Australian Journal of Forensic Sciences
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tajf20

Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms

Farhood Norouzizadeh Dezfouli[a], Ali Dehghantanha[b], Brett Eterovic-Soric[c] & Kim-Kwang Raymond Choo[c]

[a] Faculty of Computer Science and IT, University Putra Malaysia, Kuala Lumpur, Malaysia

[b] School of Computing, Science and Engineering, University of Salford, Manchester, United Kingdom

[c] Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Mawson Lakes Campus, Mawson Lakes, South Australia, Australia

Published online: 07 Aug 2015.

To cite this article: Farhood Norouzizadeh Dezfouli, Ali Dehghantanha, Brett Eterovic-Soric & Kim-Kwang Raymond Choo (2015): Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms, Australian Journal of Forensic Sciences, DOI: 10.1080/00450618.2015.1066854

To link to this article: http://dx.doi.org/10.1080/00450618.2015.1066854

Australian Journal of Forensic Sciences, 2015
http://dx.doi.org/10.1080/00450618.2015.1066854

Taylor & Francis
Taylor & Francis Group

Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms

Farhood Norouzizadeh Dezfouli[a], Ali Dehghantanha[b]*, Brett Eterovic-Soric[c] and Kim-Kwang Raymond Choo[c]

[a]*Faculty of Computer Science and IT, University Putra Malaysia, Kuala Lumpur, Malaysia;* [b]*School of Computing, Science and Engineering, University of Salford, Manchester, United Kingdom;* [c]*Information Assurance Research Group, Advanced Computing Research Centre, University of South Australia, Mawson Lakes Campus, Mawson Lakes, South Australia, Australia*

(*Received 21 April 2015; accepted 16 June 2015*)

The rapid growth in usage and application of Social Networking (SN) platforms make them a potential target by cyber criminals to conduct malicious activities such as identity theft, piracy, illegal trading, sexual harassment, cyber stalking and cyber terrorism. Many SN platforms are extending their services to mobile platforms, making them an important source of evidence in cyber investigation cases. Therefore, understanding the types of potential evidence of users' SN activities available on mobile devices is crucial to forensic investigation and research. In this paper, we examine four popular SN applications: Facebook, Twitter, LinkedIn and Google+, on Android and iOS platforms, to detect remnants of users' activities that are of forensic interest. We detect a variety of artefacts (e.g. usernames, passwords, login information, personal information, uploaded posts, exchanged messages and uploaded comments from SN applications) that could facilitate a criminal investigation.

Keywords: digital forensics; social networking application forensics; mobile app forensics; mobile device forensics; Android investigation; iOS investigation; Facebook forensics; Twitter forensics; LinkedIn forensics; Google+ forensics

1. Introduction

Social network applications allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system[1]. Currently, Facebook is the most popular SN platform with around 1.06 billion active users[2] followed by Twitter, LinkedIn and Google+. These platforms provide users with features such as email, blogging, instant messaging and photo sharing[3]. Various types of personal information are shared on SN platforms such as name, date of birth, photos, email addresses, phone numbers and group affiliations, which could provide crucial information during a crime investigation[4]. SN sites provide customised features and interactions according to each user's personal preferences, social graph, and the digital mapping of people's real-world social connections.

*Corresponding author. Email: A.Dehghantanha@salford.ac.uk

Smartphones, an increasingly popular way of accessing SN services on the go, store a wealth of personal information. For example, as many as 40% of SN users reportedly access SN applications through their mobile devices[5], and so are cyber criminals[6]. Not surprisingly, smartphones have been widely targeted by cybercriminals such as malware designed to steal users' private data[7–9]. Rapid changes in mobile device technologies and the wide range of mobile operating platforms and devices compound the challenges in a forensic investigation[10]. At the time of research, Android with 51.5% market share and Apple's iOS with 42.4% market share are the most popular mobile platforms[11]. Therefore, it is important for law enforcement and the forensic communities to keep pace with technologies[12–14].

In addition to a criminal investigation, forensic investigation of SN applications may also be required in a civil litigation or a corporate disciplinary investigation of employees suspected of breaching company policies[15,16]. However, there do not appear to be any commonly used guidelines specifically related to investigation of SN mobile applications[6,17].

1.1. Related Work

There are three methods for acquiring data from a smartphone: manual acquisition, logical acquisition and physical acquisition[18]. During manual acquisition, data are retrieved by copying a log of the actions taken on the device[19] and copying existing application data[20]. Casey[21] describes logical acquisition as a bitwise copy of files and directories that contains timestamp and location of resident files within the filesystem of the device. Logical acquisition is classified into the following categories: copying files and folders, partition imaging, content provider and recovery mode data collection[22–24]. In the physical acquisition, a bitwise copy of the internal memory of the device would be acquired, which normally includes deleted data.

Physical acquisition of Android devices is generally performed using dd, JTAG, physical extraction and DMD. Logical acquisition of Android devices can be conducted by either using mobile forensic tools such as Cellebrite UFED or by booting into recovery mode. Lessard and Kessler[25] performed physical acquisition by gaining super-ruser access and using dd on an ADB shell to retrieve images of each mtd file present in the '/dev/mtd' directory, and logical acquisition using Cellebrite UFED to extract data from the '/data/data' directory of a HTC Hero running Android 1.5. Vidas *et al.*[24] used the booting into recovery mode method to perform logical acquisition. JTAG and physical extraction methods are useful when performing logical acquisition is not possible, while booting into and collecting data in the recovery mode can be used when the device is powered off and the analyst wishes to avoid powering on the device. Sylve *et al.*[26] used the DMD method to perform physical acquisition of a rooted HTC Android device. Son *et al.*[23] utilised JTAG and recovery mode data collection methods to perform physical and logical acquisition of rooted Samsung devices running Android platforms, and concluded that both methods to acquire similar data. Do *et al.*[27] proposed an adversary model for Android covert data exfiltration which was then used to build a mobile data exfiltration technique (MDET). In this model, common communication mediums found on almost all mobile devices are used to obtain sensitive data from Android devices in a covert manner. The authors also developed two proof-of-concepts that use SMS and audio to exfiltrate data from Android devices.

With older iOS devices, physical acquisition can be conducted using the dd method, and logical acquisition is performed using Apple's iTunes. Zdziarski[28] acquired a

physical image of an iPhone 3G by booting it with a recovery toolkit called iLibrary+version 1.6, which enables the acquisition of a bitwise copy of the memory image using the dd command. Hoog and Gaffaney[29] performed physical and logical acquisitions on an iPhone 3GS (2.2 firmware) using iTunes backup and Zdziarski's physical dd method and concluded that Zdziarski's method is more accurate in terms of ease of installation and integrity of acquired data. Morrissey and Campbell[30] also used the iTunes backup to perform logical acquisition on iPhone 2G and 3GS and recovered artefacts of call logs, address book, cookies, logs, geolocation data, web browsing, SMS/MMS, multimedia and application data. Bader and Baggili[31] used the iTunes backup to perform a logical acquisition on an iPhone 3GS and classified the collected files into three categories: pList files, which store data in plaintext format; mddata files, which store data in a raw binary format; and mdinfo files, which store encoded metadata of the corresponding mddata files. Bader and Baggili[31] claimed that evidential data from iPhones were mainly obtained using logical acquisition as they are more compliant to forensic standards (NIST, 2014). However, they also noted several limitations in using logical acquisition such as limitation of interaction with iTunes versions prior to 8.2 for iPhone versions 3GS or higher.

Proffitt[32] stated that frequent changes in security mechanisms employed by iOS devices can make the task of acquiring a forensically sound image more challenging. For instance, the employed security methods on iOS version 3 and iOS version 6 are so different that methods used to acquire an image from iOS 3 do not work on iOS 6. For example in another related work, Al Mutawa *et al.*[33] extracted a database file from the Facebook application on an iOS 4 device, which was not found on iOS 6 devices[34]. Ariffin *et al.*[35] also explained the complexity of retrieving unallocated and encrypted data from iOS devices and presented an acquisition method for recovering deleted image files from the iOS journaling system in iPhone 3GS and 4, which may not work for newer iOS devices.

Jung *et al.*[36] investigated Cyworld, Me2Day, Yozm, NateOn UC, KakaoTalk, MyPeople, Twitter and Facebook applications for iOS version 3.x and 4.x, and using the logical acquisition technique, they recovered multimedia images, user-driven social media attributes and geolocation data. In another work, Said *et al.*[37] investigated Facebook and Twitter apps on iPhone 3GS, BlackBerry Bold and Samsung Omnia II and extracted friends lists and birthday notifications. Tso *et al.*[38] analysed Facebook, WhatsApp Messenger, Skype, Windows Live Messenger and Viber applications on an iPhone 4 running iOS version 4.3.5 and extracted users' personal data, messages, contact lists and posts from backup folders. Al Mutawa *et al.*[33] investigated Facebook, Twitter and MySpace applications on an iPhone 4, Blackberry Torch 9800 and Samsung GT-i9000 Galaxy S and extracted users' personal data, uploaded photos and posted comments. Quick and Choo[39–41] recovered users' personal data, login information, downloaded files, uploaded files and deleted files of Dropbox, SkyDrive and GoogleDrive cloud client applications from an iPhone 3G running iOS 4.2.1. Anglano[42] extracted a variety of artefacts such as the contact list, exchanged messages, deleted contacts, and deleted messages of the WhatsApp instant messaging application from a virtualised Android device on YouWave.

To the best of our knowledge, there are very limited works on the investigation of SN platforms on mobile devices. To contribute to this framework, we also adopt a framework based on Refs.[43] and[44] to guide our investigation of the SN applications, namely Facebook, Twitter, LinkedIn and Google+ applications on both Android and iOS platforms.

1.2. Research questions and contributions

To address the aforementioned gap, the following research questions will be answered in this paper.

(1) What artefacts of forensic value remain on Android internal memory and internal storage from using the Facebook, Twitter, LinkedIn and Google+ applications?
(2) What artefacts of forensic value reside on iOS internal storage from using the Facebook, Twitter, LinkedIn and Google+ applications?
(3) What network traffic artefacts can be detected when users are using the Facebook, Twitter, LinkedIn and Google+ applications on Android and iOS platforms?

The main contribution of this research is a systematic way of recovering artefacts of users' activities within SN applications and examining their forensic values. This paper may serve as a guideline for forensic practitioners during investigation of SN platforms including Facebook, Twitter, LinkedIn and Google+, on both Android and iOS platforms.

1.3. Investigation framework

Using ad hoc processes and tools to extract digital evidence can jeopardise the integrity and credibility of evidence[6]. Therefore, we adopt an investigation framework in this research (similar to the approach undertaken by Martini and Choo[45,46], Martini *et al.*[13,47] and Quick and Choo[39–41].

McKemmish[43] described forensic investigation in four stages: identification, preservation, analysis and presentation of digital evidence, which is similar to the four-stage NIST guidelines for mobile device forensics[48] and the four-stage cloud forensic framework[44]. In this paper, we adapt these frameworks to guide our mobile device investigation. The integrated investigation framework comprises the following.

(1) *Identification and collection.* Evidence was collected from the internal storage and internal memory of the mobile device and the network traffic was monitored. In our context, a Samsung Galaxy Tab II running Android version 4.2 and the internal storage of an Apple iPhone 5s running iOS version 7.1.2A bit-for-bit image of the internal memory and the internal storage of the Android device was acquired. On the iOS device, a backup of the internal storage using Apple's iTunes version 11.3.1 was acquired. Generated network traffic of user SN activities with both smartphones was captured using Wireshark version 1.4.4.
(2) *Preservation.* MD5 and SHA1 hash values of each acquired file were calculated and, subsequently, verified so that we can detect any modification. In our investigation, the comparisons of hash values and timestamp metadata of files uploaded and then downloaded through Facebook, Twitter, LinkedIn and Google+ applications on both Android and iOS platforms revealed that the process of downloading files modifies file content and timestamps, as shown in Table 1. As expected, created, modified and last accessed timestamps of the downloaded file are matched to the timestamps of the destination folder on the smartphone following iOS and Android file system rules. Furthermore, since SN platforms are utilising resizing and compression algorithms during uploading files, the hash value of downloaded files differs from that of the original files.

Table 1. Summary of metadata preservation on Android and iOS platforms.

	Android				iOS			
	Original file hash value	Original file timestamps	Downloaded File Hash Value	Downloaded file Timestamps	Original file hash value	Original file timestamps	Downloaded file hash value	Downloaded file timestamps
Facebook	MD5: 9547976B18E8EA 972003ADC4228A2D81	Created: 29/01/2014 12:31:49 AM	MD5: 0AB43E6B606 AB074E703B39EF70E3F51	Created: 11/11/2014 3:59:18 PM	MD5: FB40E2AA63F65F27 E135C70B70AF6C29	Created: 8/02/2014 10:51:32 AM	MD5: 5F4EA700B1DA 7301B3D5EED626B3DEC0	Created: 12/11/2014 3:34:51 PM
		Modified: 29/01/2014 12:31:50AM		Modified: 11/11/2014 3:59:18 PM		Modified: 8/02/2014 10:51:33 AM		Modified: 12/11/2014 3:34:51 PM
	SHA1: 9815DEAFE8316487 2C3767A26B77B724F5CE1095	Accessed: 29/01/2014 12:31:46 AM	SHA1: D45AB3E30E6D7D67 AE3B28AADD8674CD50D8CCEF	Accessed: 11/11/2014 3:59:18 PM	SHA1: EE6E0FFD128B854EB4ACBFA 8C8719D56D1A3433F	Accessed: 8/02/2014 10:51:29 AM	SHA1: 055FCB9FCB7610868ABC 7FB73DDB39C0C6ED7AE5	Accessed: 12/11/2014 3:34:51 PM
Twitter	MD5: 11BEC50C0CA473 DB6C58B947B57EE71A	Created: 3/02/2014 5:48:13 PM	MD5: 508A414DC171493 F60FC581D736226AD	Created: 11/11/2014 6:12:43 PM	MD5: 8CB21B388DBAC98 DA8B59FD72E492F6A	Created: 10/02/2014 2:34:48 PM	MD5: 804D3B7E7A11B2 E0D0E0F80D3897EF53	Created: 12/11/2014 4:28:14 PM
		Modified: 3/02/2014 5:48:14 PM		Modified: 11/11/2014 6:12:43 PM		Modified: 10/02/2014 2:34:49 PM		Modified: 12/11/2014 4:28:14 PM
	SHA1: 067D663D908EDC0597 ACA87AB2E7B1CFD8A5F46D	Accessed: 3/02/2014 5:48:10 PM	SHA1: E81E59F24C05C772894C40 EBB53D5F5C75140B11	Accessed: 11/11/2014 6:12:43 PM	SHA1: 24A301559AAFCBC823CFB 64D5F765F2B3F8512C8	Accessed: 10/02/2014 2:34:45 PM	SHA1: 0A9E1F93E351C79A0849A 444ADD347C33C189536	Accessed: 12/11/2014 4:28:14 PM
LinkedIn	MD5: BB1E236CF9358 15E5BB714760B23F549	Created: 5/02/2014 2:24:18 PM	MD5: 25655641659CA4 579FEBEC7844AC90F7	Created: 11/11/2014 10:34:26 PM	MD5: FAA477F70EB298 7DFECE93E4C5F2DA8B	Created: 12/02/2014 8:43:33 PM	MD5: BA92ADBF5FDE 8418BB58E3B476EFB287	Created: 12/11/2014 6:16:42 PM
		Modified: 5/02/2014 2:24:19 PM		Modified: 11/11/2014 10:34:26 PM		Modified: 12/02/2014 8:43:34 PM		Modified: 12/11/2014 6:16:42 PM
	SHA1: 395E50118BC92973 04970956C2241059EABE7F8D	Accessed: 5/02/2014 2:24:15 PM	SHA1: A4AB5E45CDBA4678B 23FF46A76C14A5B8C5DA4A0	Accessed: 11/11/2014 10:34:26 PM	SHA1: 9F5D00F5869D52269 ACAD04B1818F9DC34A2DAE2	Accessed: 12/02/2014 8:43:30 PM	SHA1: 55A4B2418EA9856E 73FE23F20A18B01AC28E9A4B	Accessed: 12/11/2014 6:16:42 PM
Google+	MD5: 47BAE32EB601F29 642F07D3C69EC3868	Created: 7/02/2014 10:51:41 AM	MD5: 5DDDD74840793 DB1E992B6FF75F3344E	Created: 12/11/2014 1:18:23 PM	MD5: D9CAA3FF494B49228 EA6E3E5D4CAB06B	Created: 14/02/2014 11:22:46 AM	MD5: 59DA5CB3FF501 63C1491D2CA39FB5F9D	Created: 12/11/2014 10:38:29 PM
		Modified: 7/02/2014 10:51:42 AM		Modified: 12/11/2014 1:18:23 PM		Modified: 14/02/2014 11:22:47 AM		Modified: 12/11/2014 10:38:29 PM
	SHA1: 50FDD9462ECAC5D34B3 F3D4B213733E9DC28D97B	Accessed: 7/02/2014 10:51:38 AM	SHA1: A5B1B059A48633D98 BF80CA434D08A8F1FE9DB77	Accessed: 12/11/2014 1:18:23 PM	SHA1: 4577622CBD434BC 3D145F1EB65456D13445BCBF9	Accessed: 14/02/2014 11:22:43 AM	SHA1: E17FAE72EF79756 CBC58C37C9B1645775F7E0D1F	Accessed: 12/11/2014 10:38:29 PM

(3) *Examination and analysis.* Collected data from internal memory and internal storage of the devices was examined to determine possible data remnants of using the Facebook, Twitter, LinkedIn and Google+ applications on Android (refer to Section 3) and iOS (refer to Section 4), and to answer the first and the second question of this research respectively. Lastly, analysis of network capture files (refer to Section 5) will answer the third research question.
(4) *Presentation.* A summary of findings and their forensic values are presented at this stage, as explained in Section 6 of this paper.

1.4. Outline

This paper is organised as follows: Section 2 describes the process of experiment setup and lists the tools and equipment used during this research. Sections 3 and 4 outline the analysis of the mentioned SN applications on Android and iOS devices respectively. Section 5 contains the examination of network traffic on both Android and iOS platforms. Finally, we conclude the paper with a summary of the results.

2. Experimental setup

A Samsung Galaxy Tab II running on Android 4.2 and an iPhone 5s running on iOS 7.1.2 were examined in this research. Both devices were wiped and restored to the default factory settings. The Facebook, Twitter, LinkedIn and Google+ applications were downloaded and installed through Play Store version 5.2.13 for Android and the built-in App Store on iOS. Two accounts were created ('bt.enron@gmail.com' on Android and 'bs.enron@gmail.com' on iOS) to conduct a total of five experiments on each platform to analyse Facebook, Twitter, LinkedIn and Google+. Each experiment includes five activities to mimic user attempts to login to the app, modify personal information, upload posts, exchange messages and upload comments. While these actions were being undertaken, Wireshark was running to capture the network traffic. The EDRM Enron Email data-set[49] was used to facilitate the research activities.

On the Samsung Galaxy Tab II, upon completion of each activity, a bit-for-bit image of the internal memory and internal storage was acquired using the *dd* command in Kali Linux version 1.1.0. Similarly, on iPhone 5s, a logical backup of internal storage was acquired upon completion of each activity, using Apple's iTunes version 11.3.1. Afterwards, MD5 and SHA1 hash values of the collected data were calculated using the md5sum and sha1sum commands built in to Kali Linux. The list of devices, applications and tools that are used in this research are provided in Table 2. AccessData FTK version 3.1.2 and HxD Hex Editor version 1.7.7.0 were used to analyse bit-for-bit images and iBackupBot version 5.1.9, Plist Editor for Windows version 1.0.1 and SQLite Database Browser version 1.3 were utilised to analyse the iOS backup files. Moreover, DCode software version 4.02 was used to decode the iOS backup files' timestamps.

A total of 20 physical images of Android internal memory, 20 physical images of Android internal storage, and 20 network captures of Android activities were created. Similarly, a total of 20 backups of iOS internal storage, and 20 network captures of iOS activities were created. The activities are named as *Type of Device (A for Android and i for iOS),* followed by *Type of SN platform (F for Facebook, T for Twitter, L for*

Table 2. Devices, applications and tools used.

Devices, applications and tools	Specification
iPhone 5s	iOS version 7.1.2
Samsung Galaxy Tab II	Android version 4.2
Facebook Application	version 4.3 for iOS, version 18.0.0 for Android
Twitter Application	version 3.2.2 for iOS, version 5.30.1 for Android
LinkedIn Application	version 1.1 for iOS, version 3.4.2 for Android
Google+ Application	version 3.1.1 for iOS, version 4.6.0 for Android
AccessData FTK	version 3.1.2
HxD Hex Editor	version 1.7.7.0
iBackupBot	version 5.1.9
Plist Editor for Windows	version 1.0.1
SQLite Database Browser	version 1.3
iTunes	version 11.3.1
Wireshark	version 1.10.1
DCode	version 4.02

LinkedIn and G for Google+) followed by *a sequential number*. For example, AF1 represents the first Facebook experiment on the Android platform. Tables 3 and 4 show details of experiments conducted on the Android and iOS platforms.

3. Findings of the Android investigations

This section explains the analysis of the internal memory and internal storage of a Samsung Galaxy Tab II to determine data remnants after performing login, modifying personal information, uploading posts, exchanging messages and uploading comments activities in each of the Facebook, Twitter, LinkedIn and Google+ applications. In order to perform the analysis of internal memory and internal storage, each of the images were opened in AccessData FTK and HxD Hex Editor. The hex editor functionality allowed us to view the acquired images. The analysis was performed manually on each file to identify the file types, determine the headers and signatures of the files, search for related SN data, and determine the location of stored artefacts. Several keywords, such as usernames, *uid*, name and email address, were used to facilitate the searching of each file in order to detect SN artefacts. Further explanation on the process of analysis and the findings of the examination are provided in the following subsections.

3.1. Login information

Logging in to the Facebook app would leave the username (in this case bt.enron@gmail.com) in internal memory which can be detected by searching for 'www.facebook.com' as shown in Figure 1. Utilising this username and further searching the internal memory led to the identification of assigned Facebook *UID* and the timestamp of the last login to the application.

Similar to Facebook, searching the internal memory with 'www.twitter.com', 'www.linkedin.com' and 'www.plus.google.com' keywords revealed the username, *UID*, and last successful login time of the respective application. However, no trace of the user password could be detected in the internal memory, for any of these platforms.

Table 3. User details and sample data for Android experiments.

SN applications on the Android device	Performed activities on the SN applications
Facebook (AF1-AF5)	AF1. Login username: bt.enron@gmail.com, password: hockey1* AF2. Modify personal info (Name: Barry Tycholiz, Friend: Hrap Gerry) AF3. Upload post (Content of Email 'End of Year Party') AF4. Exchange messages (Content of Email 'Hockey on Thurs') AF5. Upload comment (Content of Email 'Dinner Invitation')
Twitter (AT1-AT5)	AT1. Login username: bt.enron@gmail.com, password: hockey1* AT2. Modify personal info (Name: Barry Tycholiz, Friend: Hrap Gerry) AT3. Upload post (Content of Email 'Hockey on Thurs') AT4. Exchange messages (Content of Email 'Kern River 637') AT5. Upload comment (Content of Email 'Trip to Russia')
LinkedIn (AL1-AL5)	AL1. Login username: bt.enron@gmail.com, password: hockey1* AL2. Modify personal info (Name: Barry Tycholiz, Friend: Hrap Gerry) AL3. Upload post (Content of Email 'Hockey and Lunch') AL4. Exchange messages (Content of Email 'Kern River 637') AL5. Upload comment (Content of Email 'Hockey and Lunch')
Google+ (AG1-AG5)	AG1. Login username: bt.enron@gmail.com, password: hockey1* AG2. Modify personal info (Name: Barry Tycholiz, Friend: Hrap Gerry) AG3. Upload post (Content of Email 'Trip to Russia') AG4. Exchange messages (Content of Email 'Hockey on Thurs') AG5. Upload comment (Content of Email 'Trip to Russia')

3.2. *User profile information*

When analysing the Facebook app, investigation of the internal storage was undertaken using the previously identified username as a keyword. We were able to identify date of birth, location, work and education background, email addresses, phone numbers and interests of the last logged in user, as shown in Figure 2. Moreover, the user's friends list, which includes the names and email addresses of the user's friends accompanied by URLs of their profile pages, are detectable as shown in Figure 3.

Investigation of users' profile information in the Twitter, LinkedIn and Google+ apps with the usernames identified in the previous section for each application, also resulted in discovery of the user's name, date of birth, biography, location, email

Table 4. User details and sample data for iOS experiments.

SN applications on the iOS device	Performed activities on the SN applications
Facebook (iF1-iF5)	IF1. Login username: bs.enron@gmail.com, password: hockey1* IF2. Modify personal info (Name: Brian Stone, Friend: Jerry Strange) IF3. Upload post (Content of Email 'Great trip') IF4. Exchange messages (Content of Email 'Hockey on Thurs') IF5. Upload comment (Content of Email 'Dinner Invitation')
Twitter (iT1-iT5)	IT1. Login username: bs.enron@gmail.com, password: hockey1* IT2. Modify personal info (Name: Brian Stone, Friend: Jerry Strange) IT3. Upload post (Content of Email 'Trip to Russia') IT4. Exchange messages (Content of Email 'Kern River 637') IT5. Upload comment (Content of Email 'Trip to Russia')
LinkedIn (iL1-iL5)	IL1. Login username: bs.enron@gmail.com, password: hockey1* IL2. Modify personal info (Name: Brian Stone, Friend: Jerry Strange) IL3. Upload post (Content of Email 'Hockey and Lunch') IL4. Exchange messages (Content of Email 'Kern River 637') IL5. Upload comment (Content of Email 'Hockey and Lunch')
Google+ (iG1-iG5)	IG1. Login username: bs.enron@gmail.com, password: hockey1* IG2. Modify personal info (Name: Brian Stone, Friend: Jerry Strange) IG3. Upload post (Content of Email 'Trip to Russia') IG4. Exchange messages (Content of Email 'Hockey on Thurs') IG5. Upload comment (Content of Email 'Trip to Russia')

address, work background, connections, friends list and URLs to each friend's profile picture for respective applications in the internal storage.

3.3. *Uploading posts*

The investigation revealed that posts uploaded using the Facebook application, and their corresponding timestamps, are detectable in the internal storage in plain text and can be recovered by searching for the user's *UID*, username or registered name (in this case bt.enron@gmail.com and Barry Tycholiz), as shown in Figure 4.

UID, username and registered name were also used in investigations of Twitter, LinkedIn and Google+ to detect uploaded posts, sent Tweets and corresponding timestamps within the internal storage.

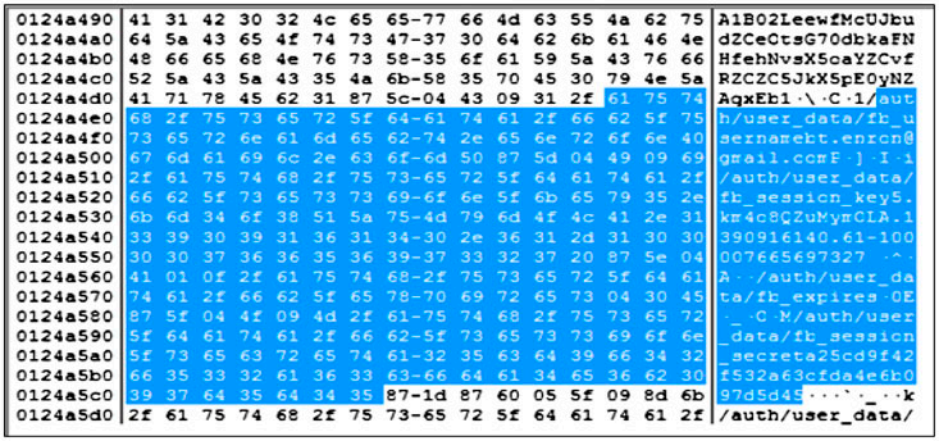

Figure 1. Facebook login artefact in Android.

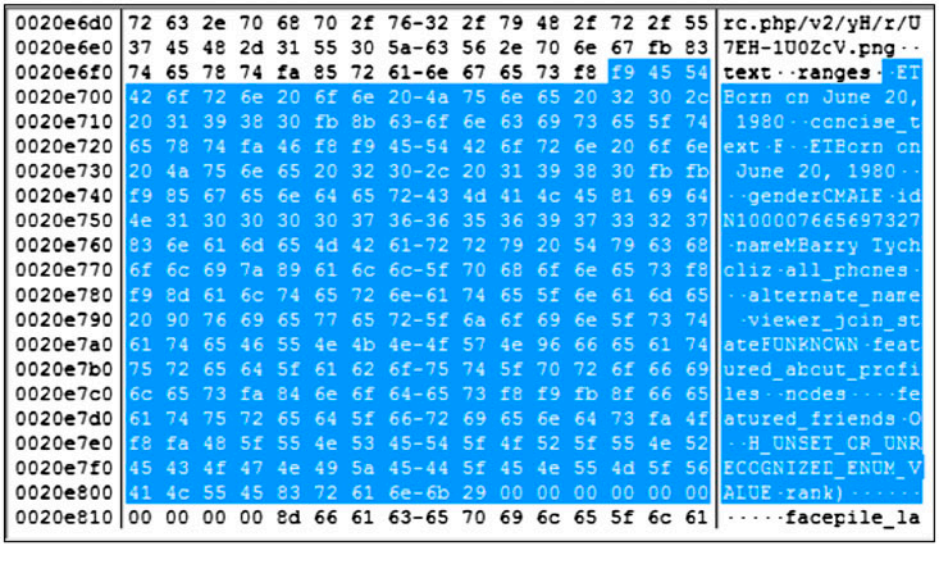

Figure 2. Facebook user's personal information artefact in Android.

3.4. *Messaging*

As shown in Figure 5, examination of the Facebook application revealed instant messages sent and received by the user, *UID* and name of the party who has received the message, along with timestamp metadata. These artefacts remain in the internal storage in plain text, and can be retrieved from the internal storage using *UID*, name, or username as search keywords.

When analysing Twitter, LinkedIn and Google+ apps, searching the internal storage with the user's *UID*, name, or username would result in locating the instant messages sent and received by the user, *UID* and name of the party who has received the message, along with timestamp metadata.

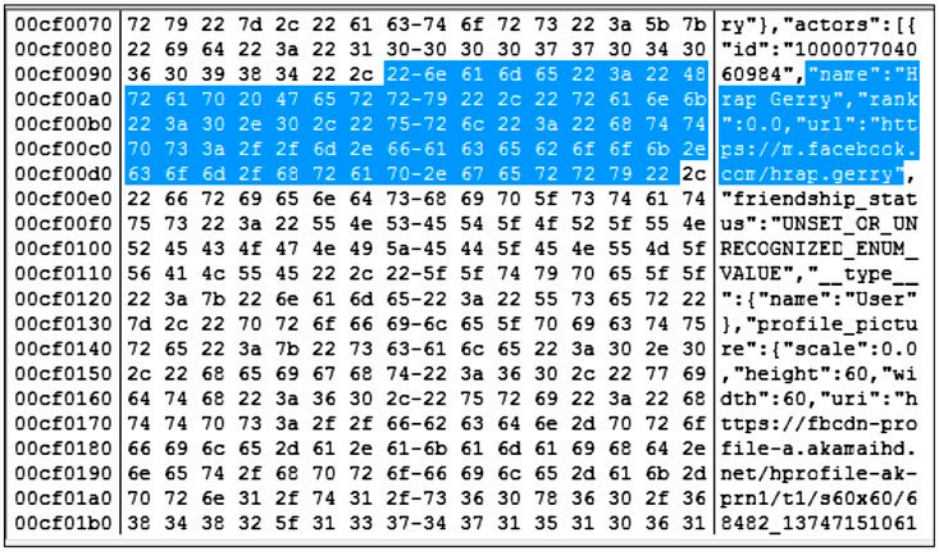

Figure 3. Facebook friend list artefact in Android. (The user's friend was identified as 'Hrap Gerry' with email address 'hg.enron@gmail.com'.)

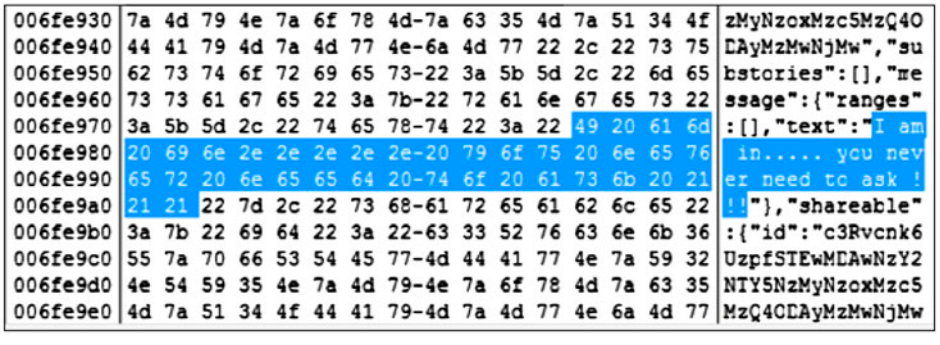

Figure 4. Facebook uploading posts Artefact in Android.

3.5. *Uploading comments*

The Facebook app stores comments uploaded or received by the user on the internal storage in plain text, along with the upload time, as shown in Figure 6. These artefacts contain the information of the person who has uploaded the comment, the relevant post that the comment was posted on, and the content of the comment itself, and can be recovered from the internal storage with a search for the *UID*, username, or email address of the user.

Similarly, searching the internal storage with the user's *UID*, username, and email address revealed the uploaded comments and timestamp metadata for Twitter, LinkedIn and Google+.

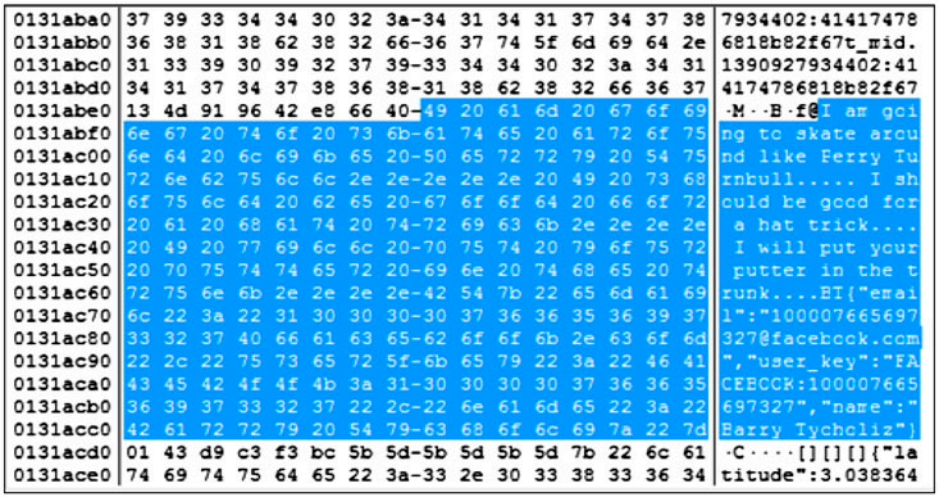

Figure 5. Facebook instant messaging Artefact in Android.

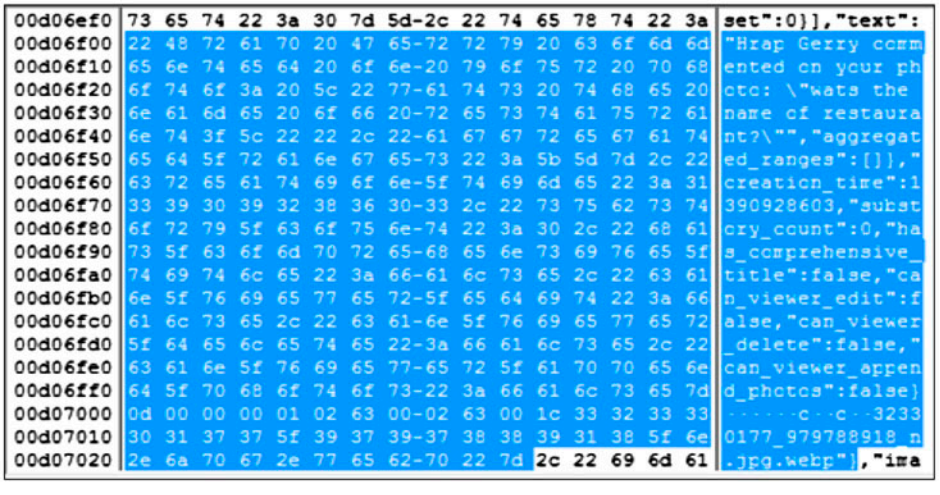

Figure 6. Facebook uploading comments Artefact in Android.

4. Findings of the iOS investigations

This section explains the analysis of acquired iOS backup files to determine the data remnants after performing login, modifying personal information, uploading posts, exchanging messages and uploading comments in the Facebook, Twitter, LinkedIn and Google+ applications. In order to perform the analysis of iOS backup files, a tool called iBackupBot was utilised. The analysis was performed manually on each file to identify the file types, determine the headers and signatures of the files, search for related SN data and determine the location of stored artefacts. The backup files contained several files with the .plist and .sqlite formats. Plist editor was used to view the .plist files content and SQLite Database Browser was utilised to open the .sqlite files. Several

keywords, such as usernames, *uid*, name and email address, were then used to search each file in order to detect SN artefacts. Further explanation on the process of analysis and the findings of the examination are provided in the following subsections.

Login information

Searching in acquired backup files using 'www.facebook.com', 'www.twitter.com', 'www.linkedin.com' and 'www.plus.google.com' revealed the username, *UID* and timestamp of the last successful login for each platform as shown for Facebook in Figure 7. These artefacts were located in the 'com.facebook.Facebook.plist', 'com.atebits.Tweetie2.plist', 'com.linkedin.LinkedIn.plist' and 'loggedInUserKey.plist' files for Facebook, Twitter, LinkedIn and Google+ respectively. However, no password remnant could be detected for any SN platform within the acquired backup files.

User profile Information

The acquired backup files were investigated using the *UID* and username for each SN platform. The Facebook app would leave the user's friend list, phone number, email address, age, place of work, city of residence, interests and GPS location in the 'kFBPrivacySettingCachedSettings-1401330084.plist' file. The Twitter app would leave the user's name, email address, place of work, biography, location, followers list, people followed by, URL pointing to user's profile picture, along with the username and email address of the other Twitter accounts that the user is following or being followed by in the 'app.acct.bsenron.typeahead_users.plist' file. In the LinkedIn app, artefacts including the user's name, email address, employer and position, along with a URL pointing to the user's profile picture were detected in the 'com.linkedin.LinkedIn.plist' file as shown in Figure 8. Users' LinkedIn connections, listed skills, resume, email addresses, connections and URLs showing the profile picture of each connection were extracted from the 'LINUSStreamV2CacheKey.plist' file. The Google+ app left the user's name, email address and a URL redirecting to the user's profile picture in a file named 'loggedInUserKey.plist'.

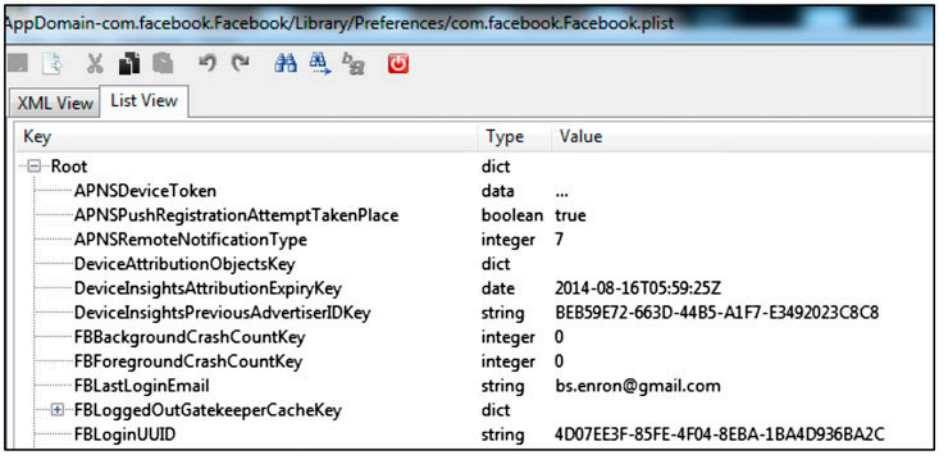

Figure 7. Facebook login artefact in iOS.

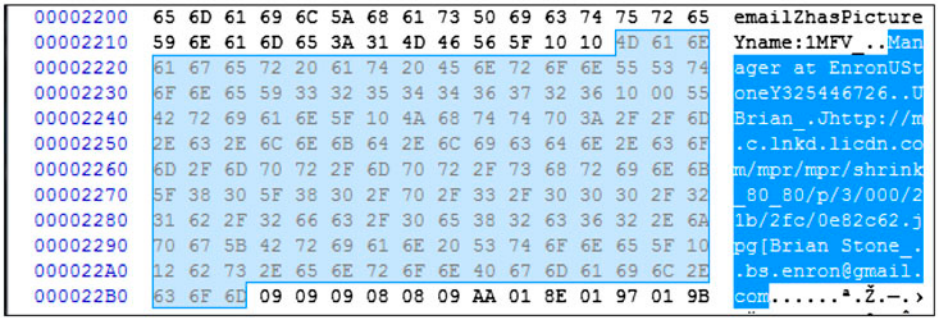

Figure 8. LinkedIn user's personal information Artefact in iOS.

Uploading posts

Investigation of backup files using the user's name and email address revealed that the Facebook app leaves URLs showing public posts uploaded by the user in a file named 'com.facebook.Facebook.plist'. The contents and timestamp of sent tweets in the Twitter app are left in a file called 'com.atebits.Tweetie2.plist'. In the LinkedIn app, uploaded posts remain in a file named 'liv2profile325446726.plist', as shown in Figure 9. In the Google+ app, a SQLite file named 'MediaUploader_d9a336d6e29858e450949fcf0b-b662b2' was detected, containing URLs pointing to pictures posted by the user along with their timestamps and hash values.

Messaging

Investigation of backup files using the *UID*, name and email address of the user revealed that the Facebook app leaves message content in a file named 'Dynamic Dictionary.dat'. This file holds snippets of text that the user types using the device's keyboard. Moreover, the time of the last inbox update, which indicates the last message received, can be retrieved from the 'com.facebook.Facebook.plist' file. The Twitter and LinkedIn apps also reveal the content of messages sent by the user (in this case Brian

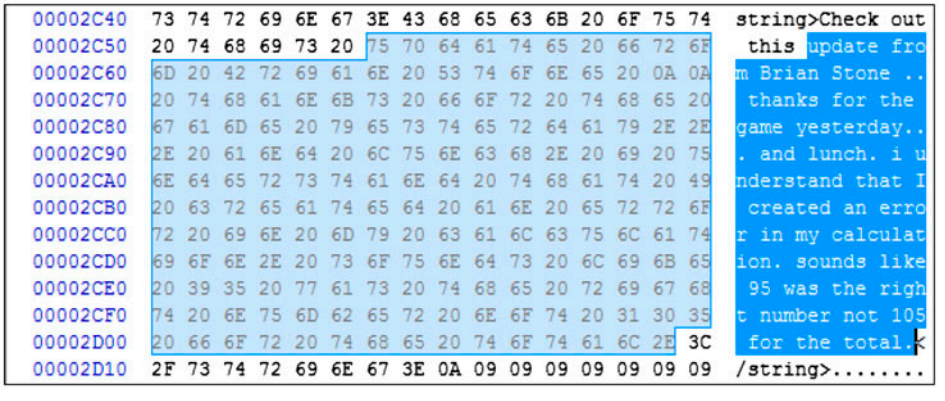

Figure 9. LinkedIn uploading posts Artefact in iOS.

Stone) in the 'DynamicDictionary.dat' file as shown in Figure 10. The Google+ app did not leave any artefacts with forensic value relevant to messages sent and received by the user that we could detect in the iOS backup files.

Uploading comments

Investigation of iOS backup files using the user's *UID*, name and email address revealed that the Facebook app leaves URL artefacts showing comments uploaded on public posts in the 'com.facebook.Facebook.plist' file. The Twitter app leaves detectable content of uploaded comments in the 'DynamicDictionary.dat' file. As shown in Figure 11, the LinkedIn app would leave the full content of uploaded comments, upload time, comments received by the user and the name of the other users (in this case Jerry Strange) who commented on posts uploaded by the user (Brian Stone) in the 'liv2profile325446726.plist' file. Examination of iOS backup files in relation to comments uploaded by the user of the Google+ app did not result in detection of any artefacts with forensic value.

5. Network traffic analysis

When accessing the Facebook, Twitter, LinkedIn and Google+ applications, a session would be established over TCP port 443. Facebook's SSL certificates for api.facebook.com and LinkedIn's SSL certificates for api.lcdn.com are issued by DigiCert, while Twitter's SSL certificates for api.twitter.com and Google+ SSL certificates for api.plus.google.com are issued by Verisign. A session with Google Analytic services is established during the sign-in process for all four applications.

All application layer data are encrypted with TLSv1.2 protocols. Therefore, only the user's IP address, domain name of connected SN sites and corresponding session timestamps can be determined from network traffic. However, the corresponding timestamp can be used as an aid in timelining user activities. Figure 12 shows the network packet containing Facebook's SSL certificate from DigiCert and Facebook's application data encrypted using TLSv1.2.


```
00000040  74 65 63 74 75 72 65 00 03 00 61 72 6F 75 6E 64  tecture...around
00000050  00 01 00 61 73 6B 00 01 00 61 74 74 65 6E 64 00  ...ask...attend.
00000060  01 00 61 75 74 68 65 6E 74 69 63 61 74 69 6F 6E  ..authentication
```

Figure 10. Twitter instant messaging Artefact in iOS.

```
000041C0  78 74 3C 2F 6B 65 79 3E 0A 09 09 09 09 09 09 09  xt</key>........
000041D0  09 09 09 3C 73 74 72 69 6E 67 3E 64 6F 65 73 20  ...<string>does
000041E0  6E 6F 74 20 6D 61 74 74 65 72 2C 20 61 6E 64 20  not matter, and
000041F0  79 6F 75 72 20 77 65 6C 63 6F 6D 65 2E 20 6E 65  your welcome. ne
00004200  78 74 20 74 69 6D 65 20 6C 65 74 20 6D 65 20 63  xt time let me c
00004210  6F 75 6E 74 20 6C 6F 6C 3C 2F 73 74 72 69 6E 67  ount lol</string
00004220  3E 0A 09 09 09 09 09 09 09 09 09 3C 2F 64 69 63  >..........</dic
```

Figure 11. LinkedIn uploading comments Artefact in iOS.

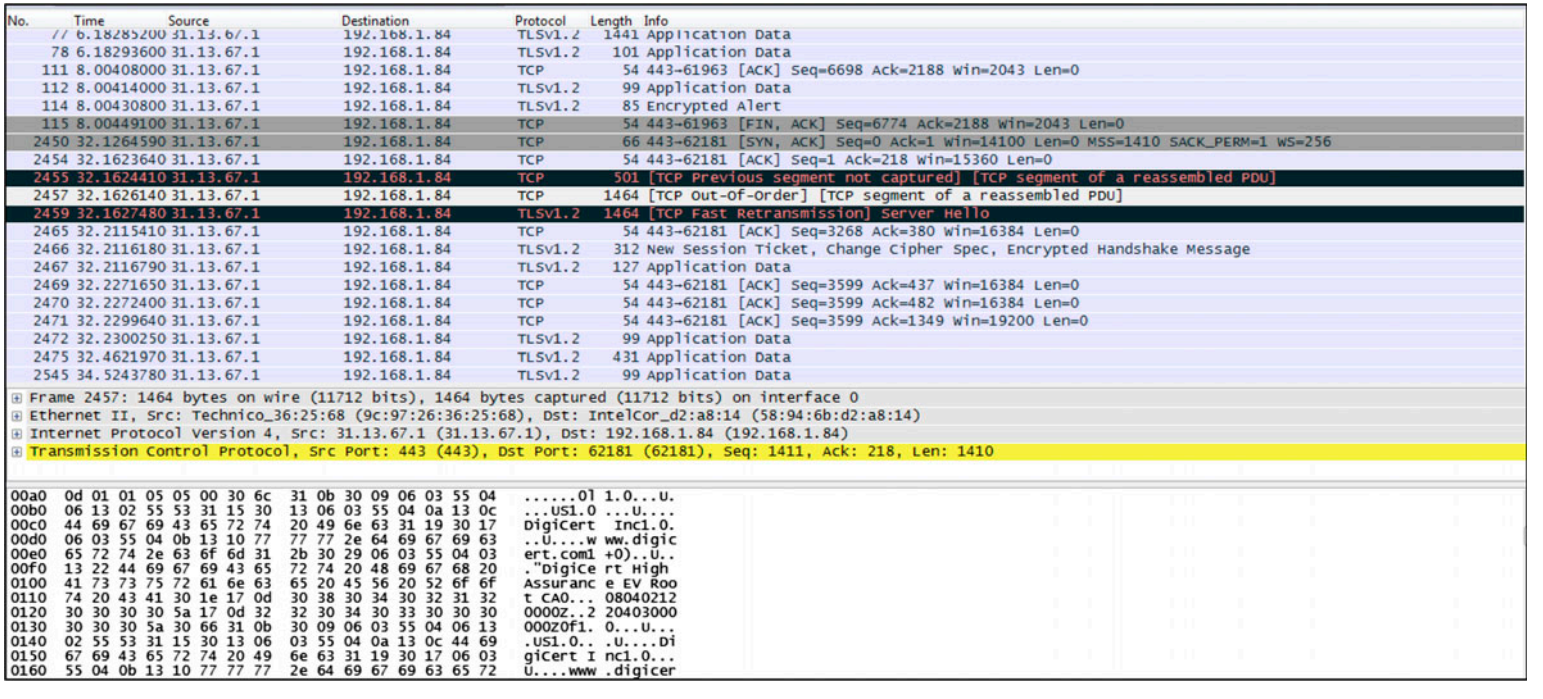

Figure 12. Facebook network traffic artefact.

Table 5. Summary of findings from investigation of SN applications on Android and iOS platforms.

	Facebook		Twitter		LinkedIn		Google+	
	Android	iOS	Android	iOS	Android	iOS	Android	iOS
Login	✓	✓	✓	✓	✓	✓	✓	✓
Username	✓	✓	✓	✓	✓	✓	✓	✓
Password	✗	✗	✗	✗	✗	✗	✗	✗
Name	✓	✓	✓	✓	✓	✓	✓	✓
Contact information	✓	✓	✓	✓	✓	✓	✓	✓
Profile picture	✓	✓	✓	✓	✓	✓	✓	✓
Work and education	✓	✓	✓	✓	✓	✓	✓	✗
Location	✓	✓	✓	✓	✓	✓	✓	✗
Friend list	✓	✓	✓	✓	✓	✓	✓	✗
Posts	✓	✓	✓	✓	✓	✓	✓	✓
Messages	✓	✓	✓	✓	✓	✓	✓	✗
Comments	✓	✓	✓	✓	✓	✓	✓	✗
IP address	✓	✓	✓	✓	✓	✓	✓	✓

6. Concluding remarks

Investigations of the Facebook Android app showed that the username, assigned *UID* and time of last successful login could be recovered from the device's internal memory. The user's name, email address, phone number, profile picture, work and education background, location, friend list, posts, messages, comments and timestamp metadata could be found in the device's internal storage. The Facebook application would leave in iOS backup files the username, assigned *UID,* name, email addresses, phone numbers, profile picture, work and education background, location, friend list, posts, content of messages, comments and timestamp metadata for last successful login, uploaded posts, uploaded comments and the last time the user's inbox was updated.

The Twitter app for Android leaves the assigned *UID* and time for last successful login on the device's internal memory, and the user's name, email address, phone number, profile picture, work and education background, location, friend list, posts, messages, comments and timestamp metadata on the device's internal storage. The Twitter app for iOS leaves in the iOS backup files the username, assigned *UID,* name, email addresses, phone numbers, profile picture, work and education background, location, friend list, content of the tweets, content of the messages and comments, and timestamp metadata for successful login and recent tweets.

The remnants of the Android LinkedIn application found on the device's internal memory are the username, assigned *UID* and time for the last successful login, and artefacts found in the Android internal storage are the user's name, email addresses, phone numbers, profile picture, work and educational background, location, friend list, posts, messages, comments and timestamp metadata on Android internal storage. Remnants of the LinkedIn application to be found on iOS backup files include the username, assigned *UID,* name, email addresses, phone numbers, profile picture, work and education background, location, friend list, posts, comments, content of messages, and timestamp metadata for the last successful login, uploaded posts and uploaded comments.

The remaining artefacts of Android's Google+ application are the username, assigned *UID* and the time for last successful login on the device's internal memory; and the user's name, email address, phone number, profile picture, work and education

background, location, friend list, posts, messages, comments and timestamp metadata on the internal storage. On iOS, the Google+ app leaves the username, assigned *UID,* name, profile picture, email addresses, phone numbers, pictures uploaded in posts and timestamp metadata for the last successful login.

A summary of findings from our investigations is presented in Table 5.

In this research we have adapted the cloud forensic framework of Martini and Choo[44], which integrates both McKemmish [43] and Ayers *et al*.[48] general digital forensic frameworks, to guide our investigations. Using our case study applications, we have demonstrated the utility of the framework for non-cloud investigations. We also believe that the framework of Martini and Choo[44] could be used to investigate other similar smartphone platforms. However, the reported findings are likely to differ between applications and platforms. For example, different smartphones use different operating systems (or different versions) and differences in operating system structure and file types may affect the type and amount of artefacts that could be recovered. Smartphone applications are also constantly being updated by their vendors, which will change the structure, functionality and features of newer applications. Therefore, this could affect the type and amount of artefacts that could be recovered. Tools would also have to keep pace with hardware and software changes, which could result in different artefacts recovered. Therefore, future research would include extending this research to other newer applications such as games, health, instant messengers, cloud storages and banking, and newer smartphone platforms, which would allow us to have an up-to-date forensic understanding.

Acknowledgements

This work is partially supported by FP7-PEOPLE-2013-IIF project 625402. The views and opinions expressed in this article are those of authors alone and not the organizations with whom authors are or have been associated/supported. We thank the editor and the anonymous reviewers for providing constructive and generous feedback. Despite their invaluable assistance, any errors remaining in this paper are solely attributed to the authors.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

1. Boyd DM, Ellison NB. Social network sites: definition, history, and scholarship. J Comput-Mediat Comm. 2007;13(1):210–230.
2. Statista. Facebook: figures of monthly active users 2014 | Statistic. Statista [Internet]. 2015 [cited 1 April 2015]. Available from: http://www.statista.com/statistics/264810/number-of-monthly-active-facebook-users-worldwide/
3. Taylor M, Haggerty J, Gresty D, Almond P, Berry T. Forensic investigation of social networking applications. Network Secur. 2014;2014(11):9–16.
4. Kisekka V, Bagchi-Sen S, Raghav Rao H. Extent of private information disclosure on online social networks: an exploration of Facebook mobile phone users. Comput Human Behav. 2013;29(6):2722–2729.
5. Brenner J. Social networking fact sheet. Pew Research Center's Internet & American Life Project [Internet]. 2013 [cited 20 November 2014]. Available from: http://www.pewinternet.org/fact-sheets/social-networking-fact-sheet/
6. Zainudin NM, Merabti M, Llewellyn-Jones D. A digital forensic investigation model for online social networking. Proceedings of the 11th annual conference on the convergence of telecommunications, Networking & Broadcasting, Liverpool; 2010. p. 21–22.

7. Damopoulos D, Kambourakis G, Gritzalis S. iSAM: an iPhone stealth airborne malware. In: Camenisch J, Fischer-Hübner S, Murayama Y, Portmann A, Rieder C, editors. Future challenges in security and privacy for academia and industry. IFIP Advances in Information and Communication Technology. Springer: Berlin Heidelberg; 2011. p. 17–28.
8. Damopoulos D, Kambourakis G, Gritzalis S. From keyloggers to touchloggers: take the rough with the smooth. Comput Secur. 2013;32:102–114.
9. Damopoulos D, Kambourakis G, Gritzalis S, Park SO. Exposing mobile malware from the inside (or what is your mobile app really doing?) P2P Networking App. 2012;1–11.
10. Barmpatsalou K, Damopoulos D, Kambourakis G, Katos V. A critical review of 7 years of mobile device forensics. Digital Invest. 2013;10(4):323–349.
11. comScore Inc. comScore Reports July 2014 U.S. Smartphone Subscriber Market Share. comScore, Inc [Internet]. 2014 [cited 4 March 2015]. Available from: http://www.comscore.com/Insights/Market-Rankings/comScore-Reports-July-2014-US-Smartphone-Subscriber-Market-Share
12. MBaca, Cosic J, Cosic Z. Forensic analysis of social networks (case study). In: Information Technology Interfaces (ITI), Proceedings of the ITI 2013 35th International Conference on. IEEE; 2013. p. 219–223.
13. Martini B, Do Q, Choo K-KR. Chapter 14-Conceptual evidence collection and analysis methodology for Android devices. In: Choo K-KR, editor. The Cloud Security Ecosystem. Boston, MA: Syngress; 2015. p. 285–307.
14. Shariati M, Dehghantanha A, Martini B, Choo K-KR. Chapter 19-Ubuntu One investigation: detecting evidences on client machines. In: Choo K-KR, editor. The Cloud Security Ecosystem. Boston, MA: Syngress; 2015. p. 429–446.
15. Hutchings C. Commercial use of Facebook and Twitter – risks and rewards. Comput Fraud Secur. 2012;2012(6):19–20.
16. Sipior JC, Ward BT, Volonino L, MacGabhann L. A framework for the E-Discovery of social media content in the United States. Inf Sys Manag. 2013;30(4):352–358.
17. Angelopoulou O, Vidaliss S. Towards "crime specific" digital investigation frameworks. In: School of Computer Science, University of Cardiff, Cardiff; 2013.
18. Tassone C, Martini B, Choo K-KR, Slay J. Mobile device forensics: a snapshot. Trends Issues Crime Criminal Justice. 2013(460):1–7.
19. Grispos G, Storer T, Glisson WB. A comparison of forensic evidence recovery techniques for a windows mobile smart phone. Digital Invest. 2011;8(1):23–36.
20. Mokhonoana PM, Olivier MS. Acquisition of a Symbian smart phone's content with an on-phone forensic tool. In: Proceedings of the Southern African Telecommunication Networks and Applications Conference 2007 (SATNAC 2007). Sugar Beach Resort, Mauritius; 2007.
21. Casey E. Digital evidence and computer crime: forensic science, computers and the Internet. Baltimore: Academic Press; 2011.
22. Hoog A. Chapter 6-android forensic techniques. In: Hoog A, editor. Android forensics: investigation, analysis and mobile security for Google Android. Boston, MA: Syngress; 2011. p. 195–284.
23. Son N, Lee Y, Kim D, James JI, Lee S, Lee K. A study of user data integrity during acquisition of Android devices. Digital Invest. 2013;10(Supplement):S3–S11.
24. Vidas T, Zhang C, Christin N. Toward a general collection methodology for Android devices. Digital Invest. 2011;8(Supplement):S14–S24.
25. Lessard J, Kessler G. Android forensics: simplifying cell phone examinations. Small Scale Digital Device Forensics J. 2009;4(1):1–12.
26. Sylve J, Case A, Marziale L, Richard GG. Acquisition and analysis of volatile memory from android devices. Digital Invest. 2012;8(3–4):175–184.
27. Do Q, Martini B, Choo K-KR. Exfiltrating data from Android devices. Comput Secur. 2015;48:74–91.
28. Zdziarski J. iPhone forensics: recovering evidence, personal data, and corporate assets. 1 edition ed. Sebastopol, CA: O'Reilly Media; 2008.
29. Hoog A, Gaffaney K, iPhone Forensics Whitepaper Introduction. NowSecure Blogs [Internet]. 2009 [cited 2015 July 22]. Available from: https://www.nowsecure.com/blog/2009/07/11/iphone-forensics-whitepaper-introduction/
30. Morrissey S, Campbell T. iOS forensic analysis: for iPhone, iPad, and iPod touch. 2010 ed. Berkeley, Calif. : New York: Apress; 2010.

31. Bader M, Baggili I. iPhone 3GS forensics: Logical analysis using apple iTunes backup utility. Small Scale Digital Device Forensics J. 2010;4(1):1–15.
32. Proffitt T, Forensic Analysis on iOS Devices – Forensic-analysis-ios-devices-34092. 2012 [cited 2015 July 22]. Available from: https://www.sans.org/reading-room/whitepapers/forensics/forensic-analysis-ios-devices-34092
33. Al Mutawa N Baggili I, Marrington A. Forensic analysis of social networking applications on mobile devices. Digital Invest. 2012;9:S24–S33.
34. Engman M. Forensic investigations of Apple's iPhone. HALMSTAD: Sektionen för informationsvetenskap; 2013.
35. Ariffin A, D'Oorazio C, Choo K-KR, Slay J. iOS forensics: how can we recover deleted image files with timestamp in a forensically sound manner? In: 2013 Eighth International Conference on Availability, Reliability and Security (ARES); 2013. p. 375–382.
36. Jung J, Jeong C, Byun K, Lee S. Sensitive privacy data acquisition in the iPhone for digital forensic analysis. In: Park JJ, Lopez J, S-SYeo, Shon T, Taniar D, editors. Secure and Trust Computing, Data Management and Applications. Communications in Computer and Information Science. Berlin Heidelberg: Springer; 2011. p. 172–186.
37. Said H, Yousif A, Humaid H. IPhone forensics techniques and crime investigation. in: Current Trends in Information Technology (CTIT), 2011 International Conference and Workshop on. IEEE; 2011. p. 120–125.
38. Tso Y-C, Wang S-J, Huang C-T, Wang W-J. iPhone social networking for evidence investigations using iTunes forensics. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication. ICUIMC '12. New York, NY: ACM; 2012. p. 62:1–62:7.
39. Quick D, Choo K-KR. Digital droplets: Microsoft SkyDrive forensic data remnants. Future Gener Comput Syst. 2013;29(6):1378–1394.
40. Quick D, Choo K-KR. Dropbox analysis: data remnants on user machines. Digital Invest. 2013;10(1):3–18.
41. Quick D, Choo K-KR. Google drive: forensic analysis of data remnants. J Network Comput App. 2014;40:179–193.
42. Anglano C. Forensic analysis of WhatsApp Messenger on android smartphones. Digital Invest. 2014;11(3):201–213.
43. McKemmish R. What is forensic computing? Trends & issues in crime and criminal justice Australian Inst Criminology; 1999;118:1–6.
44. Martini B, Choo K-KR. An integrated conceptual digital forensic framework for cloud computing. Digital Invest. 2012;9(2):71–80.
45. Martini B, Choo K-KR. Distributed filesystem forensics: XtreemFS as a case study. Digital Invest. 2014;11(4):295–313.
46. Martini B, Choo R. Remote programmatic vCloud forensics: a six-step collection process and a proof of concept. In: Proceedings of 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2014). IEEE Computer Society Press; 2014. p. 935–942.
47. Martini B, Do Q, Choo K-KR. Mobile cloud forensics: an analysis of seven popular Android apps. In: Choo K-KR, editor. The cloud security ecosystem. Boston, MA: Syngress; 2015. p. 309–345.
48. Ayers R, Brothers S, Jansen W. Guidelines on mobile device forensics. National Institute of Standards and Technology; 2014.
49. EDRM LLC, EDRM Enron Email Data Set. EDRM LLC [Internet]. 2013 [cited 2014 November 24]. Available from: http://www.edrm.net/resources/data-sets/edrm-enron-email-data-set