

Computer Systems Science & Engineering DOI:10.32604/csse.2021.014472 Article

Instagram Mobile Application Digital Forensics

Muhammad Asim Mubarik¹, Zhijian Wang¹, Yunyoung Nam^{2,*}, Seifedine Kadry³ and Muhammad Azam waqar⁴

¹College of Information and Computer Science, Hohai University, Nanjing, 210098, China ²Department of Computer Science and Engineering, Soonchunhyang University, Asan, 336811, Korea ³Department of Mathematics and Computer Science, Beirut Arab University, Beirut, 000000, Lebanon ⁴School of Business Management, NFC Institute of Engineering & Fertilizer Research, Faisalabad, 38800, Pakistan *Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr Received: 22 September 2020; Accepted: 27 November 2020

> Abstract: In this research, we developed a plugin for our automated digital forensics framework to extract and preserve the evidence from the Android and the IOS-based mobile phone application, Instagram. This plugin extracts personal details from Instagram users, e.g., name, user name, mobile number, ID, direct text or audio, video, and picture messages exchanged between different Instagram users. While developing the plugin, we identified resources available in both Android and IOS-based devices holding key forensics artifacts. We highlighted the poor privacy scheme employed by Instagram. This work, has shown how the sensitive data posted in the Instagram mobile application can easily be reconstructed, and how the traces, as well as the URL links of visual messages, can be used to access the privacy of any Instagram user without any critical credential verification. We also employed the anti-forensics method on the Instagram Android's application and were able to restore the application from the altered or corrupted database file, which any criminal mind can use to set up or trap someone else. The outcome of this research is a plugin for our digital forensics ready framework software which could be used by law enforcement and regulatory agencies to reconstruct the digital evidence available in the Instagram mobile application directories on both Android and IOS-based mobile phones.

> **Keywords:** Digital forensics; Instagram; mobile application forensics; antiforensics; forensics framework plugin

1 Introduction

Since the introduction of Facebook, online social networks have evolved (over the last decade) and a countless number of applications, that provide different features, have surfaced on the Internet [1]. These applications vary from generic social network services, to image-sharing, and video sharing, social networking services. Their primary purpose is help people from different continents to stay connected with one another.



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Among the most popular social networking sites is Instagram, (primarily used via its mobile application) [2] which has over a billion registered users [3]. It is a social networking application that lets users capture and share photos and videos within their social circles. Users of Instagram, register themselves with a unique user ID and password. They also have the option of connecting their Instagram accounts with their Facebook accounts, which help users share their photos and videos with their audience (social circles) more effectively. Similar to Facebook's functionality, every Instagram user has their own personal newsfeed, which functions as a personal notice board, displaying content shared by individuals or pages that the user has subscribed to ("follows"). This visual sharing platform has become more popular nowadays because Instagram provides users with additional features including video editing and photo editing tools (enabling users to add filters, animals, and shapes, to their photos and videos). The "story" feature of Instagram allows users to attract more audience, which consequently helps them increase interaction on their regular posts. This results in users accumulating more highly desired "likes" on their posts. The "story" functions as a secondary newsfeed, situated atop the main personal newsfeed of every Instagram user. Through such activities, the platforms helps people from across the globe, to connect and form new relationships, in a very interactive manner. However, unbeknownst to its users, applications such as Instagram provide personal information of its users to other users, which can be potentially dangerous. In today's technologically advanced era, people are using platforms such as Instagram to find their future spouses online. Couples connect with each other freely on Instagram, without comprehending the potential danger they are exposed to. Cases of online theft, copyright infringement, extortion, kidnapping [4] and even rape, have been reported, which occurred because the victim had trusted a friend from their social network friend list, and shared personal information online.

In recent times, more and more cases are being reported to investigation agencies, which involve criminal activity caused by the misuse of social media platforms. These investigation agencies employ various digital forensics tools to extract key evidence from the mobile devices of culprits, to help get them convicted in the court of law. The problem here is that there are so many different devices and applications generating such large amounts of data, that its difficult for digital forensics experts keep themselves updated on latest digital forensics tools [5]. Keeping in mind the significance of social networking applications and digital forensics, national governments are now updating their standards and training their staff to detect drug-related crimes and stop drug trafficking operations [6].

"So many applications and so many technologies are being created and continuously updated, that forensic investigators cannot keep up" [7].

"Our Digital Forensics research group here at Shanghai Forensic Research Center keeps adding automated forensics plugins for every new app that is popular among the masses so that our law enforcement officers can extract evidence from all the apps available on mobile phones whether it's an Android phone or an IOS phone."

In this paper, we have studied the forensic artifacts of the Instagram application on both Android and IOS phones. We implemented code to automatically extract these forensics artifacts using our forensic framework environment, which is capable of extracting evidence from more than seventy different Android and IOS applications. At the end of this paper, we have discussed an anti-forensics experiment conducted on the Android Instagram application, to gauge its effectiveness. We have also presented a few privacy issues that we found in both versions of the Instagram application (Android and IOS).

2 Related Work

Forensics analysis of the Instagram mobile application is not a very popular topic. Several researchers have done the forensics analysis of Instagram, but none of them have helped digital forensics investigators to reconstruct the data from forensic artifacts available in mobile devices. Instagram has always been a

successful and popular application. Within only two months of its inception, it had 2 million users registered worldwide. The popularity of the Instagram application coupled with the abundant forensics artifacts it leaves behind is why forensics experts find the application so interesting. Consequently, standalone applications like Instagram that run on mobile phones can provide add-on forensic information to help identify criminal suspects. A suspect's geolocation near a crime scene and their digital signature available on their installed Instagram application, can help resolve many cases. Such applications nowadays also provide a very convenient one-tap login facility, which allows you to set your mobile phone as your default device.

Reema Al Mushcab et al. performed forensics analysis of the iPhone 5s Instagram application [8]. Their focus was primarily on "write-blockers" rather than the forensics artifacts of the Instagram application. A problem that we identified in this research is that they could not locate the direct messages database in the Instagram directory "com.instagram.android" of the IOS application, where the main evidence usually resides. This is the main focus of our research. We want to extract evidence (messages exchanged between the victim and the accused directly using this application) of a criminal nature, which will prove the involvement of suspects in kidnapping, murdering, bombing, raping, or financial corruption cases.

In 2015, Ming [9] employed evidence-gathering techniques on Instagram using the Windows 10 Operating System. The focus of Ming Sang Chang's research matches our own research objectives which is to capture extensive evidence from social networking services that can be used to help deter people from committing crimes such as spreading slander, cyberstalking, cyberbullying, hacking, copyright infringement, rape, murder, and financial corruption. Chang's research involved capturing evidence from Internet Explorer and Google Chrome, using the SQLite database and WinHex to find data remnants of user's account IDs and passwords. The researchers created a scenario in which, after performing a criminal activity, the user tried to remove the digital evidence using Eraser Portable v5.8.8.1 and CCleaner V5.19.5633. The researchers subsequently performed digital forensics analysis of the user's hard drive to uncover any evidence. Unlike Mr. Chang's research, our own research focuses mainly on forensics analysis of the Android and IOS mobile applications.

Wong et al. (security researcher at Valkyrie-X Security Research Group) [10] conducted a detailed digital forensics study of Facebook's web application and mobile application. This research was conducted on the iPhone 3GS IOS version 4.3 which is why it's very old and outdated. The research is also limited only to IOS forensics, neglecting the large majority of Android device users.

Yusoff et al. [11], conducted a forensic investigation of social media and instant messaging services on Firefox OS, including Instagram, in 2017. This research involved using Forensics Toolkit (FTK) version 3.1.2 and HxD Hex Editor 1.7.7.0 to capture and analyze memory images on a phone called Peak (Geeksphone, 2013) running Mozilla FxOS. The limitation in this research is that volatile memory cannot hold data for long periods and forensic artifacts can only be found on the phone while Mozilla FxOS is running. If forensics analysis is performed after the phone has restarted, all the digital forensics evidence will have evaporated out of the phone's volatile memory. More importantly, every social networking service has now launched their own customized application, and registered users prefer accessing this application, rather than browsing the social media site it is affiliated with, on the mobile browser. The main reason for this is that users no longer need to re-login every time to use their desired social networking site, instead they can use the application's one-tap login feature. The mobile applications enable this feature by saving the registered user's account credentials on the mobile device. Mobile apps such as Instagram have their directory structures saved onto the users mobile device's physical storage. Such mobile applications store crucial information on persistent storage of the device aswell. In this research (of ours) we will show that the complete SQL lite database file can be found in different directories in both Android and IOS devices.

In 2015, Daniel Walnysky et al. [12], published their research on direct messaging mobile applications and established how evidentiary traces allow reconstruction of data, and permit reconstruction of activities performed by users and applications. Their work mainly focuses on only Android applications. In their research, they have also suggested that the automated reconstruction of data is also possible, mentioning it as the basis of their future work. This automated reconstruction of data is the main outcome of our own research. In our research's analysis and results section we will discuss how our framework reconstructs the forensic remnants available inside the mobile phone and the concerned application directories for both Android and Apple devices. Walnysky et al. [12], also performed similar work on Facebook forensics analysis [13,14]. Our own research group has also conducted similar extensive research on Mobile application's digital forensics. Until now, our group has performed digital forensics analysis on more than seventy popular mobile applications used by the masses in China and elsewhere. The fruits of our labor will enable law enforcement personnel to use our digital forensics-ready workstation to perform necessary forensics analysis in ongoing cases, and present the evidence in a court of law.

This paper discusses our research and development of a forensics plugin for the Instagram mobile application.

3 Test Environment and Requirements

This framework was developed in Visual Studio with devexpress tools installed. Below is a complete list of all the hardware and software tools used to perform the forensics analysis of both Android and IOS based mobile applications:

- iPhone 7Plus (v. 12.0.1)
- Meizu Note 6
- Instagram (v. 69.0)
- Apple iTunes application (v. 12.1)
- Android Debug Bridge (ADB)
- Microsoft Visual Studio Professional 2012 (v. 11.0.50727.1 RTMREL)
- DEVEXPRESS
- SQLite Expert Professional (v. 3.5.21.2440)
- DATABASE Browser for SQL open-source (Version 3.10.1)
- Plist Editor Pro (v. 2.0)
- Win Hex/X-Ways Forensics software
- Apple's iPhone 7Plus USB data cable
- Windows Photo Viewer
- VLC Media player (v. 2.1.3)

4 Forensics Analysis of Android-Based Instagram Application

4.1 Retrieval of the Instagram Directory Structure

We selected the Instagram application because our research group had already performed digital forensics for almost all the popular social media networking applications including Facebook, WhatsApp, Line, Weibo, and so many others. A picture of the front end for our framework can be seen in Fig. 1.

We chose a mobile application based on popularity in China because this framework is designed for the use of Chinese law enforcement agencies and public security organizations to help solve cases more easily, and hence provide a more safe and secure social life within Chinese cities. We performed forensics analysis

of the Instagram application for both Android and Apple IOS versions. For this activity, we first installed the Instagram application on an Android device as well as an Apple IOS device. In this specific experiment, we used an Android-based Meizu Note 5 phone and an Apple IOS 7 plus mobile device with IOS version 12.1.1.

ase enter the content to be searched for								
入关键字之后自动搜索	Q.							
ect the evidence to be searched		Application s	earch OSearc	h for multimedia infor	mation			
Evidence type		Select all						
	-	□ 终端信息	🗌 文件系统	🗌 通话记录	🗌 短信彩信	🗌 通信录	日历	
		备忘录	□ 多媒体	🗌 已安装软件	₩ifi热点	□ 蓝牙记录	□ 同步账号	
		即时通讯						
		ହତ	□ 微信	□ 易信	点点虫	飞信	田信	
		β百β百	- 米聊	TA AN	Skype	111	line	
		Facebook		INSTAGRAM				
		暴恐类软件						
		TalkBox	Voxer	Viber	Coco	GTalk	Telegram	
		WhatsApp	Zalo	Zello	imo班聊	IM+		
		微博和浏览器						
		🗌 浏览器	─ 新浪微博	□ 騰讯微博	Twitter			
		邮件						
		□ 邮件	139邮箱	189邮箱	□ 沃邮箱	QQ邮箱	Gmail邮箱	
	-	网易邮箱	□邮箱大师					
re are 0 search results								
No. Evidence Function		Title		时间		Conten		
No. LVIGENCE Function		fitle		H2181		Lonten		

Figure 1: Forensics framework front end

4.2 Android Devices Data Extraction

The second stage was to utilize the "pull" command of Android Debug Bridge (ADB), which helped us extract the contents of "com.instagram.android." We did this so that we could perform manual analysis of the changes in the contents of the package upon performing different activities (creating user, sending a message, and sharing pictures) via the mobile application. Fig. 2 shows the flow diagram of our experiment analysis process that we followed.

5 Key Forensics Artifacts Identification of Android Instagram App

After acquiring the directories from an Android device, we performed a manual analysis of the application and attempted to locate the files of interest. Tab. 1 elaborates the information cum evidence that we wanted to locate from these devices; for this purpose, we performed an activities in the Instagram app so that data is generated and stored into the Instagram directory structure. We created two user names on Instagram to generate the forensics artifacts in the Instagram mobile application database. Tab. 1 elaborates on the activities performed in the mobile application to generate the data that would be extracted as evidence later on by our forensics framework.

Examination of Instagram on Android (com.instagram.android): Instagram creates com.instagram. android in data/app/ to store all the directories and files. Directories of com.instagram.android are shown in Fig. 3.

The "databases" directory and "shared_prefs" directory hold important forensics artifacts for digital forensics analysis of Instagram. "shared_prefs" directory contains XML files that hold 'full name', 'user

name', 'id', location, and other important information about the user of the application. The databases directory contains the direct messages exchanged between the user and other Instagram users.

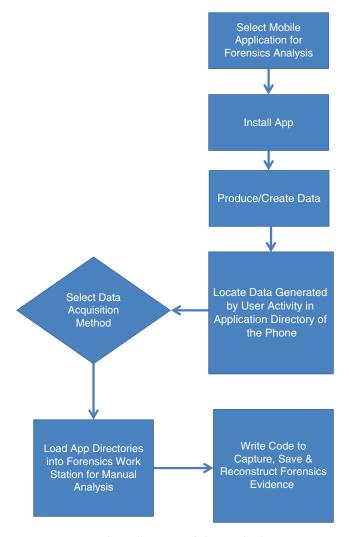
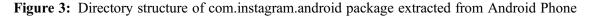


Figure 2: Flow diagram of the analysis process

Table 1:	Activities	performed
----------	------------	-----------

Mobile & Application	Activity
1. Meizu Android Instagram.apk	1. New user registration
2. IOS iPhone 7plus Instagram	2. Following another Instagram user
	3. Send text message
	4. Send picture message
	5. Send audio message
	6. Send video clip

Name	Date modified	Туре
app_acra-reports	11/4/2018 5:14 PM	File folder
app_batch_counter	11/4/2018 5:14 PM	File folder
app_browser_proc_webview	11/4/2018 5:14 PM	File folder
📊 app_funnel_backup	11/4/2018 5:14 PM	File folder
app_ig_analytics_beacon	11/4/2018 5:14 PM	File folder
app_light_prefs	11/4/2018 5:14 PM	File folder
🔒 app_minidumps	6/1/2018 9:52 AM	File folder
app_modules	11/4/2018 5:14 PM	File folder
📊 app_overtheair	11/4/2018 5:14 PM	File folder
app_textures	6/1/2018 9:52 AM	File folder
app_webview	11/4/2018 5:14 PM	File folder
📊 cache	11/4/2018 5:14 PM	File folder
code_cache	11/4/2018 5:14 PM	File folder
databases	11/4/2018 5:14 PM	File folder
files	11/4/2018 5:14 PM	File folder
📙 lib-main	11/4/2018 5:14 PM	File folder
shared_prefs	11/4/2018 5:14 PM	File folder



In the following section, we will describe the anatomy of the "com.instagram.android_preferences.xml" file. Snapshot of this file is presented in Fig. 4, and we have listed important forensics artifacts in Tab. 2.

xml version="1.0" encoding="UTF-8" standalone="true"?
<map></map>
<string name="App Restrictions">AAAAAA== </string>
<int name="carousel_nux_impressions" value="25"></int>
<string name="google_ad_id">abf68bb6-70af-4438-a72b-822e83f188f0</string>
<string name="user_access_map">[{"user_info":</string>
{"id":"7526247127","blocking":false,"blocking_reel":false,"follower_count":13,"following_count":null,"follow_status":"Fo sit4-1.cdninstagram.com/vp/b38273e465e0bb2777b6845dea25cbd2/5BB665AE/t51.2885-
19/s150x150/30078315_2120078654888308_1563912706187067392_n.jpg","profile_pic_id":"1759213689388687079_7 followers","coeff_weight":0.0,"can_see_organic_insights":false_gagregate_promote_engagement':false,"can_boost_post {"mode":0,"gradient":2,"emoji":"@","selfie_sticker":0,"selfie_un"gruinut:1","time_accessed":1527749843766) <</th
<pre>string name="preference hardware id">serie_stoke: o, serie_un: nump;; time_accessed .1327749645760;]</pre>
<pre><string name="puelerence_naruware_nu"></string></pre>
<pre>string name= pusi_reg_dateanorud_mqtt value= 1527/4936/114/5</pre>
["upsell seen before";false,"allow non fb sso":true,"rejected sso upsell";false,"one tap upsell after login count":0.
[1 upsel_seen_before_inase, anow_non_base .true, rejected_sso_upsen_inase, one_tap_upsen_arter_iogin_count_to, sit4-1.cdninstagram.com/vp/be5446832099264ae250fdc166ff7a/58C4F2EC/t51.2885-
19/s150x150/30591478_1604198099628650_1119136862359781376_n.jpg"}]}
<string name="cm_last_bandwidth">ConnectionManagerHistoricalData:mData=54.34873103962058,</string>
minestamp=1527753149029
<int name="number of carousels swiped" value="1"></int>
long name="b_LAST_REQUEST" value="1527749698"/>
 soolean name="show tos" value="false"/>
<boolean name="used double tap" value="true"></boolean>
<string name="fb attribution id">a3ea46e6-dc0b-4da0-949b-6b49d64db586</string>
<string name="cm_last_latency">ConnectionManagerHistoricalData:mData=250.30865273515246,</string>
mTimestamp=1527753149029
<string name="current"></string>
{"id":"7526247127","blocking":false,"blocking_reel":false,"follower_count":13,"following_count":null,"follow_status":"Fasit4-1.cdninstaaram.com/vp/b38273e465e0bb2777b6845dea25cbd2/5BB665AE/t51.2885-
19/s150x150/30078315_2120078654888308_1563912706187067392_n.jpg","profile_pic_id":"1759213689388687079_7
followers","coeff weight":0.0,"can see organic insights":false,"aggregate promote engagement":false,"can boost post
{"mode":0,"gradient":2,"emoji":"@","selfie_sticker":0,"selfie_url":null}}
<boolean name="google_ad_id_logged" value="true"></boolean>
<boolean name="opt_out_ads" value="false"></boolean>
<boolean name="written_cache_dummy_file" value="true"></boolean>
<int name="used_double_tap_hint_impressions" value="3"></int>
<boolean name="com.facebook.sdk.appInstallEvent" value="true"></boolean>

Figure 4: Contents of com.instagram.android.xml

This XML file contains two important tags that store information regarding the user of the application; the information is stored in a key-value pairs format, which can be easily extracted using any programming technique. In our experiment, as we mentioned above, we utilized Visual Studio with devexpress tool, to program the extraction of these forensics artifacts. The rest of the XML files and directories contain user bootstrap services information, cookies, etc. The next directory of our interest is the databases directory, which contains direct messages (in the file named 'direct.db') exchanged between the registered user of the mobile application, and other Instagram users. Fig. 5 shows the relations and tables in the direct.db file of Instagram; In these tables, the messages table, contains the direct messages exchanged between different users and has great significance as digital forensics information.

Forensic Information	Key	Value
Unique ID by Instagram	Id	7526247127
Total number of people following this ID	follower_count	13
Count of IDs this user is following	following_count	Null
Full name of user	full_name	DarkTest
URL of user profile picture	profile_pic_url	https://scontent-sit4-1.cdnInstagram.com/vp/ b38273e465e0bb2777b6845dea25cbd2/5BB665AE/ t51.2885-19/s150x150/30078315_ 2120078654888308_1563912706187067392_n.jpg
Unique ID by Instagram for this profile picture	profile_pic_id	1759213689388687079_7526247127
User name registered on Instagram	Username	dark2539
Selfie picture URL	selfie_url	Null
Last online time	time_accessed	1527749843766

Table 2: Important forensics artifacts available within User Access Map tag com.instagram.android.xml

Name		Туре	Schema							
• 🔳 •	Tables (6)									
>	android_metadata		CREATE TABLE android_metadata (locale TEXT)							
~	messages		CREATE TABLE messages(_id integer primary key autoincrement, user_id text, server_item_id text, client_item_id text, thread_id							
	📄 _id	integer	`_id` integer PRIMARY KEY AUTOINCREMENT							
	user_id	text	`user_id`text							
	server_item_id	text	`server_item_id` text							
	📄 client_item_id	text	`client_item_id` text							
	thread_id	text	`thread_id` text							
	recipient_ids	text	`recipient_ids` text							
	📄 timestamp	integer	`timestamp` integer NOT NULL							
	message_type	text	`message_type` text NOT NULL							
	Lext	text	`text`text							
	📄 message	text	`message` text NOT NULL							
>	mutations		CREATE TABLE mutations(_id integer primary key autoincrement, user_id text, mutation_type text not null, mutation text not n							
>	session		CREATE TABLE session(user_id TEXT PRIMARY KEY, value TEXT NOT NULL)							
>	sqlite_sequence		CREATE TABLE sqlite_sequence(name,seq)							
>	threads		CREATE TABLE threads(_id integer primary key autoincrement, user_id text, thread_id text, recipient_ids text, last_activity_time i							

Figure 5: Table relation in Direct.db

Here, we can observe that the message-id and user-id are the current user's id to identify the user of the current Instagram mobile application. As Instagram stores a copy of the contents on the server side, so every message is assigned a server_item_id whereas contents that reside in mobile application directories are assigned client_item_id, while recipient_id timestamp is for when the message was received.

Message_type distinguishes whether the message is a text message or an audio-video message. In case it is a text message, the text is stored in the 'text' field of this table. However, if the message is a picture or video message, then the link of the multimedia message is stored in the message field while the 'text' field is kept empty and has no value stored in it; this has been shown in Fig. 5 and Fig. 6b by highlighting the message field data in these pictures. The Message field is an important element of this table and stores the most valuable forensics information for all kinds of messages, whether it is a text, video, or audio message.

CSSE, 2021, vol.37, no.2

0	ew Data	oase 🛛 👼 Open I	Database 🛛 📋 Wr	ite Chang	es 🥴	Revert Chang	ges									
ata	base Str	ucture Browse	Data Edit Pragmas	s Exe	cute SQI							Edit Database	Cell			
able	: 🔲 n	nessages					• 🔁 🍾			New Record D	elete Record	Mode: Text				t as <u>N</u> L
	_id	user_id	server_item_id	nt_item	read	recipient_ids	timestamp	message_type	text	message	^			bled":false},"item_ty		am
		Filter	Filter			Filter	Filter	Filter	Filter	Filter			213965211439525 cf-007a-40d4-83c	67642387187236864	","client_cont	ext
1	1469	7526247127	28121395519	NULL	340	7575387	15244639057	text	this is Asim	j		a963dda8	368c5","timestamp	":"152446396007752	9","timestam	p_i
5	1470	7526247127	28121395961	519	340	7575387	15244639297	text	hi	{"content_typ		n_micro"		d":"7526247127" "te	vt"-"thic ic	
5	1471	7526247127	28121396521	8f7e	340	7575387	15244639600	text	this is darktes	{"content_typ		darktest :	2539 from m5", "hid	de_in_thread":false,'	'thread_key":	
7	1472	7526247127	28121397040	NULL	340	7575387	15244639882	text	nice to meet y	{"content_typ				17103009491281839 7"]},"seen count":	03867454875	","
3	1473	7526247127	28121397266	4fc0	340	7575387	15244640005	text	same here Asim	{"content_typ				lay_expiring_at_us":	0}	
,	1474	7526247127	28121397583	NULL	340	7575387	15244640176	text	lets try to pull	{"content_typ		Type of data	currently in cell: Text /	Numeric		
10	1475	7526247127	28121397699	6cf2	340	7575387	15244640239	text	ok	{"content_typ		2220 char(s)				Apply
1	1476	7526247127	28121398133	161c	340	7575387	15244640474	media_share	NULL	{"content_typ		Remote				
2	1477	7526247127	28121398322	03e	340	7575387	15244640577	media_share	NULL	{"content_typ		Identity	-			
3	1478	7526247127	28122685914	NULL	340	7575387	15245338582	raven_media	NULL	{"content_typ		Name		Last modified	Size	
4	1479	7526247127	28126056259	7a4	340	7575387	15247165649	text	dark say hello	{"content_typ		Name	Commit	Last modified	Size	
5	1480	7526247127	28126056451	945	340	7575387	15247165753	text	how are you	{"content_typ						
6	1481	7526247127	28126063065	146	340	7575387	15247169339	raven_media	NULL	{"content_typ						
7	1482	7526247127	28182014619	122	340	7575387	15277500737	text	hi	{"content_typ						
8	1483	7526247127	28182015465	475	340	7575387	15277501196	text	asim how are	{"content_typ						
9	1484	7526247127	NULL	00b	340	7575387	15277507734	text	pplease reply	{"content_typ						
20	1485	7526247127	28122828725		340	1068207	15245416000	text	Pleasure is mi	{"content typ						

Da	tabase Si	tructure Browse	Data Edit Pragmas	s Exe	cute SQL						E	Edit Database Ce	1				
Ta	ole:	messages					- 🔁 🈼			New Record D	elete Record	Mode: Text	•	Import E	xport Set		
Γ	_id	user_id	server_item_id	nt_item	read_	recipient_ids	timestamp	message_type	text	message	^			ZmY5YjA3NDA0ZmY0			
		Filter	Filter			Filter	Filter	Filter	Filter	Filter]	Dg1MDZ8MTc2Mjk1NDk1I	Tc2Mjk1NDk1M	r3MjY0MiIsInNlcnZlcl90b2tlbiI6IjE1Mjc3NTA3 1NDk1MDU0MjY3MjY0Mnw3NTI2MjQ3MTI3fD			
4	1469	7526247127	28121395519	NULL	340	7575387	15244639057	text	this is Asim	j				IWM0ZThjMWU5MWIwZTcxOTM2ZmVhZ wMTU2MWM2N2NmYjJkNjqifSwic2lnbmFi			
5	1470	7526247127	28121395961	519	340	7575387	15244639297	text	hi	{"content_typ		XJIIjoiIn0=	<pre>JIIjoiIn0=","store_map_zoom_level": ,"is_aymf_media":false,"is_ad4ad":false,"can_viewer_reshare rue}["text":""}, hide_in_thread":false,"thread_key":</pre>				
6	1471	7526247127	28121396521	8f7e	340	7575387	15244639600	text	this is darktes	{"content_typ							
7	1472	7526247127	28121397040	NULL	340	7575387	15244639882	text	nice to meet y	{"content_typ			_id":"340282366841710300949128183903867454875" t_ids":["7575387457"]},"seen_count":				
8	1473	7526247127	28121397266	4fc0	340	7575387	15244640005	text	same here Asim	{"content_typ				play_expiring_at_us"	:0}		
9	1474	7526247127	28121397583	NULL	340	7575387	15244640176	text	lets try to pull	{"content_typ		Type of data cu	rently in cell: Text	/ Numeric	_		
1	0 1475	7526247127	28121397699	6cf2	340	7575387	15244640239	text	ok	{"content_typ		4645 char(s)	renty in cent rent	, remaine			
1	1 1476	7526247127	28121398133	161c	340	7575387	15244640474	media_share	NULL	{"content_typ	B	Remote					
1	2 1477	7526247127	28121398322	03e	340	7575387	15244640577	media_share	NULL	{"content_typ		Identity	-				
1	3 1478	7526247127	28122685914	NULL	340	7575387	15245338582	raven_media	NULL	{"content_typ				lar mara	1.22		
1	4 1479	7526247127	28126056259	7a4	340	7575387	15247165649	text	dark say hello	{"content_typ		Name	Commit	Last modified	Size		
1	5 1480	7526247127	28126056451	945	340	7575387	15247165753	text	how are you	{"content_typ							
1	6 1481	7526247127	28126063065	146	340	7575387	15247169339	raven_media	NULL	{"content_typ							
1	7 1482	7526247127	28182014619	122	340	7575387	15277500737	text	hi	{"content_typ							
1	8 1483	7526247127	28182015465	475	340	7575387	15277501196	text	asim how are	{"content_typ							
1	9 1484	7526247127	NULL	00b	340	7575387	15277507734	text	pplease reply	{"content_typ							
2	0 1485	7526247127	28122828725	NULL	340	1068207	15245416000	text	Pleasure is mi	{"content_typ							
2	1 1486	7526247127	28113214462	cfe6	340	2327261	15240204097	text	this is test mon	("content him	~						

Figure 6: Messages table of Direct.db (b) Contents of message field messages table of Direct.db

This information is stored in JSON format. Contents of the message field for text message and visual (image or video) messages are listed in Fig. 7.

Once we had identified the information available in the different files and directories, we extracted this information in our digital forensics ready framework software. For this purpose, we created our local databases to store the forensics artifacts in. In the results section, we have shown how we extracted valuable information from different files to be stored permanently in our forensics workstation evidence database.

177

{"content_type":"TEXT","status":"UPLOADED","user":{"username":"dark2539","full_name":"DarkTest","profile_pic_url":"https://scontent-sit4-1.cdninstagram.com/vp/b38273e46 ^
"profile_pic_id":"1759213689388687079_7526247127","hd_profile_pic_url_info":{"url":"https://scontent-sit4-1.cdninstagram.com/vp/8ff5fbeb9abde2f5d483404c30eaba04/588CE9C
"width":1080,"height":1080,"type":0),"has_anonymous_profile_picture":false,"id":"7526247127","is_favorite":false,"is_profile_action_needed":false,"usertag_review_enable
"external_jux_url":"","follower_court":14, "following_court":18,"besties_court":0,"necently_bestied_by_court":0," media_court":18,"splay="back","aggregate_promc
"can_convert_to_business":true,"can_see_organic_insights":false,"is_unpublished":false,"allow_contacts_sync":true,"sonbusine
"can_link_entities_in_bio":true,"reel_auto_archive":"une:","medie":","modie":","selfie_sticker":0),"sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefore.","sieffacefo

	Figure 7	7:	JSON	Contents	of the	message	field
--	----------	----	------	----------	--------	---------	-------

Forensic Information	Key	Value
User information	User	<pre>"username":"dark2539", "full_name":"DarkTest", "profile_pic_url":"https://scontent-sit4-1.cdnInstagram.com/vp/ b38273e465e0bb2777b6845dea25cbd2/5BB665AE/t51.2885-19/ s150x150/30078315_2120078654888308_ 1563912706187067392_n.jpg, "profile_pic_id": "1759213689388687079_7526247127", "hd_profile_pic_url_info": {"url":"https://scontent-sit4-1.cdnInstagram.com/vp/ 8ff5fbeb9abde2f5d483404c30eaba04/5BBCE9D6/t51.2885-19/ 30078315_2120078654888308_1563912706187067392_n.jpg"</pre>
Instagram ID	Id	7526247127
Follower count	follower_count	14
	following_count	18
	besties_count	0
	Timestamp	1524463960077529
	Timestamp	1524463960077529
	Text	this is darktest 2539 from m5

Table 3: Important forensics artifacts available within in User Access Map tag

6 Forensics Analysis of IOS-Based Instagram Application

For the logical acquisition of the iPhone image from Apple devices, iTunes is the best authentic software available. Many research articles suggest and recommend the use of iTunes for the logical acquisition of Apple device contents; in their research, Bader et al. [15] described in detail how the logical acquisition of a device image using iTunes, with auto synchronization disabled, ensures that the acquired logical image of the device is forensically sound. Once the whole directory structure containing the data for forensics analysis was extracted into our forensics workstation, we started analyzing the contents of the directories and files manually to find the evidence we needed so we could code our framework. The purpose of this framework would be to extract similar forensics artifacts automatically later by just the click of a button.

7 Key Forensics Artifacts Identification of Instagram in IOS-based Device

Logical acquisition of an IOS device using the backup facility, provides a wealth of information for forensic analysis [16]. After being installed on the IOS device, the Instagram application creates the following directory structure as shown in Fig. 8. In the AppDomain directory of the iPhone, a directory with "com.burbn.Instagram" name is created to store the data of Instagram on the IOS device. We extracted

the entire directory structure from the IOS device using the iTunes backup facility [17]. After manual analysis of this directory structure, we noted that the database file containing direct messages exchanged between users of Instagram, was stored in the "AppDomain\com.burbn.Instagram\Library\Application Support \DirectSQLiteDatabase\7463799528.db" file. Another important fact we noted was that the direct messages file was named differently from that of the Android database file name counterpart. This number 7463799528 is the unique identifier for each Instagram user. As seen in the "com.instagram, android_preferences.xml" file of the Android Instagram version, this kind of unique identifier is being used to identify users of Instagram as well. Similarly in the IOS, the database file name is also assigned a congruent identifier to distinguish between the messages and databases of different users. We will now explain how we extracted the messages exchanged between users from this "7463799528.db" file.

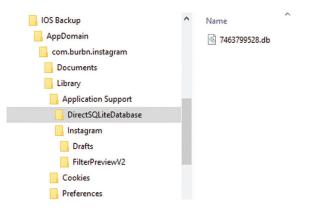


Figure 8: Directory structure of com.burbn.Instagram package extracted from Apple iPhone

The second important file that contains significant digital artifacts related to the Instagram user in IOS devices is the "com.burbn.Instagram.plist," which is located in the "AppDomain\com.burbn.Instagram \Library\Preferences" directory.

Fig. 9 reveals all the information stored within the "com.burbn.Instagram.plist" file. In this file we are only interested in the key data which can serve as evidence and give away the personal details of the mobile application's user (name, user name, phone number, email address).

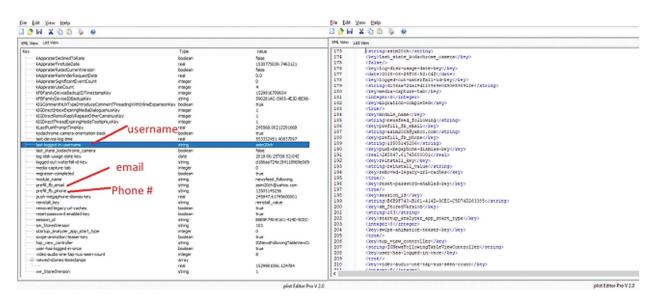


Figure 9: Key forensics artifacts stored in com.burbn.Instagram.plist

After we had identified the personal information associated with the user, we wanted to extract the personal messages exchanged through this device using the Instagram application. This information is available in the "7463799528.db" file. Analysis of this file showed that the archive column of the messages table contains important information stored in the Binary Large Object (BLOB) format.

In Fig. 10 you can see the contents of the database file and the contents of its corresponding messages table and "archive" column. It is important to note that pictures and video messages are not stored locally on the device, instead, if the user shared any visual message, only its URL would be stored in this local database file.

Data	abase Structure	Browse Data Edit	Pragmas Execut	e SOL			Edit Database Cell	
able	e: messages	•	8		New F	ecord Delete Record	Mode: Binary 🔻	Import Export Set as NU
	message_id	read_id_publishe	nread_id_pendin	archive	class_name	row_id		bpl i st <mark>00Ĉ</mark>
	Filter	Filter	Filter	Filter	Filter	Filter	0010 04 00 05 00 06 01 70 01 71 58 24 76 65 72 73 69 0020 6f 6e 58 24 6f 62 6a 65 63 74 73 59 24 61 72 63	p. qX\$versi onX\$objectsY\$arc
i l	28223138425	34028236684		BLOB	IGDirectText	192	0030 68 69 76 65 72 54 24 74 6f 70 12 00 01 86 a0 af	hi ver T\$t op
2	28223138789	34028236684		BLOB	IGDirectText	193	0040 10 30 00 07 00 08 00 17 00 21 00 27 00 28 00 29 0050 00 30 00 31 00 f9 00 fa 01 00 01 01 01 04 01 05	. 0 ! . ' . (.) . 0. 1. ù. ú
3	28223155246			BLOB	IGDirectVisual		0060 01 11 01 12 01 13 01 14 01 15 01 16 01 19 01 1a	
							0070 01 1b 01 1c 01 20 01 21 01 25 00 16 00 11 01 26 0080 01 3c 01 3d 01 42 01 4e 01 4f 01 50 01 51 01 52	!.%& . <. =. B. N. C. P. C. R
1	28223157092		NULL	BLOB	IGDirectPhoto	195		. <. =. B. N. U. P. Q. K . S. V. Z. ^. a. e. f. i
5	28223157558	34028236684	NULL	BLOB	IGDirectPhoto	196		. I U\$nul I ×
5	28239109944	34028236684	NULL	BLOB	IGDirectText	197	00b0 0c 00 0d 00 0e 00 0f 00 10 00 11 00 12 00 13 00 00c0 14 00 15 00 16 56 24 63 6c 61 73 73 5a 6c 69 6b	Vści assZlik
7	28226095927	34028236684	NULL	BLOB	IGDirectThrea	198	00d0 65 5f 63 6f 75 6e 74 59 72 65 61 63 74 69 6f 6e	e_count Yr eact i on
3	28223162439	34028236684	NULL	BLOB	IGDirectText	199		sTt ext Zupl oadI nf oXnet adat aYt ext
9	28223162577	34028236684		BLOB	IGDirectPhoto	200		type 1/ 1, 1. 1.
10	28223163216			BLOB	IGDirectText	201	0110 10 01 d5 00 18 00 19 00 1a 00 1b 00 09 00 1c 00 0120 1d 00 1e 00 1f 00 20 58 73 65 6e 74 44 61 74 65	ČXsent Dat e
-	28223163298		NULL	BLOB	IGDirectText	202		SkeyVsender Xt hr e
								adId ') 1. 1. 1. 1+ć.
	28223164184		NULL	BLOB	IGDirectVideo	203	0150 22 00 23 00 09 00 24 00 25 00 26 58 73 65 72 76 0160 65 72 49 64 58 63 6c 69 65 6e 74 49 64 80 04 80	". #\$. %. &Xserv erIdXclientId %. %
13	28223164352	34028236684	NULL	BLOB	IGDirectPhoto	204	0170 05 80 06 5f 10 23 32 38 32 32 33 31 33 38 37 38	. 1 #2822313878
14	28223166675	34028236684	NULL	BLOB	IGDirectText	205		9546310718945443 935420416 \$6010
15	28223167285	34028236684		BLOB	IGDirectText	206		CBCC- E263- 4F88- 9

Figure 10: Table relation in 7463799528.db

Upon analyzing the archive column's BLOB object, and contents of the binary data, it was found that this BLOB contains data in the bplist format as shown in Fig. 11. The first 6 offsets of every BLOB object is 62 72 6c 69 73 74 which corresponds to a bplist format of data.

																	bpl i st <mark>00Ĉ</mark>
0010	04	00	05	00	06	01	70	01	71	58	24	76	65	72	73	69	p. qX\$ver si
0020	6f	6e	58	24	6f	62	6a	65	63	74	73	59	24	61	72	63	onX\$obj ect sY\$ar c
0030	68	69	76	65	72	54	24	74	6f	70	12	00	01	86	a0	af	hi ver T\$t op
0040	10	30	00	07	00	08	00	17	00	21	00	27	00	28	00	29	.0!.'.(.)
0050	00	30	00	31	00	f9	00	fa	01	00	01	01	01	04	01	05	. 0. 1. ù. ú

Figure 11: Pblist Hexadecimal Header in BLOB object

In the next step, we exported the binary data into the bplist file. After this, we opened the bplist file with WinHex/X-Ways Forensics software, and were able to view the contents of a text message in hex file as shown in Fig. 12.

Further analysis of this binary data revealed the contents of text messages, which were found at offsets 11B3 to 11F7. For visual messages, the URL of the contents are also stored in a BLOB. This bplist file contains a directory named "root" which has four keys stored in its directory, namely: \$archiver, \$objects, \$top, and \$version (as shown in Fig. 13).

WinHex - [text2.plist]		- 🗆 X
📓 文件(E) 编辑(E) 搜索(S)) 导航(N) 查看(V) 工具(I) 专业工具(I) 选项(Q) 窗口(W) 帮助(H)	18.5 SR-3 x64 🚊 🖅
 保件数据	📄 🗅 🥶 🔄 🖾 🛎 🔚 🗠 🖻 🕲 🛱 🛯 🏘 🌺 🖊 🍇 🍇 🔛 🗎 → 🕀 🔶 🍩 🛲 🔎 🦈	4)*
文件(L) 编辑(D)	text2.plist	
		text2.plist
		text2.plist C:\Users\MuhammadAsim\Desi
	000010E0 5F 73 74 69 63 6B 65 72 62 D8 3D DE 00 D2 00 2A stickerb@=> 0 *	C. (Osers (MunanimadAsim (Des
	000010F0 00 2B 01 54 01 55 5F 10 13 4E 53 4D 75 74 61 62 + T U_ NSMutab	文件大小: 5.4 k
	00001100 6C 65 44 69 63 74 69 6F 6E 61 72 79 A3 01 54 01 leDictionaryf T	5,480 字=
	00001110 17 00 2F D2 00 2A 00 2B 01 57 01 58 56 49 47 55 /Ò * + W XVIGU	缺省编辑模式
		状态: 原如
	00001130 74 6F 72 61 62 6C 65 4F 62 6A 65 63 74 D2 01 5B torableObject0 [+	撤消级别:
		反向撤消: 不可
	00001150 C0 70 EF 63 D8 AD 26 80 2A D2 00 2A 00 2B 01 5F ÅpïcØ-& *Ò * +	
	00001160 01 60 56 4E 53 44 61 74 65 A2 01 5F 00 2F D2 00 `VNSDate¢ _ /Ò	创建时间: 11/29/20
	00001170 2A 00 2B 01 62 01 63 5F 10 17 49 47 44 69 72 65 * + b c_ IGDire	14:37:2
	00001180 63 74 4D 65 73 73 61 67 65 4D 65 74 61 64 61 74 ctMessageMetadat	最后写入时间: 11/29/20
	00001190 61 A2 01 64 00 2F 5F 10 17 49 47 44 69 72 65 63 a¢ d /_ IGDirec	14:37:
		属性:
		图标:
	000011C0 6F 72 20 6D 75 62 61 72 69 6B 5F 6D 75 68 61 6D or mubarik_muham	模式: 文:
		字符集: ANSI ASG
		偏移量: 十六进
		每页面字节数: 27x16=4
		当前窗口号:
		窗口总数:
	00001220 6D 01 6E 5C 49 47 44 69 72 65 63 74 54 65 78 74 m n\IGDirectText	剪贴板: 可,
	00001230 A2 01 6r 00 2r 5C 49 47 44 69 72 65 65 74 54 65 C 6 / (GDIFectie	
		临时文件夹: 103 GB 空; JHAMM~1\AppData\Local\Tem
	00001250 69 76 65 72 D1 01 72 01 73 54 72 6F 6F 74 80 01 iverÑ r sTroot	repputate Cocal Ten

Figure 12: The text message is visible along with related metadata in the BLOB object of the archive column (in the messages table)

<u>ile E</u> dit <u>V</u> iew <u>H</u> elp		
🗋 🤔 🖩 👗 🚡 🚳	0	
XML View List View		
Кеу	Туре	Value
Root	dict	
\$archiver	string	NSKeyedArchive
\$objects	array	
\$top	dict	
\$version	integer	100000

Figure 13: Bplist list view

\$Objects has sub directories which store the text messages exchanged within them, or the URL of picture messages exchanged between different users. Fig. 14 shows the hierarchy of directories in this bplist, \$Objects are stored as BLOB objects in the archive column, of the messages table, in the user direct messages database file. In the next section, we will discuss the forensics results and data reconstructed from these files.

8 Evidence Retrieval, Plugin Implementation, and Results

As elaborated in previous sections, we successfully identified and located the valuable information which could serve as potential evidence. In this section, we will now present the extracted evidence from the Instagram mobile application. Instagram forensics was implemented in our forensics-ready framework as a dynamic link library. A unique identifier was created, and in this folder, forensics data of each application was saved into a separate folder (named according to the corresponding mobile application). In case of Instagram, the IMINSTAGRAM directory stored a copy of original files extracted from the Instagram app. Briefly explaining how the forensics software operated; it first extracted, and made a copy of the files containing digital forensics artifacts ("com.instagram.android," "direct.db" and "burbn.Instagram. plist," "7463799528.db") for both the IOS and Android devices respectively, into the IMInstagram directory, as shown in Fig. 15. Then, the software created a results folder, in which it stored a database file, containing the reports of all relevant information and direct messages exchanged between the current user and other Instagram users. Figs. 16 and 17 exhibit the extraction of the currently logged in user's information, and messages exchanged between this user and different users within this application.

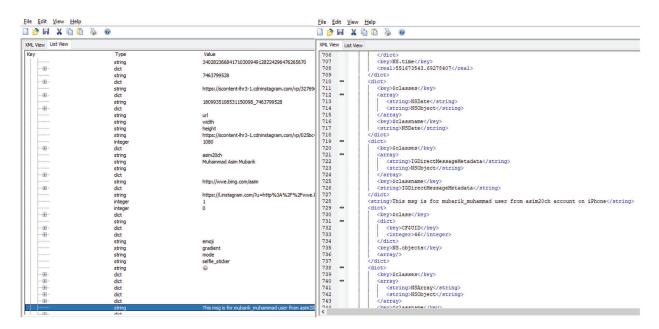


Figure 14: List view & XML view of the bplist extracted from the BLOB object of the archive column in the messages table

- → × ↑ 📙 « f71a048815588c2	838fd061ead5f04bb1544417437448.4 > IMInstagram	~ ē	Search IMInstag	ram 🔎
 	▲ Name ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲ ▲	D 1	Search INIInstag Nate modified 2/10/2018 1:30 PM /24/2018 10:27 PM	ram p Type Data Base File Property List File
 a adccPlcutes b case20181103-1416 njh nmwhfy Not Used Offten 				

Figure 15: Forensics workstation evidence files local copy

R Ste) 取正先結一综合手机取证 _ の Start Case 201901116160132 - Android Report Global Search Tool エーター							
	账号信息 浏览历史	2 搜索历史 聊天信息 位置信息	共26条记录		输入关键字之后自动搜索 Q ▼ ▶ 导出			
	编号	好友名称	账号名称	发送时间	发送内容			
	1	1758953528	7526247127	2019-01-16 02:32:39	hi brother.bahira this is Dark test i am running a tear on Instagram plea.			
	2	1758953528	7526247127	2019-01-16 02:32:42	dark test and team			
	3	1758953528	7526247127	2019-01-16 02:32:54	ansari kesa hai tu			
	4	1758953528	7526247127	2019-01-16 15:58:10	asim sb			
	5	2122550024	7526247127	2019-01-16 12:10:41	Who?			
	6	6215026012	7526247127 2019-01-16 02:58:46					
	7	6215026012	7526247127 2019-01-16 10:40:29		Hey			
	8	4581732344	7526247127	2019-01-16 03:03:44	hi waseem how u doing this.morning?			
	9	4581732344	7526247127	2019-01-16 03:22:05	I'm fine			
	10	4581732344	7526247127	2019-01-16 03:22:13	Who are you Sorry			
	11	1429506193	7526247127	2019-01-16 03:14:16	saim sb kaisay hain. ap?			
	12	8604168394	7526247127	2019-01-16 03:13:29	shahrad kider ho. ajkal regds Darktest			
	13	8646329902	7526247127	2019-01-16 03:13:01	hamza kider hoaj kal?			
	14	1068207656, 1345517946, 1758953528, 1769089714, 2030089807…	7526247127	2019-01-16 03:12:53	Midark2539who is this ?			
	15	7600728161	7526247127	2019-01-16 03:10:55				
F	16	637E10E64E	7596947197	2019-01-16 03-10-19	Farar has no un this is don't toot			
nsnj	C:\Vsers\MuhammadAsim\D	esktop\New folder (3)\nsmj			Case resources分析进度: 100% 2019/01/17 07:15:22			

Figure 16: Direct messages reconstructed from Android application

	Start Case 20190116160192 - Android Report	取证先键一综合手机取证 Global Search Tool		- & × & * ?
Junctified OPCODE DESTRICT DESTRI				
	账号信息 浏览历史 搜索历史 聊天信	息 位置信息 共1条记录	输入关键字之后自动搜索	Q - 📑 号出
2019-01-10 16:00:35 dw1/2539	□ 最后登录时间 账户名称	眼眼称		
	2019-01-16 16:00:35 dark2539	DarkTest		<u>^</u>
New Case C:\Users)Muhammaddsim\Desktop\New folder (3)\New Case 2019/01/17 05:54:31	New Case C - Marca Markawa Ja - 10, a kawa New Ed Jac (2010) and		Concernance of HERITY 1000	2019/01/12 DE-EF-11

Figure 17: User information retrieved from Meizu Android-based application

9 Instagram Anti-Forensics on Android

In IOS-based devices, it is not possible to perform any anti-forensics techniques. However, for Androidbased devices, if suspected culprits are aware of currently employed digital forensics techniques that could get them convicted, the culprits can potentially get away with their criminal activities by simply corrupting the evidence in the application directory. This can be done easily by simply extracting the database file located in the "\com.instagram.android \databases\direct.db" path of Instagram, and alter its contents, then subsequently save them back into the directory.

To confirm the presented hypothesis, we simulated this scenario, and attempted to change the contents of the text messages placed in "direct.db," and save this direct.db file onto the following path "\com.instagram. android\databases\direct.db."

Here are the details of this experiment.

We sent a message from the Android-based Meizu phone hosting "Dark2539" user to "Mubarik Muhammad" Instagram user on the iPhone7 device.

Message: "I am DarkTest from mezu"

This message is located in "\com.instagram.android\databases\direct.db" file.

We extracted the "direct.db" file, opened it in the database software, and changed the contents of the message to

"I am wangle from Meizu"

In the next step, we restored this altered 'direct.db' file into the phone.

The file was successfully restored on the Android phone.

Next, we connected the device to the internet, went online, and opened the Instagram Application, to check if our altered text is visible in the application as well.

As expected Instagram displayed the original text, "I am DarkTest from Meizu."

This shows that Instagram maintains a copy of chat on their own servers, and updates the data when users appear online.

We concluded that a criminal may be smart enough to leave his/her phone offline. However another reason that invalidates this anti-forensics technique, the "direct.db" file also displays the date it was modified alongside it, so if the data is changed, it will be quite evident to a forensics analyst.

10 Privacy Issues of Instagram

During the forensics investigation of both the Android and IOS versions of the Instagram Application, we have found a serious privacy issue regarding user's multimedia content stored on the server-side. As in the previous sections, we have explained that pictures and videos shared by Instagram users are not stored in the local directory structure of the application, instead, the messages table of the database file stores only the URL link of the multimedia messages. A person with very little knowledge of digital forensics can extract this URL of multimedia messages and have access to a user's images and videos directly, using any web browser, and without having to verify or input any critical credentials (username or passwords). To test this, we experimented. The user Dark2539 shared a few pictures with the Mubarik_Muhammad account and we extracted the URL of these pictures from the messages table of the database file. All the URLs were available, so we accessed the URL in the Chrome web browser and successfully retrieved all the pictures.

This depicts that the invalidity of user privacy on Instagram; anyone can have access to your personal multimedia information if they has access to these URLs. When Dark2539 shared a picture of the Shanghai River with the Mubarik_Muhammad account, we extracted the URL from the messages table of the database and accessed the image in the Chrome browser. We successfully retrieved the image without inputting any user verification information. However (over time), as we were developing this plugin and writing this article, Instagram updated their application and the URLs extracted from the messages database table do

not work anymore. We noticed a visible change in both URLs saved in an older version and a newer version of Instagram.

URL of Pictures shared in the old version.

https://scontent-sit4-1.cdnInstagram.com/vp/8e2ae39215d58a78971a0fa373565e5d/5BB77629/ t51.2885-19/s150x150/35509004 417214398762526 2675262676875083776 n.jpg

URL of Pictures shared in New Version.

https://scontent-lhr3-1.cdnInstagram.com/vp/32769d90c55fe39e08da1cd97c5deb98/5BDF0329/ t51.2885-19/s150x150/35509004_417214398762526_2675262676875083776_n.jpg? efg=eyJ1cmxnZW4iOiJ1cmxnZW5fZnJvbV9pZyJ9

Old URLs were un-signed but now Instagram is using signed URLs for their pictures in the new version. URL signing is a way to control time-limited access to HTTP resources which are the pictures in our experiment. In the new URLs, Instagram has added a URL parameter 'efg' which has an encrypted value; this acts as a URL signature. When we open the image URL, the Instagram server decrypts the signatures, and decides whether it is expired or not, based on the timestamp used while creating the URLs originally. Once the URL has expired, the Instagram app receives a new URL with updated signatures while the app is authenticated (logged in). If we try to run an expired URL in the Chrome browser (outside Instagram app scope), the Instagram server will not send us a new URL because we're not logged in. This resolves the privacy issues we found in the old version of Instagram.

11 Conclusion

This was a great learning experience on how Instagram organizes their application on both Android and IOS based devices. As technology keeps enhancing, Instagram also has updated its application over time. We intend to add versioning in our Instagram forensics plugin, so it keeps track of changes and keeps extracting data from all the versions of the Instagram application. In conclusion, I would like to extend my gratitude to "Chen Star Electronic Data Forensic Research Center" Ministry of Public Security Beijing China for their help and guidance in conducting this research.

Acknowledgement: This research article is part of my Ph.D. work by the Chinese scholarship council (CSC) 'www.csc.edu.cn' conducted under the collaboration between "College of Information and Computer Science Hohai University Nanjing China" And "Chen Star Electronic Data Forensic Research Center" Ministry of Public Security Beijing China.

Funding Statement: This research was supported by the Korea Institute for Advancement of Technology (KIAT) Grant Funded by the Korea Government (MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] W. McIver, W. Birdsall and M. Rasmussen, "The Internet and the right to communicate," *First Monday*, vol. 8, no. 12, 2003. [Online]. Available: https://www.firstmonday.org/article/view/1102/1022/.
- [2] P. Kallas, "Top 15 most Popular Social Networking Sites and Apps," DreamGrow, 2018. [Online]. Available: https://www.dreamgrow.com/top-15-most-popular-social-networking-sites/.
- [3] Anonymous, "Instagram active users," Statista, 2018. [Online]. Available: https://www.statista.com/statistics/ 253577/number-of-monthly-active-instagram-users/.

- [4] J. M. Olsen, "Norwegian billionaire's wife kidnapped, ransom demanded: police," HuffPost, 2019. [Online]. Available: https://www.huffingtonpost.com/entry/norway-businessman-s-wife-abducted-ransom-demanded_us_ 5c3629bde4b00c33ab5f12d9.
- [5] S. G. Punja and R. P. Mislan, "Mobile device analysis," Small Scale Digit Device Forensics Journal, 2008. [Online]. Available: https://pdfs.semanticscholar.org/279c/06abd3861704ed20883fbd49b7666b00113b.pdf.
- [6] Y. Fedotov, "The drug problem and organized crime, illicit financial flows, corruption, and terrorism 5 world drug report," 2017. [Online]. Available: https://www.unodc.org/wdr2017/field/Booklet 5 NEXUS.pdf.
- [7] I. Baggili, "Cyber forensics team launches digital forensics database," University of New Haven, 2019.
 [Online]. Available: http://www.newhaven.edu/news/releases/2017/cyber-forensics-team-launches-digital-forensics-database.php.
- [8] R. Al Mushcab and P. Gladyshev, "Forensic analysis of Instagram and path on an iPhone 5s mobile device," in IEEE Symposium Computers and Communication ISCC, pp. 146–151, 2015.
- [9] S. C. Ming, "Chang Evidence Gathering of Instagram on Windows 10," *International Journal of Innovative Science, Engineering & Technology*, vol. 3, no. 10, pp. 2348–7968, 2016.
- [10] K. Wong, A. C. T. Lai, J. C. K. Yeung, W. L. Lee and P. H. Chan, "Facebook forensics," 5, Valkyrie-X Secur. Res. Group, 2011. [Online]. Available: https://www.fbiic.gov/public/2011/jul/Facebook Forensics-Finalized.pdf.
- [11] M. N. Yusoff, A. Dehghantanha and R. Mahmod, "Forensic investigation of social media and instant messaging services in Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp, and Line as case studies," *Contemporary Digital Forensic Investigations Of Cloud And Mobile Applications Elsevier*, pp. 41–62, 2017.
- [12] D. Walnycky, I. Baggili, A. Marrington, J. Moore and F. Breitinger, "Network and device forensic analysis of android social-messaging applications," *Digital Investigation*, vol. 14, pp. S77–S84, 2015.
- [13] N. Al Mutawa, I. Baggili and A. Marrington, "Forensic analysis of social networking applications on mobile devices," *Digital Investigation*, vol. 9, pp. S24–S33, 2012.
- [14] N. Al Mutawa, I. Al Awadhi, I. Baggili and A. Marrington, "Forensic artifacts of Facebook's instant messaging service," in *Int. Conf. for Internet Technology and Secured Transactions*, pp. 771–776, 2011.
- [15] M. Bader and I. Baggili, "iPhone 3GS forensics: Logical analysis using apple iTunes backup utility," Small Scale Digital Device Forensics Journal, vol. 4, no. 1, pp. 1–15, 2010.
- [16] C. Carpene, "Looking to iPhone backup files for evidence extraction," in *Proc. of the 9th Australian Digital Forensics Conf.*, 2011.
- [17] Anonymous, "Locate backups of your iPhone, iPad, and iPod touch—Apple Support," 2019. [Online]. Available: https://support.apple.com/en-us/HT204215.