Arghadeep Lahiri/MSc CyberSecurity/  
Digital Forensic Analysis of Smartphone Application - Facebook

**Executive Summary:**In the era of digital advancement, the use of social media is a very popular choice among humans in their everyday lives, making this is very easy target for social engineering attacks, extracting user data pertaining to username, email address, phone number, social security numbers and several other personal information. One such mobile application, popular these days is Facebook founded in 2004 by Mark Zuckerberg, Eduardo Saverin, Andrew McCollum, Dustin Moskovitz, and Chris Hughes [1]. The mobile application was launched in 2007, making it widely available to users with smartphones, has become a lucrative source for cyber bullying, data theft and copyright infringement. Thus, the need of digital forensics for Facebook application is in demand in extracting extensive amount of data essential in mitigating cybercrimes, data theft and unauthorized data usage. This report focuses on key findings like **extracted images, conversation threads, username, Friend list, Activities of the user gathered from the database** of a test Facebook account deployed on a virtual emulator.

**Keywords**: Facebook forensics, social network applications forensics, mobile device investigation, Android Investigation.

**Methodology:**

**Literature review**Previous investigation [2] performed by several forensic analysts revealed that a substantial amount of data is left behind including photos, videos, chat message, audio messages exchanged between parties, can be reconstructed as produced as evidence.   
This investigation focuses on relevant artefacts gathered from an android free BSD version emulated virtual machine, spun using VMware workstation Pro for Windows. It predominantly focuses on capturing images, message threads, friend list, phone numbers, username and password of the account used as a test. As cited previously,[3], there are three methods of data acquisition: manual, logical and physical acquisitions. The manual acquisition focuses on gathering data via copying data from a smartphone and storing in a secure environment for analysis. As for logical acquisition, Casey[4] describes it as a bitwise copy of files and directories that contains timestamp and location of resident files within the filesystem of the device. For physical acquisition, it mainly involves recovering an internal or memory card storage, that potentially contains deleted data. Sangju Jeon[5] in their research talk about the opensource database analyser tool SQLite which can be used to easily dissect database artifacts and gather information such as friend list, message thread, images, conversation threads. Simultaneously, it is also capable of reading deleted data which is stored in the cache can be recovered to be used as evidence. In a study, [6] tools like, Oxygen Forensic, Paraben E3:DS, Axiom Magnet were employed for manual extraction of forensic data retrieval, however in this research, the usage of tools is limited to Android Debug Bridge (ADB), APKtool, Jadx, Autopsy, Wireshark deployed in a Virtual Environment (VMware). The machines comprises of an android free Android BSD machine, Kali Linux and Windows Server downloaded from osboxes.org as vmdk images and directly deployed.  
A comprehensive analysis of a test(fake) Facebook account is conducted to mine data in the Facebook database, Android manifest.xml file, and other relevant location from where images, message conversation thread is identified and put up as evidence.

**Details of the app files and artefacts gathered from the device:**

Files and folder locations that were explored:  
From the android device using adb shell, the following directories is explored. Primary directory -- **/data/data/com.facebook.katana**  
Under this directory we have the following structure:

**/data/data/com.facebook.katana   
├── app\_\* (analytics, components, bootstrap, call\_stats, ce, gatekeepers, uploads, webview, etc.)   
├── cache, code\_cache, crash\_lock, crash\_log   
├── databases, dex, files, flags   
├── insta\_crash\_log, lib-compressed, modules   
├── native\_deps, no\_backup, shared\_prefs**Dissecting the folder structure, evidence of **images, messages, Friend list, User ID, Username details, Potential IP address, Location of downloaded data** were found using Autopsy tool. The following location are interesting which contains relevant user data:   
**Images**: com.facebook.katana\cache\image\_scoped\100010618894220  
**Friends list**: com.facebook.katana\databases\ ssus.100010618894220.android\_facebook\_contacts\_db  
**Messages:** com.facebook.katana\app\_mib\_msys\v2\100010618894220\ msys\_database\_100010618894220  
**Username Details:** com.facebook.katana\app\_graph\_service\_cache\100010618894220\ G1%3a40549644413844672135088385235%3a13219296739839175214  
**UserID:** com.facebook.katana\app\_mib\_msys\v2\100010618894220\ crypto\_db\_100010618894220  
**List of phone number:** com.facebook.katana\app\_graph\_service\_cache\100010618894220\ G1%3a1053734612793240184803451798%3a4758716765201137930  
**List of IP’s**: com.facebook.katana\app\_graph\_service\_cache\100010618894220\ G1%3a1053734612793240184803451798%3a4758716765201137930  
  
The analysis reveals interesting facts like **message conversation** between friends, **Image shared on the profile**, which is discussed later in the **App Investigation and Findings** section.

**Forensic analysis method and guideline:**Following the Digital Forensics and Incident Response Framework as mentioned in NIST, this investigation will be monitored, which covers: Collection of Data, Analysing the Data, Interpretation of data to support Investigation related to cyber incidents, legal implication, Documentation of the evidence gathered[7].Following this framework[8],aligning with ACPO principle[9]; ADB tool is used initially, ensuring the Virtual Android Free BSD version has USB debugging and root access enabled. Using ABD tool in Kali Linux a shell connection is created, and Facebook database is pulled locally to the Kali for analysis. The analysis involves dissecting the database; to understand it’s structure and possible location of data residence. Following which APKtool is used to decompile the Facebook application to convert the software codes to human friendly readable format. This method reveals the smali code used to construct the architecture of the application, along with the AndroidManifest.xml file which tells us about the permission user has inside the application and the resources and assets it contains. This method is followed by converting the Dex files obtained using the APKtool conversion, to JAR files for better readability. Also using Autopsy tool, the database and application infrastructure is explored to find relevant sources and locations for evidence. Finally, using a network capture tool like Wireshark the network traffic is investigated to understand the dynamic nature of the application and its connection preferences, ports, protocols used.

**Description of the test environment:**The following environment is used for study and analysis of the application

|  |  |
| --- | --- |
| **Software** | **Usage** |
| Virtualization interface – Vmware workstation Pro 17 | To setup up virtual Operating Systems. |
| Kali Linux | VM setup for using ADB, Apktool, Sqlite |
| Android Free BSD 9 version – Oreo | Setting up test Facebook account |
| Windows Server 2025 | Using Autopsy tool – Facebook file analysis |
| **Tools** | **Usage** |
| Android Debug Bridge | Establish shell with Android emulator and transfer data |
| APKTool | Decompile Facebook APK file |
| JadX | Converting hex code to JAR file |
| SQLITE Browser | Analyse Facebook database |
| Autopsy tool | Analyse file structure and finding artefacts inside databases |
| Wireshark | Real time network packet analysis when Facebook connecting to internet |

**App Findings and Investigation:**  
After creating a test facebook account, several activities, like messaging, sharing and uploading pictures in the Facebook profile is performed, including saving videos and reels. Later the facebook database is pulled from the Android emulator into Windows Server and Kali Linux machine. The following command was used to pull data from android: Using a Kali CLI command- **adb pull /data/data/com.facebook.katana** **/home/argha/ForensisAnalysis** the entire data was pulled. Additionally the Facebook base.apk was also pulled for decomplication. Command used is: **apktool d /home/argha/ForensicAnalysis/base.apk -o /home/argha/ForensicAnalysis/FacebookDecompiled**  
  
This creates the following folder structure :  
/home/argha/ForensicAnalysis/FacebookDecompiled   
├── AndroidManifest.xml, apktool.yml   
├── assets, kotlin, lib, META-INF, original, res, smali, smali\_assets, unknown  
The interesting file and folder to investigate are **AndroidManifest.xml** and **Smali** folder  
  
**Smali** folder reveals the coding logic behind the application.   
Doing a simple grep search using Kali Linux *grep -Ri 'http\|https'* will show the backend server and network calls the app is performing. Example output : ***X/000.smali: const-string p0,*** [***http://schemas.android.com/apk/res/android***](http://schemas.android.com/apk/res/android)Also performing another grep search : grep -Ri ‘token\|authentication\secret’ can reveal hardcoded authentication data which can be analysed later to find password of an account.  
From the android.manifest.xml file in the root directory of FacebookDecompiled code, permissions of user, services are revealed. For example:   
  
**AndroidManifest.xml: <uses-permission android:name="android.permission.ACCESS\_MEDIA\_LOCATION"/>  
AndroidManifest.xml: <uses-permission android:name="android.permission.READ\_MEDIA\_VISUAL\_USER\_SELECTED"/>  
AndroidManifest.xml: <uses-permission android:name="android.permission.READ\_MEDIA\_VIDEO"/>  
AndroidManifest.xml: <uses-permission android:name="android.permission.READ\_MEDIA\_IMAGES"/>  
AndroidManifest.xml: <uses-permission-sdk-23**Example of service permission:  
**AndroidManifest.xml:   
<provider android:authorities="com.facebook.katana.ClientMessagePushDedupInfoProvider" android:enabled="false" android:exported="true" android:name="com.facebook.messaging.push.dedup.provider.ClientMessagePushDedupInfoProvider" android:permission="com.facebook.permission.prod.FB\_APP\_COMMUNICATION"/>**

Additionally using a Jadx viewer the source code and signature can be verified.   
See **Appendix A** for the APK signature verification result.  
Now focusing on the database and file structure **/data/data/com.facebook.katana** reveals extraordinary evidence. Locations of the artefacts are mentioned under section **Details of the app files and artefacts gathered from the device:** Correlating the directory location and the image as mentioned in appendix will give a clear idea of the evidence gathered.

1. **Images**: Check **Appendix B and C** for image  
2. **Friends list**: Check **Appendix D** for image  
3. **Messages:** See **Appendix E** for image   
4. **Username Details:** See **Appendix F** for image  
5. **UserID:** See **Appendix G** for image  
6. **List of phone number:** See **Appendix H** for image  
7. **List of IP’s**: See **Appendix I** for image

Additional 4 images were retrieved from the location com.facebook.katana/app\_uploads/af345691-eb04-4628-a6d9-a27486171dce/ are also attached as evidence along with the file name in the Appendix **J, K, L, M**

**Application architecture:**  
The Facebook application consists of 2 vital infrastructure – The backend and the Frontend.   
The backend consists of Data storage, Cache, RealTime Systems, DataCenter Infrasturcture, Machine Learning, Security/Privacy server.  
Here is a simple representation of the architecture represented below:  
Using the help of online websites available for free the below flowchart was created.

A diagram of a network

AI-generated content may be incorrect.

**App Behaviour (Dynamic Analysis using Wireshark):**  
  
It is evident the Android VM 192.168.66.133 is attempting a connection to graph.facebook.com using TLVv1.3 on port 445 to destination address 31.13.73.35  
**554 58.324640 192.168.66.133 31.13.73.35 TCP 74 50558 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK\_PERM TSval=3303720657 TSecr=0 WS=128  
561 58.352067 192.168.66.133 31.13.73.35 TLSv1.3 456 Client Hello (SNI=b-graph.facebook.com)**

The server sends a response back to the client and server hello message is sent back, denoting a successful communication between the client and server   
**584 58.411027 31.13.73.35 192.168.66.133 TLSv1.3 279 Server Hello, Change Cipher Spec, Application Data**

Another example would be for lookaside.facebook.com SNI request  
Using QUIC protocol the android VM tries to connect to lookaside.facebook.com via IP address 31.13.73.22

**1699 65.648280 192.168.66.133 31.13.73.22 QUIC 90 Protected Payload (KP0), DCID=9724001b4fc9fdbd**  
Client Hello initiated   
**1942 65.709865 192.168.66.133 31.13.73.22 TLSv1.3 458 Client Hello (SNI=lookaside.facebook.com)**  
Server hello completed  
**1958 65.729046 31.13.73.22 192.168.66.133 TLSv1.3 266 Server Hello, Change Cipher Spec, Application Data**

**Recommendation:**From an organisation perspective, storing vital data on social media applications can be detrimental to an industry data integrity as it can cause data breach. If someone is suspected of using social media applications such as facebook to share organisation data outside a company, the disgruntled employee’s database could be used to gain insights on substantial data shared outside the organisation, thus alerting the security team to secure data and avoid mass leakage. Hence, creating stringent security polices for using social media apps in secured environment must be in place, in order for restrict data transfer from official devices to personal social media accounts.

**Conclusion**: From the above analysis we can derive the following:

* Using Facebook database, gathered from a physical or virtual image, critical evidence such as Phone numbers, email addresses, images, videos, username, authentication token can be retrieved.
* Using these data tracing back to a cybercrime evidence could be easy in order to incriminate someone or can be presented in a court of law.
* Even if Facebook claims, the data transferred in encrypted, however from the above analysis, it can be said that it is partially true as **message conversation** can be seen in plain text.
* From the database analysed, **usernames** are also showing in plain text, which means, there is a possibility of enumeration and identity theft.
* Privately shared images could also easily be identified, which can again, hinder an individual’s privacy, leading to anonymous blackmail.
* If there was more time to investigate, there was a possibility of decrypting the authentication token, to look for hidden password and in some cases, credit card details associated with an account.
* Tools like Volatile could have been used to analyse RAM data, which could reveal login information stored in the cache memory.
* Lastly, if there were more time and resources, industry standard tools like, Axiom magnet, Oxygen Forensics tool could have been used to simplify the investigation and dive deeper into each database segment.

References:

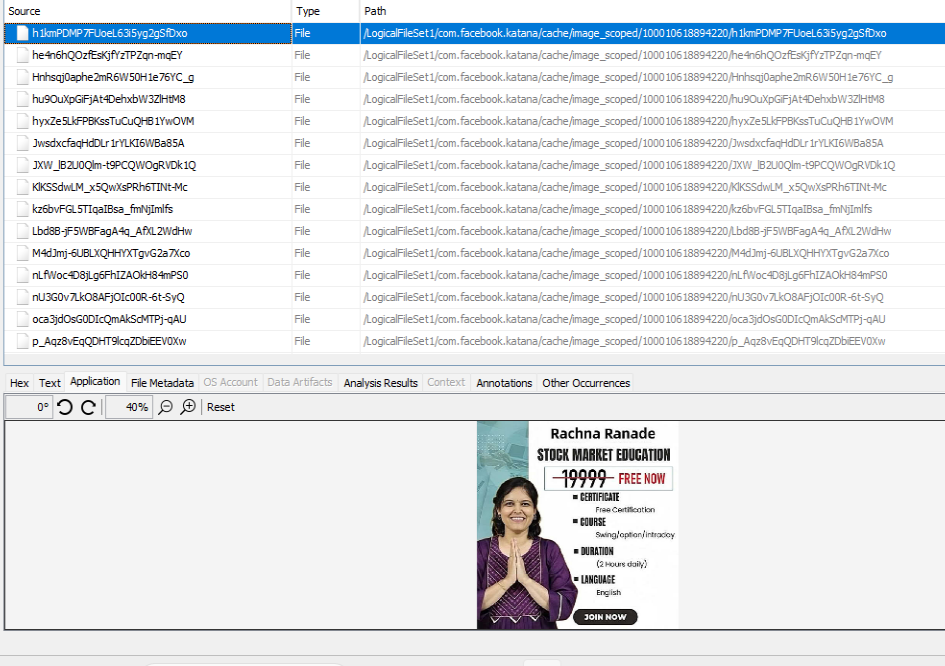
1. “Britannica money,” Feb. 26, 2025. https://www.britannica.com/money/Facebook
2. Daniel Walnycky et al., "Network and device forensic analysis of Androidsocial-messaging applications", Digital Investigation, vol. 14, pp. S77-S84, 2015.
3. F. N. Dezfouli, A. Dehghantanha, B. Eterovic-Soric, and K.-K. R. Choo, “Investigating Social Networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms,” Australian Journal of Forensic Sciences, vol. 48, no. 4, pp. 469–488, Aug. 2015, doi: 10.1080/00450618.2015.1066854.
4. Casey E. Digital evidence and computer crime: forensic science, computers and the Internet.  
   Baltimore: Academic Press; 2011.
5. Sangjun Jeon et al., A recovery method of deleted record for SQLite database, Springer-Verlag London Limited, 2011.
6. Ç. Hüseyin and K. Hatice, “Analysis and comparison of social media applications using forensic software on mobile devices,” Journal of Forensic Science and Research, vol. 8, no. 1, pp. 058–063, Oct. 2024, doi: 10.29328/journal.jfsr.1001065.
7. K. Kent, S. Chevalier, T. Grance, H. Dang, National Institute of Standards and Technology, and Booz Allen Hamilton, Guide to Integrating Forensic Techniques into Incident Response. National Institute of Standards and Technology, 2006. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-86.pdf
8. Digital Forensic Research Workshop (DFRWS), Utica, New York:A Road Map for Digital Forensic Research, August 2001.
9. "ACPO Good Practice Guide for Digital Evidence", Available, 2012, [online] Available: http://www.acpo.police.uk/documents/crime/2011/201110-cba-digital-evidence-v5.pdf

Appendix:  
A - A screenshot of a computer error

AI-generated content may be incorrect.

B- A screenshot of a computer

AI-generated content may be incorrect.

C- 

D- A screenshot of a computer

AI-generated content may be incorrect.

E- A screenshot of a computer

AI-generated content may be incorrect.

F- A screenshot of a computer

AI-generated content may be incorrect.

G - A screenshot of a computer

AI-generated content may be incorrect.

H- A screenshot of a computer

AI-generated content may be incorrect.

I - A screenshot of a computer

AI-generated content may be incorrect.

J - (**4ebbc4fa-5eb35afa2f0fa19f.tmp)**

K - A person with long hair and a black shirt

AI-generated content may be incorrect.(**5c270b7f-8798c896df42863a.tmp**)

L - (**223d03bb-96fb17103d0c4b20.tmp**)

M - (**f1ecbf7a-f9924f0628027935.tmp**)