CBC1720 | 30 4D 44 55 77 4D 54 49 77 4E 44 67 77 76 4E 6F | 0MDUWTIwDwVNw |
| 9CBC1730 | 58 44 54 45 31 4D 54 49 7A 4D 54 45 77 4D 54 45 | XbTE1MTIHtPHTe |
| 9CBC1740 | 77 4D 6C 6F 77 50 44 45 68 4D 42 38 47 41 51 55 | wbIowFDEmHoBGAE |
| 9CBC1750 | 45 43 78 4D 59 52 47 39 74 59 57 6C 76 49 45 4E | ECmMYR9g3YLu1Z8 |
| 9CBC1760 | 76 62 6E 52 79 62 32 77 67 56 6D 41 73 61 57 52 | vbrNy2wg0mFzaWR |

Figure 12 - OpenWapp Communication String Found in Memory Image
The search were continued with other instant messaging communication activities and the phone number used in Telegram and Line. The summary in Table 9 shows the result of what evidences can be retrieved from both phone images and memory images.

| Evidences                  | Telegram Application | Telegram Web | OpenWapp Application | Line Application |
|----------------------------|----------------------|--------------|----------------------|------------------|
| Keyword after install      | ✓                    | N/A          | ✓                    | ✓                |
| Web URL                    | N/A                  | ✓            | N/A                  | N/A              |
| Number after register      | ✓                    | ✓            | ✓                    | ✓                |
| Registration code          | ✓                    | ✓            | ✓                    | ✓                |
| SMS/Call verification      | ✓                    | ✓            | ✓                    | ✓                |
| Activities                 | x                    | x            | ✓                    | x                |
| Uninstalled data remnant   | ✓                    | N/A          | ✓                    | ✓                |

The finding of this analysis has shown that, all registration phone numbers that were not able to be seen in phone images, were able to be traced in memory images. As for the communication activities, we have only manage to find the communication string in OpenWapp and still do not find any keyword matching in Telegram and Line. We then randomly checked with Telegram and Line memory images to find any communication occurrence other than what we recorded, but also failed to find any. The result of the studies suggested that the communication activities in Telegram and Line for FxOS were either encrypted or stored in their respective server only.

6. Conclusion

In this research, we have successfully acquired 24 forensic images, each from FxOS internal phone and volatile memory. The acquired images were then extracted based on the action performed as per documented in detail steps, and were named accordingly. These images were then analyzed and the result was presented and tabled. The result of this research has indicated that most of valuable forensic information were residing in volatile memory. The finding of this study also suggested that memory in FxOS phone were not encrypted, hence it was readable with our forensic tools. Therefore, we managed to recover and trace social media account credential especially on Facebook and Twitter services. In contrary, all untraced information in phone images such as profile name on Google+ and phone number used during registration for Telegram and Line, were traceable in memory images.

The other conclusion drawn from this research were, FxOS applications behaving the same way of its counterpart in mobile web. Therefore, we managed to get the exact same forensic traces and evidences when we analyzed the same services, both in application and mobile web platform. For example Facebook, Twitter and Telegram applications are producing the same forensic trace with its mobile web. In short, these findings have significantly enhanced our understanding of the similarity of FxOS application which design based on WebAPI, with its mobile web platforms. By using WebAPI, FxOS application can executed under very minimal memory requirement, just like opening the web browser.
On the other hand, the analysis of the acquired images has shown that application artifacts were remained in FxOS device after it has been uninstalled. The data remnant and left over from the application folders and browser history will be value added to the investigation and many more information have been successfully retrieved. Further, the findings from this research have also make several contributions to the current set of mobile forensic investigation standard. It is recommend that further research to be undertaken with more broad application and mobile web platform. More forensic investigation of FxOS application would help us to establish greater degree of accuracy in this study.

References

Accessdata, 2007. Forensic Toolkit (FTK).
Al Mutawa, N., Baggili, I., Marrington, A., 2012. Forensic analysis of social networking applications on mobile devices. Digit. Investig. 9, S24–S33. doi:10.1016/j.diin.2012.05.007
Anglano, C., 2014. Forensic analysis of WhatsApp Messenger on Android smartphones. Digit. Investig. 11, 1–13. doi:10.1016/j.diin.2014.04.003
Chen, S., Hao, X., Luo, M., 2009. Research of Mobile Forensic Software System Based on Windows Mobile, in: 2009 International Conference on Wireless Networks and Information Systems. IEEE, pp. 366–369. doi:10.1109/WNIS.2009.32
Damshenas, M., Dehghantanha, A., Mahmoud, R., 2014. A Survey on Digital Forensics Trends. Int. J. Cyber-Security Digit. Forensics 3, 1–26.
Dezfouli, F.N., Dehghantanha, A., Eterovic-Soric, B., Choo, K.-K.R., 2015. Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms. Aust. J. Forensic Sci. 0618, 1–20. doi:10.1080/00450618.2015.1066854
Dezfouli, F.N., Dehghantanha, A., Mahmod, R., Mohd Sani, N.F., Shamsuddin, S., 2013. A Data-centric Model for Smartphone Security. Int. J. Adv. Comput. Technol. 5, 9–17. doi:10.4156/ijact.vol5.issue9.2
Geeksphone, 2013. Geeksphone Peak [WWW Document]. URL http://www.geeksphone.com/other-devices-2/
Gómez-Miralles, L., Arnedo-Moreno, J., 2012. Versatile iPad forensic acquisition using the Apple Camera Connection Kit. Comput. Math. with Appl. 63, 544–553. doi:10.1016/j.camwa.2011.09.053
Husain, M.I., Sridhar, R., 2010. iForensics : Forensic Analysis of Instant Messaging on. Lect. Notes Inst. Comput. Sci. Soc. Informatics Telecommun. Eng. - Digit. Forensics Cyber Crime 31, 9–18. doi:10.1007/978-3-642-11534-9_2
Iqbal, A., Marrington, A., Baggili, I., 2014. Forensic artifacts of the ChatON Instant Messaging application. Int. Work. Syst. Approaches Digit. Forensics Eng., SADFE. doi:10.1109/SADFE.2013.6911538
Iqbal, B., Iqbal, A., Obaidli, H. Al, 2012. A novel method of iDevice (iPhone, iPad, iPod) forensics without jailbreaking, in: 2012 International Conference on Innovations in
Irwin, D., Hunt, R., 2009. Forensic information acquisition in mobile networks, in: 2009 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. IEEE, pp. 163–168. doi:10.1109/PACRIM.2009.5291378

Jung, J., Jeong, C., Byun, K., Lee, S., 2011. Sensitive Privacy Data Acquisition in the iPhone for Digital Forensic Analysis. Commun. Comput. Inf. Sci. - Secur. Trust Comput. Data Manag. Appl. 186, 172–186. doi:10.1007/978-3-642-22339-6_21

Karpisek, F., Baggili, I., Breitinger, F., 2015. WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages. Digit. Investig. 1–9. doi:10.1016/j.diin.2015.09.002

Lessard, J., Kessler, G.C., 2010. Android Forensics : Simplifying Cell Phone Examinations. Small Scale Digit. Device Forensics J. 4, 1–12.

Maël Hörz, 2009. HxD Hex Editor.

Mohtasebi, S., Dehghantanha, A., 2013. Towards a Unified Forensic Investigation Framework of Smartphones. Int. J. Comput. Theory Eng. 5, 351–355. doi:10.7763/IJCTE.2013.V5.708

Mohtasebi, S., Dehghantanha, A., 2011a. Defusing the Hazards of Social Network Services. Int. J. Digit. Inf. Wirel. Commun. 1, 504–516.

Mohtasebi, S., Dehghantanha, A., 2011b. A Mitigation Approach to the Privacy and Malware Threats of Social Network Services, in: Digital Information Processing and Communications. pp. 448–459. doi:10.1007/978-3-642-22410-2_39

Mohtasebi, S., Dehghantanha, A., Broujerdi, H.G., 2012. Smartphone Forensics : A Case Study with Nokia E5-00 Mobile Phone. Int. J. Digit. Inf. Wirel. Commun. 1, 651–655.

Pooters, I., 2010. Full user data acquisition from Symbian smart phones. Digit. Investig. 6, 125–135. doi:10.1016/j.diin.2010.01.001

QtADB, 2011. QtADB Android Manager [WWW Document]. URL http://qtadb.wordpress.com

Said, H., Yousif, A., Humaid, H., 2011. IPhone forensics techniques and crime investigation, in: The 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT 11). IEEE, pp. 120–125. doi:10.1109/CTIT.2011.6107946

Sylve, J., Case, A., Marziale, L., Richard, G.G., 2012. Acquisition and analysis of volatile memory from android devices. Digit. Investig. 8, 175–184. doi:10.1016/j.diin.2011.10.003

Taylor, M., Hughes, G., Haggerty, J., Gresty, D., Almond, P., 2012. Digital evidence from mobile telephone applications. Comput. Law Secur. Rev. 28, 335–339. doi:10.1016/j.clsr.2012.03.006

Tso, Y.-C., Wang, S.-J., Huang, C.-T., Wang, W.-J., 2012. iPhone social networking for evidence investigations using iTunes forensics, in: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication - ICUIMC ’12. ACM Press, New York, New York, USA, p. 1. doi:10.1145/2184751.2184827

Walnycky, D., Baggili, I., Marrington, A., Moore, J., Breitinger, F., 2015. Network and device forensic analysis of Android social-messaging applications. Digit. Investig. 14, S77–S84. doi:10.1016/j.diin.2015.05.009
Yang, T.Y., Dehghantanha, A., Choo, K.-K.R., Muda, Z., 2016. Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies. PLoS One 11, e0150300. doi:10.1371/journal.pone.0150300

Yusoff, M.N., Mahmod, R., Abdullah, M.T., Dehghantanha, A., 2014a. Mobile Forensic Data Acquisition in Firefox OS, in: The Third International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec2014). pp. 27–31.

Yusoff, M.N., Mahmod, R., Abdullah, M.T., Dehghantanha, A., 2014b. Performance Measurement for Mobile Forensic Data Acquisition in Firefox OS. Int. J. Cyber-Security Digit. Forensics 3, 130–140.

Yusoff, M.N., Mahmod, R., Dehghantanha, A., Abdullah, M.T., 2014c. An Approach for Forensic Investigation in Firefox OS, in: The Third International Conference on Cyber Security, Cyber Warfare, and Digital Forensic (CyberSec2014). IEEE, pp. 22–26.

Yusoff, M.N., Mahmod, R., Dehghantanha, A., Abdullah, M.T., 2014d. Advances of Mobile Forensic Procedures in Firefox OS. Int. J. Cyber-Security Digit. Forensics 3, 141–157.