Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2017; 5(11):644-654 **ISSN 2347-9523 (Print)**
©Scholars Academic and Scientific Publisher
(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Forensics Analysis of Dcard Application on Android Smartphone

Ching-Yu Lin[1], Ming-Sang Chang[2*]
[1,2]Department of Information Management, Central Police University, Taoyuan, Taiwan

***Corresponding author**
Ming-Sang Chang

Article History
Received: 13.11.2017
Accepted: 18.11.2017
Published: 30.11.2017

DOI:
10.21276/sjet.2017.5.11.7

Abstract: Nowadays, mobile devices are considered as an important necessity in our daily lives. Mobile devices can keep us connecting with family, friends, colleagues and so on. Moreover, combining with the sprung up social networking sites, lots of people have permanently altered their way of living habits. For example, more and more new mobile devices have successfully launched in the market, people in the world can easily contact with anyone everywhere at any time. However, there are some problems we should concern. Thanks to the new technology and cyber worlds are prosperous, the mobile phones and social networking sites have become a tool or medium for perpetrators to commit a crime in recent years. As a result, the investigators should collect important evidences from all the digital devices at the crime scene, such as computer, mobile phone, tablet and so forth. In this paper, we focus on the mobile forensics of Dcard application running on the mobile phone. In the experiment, we strive to search the digital evidence that user has been done on the mobile phone. We adopt "dd" instruction to create an image file for the physical memory. The physical memory is also called non-volatile memory. The data stored in the non-volatile memory would not be vanished when the power is cutting off. Thereafter, we make use of authoritative computer forensic tools which is called FTK Imager to analyze these significant evidences and the correlation between them in detail. In addition, we also adopt SQLite editor to examine and analyze the database on the mobile phone. The analysis of experiment was performed with the aim of determining what information can be found on the device. We find that which behavior of suspect will leave what kind of evidences in the mobile phone. These findings could be an important reference for law enforcement agency to investigate a cybercrime.

Keywords: social networking sites, mobile forensics, crime investigation, Dcard

INTRODUCTIOIN

Thanks to the prosperous of Internet technology, the use of mobile devices in crime was widely recognized for some years. Therefore, a proliferation of mobile devices on the market caused a demand for forensic examination [1]. In recent years, the social networking site has been a significant medium for people to enhance their interpersonal relationships [2]. The popularity of social networking site has given rise to the number of social networking users for recreation, business or any other purposes. The definition of "social network" is a community where people across the globe world online that can develop network with different individuals for a specific purpose [3]. People can make use of social networking sites to build up their profile. A profile is a list of identifying information that can portray users' online identity, including name, photograph, hometown, birthday, personal preferences and so forth [4]. The prevalence of social networking websites has changed the living habits of many people. Many people share their emotion or daily life with their friends via texting, photographing or videoing. All the sharing leaves behind informative trails about the personalities, friends, behaviors, motivations and activities of a person [5]. As a result, social networking sites can connect people and maintain relationships from all parts of lives [6]. On the other hand, people can browse other people's news feed on social networking sites to relieve their working pressure or any other kinds of pressures in their daily life. There is no doubt that people have incorporated social networking sites into their lives and made using social networking sites a frequent daily activities.

According to the survey of STATISTA, social networking sites are popular nowadays, such as Facebook, YouTube, WhatsApp, Instagram, Twitter and so forth [7]. Many of them have over than several million members, a quite large number for the time. However, there are still a lot of distinguished social networking sites in Taiwan, such as Plurk, Dcard, Xuite and so on. It is worth noting a thing, the young between at the age of twelve and twenty-four, the visiting ratio of Dcard, Instagram and YouTube is much higher than the other age. This situation indicates that Dcard is noted for the college students in Taiwan nowadays. The registration members have over than one million people and the average of posts are created per every ten seconds. Therefore, we can predict that Dcard will be the mainstream of the social networking sites without dispute in the feature.

Owing to the advancement of scientific technology, many kinds of crimes are getting much more complexity than before. Currently, the traditional crimes such as killing people, domestic violence, stealing and robbing are on the decrease now. In other words, the high technology crimes such as cybercrime and computer crime are increasing nowadays. There are more and more criminals using social networking sites to commit the cybercrime because of its convenience and anonymity characteristics. Therefore, the cybercrime and computer crime have already become the mainstream of all the crimes nowadays. However, cybercrime refers to a perpetrator that abused or destroyed a computer to commit a crime. The cybercrime is definitely different to the traditional crime.

In recent years, various kinds of cybercrimes have emerged endlessly due to the anonymity characteristic of the Internet. In other words, anonymity is largely tied to the cybercrime nowadays. Moreover, it is also claimed that the anonymity characteristic allows criminals to use the Internet without the possibility of detection. According to the survey of National Police Agency, Ministry of the Interior Republic of China, the statistics show the cybercrimes happened in Taiwan between January and June in 2017, there are 6,567 cybercrime cases occurred. The cybercrime ratio increases 4.39 percentages relative to the same period of last year. However, the perpetrators who are at the age of 18 to 23 called adolescents are increasing 28.07 percentages relative to the same period of last year. The victims who are more than 50 years old are increasing 43.54 percentages relative to the same period of last year [8]. Furthermore, Catherine D. Marcum, *et al*. categorized several types of social networking criminality, for example, identity theft, digital piracy, cyberbullying, sexual violence and so on [9]. Therefore, we can realize that the social networking websites have seriously become a hotbed of cybercrimes.

The rest of this paper is organized as follows. In the next section, we present our related work. We'll introduce Dcard social networking site, the concept of rooting a mobile phone, the related background of ADB tool and Buxybox, and the forensic tools we used. In the section 3, we present our methodology and system architecture. We'll describe our mobile forensic methods and how to extract the digital evidence in physical memory without commercial forensic tools. In the section 4 and section 5, we present the results and findings of our experiment. We'll describe how to extract important information from physical memory and log file. Finally, we summarize our conclusions and future work.

RELATED WORK

Dcard Social Networking Site

Nowadays, there are many kinds of social networking sites exist in the world. Social networking sites have already made the distance of everyone much closer than before. Now, there is a particular social networking site emerges in recent years. Dcard is a popular and free social networking site service for college students in Taiwan. Dcard application was launched by a Taiwan university student called Chin Yu, Chien on December 16, 2011 [10]. Up to present, Dcard was largely accepted and loved by Taiwan college students. According to the survey of Alexa, Dcard was ranked 26th relative to other websites in Taiwan and 1,050th relative to other websites in the world [11]. It's worth noting that Dcard had already permitted more college students to register it last year. Up to now, there are 169 universities involve in the Dcard in Taiwan. Therefore, we predict that Dcard will come out on top in the next few years. Now, Dcard offers an additional service, allowing foreign students of overseas countries to take part in the activities of Dcard social networking sites.

Dcard provides users to write posts for the sake of expressing their feelings or sharing their daily activities with their friends or other users. On the contrary, other users can make any comments with anonymity on authors' news feed. However, one of Dcard advantages is anonymity characteristic. In other words, when you write a post on the news feed, nobody knows who you are. Owing to this advantage of Dcard, it would easily cause a person with bad intention to commit a cybercrime. Hence, it is hard for investigators to investigate cybercrimes because of the anonymity characteristic. On the other hand, Dcard allows users to chat with their friends in the chat room. However, they cannot chat with strangers. They can just only chat with friends they added in the past. In the midnight, college students can draw a card to make a chance for meeting a new friend. If both of them like each other and send the friend request to each other, then they can be friends. Otherwise, they will no longer meet with each other again on Dcard. The goal of Dcard is to let all the college students in Taiwan have a chance to acquaint with each other.

However, there are many literatures focus on the forensic analysis of social networking sites. Abdullah Azfar *et al.* proposed the utility model for the evidence extraction of five social networking applications, including Twitter, POF Dating, Snapchat, Fling and Pinterest [12]. Thakur focused on the forensic analysis of WhatsApp application on storage devices and volatile memory [13]. Mutawa et al. focused on the forensic analysis of three popular social networking sites, including Facebook, Twitter and Myspace [14]. Yet, the technical literature about Dcard forensics is relatively scarce. From the point of this view, this paper focuses on the evidence extraction and crime analysis of Dcard application. In this paper, we study the behavior of a user who logins into the Dcard on the mobile phone. We strive to find out adding friend evidence, creating post evidence, making comment evidence, browsing behavior evidence, chatting record evidence and so forth. Moreover, we analyze the correlation between these evidences and discuss how these evidences can assist law enforcement agencies to investigate a crime.

Root

Rooting is a process of allowing users to gain privileged control which is known as root over the various Android systems. The devices include mobile phones, tablets or any other electronic device that is running Android mobile operating system could obtain highest authority when they rooted the phone. Rooting is often carried out with the aim of overcoming limitations that mobile operators and developers put on some devices. Therefore, rooting gives capability to alter system settings, run special applications that require administrator-level permissions [15]. No matter which brand of mobile phones, the manufacturer would limit some usage functions for the sake of protecting its mobile phone system. The protection mechanism can avoid users to alter or delete some important files, preventing its system being crashed. However, this protection mechanism obstructs investigators to conduct mobile forensics simultaneously. Therefore, in order to obtain more information on the mobile phone, the investigators should execute a series of rooting processes before examining a mobile phone. There are a lot of methods for rooting a mobile phone now. In the experiment, we make use of the third part ROM package to override its original system. ROM package is a modified version of Android operating system. It provides the possibility to use an unreleased or newer version of Android on the phone that might usually not be available from the device because of the restrictions from manufacturers. After we root the mobile phone, we can read and write some special applications that require administrator-level permissions.

Android Debug Bridge

Android debug bridge (ADB) is a versatile command-line tool that lets users communicate with connected Android devices or emulators. Android debug bridge command also facilitates a variety of devices actions, for example, installing or debugging applications. It provides access to a Unix shell that can be used to run a variety of commands on a device.

Busybox

Busybox is software that offers several stripped-down UNIX tools in a single executable file. It runs in a variety of portable operating system interface environments such as Linux, Android and FreeBSD. It provides minimalist replacements for the most common utilities such as ls, cp, mv, mount and so on. Therefore, Busybox will implement more commands that are necessary for some root applications to work properly. In other words, there are limited instructions in ADB environments. If users want to execute more commands, they can install Busybox on the Android devices.

Tools

There are a lot of mobile forensics related literatures discussing about the digital evidence which is extracted on the phone by using the commercial mobile forensic tools. Mahajan *et al.* conducted the mobile forensics for the WhatsApp and Viber applications by using Cellebrite UFED forensic tool. They strove to find the evidences of chat messaging, send and received image, video files and so on for these two different applications on the Android mobile device. However, when they examined the Viber application by using UFED Physical Analyzer, no traces, no artifacts or any message history was found [16]. Faheem et al. conducted the mobile forensics for the Viber application by using Android Debug Bridge tool instead of using commercial forensic tool. They can find out the chatting message, timestamp, account information, picture and so on [17]. As a result, we can realize that the investigators can’t just only rely on the commercial forensic tools to investigating a crime. The function of commercial forensic tools may be different according to the brand and type of an item, and its system version. Furthermore, the version update speed of commercial forensic tool is far behind than the production of new mobile phone. On the other hand, the high price of commercial forensic tools is a major consideration for the law enforcement agency. Not every law enforcement agency can afford these high cost commercial forensic tools. Therefore, these reasons may be an obstacle for law enforcement agencies to conduct an investigation.

K.K. Arthur *et al*. conducted an investigation into some of forensic tools, including PC Inspector File Recovery, EnCase, Forensic Toolkit and FTK Imager. However, the main function of FTK Imager is to view and to image storage devices [18]. Forensic Toolkit is forensic software made by AccessData. In light of these advantages, we adopt AccessData FTK Imager V4.1.1 to analyze image files. The toolkit comprises a standalone disk image program called FTK Imager. In the experiment, in order not to influence the integrity of digital evidence, we try to use Android Debug Bridge tool to create image files for the physical memory on the mobile phone. Thereafter, we make use of AccessData FTK Imager V4.1.1 to analyze the image files on another clean computer. On the other hand, we also make use of SQLite editor to inspect the database on the mobile phone. We strive to find out the evidence of creating post, browsing behaviors, making comments, clicking “Like” button and so forth.

In this paper, all the experiments were conducted on the real system. The mobile phone system was installed Android 5.0. The central processing unit is 2.3 GHz. The memory size is 4 Gigabytes. All the specifications of the devices we used are list in the Table 1.

Table-1: List of hardware and software used for analysis

Devices/Tools	Introduction	Specification/Versions
ASUS Zenfone 2	Android mobile phone	ASUS_Z00AD Android 5.0 CPU 2.3GHz Memory 4G GPS 29.19.15.220149 WiFi 6.37.32.RC23.34.22 Bluetooth V10.00.01 Battery Z2N3 01040003 1521
Busybox	Providing a series of standard Unix instructions	Version 1.20.1
Android Debug Bridge	A debugging tool for Android system	Version 26.0.1
FTK Imager	Forensic tool	Version 4.1.1
SQLite Editor	Database Editor	Version 7.0

METHODOLOGY

Experiment elaboration

In the experiment, we download and install Dcard application from Google Play on the mobile phone. Then, we login into the Dcard application with personal account and password. After we login into the Dcard application, we do a series of normal behaviors, for example, adding friends, chatting with friends, writing posts, making comments, clicking “Like” button, clicking “collect” button and so forth. Afterwards, we create an image file for physical memory on the mobile phone. Finally, we make use of Forensic Toolkit Imager to extract and analyze important digital evidences from image file. On the other hand, we make use of SQLite editor to inspect the database on the mobile phone. We strive to find out the evidence of creating post, browsing behaviors, making comments, clicking “Like” button and so forth.

However, before we create an image file, we will show how to get the highest administrator-level permissions on the mobile phone first. After we get the administrator-level permission on the phone, we download and install Busybox on the mobile phone. Simultaneously, we download and install Android Debug Bridge on the computer. After we are ready for these woks, we connect the mobile phone with the computer. And then, we use “dd” instruction to extract data which is stored in the physical memory. The execution process is shown in the following steps.

Obtain administrator-level permissions

When we conduct the mobile forensic, we need to gain the highest administrator-level permissions for the sake of acquiring more important information on the phone. Otherwise, we can only get some limited information. There are many kinds of rooting instructions on the Internet. Every mobile phone has its unique rooting method. In our experiment, our mobile device is ASUS Zenfone 2 which is equipped with Android 5.0 operating system. The rooting processes are showing as follows:

- Download ASUS Zenfone 2 package which can be downloaded from https://www.fairyhorn.cc/172/asus-root-z551mlandroid-50.
- Click “Settings” and there is an “About” option. Go into “About” and keep clicking “Software version” until become developer mode.
- Click “settings” and there is a “Developer options” option. Go into “Developer options” and check the box “USB debugging”.
- Connect the phone with the computer.
- Open the package and conduct the program.
- After we root the phone, we download and install SuperSU application from Google Play. We can use this application to control the permissions for any applications.

Download ADB tool and Busybox

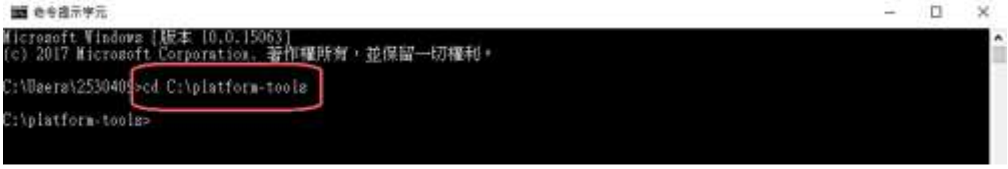
There are two main parts we need to concern, one is ADB tool that we need to install on the computer, and the other one is Busybox that we need to install on the mobile phone. Therefore, we divide this procedure into two parts which are shown as follows.

Mobile phone

In the experiment, we always need to make use of much more Linux instruction sets. Therefore, we need to download and install Busybox to help us execute more Linux instructions such as “dd” instruction. We can download Busybox application on Google Play.

Computer

- Download the ADB tool from the Internet which can be available from https://developer.android.com/studio/releases/platform-tools.html. There are three options for users, including SDK Platform-Tools for Windows, SDK Platform-Tools for Mac and SDK Platform-Tools for Linux. Users can choose anyone that is suit for their computer environment.
- After we download the SDK Platform-Tools, we decompress the zip file. There are many kinds of files in the folder, including .exe files, .dll files, .txt files and so on. However, we are interested in adb.exe file.
- Now, we put this “platform-tools” folder into “C:\”. Then, open the Windows command line and use “cd” command to enter in the relative position. For example, we enter the command “cd C:\platform-tools” to enter in the relative position, as shown in the Fig. 1.



```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\2530409>cd C:\platform-tools

C:\platform-tools>
```

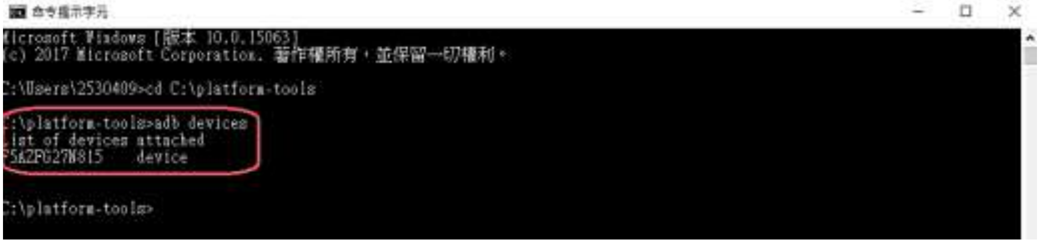
Fig–1: Enter into the relative position.

Extract data and create image file from physical memory

Now, we connect the mobile phone with the computer by using the phone cable. First, we enter “adb devices” command to connect the two devices. If the two devices are connected, the message will show the list of devices attached, as shown in the Fig. 2. Next, we enter “adb shell” command to execute remote control and now the mark sign will become “$”, as shown in the Fig. 3. Then, we need to obtain the administrator-level permissions. Thus, we enter “su’ command and the mark sign will become “#”, as shown in the Fig. 4.

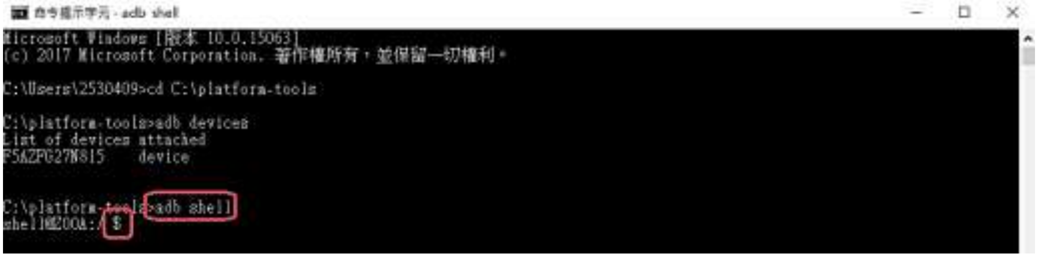
Now, we can enter “busybox df -h” command to inspect the system partition, path, volume, space usage, available space and so on, as shown in the Fig. 5. However, most of the application data installed and stored on the phone would locate at the data partition. Therefore, we are interested in this data partition. The path of data partition is “/dev/block/by-name/data”. Now, we use “dd” command to create an image file for this partition. “dd” command can perform physical imaging by adopting bit-by-bit method. We enter “busybox dd if=/dev/block/by-name/data of=/storage/MicroSD/dataexperiment conv=noerror bs=4096” command to create an image file, as shown in the Fig. 6. The string behind “if” is a partition that we would create an image file. The string behind “of” is image output path. In the experiment, we name the output image file “dataexperiment.img” and store it on the external SD card. The “conv=noerror” represents that there is no interruption when there is an error occurs. The “bs” represents the block size that we would write and read per time.

After we complete the image file creation, we need to put this image file into the computer in order to facilitate analyzing it. Therefore, we enter “adb pull /storage/MicroSD/dataexperiment.img C:\ImageFile”. We put this image file into the “ImageFile” folder on the drive C, as shown in the Fig. 7. After that, we make use of FTK Imager to analyze this image file.



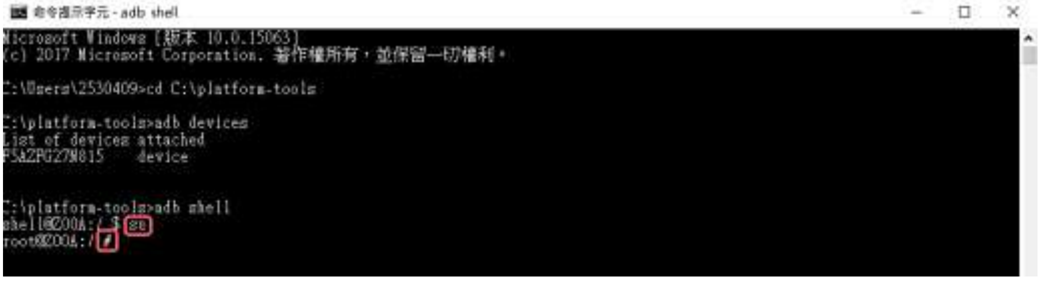
```
icrosoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。

:\Users\2530409>cd C:\platform-tools

:\platform-tools>adb devices
ist of devices attached
5A2PG27N815     device

:\platform-tools>
```

Fig–2: Connect the computer with the mobile phone



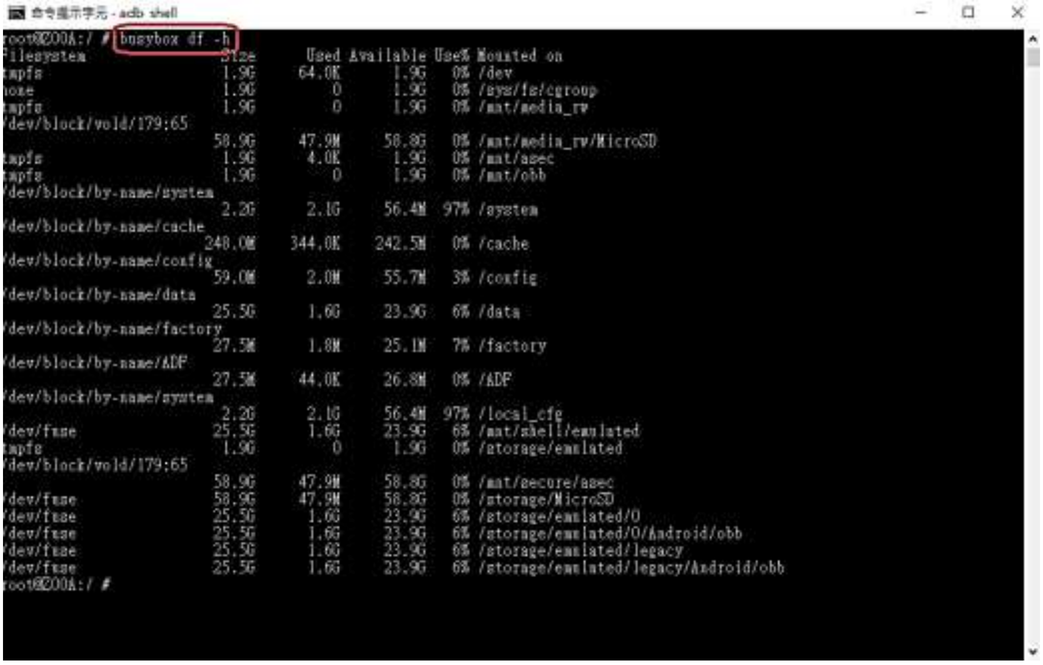
```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\2530409>cd C:\platform-tools

C:\platform-tools>adb devices
List of devices attached
F5AZFG27N815    device


C:\platform-tools>adb shell
shell@Z00A:/ $
```

Fig–3: Start ADB server



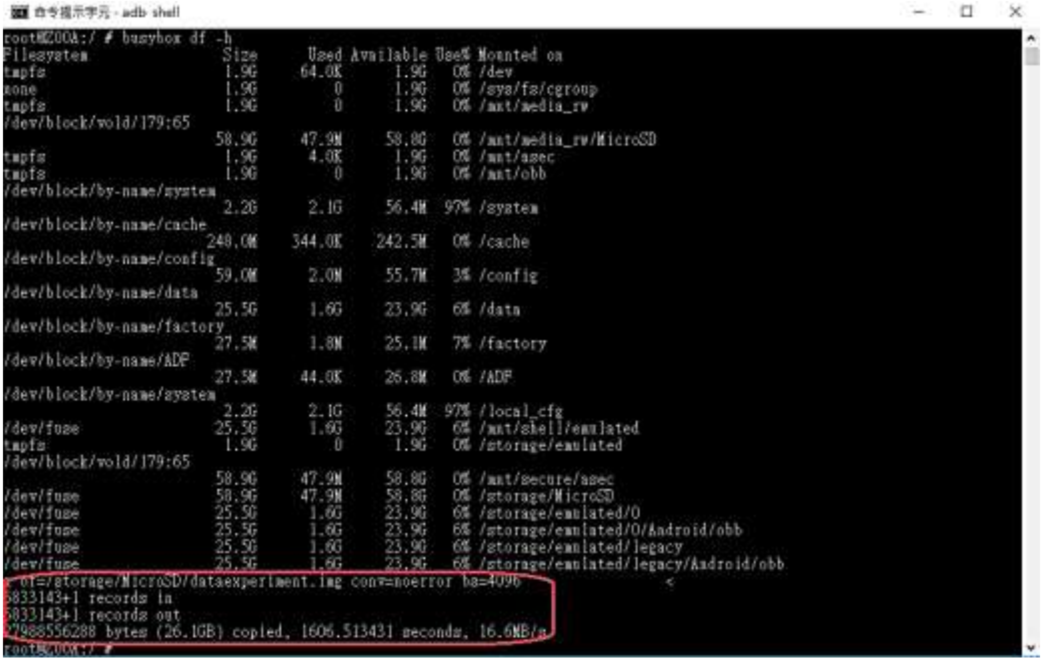
```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\2530409>cd C:\platform-tools

C:\platform-tools>adb devices
List of devices attached
F5AZFG27N815    device


C:\platform-tools>adb shell
shell@Z00A:/ $ su
root@Z00A:/ #
```

Fig–4: Administrator-level permission



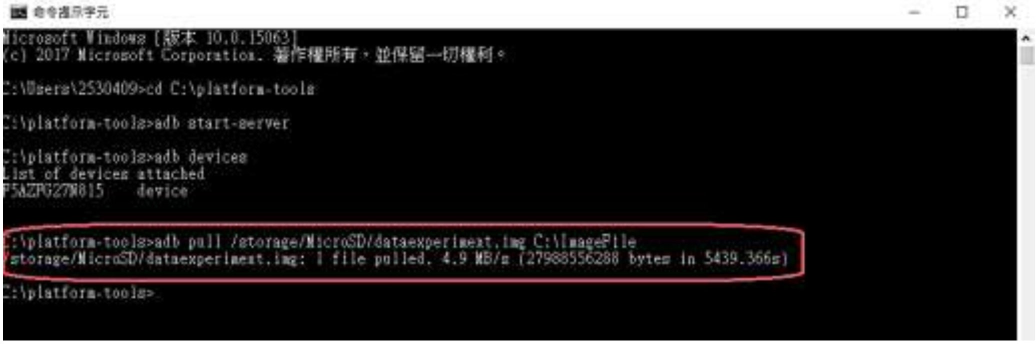
```
root@Z00A:/ # busybox df -h
Filesystem                Size      Used Available Use% Mounted on
tmpfs                     1.9G     64.0K      1.9G   0% /dev
none                      1.9G         0      1.9G   0% /sys/fs/cgroup
tmpfs                     1.9G         0      1.9G   0% /mnt/media_rw
/dev/block/vold/179:65
                         58.9G     47.9M     58.8G   0% /mnt/media_rw/MicroSD
tmpfs                     1.9G      4.0K      1.9G   0% /mnt/asec
tmpfs                     1.9G         0      1.9G   0% /mnt/obb
/dev/block/by-name/system
                          2.2G      2.1G     56.4M  97% /system
/dev/block/by-name/cache
                        248.0M    344.0K    242.5M   0% /cache
/dev/block/by-name/config
                         59.0M      2.0M     55.7M   3% /config
/dev/block/by-name/data
                         25.5G      1.6G     23.9G   6% /data
/dev/block/by-name/factory
                         27.5M      1.8M     25.1M   7% /factory
/dev/block/by-name/ADF
                         27.5M     44.0K     26.8M   0% /ADF
/dev/block/by-name/system
                          2.2G      2.1G     56.4M  97% /local_cfg
/dev/fuse                25.5G      1.6G     23.9G   6% /mnt/shell/emulated
tmpfs                     1.9G         0      1.9G   0% /storage/emulated
/dev/block/vold/179:65
                         58.9G     47.9M     58.8G   0% /mnt/secure/asec
/dev/fuse                58.9G     47.9M     58.8G   0% /storage/MicroSD
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/0
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/0/Android/obb
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/legacy
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/legacy/Android/obb
root@Z00A:/ #
```

Fig–5: System partition



```
rootMZ00A:/ # busybox df -h
Filesystem                Size      Used Available Use% Mounted on
tmpfs                     1.9G     64.0K      1.9G   0% /dev
none                      1.9G         0      1.9G   0% /sys/fs/cgroup
tmpfs                     1.9G         0      1.9G   0% /mnt/media_rw
/dev/block/vold/179:65
                         58.9G     47.9M     58.8G   0% /mnt/media_rw/MicroSD
tmpfs                     1.9G      4.0K      1.9G   0% /mnt/asec
tmpfs                     1.9G         0      1.9G   0% /mnt/obb
/dev/block/by-name/system
                          2.2G      2.1G     56.4M  97% /system
/dev/block/by-name/cache
                        248.0M    344.0K    242.5M   0% /cache
/dev/block/by-name/config
                         59.0M      2.0M     55.7M   3% /config
/dev/block/by-name/data
                         25.5G      1.6G     23.9G   6% /data
/dev/block/by-name/factory
                         27.5M      1.8M     25.1M   7% /factory
/dev/block/by-name/ADF
                         27.5M     44.0K     26.8M   0% /ADF
/dev/block/by-name/system
                          2.2G      2.1G     56.4M  97% /local_cfg
/dev/fuse                25.5G      1.6G     23.9G   6% /mnt/shell/emulated
tmpfs                     1.9G         0      1.9G   0% /storage/emulated
/dev/block/vold/179:65
                         58.9G     47.9M     58.8G   0% /mnt/secure/asec
/dev/fuse                58.9G     47.9M     58.8G   0% /storage/MicroSD
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/0
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/0/Android/obb
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/legacy
/dev/fuse                25.5G      1.6G     23.9G   6% /storage/emulated/legacy/Android/obb
 of=/storage/MicroSD/dataexperiment.img conv=noerror bs=4096                          <
6833143+1 records in
6833143+1 records out
27988556288 bytes (26.1GB) copied, 1606.513431 seconds, 16.6MB/s
rootMZ00A:/ #
```

Fig-6: Create an image file



```
Microsoft Windows [版本 10.0.15063]
(c) 2017 Microsoft Corporation. 著作權所有，並保留一切權利。

C:\Users\2530409>cd C:\platform-tools

C:\platform-tools>adb start-server

C:\platform-tools>adb devices
List of devices attached
F5AZFG27W815    device

C:\platform-tools>adb pull /storage/MicroSD/dataexperiment.img C:\ImageFile
/storage/MicroSD/dataexperiment.img: 1 file pulled. 4.9 MB/s (27988556288 bytes in 5439.366s)

C:\platform-tools>
```

Fig-7: Pull the image file from mobile phone into computer

RESULTS ANALYSIS

We login into the Dcard application by entering the email account and password on the mobile phone. Then, we perform a series of user normal behaviors, for instance, writing a post, making comments, browsing other user's post, adding friends, chatting with friends and so forth. After we perform these normal behaviors, we create an image file for the physical memory on the mobile phone. Thereafter, we make use of quick search keyword function of FTK Imager to analyze the image file. Moreover, we adopt SQLite Editor to examine the database on the mobile phone. The following are the analysis and description of our forensic results.

Account and password

In the mobile phone, there are various kinds of evidence we can extract. First, we analyze the image file which was created from the physical memory on the mobile phone. We can find out the login information by searching the keyword "login", as shown in the Fig. 8. When we analyze its contexts, we can find that there is an email account. Now, we keep looking down the contexts, we can find that there is a keyword "dcard". As a result, we can realize that the e-mail account must be the login information of Dcard application. However, when we strive to find the password information and try to search the keyword "password", there is no clue we can trace. Therefore, we infer that the password may be stored in the database.

Now, we start to examine the login information in the database. We make use of SQLite Editor to examine the database. Most of the application services installed on the phone, its related data would store in a data folder which is located at "/data/data/". There are many kinds of folders and files stored in this data folder. However, we take Dcard application as our experiment target in our experiment. Therefore, we need to search the files or folders related to the keyword "dcard". In the path of "/data/data/", we can find that there is a "com.sparkslab.dcardreader" folder. We open this folder, and we can see there is a database folder. Now, we open this database folder, we can see there are three database files, including "bilanx.db", "google_analytics_v4.db" and "google_app_measurement_local.db". However, most of the information is stored in the "bilanx.db" file. After we examine this file, we can't find the login account and password information. Therefore, we guess this kind of important information may be stored in the server side.

Posting evidence

Every posting has its classification and its post ID. The post ID is a unique string of number. The classification can be divided into several groups, such as boyslove, fitness, relationship, girl, makeup, dressup, entertainer, sport, funny, vehicle, talk, marvel, horoscopes, food, pet, handicrafts, trending, mood, movie, music, game, boy, photography, job, travel, book, language, abroad, literature, exam, course, sex, and so forth. When a user writes a post on Dcard application in the mobile phone, the system will automatically assign a unique ID. This unique ID is a post ID, for example, 227603749. Hence, when we analyze the image file on the FTK Imager, we can search the keyword "post" or "postCreated" to find the creating record, creating date of posting and posting title in the physical memory, as shown in the Fig. 9.

In the database, most of the data information is stored in the "bilanx.db" file. Therefore, we aim at this file to find out the creating posts evidence. The situation is the same in the physical memory, we can also find out the posting record by searching the keyword "post" or "postCreated" in the "bilanx.db" file.

Making comment evidence

Dcard allows any people to make any comments on any articles with anonymity. In the physical memory, we can find the making comment record by searching the keyword "postCommentCreated", as shown in the Fig. 10. However, due to the anonymity characteristic of Dcard application, we cannot find out the user's ID who left messages on the posting. We infer that the ID may be hidden by the application. Therefore, we actually don't know who makes comments on the posting. However, we can't find the record that other users make comments on our own posting as well, but we can find the record that we make comments on the other users' posting.

In the database, the situation is the same in the physical memory, we can find out the making comment record in the "bilanx.db" file. However, we can only find the record that we make comments on the other users' posting, but we can't find the record that other users make comments on our own posting.

Browsing evidence

When a user browsed the other users' posting, the devices would record browsing history in the physical memory. All of the browsing behaviors would leave browsing evidence in the physical memory. Therefore, when we search the keyword "postViewed", there is an obvious post ID we can see in the contexts, as shown in the Fig. 11. By examining the post ID, we can easily realize that the user must have browsed that posting in the past. If the post ID is same to my own post ID, this situation represents that I have browsed my own posting in the past. On the contrary, if the post ID is different to my own post ID, this situation represents that I have browsed the other users' posting in the past.

In the database, the situation is the same in the physical memory, we can also find out the browsing evidence by searching the keyword "postViewed" in the "bilanx.db" file.

Chatting records

In the physical memory, we can extract the chatting record evidence. Every user has their personal ID which was assigned when their personal account was created. Moreover, this ID is unique to every user. When a user chatted with their friends, the system would automatically record the friend's ID in the physical memory. Therefore, when we search the keyword "messageViewed" and "messageSent", we can easily find out the chatting record evidence.

In the database, the situation is the same in the physical memory, we can also find out the chatting record evidence by searching the keyword "messageViewed" and "messageSent" in the "bilanx.db" file.

However, when we strive to find the chatting content in the physical memory and in the database, we can't find the contents by searching the key string. As a result, we guess that the chatting content may be stored in the server side.

Clicking "Like" button evidence

There is a function on the Dcard application called "Like". If people like an article, they may click "Like" button on that article. We can find out the clicking "Like" evidence by searching the keyword "postLiked" in the physical memory. Therefore, the investigator can realize the preference of a perpetrator by analyzing the clicking "Like" articles.

In the database, the situation is the same in the physical memory, we can also find out the clicking "Like" evidence by searching the keyword "postLiked" in the "bilanx.db" file.

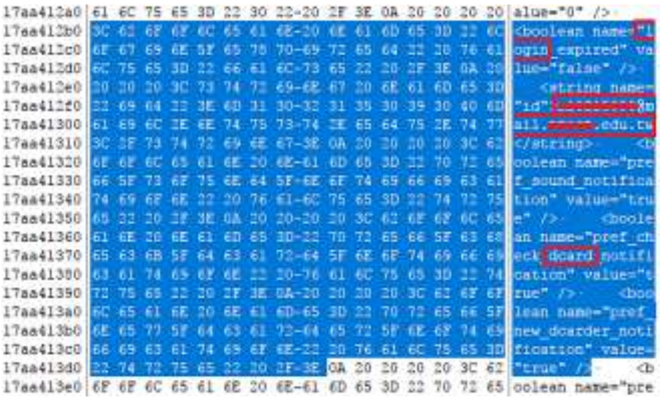
Clicking "Collect" button evidence

In addition, there is also a function on the Dcard application called "Collect". If people like an article, they can click "Collect" button to collect this article in their personal page. After that, they can view the posting they like in their personal page at any time. In the physical memory, we found the clicking "Collect" evidence by searching the keyword "postCollected". Afterwards, the investigator can also realize the preference of a perpetrator by analyzing the clicking "Collect" articles.

In the database, the situation is the same in the physical memory, we can also find out the clicking "Collect" evidence by searching the keyword "postCollected" in the "bilanx.db" file.

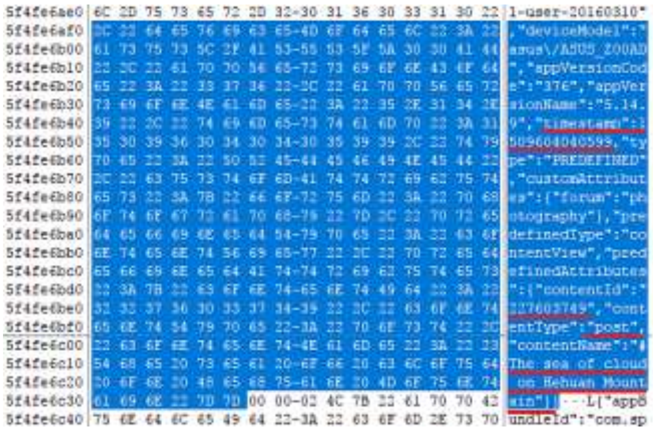
Friend list and friend request

However, when we searched the keyword in the physical memory, for example, "friend request", "friend list", "request", "list" and so on, we can't find out the friend request and friend list in the mobile phone. Also, we adopt SQLite Editor to examine the local database, we can't find out the friend list and friend request. We infer that the friend list and friend request record are stored in the server side.



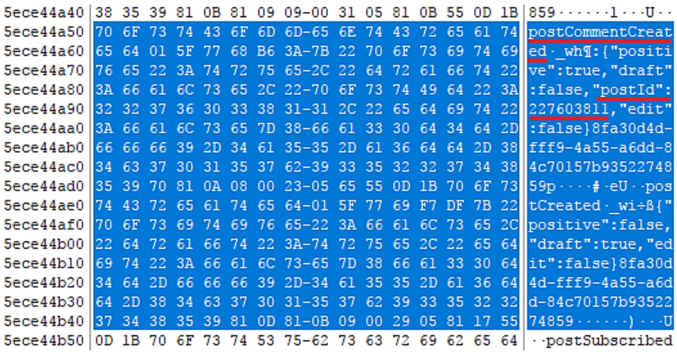
```
17aa412a0 61 6C 75 65 3D 22 30 22-20 2F 3E 0A 20 20 20 20 alue="0" />
17aa412b0 3C 62 6F 6F 6C 65 61 6E-20 6E 61 6D 65 3D 22 6C <boolean name="l
17aa412c0 6F 67 69 6E 5F 65 78 70-69 72 65 64 22 20 76 61 ogin_expired" va
17aa412d0 6C 75 65 3D 22 66 61 6C-73 65 22 20 2F 3E 0A 20 lue="false" />
17aa412e0 20 20 20 3C 73 74 72 69-6E 67 20 6E 61 6D 65 3D    <string name=
17aa412f0 22 69 64 22 3E 6D 31 30-32 31 35 30 39 30 40 6D "id">[redacted]@m
17aa41300 61 69 6C 2E 6E 74 75 73-74 2E 65 64 75 2E 74 77 ail.[redacted].edu.tw
17aa41310 3C 2F 73 74 72 69 6E 67-3E 0A 20 20 20 20 3C 62 </string>    <b
17aa41320 6F 6F 6C 65 61 6E 20 6E-61 6D 65 3D 22 70 72 65 oolean name="pre
17aa41330 66 5F 73 6F 75 6E 64 5F-6E 6F 74 69 66 69 63 61 f_sound_notifica
17aa41340 74 69 6F 6E 22 20 76 61-6C 75 65 3D 22 74 72 75 tion" value="tru
17aa41350 65 22 20 2F 3E 0A 20 20-20 20 3C 62 6F 6F 6C 65 e" />    <boole
17aa41360 61 6E 20 6E 61 6D 65 3D-22 70 72 65 66 5F 63 68 an name="pref_ch
17aa41370 65 63 6B 5F 64 63 61 72-64 5F 6E 6F 74 69 66 69 eck_dcard_notifi
17aa41380 63 61 74 69 6F 6E 22 20-76 61 6C 75 65 3D 22 74 cation" value="t
17aa41390 72 75 65 22 20 2F 3E 0A-20 20 20 20 3C 62 6F 6F rue" />    <boo
17aa413a0 6C 65 61 6E 20 6E 61 6D-65 3D 22 70 72 65 66 5F lean name="pref_
17aa413b0 6E 65 77 5F 64 63 61 72-64 65 72 5F 6E 6F 74 69 new_dcarder_noti
17aa413c0 66 69 63 61 74 69 6F 6E-22 20 76 61 6C 75 65 3D fication" value=
17aa413d0 22 74 72 75 65 22 20 2F-3E 0A 20 20 20 20 3C 62 "true" />    <b
17aa413e0 6F 6F 6C 65 61 6E 20 6E-61 6D 65 3D 22 70 72 65 oolean name="pre
```

Fig–8: The evidence of login



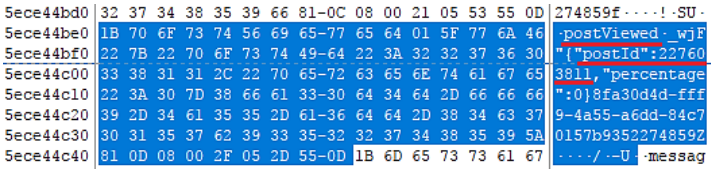
```
5f4fe6ae0 6C 2D 75 73 65 72 2D 32-30 31 36 30 33 31 30 22 l-user-20160310"
5f4fe6af0 2C 22 64 65 76 69 63 65-4D 6F 64 65 6C 22 3A 22 ,"deviceModel":"
5f4fe6b00 61 73 75 73 5C 2F 41 53-55 53 5F 5A 30 30 41 44 asus\/ASUS_Z00AD
5f4fe6b10 22 2C 22 61 70 70 56 65-72 73 69 6F 6E 43 6F 64 ","appVersionCod
5f4fe6b20 65 22 3A 22 33 37 36 22-2C 22 61 70 70 56 65 72 e":"376","appVer
5f4fe6b30 73 69 6F 6E 4E 61 6D 65-22 3A 22 35 2E 31 34 2E sionName":"5.14.
5f4fe6b40 39 22 2C 22 74 69 6D 65-73 74 61 6D 70 22 3A 31 9","timestamp":1
5f4fe6b50 35 30 39 36 30 34 30 34-30 35 39 39 2C 22 74 79 509604040599,"ty
5f4fe6b60 70 65 22 3A 22 50 52 45-44 45 46 49 4E 45 44 22 pe":"PREDEFINED"
5f4fe6b70 2C 22 63 75 73 74 6F 6D-41 74 74 72 69 62 75 74 ,"customAttribut
5f4fe6b80 65 73 22 3A 7B 22 66 6F-72 75 6D 22 3A 22 70 68 es":{"forum":"ph
5f4fe6b90 6F 74 6F 67 72 61 70 68-79 22 7D 2C 22 70 72 65 otography"},"pre
5f4fe6ba0 64 65 66 69 6E 65 64 54-79 70 65 22 3A 22 63 6F definedType":"co
5f4fe6bb0 6E 74 65 6E 74 56 69 65-77 22 2C 22 70 72 65 64 ntentView","pred
5f4fe6bc0 65 66 69 6E 65 64 41 74-74 72 69 62 75 74 65 73 efinedAttributes
5f4fe6bd0 22 3A 7B 22 63 6F 6E 74-65 6E 74 49 64 22 3A 22 ":{"contentId":"
5f4fe6be0 32 32 37 36 30 33 37 34-39 22 2C 22 63 6F 6E 74 227603749","cont
5f4fe6bf0 65 6E 74 54 79 70 65 22-3A 22 70 6F 73 74 22 2C entType":"post",
5f4fe6c00 22 63 6F 6E 74 65 6E 74-4E 61 6D 65 22 3A 22 23 "contentName":"#
5f4fe6c10 54 68 65 20 73 65 61 20-6F 66 20 63 6C 6F 75 64 The sea of cloud
5f4fe6c20 20 6F 6E 20 48 65 68 75-61 6E 20 4D 6F 75 6E 74  on Hehuan Mount
5f4fe6c30 61 69 6E 22 7D 7D 00 00-02 4C 7B 22 61 70 70 42 ain"}}···L{"appB
5f4fe6c40 75 6E 64 6C 65 49 64 22-3A 22 63 6F 6D 2E 73 70 undleId":"com.sp
```

Fig–9: The evidence of writing a post



```
5ece44a40 38 35 39 81 0B 81 09 09-00 31 05 81 0B 55 0D 1B 859······1···U··
5ece44a50 70 6F 73 74 43 6F 6D 6D-65 6E 74 43 72 65 61 74 postCommentCreat
5ece44a60 65 64 01 5F 77 68 B6 3A-7B 22 70 6F 73 69 74 69 ed·_wh¶:{"positi
5ece44a70 76 65 22 3A 74 72 75 65-2C 22 64 72 61 66 74 22 ve":true,"draft"
5ece44a80 3A 66 61 6C 73 65 2C 22-70 6F 73 74 49 64 22 3A :false,"postId":
5ece44a90 32 32 37 36 30 33 38 31-31 2C 22 65 64 69 74 22 227603811,"edit"
5ece44aa0 3A 66 61 6C 73 65 7D 38-66 61 33 30 64 34 64 2D :false}8fa30d4d-
5ece44ab0 66 66 66 39 2D 34 61 35-35 2D 61 36 64 64 2D 38 fff9-4a55-a6dd-8
5ece44ac0 34 63 37 30 31 35 37 62-39 33 35 32 32 37 34 38 4c70157b93522748
5ece44ad0 35 39 70 81 0A 08 00 23-05 65 55 0D 1B 70 6F 73 59p····#·eU··pos
5ece44ae0 74 43 72 65 61 74 65 64-01 5F 77 69 F7 DF 7B 22 tCreated·_wi÷ß{"
5ece44af0 70 6F 73 69 74 69 76 65-22 3A 66 61 6C 73 65 2C positive":false,
5ece44b00 22 64 72 61 66 74 22 3A-74 72 75 65 2C 22 65 64 "draft":true,"ed
5ece44b10 69 74 22 3A 66 61 6C 73-65 7D 38 66 61 33 30 64 it":false}8fa30d
5ece44b20 34 64 2D 66 66 66 39 2D-34 61 35 35 2D 61 36 64 4d-fff9-4a55-a6d
5ece44b30 64 2D 38 34 63 37 30 31-35 37 62 39 33 35 32 32 d-84c70157b93522
5ece44b40 37 34 38 35 39 81 0D 81-0B 09 00 29 05 81 17 55 74859······)···U
5ece44b50 0D 1B 70 6F 73 74 53 75-62 73 63 72 69 62 65 64 ··postSubscribed
```

Fig–10: The evidence of making comment



```
5ece44bd0 32 37 34 38 35 39 66 81-0C 08 00 21 05 53 55 0D 274859f····!·SU·
5ece44be0 1B 70 6F 73 74 56 69 65-77 65 64 01 5F 77 6A 46 ·postViewed·_wjF
5ece44bf0 22 7B 22 70 6F 73 74 49-64 22 3A 32 32 37 36 30 "{"postId":22760
5ece44c00 33 38 31 31 2C 22 70 65-72 63 65 6E 74 61 67 65 3811,"percentage
5ece44c10 22 3A 30 7D 38 66 61 33-30 64 34 64 2D 66 66 66 ":0}8fa30d4d-fff
5ece44c20 39 2D 34 61 35 35 2D 61-36 64 64 2D 38 34 63 37 9-4a55-a6dd-84c7
5ece44c30 30 31 35 37 62 39 33 35-32 32 37 34 38 35 39 5A 0157b9352274859Z
5ece44c40 81 0D 08 00 2F 05 2D 55-0D 1B 6D 65 73 73 61 67 ····/·-U··messag
```

Fig‒11: The evidence of browsing posts

RESEARCH FINDINGS

After we analyze our experiment results, we draw a table to clearly describe our findings in the physical memory and the source of digital evidence stored in the local database. As list in the Table 2, most of the evidence we can find in the physical memory. However, we can't find out the password in the physical memory. Because the password is confidential information, we guess that it must be hided or encrypted. In the Table 3, we can realize that most of information is stored in the same database which is called "Bilanx.db'. In this database, we can find out many kinds of information as our significant evidence to prove a cybercrime on the court. However, there is still some information we can't find in the local database. We infer that they are stored in the server side. Nevertheless, most of the evidence can be found in the local database. This advantage would facilitate investigator to investigate a crime.

Table‒2: Findings in the physical memory

Evidence	**ASUS Zenfone 2**
Account	Found
Password	None
Posting evidence	Found
Posting timestamp	Found
Posting title	Found
Making comment evidence	Found
Browsing evidence	Found
Clicking "Like" button evidence	Found
Clicking "Collect" button evidence	Found
Chatting records	Found
Chatting contents	None
Friend list	None
Friend request	None

Table‒3: Source of digital evidence in the local database

Evidence	**Path**	**File**
Account	None	None
password	None	None
Posting evidence	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Making comment evidence	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Browsing evidence	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Click "Like" button evidence	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Click "Collect" button evidence	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Chatting records	/data/data/com.sparkslab.dcardreader/databases	Bilanx.db
Chatting contents	None	None
Friend list	None	None
Friend request	None	None

CONCLUSIONS

Nowadays, due to the rapid development of new technologies, thousands of new social networking sites have sprung up over the past few years, such as Facebook, Twitter, Instagram, Dcard and so forth. Simultaneously, various kinds of cybercrime also emerge endlessly in recent years. For the sake of assisting investigators to investigate cybercrimes, this paper proposes a forensic way to investigate a perpetrator that commits a crime via Dcard application on the mobile phone. Therefore, our main goal is to extract digital evidences that perpetrators left on the mobile devices.

We did a series of normal behaviors that users may operate it on the mobile phone. After we complete these behaviors, we make use of "dd" instruction tool to create image file for the physical memory on the mobile phone. Thereafter, we adopt FTK Imager to analyze the image file on the computer. On the other hand, we also make use of SQLite Editor to inspect the database. In our experiment, we can find many kinds of evidences on the mobile phone, for example, writing a post, making comments, browsing evidence, chatting records, clicking "Like" or "Collect" button on the postings and so forth. All of the findings can be used for the crime investigation. The investigators can organize and analyze daily behaviors or preference of a perpetrator based on these significant information. Furthermore, if the crime happened, all of the evidences extracted and analyzed by the investigator can be a crucial admission on the court.

REFERENCES

1. Benkhelifa E, Thomas BE, Jararweh Y. Framework for Mobile Devices Analysis. Procedia Computer Science. 2016 Dec 31;83:1188-93.
2. Norouzizadeh Dezfouli F, Dehghantanha A, Eterovic-Soric B, Choo KK. Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms. Australian journal of forensic sciences. 2016 Jul 3;48(4):469-88.
3. Abhyankar A. Social networking sites. SAMVAD. 2011;2:18-21.
4. Dwyer C, Hiltz S, Passerini K. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. AMCIS 2007 proceedings. 2007 Dec 31:339.
5. Jennifer Golbeck. Introduction to Social Media Investigation, ELSEVIER, AMSTERDAM, Netherlands, 2015, pp. 1-15.
6. Ellison NB. Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication. 2007 Oct 1;13(1):210-30.
7. STATISTA, Most famous social networking sites worldwide as of September. 2017.
8. Kshetri N, Kshetri N. Cybersecurity in India: Regulations, governance, institutional capacity and market mechanisms. Asian Research Policy. 2017;8(1):64-76.
9. Marcum CD, Higgins GE, editors. Social Networking as a Criminal Enterprise. CRC Press; 2014 Apr 28.
10. Wiki, Dcard, 2016.08, https://zh.wikipedia.org/wiki/Dcard, Access date: Jul. 22, 2017
11. Alexa, http://www.alexa.com/siteinfo/dcard.tw, Access date: Jul. 22, 2017
12. Azfar A, Choo KK, Liu L. An android social app forensics adversary model. InSystem Sciences (HICSS), 2016 49th Hawaii International Conference on 2016 Jan 5 (pp. 5597-5606). IEEE.
13. Thakur NS. Forensic analysis of WhatsApp on Android smartphones.2013.
14. Al Mutawa N, Baggili I, Marrington A. Forensic analysis of social networking applications on mobile devices. Digital Investigation. 2012 Aug 31;9:S24-33.
15. Wiki, Rooting(Android), https://en.wikipedia.org/wiki/Rooting_(Android), Access date: Oct. 21, 2017
16. Mahajan A, Dahiya MS, Sanghvi HP. Forensic analysis of instant messenger applications on android devices. arXiv preprint arXiv:1304.4915. 2013 Apr 17.
17. Faheem M, Le-Khac NA, Kechadi T. Smartphone Forensic Analysis: A Case Study for Obtaining Root Access of an Android Samsung S3 Device and Analyse the Image without an Expensive Commercial Tool.
18. Arthur KK, Venter HS. An Investigation Into Computer Forensic Tools. InISSA 2004 Jun (pp. 1-11).