iJRASET
International Journal For Research in Applied Science and Engineering Technology



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 11 **Issue:** IV **Month of publication:** April 2023

**DOI:** https://doi.org/10.22214/ijraset.2023.49982

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 11 Issue IV Apr 2023- Available at www.ijraset.com*

# Setting the Tone for Sound Forensic Investigations on Android-Based Social Media Platforms

Austin Makate[1], Martin Muduva[2], Naveen Kumar Chaudhary[3], Ronald Chiwariro[4]
[1, 3]*National Forensic Science University, India*
[2]*Zimbabwe National Defence University, Zimbabwe*
[4]*Jain University, India*

***Abstract:*** ***Android Social Media Applications have become a yardstick in facilitating a platform for human socialization on cyber space. They are an inevitable alternative, which is fast replacing most traditional ways that lacks full multimedia interaction adored by many. These applications are of forensic value as they account for most activities helpful in either incriminating or exonerating suspects in cases of adverse events. By default, most social applications store activity data in specific directories they create at the background of the hosting Android devices. Through expertise, this data can be extracted and analyzed to come up with meaningful insights useful in an inquiry of digital evidence interest. This study focused on forensics of Twitter and Clubhouse android based social media applications. The approach taken was to install these applications on emerging Android devices using the Samsung Galaxy S20+ (SMGS20+) and Samsung Galaxy Tab A7 (SMGTA7), populate known test data, perform data acquisition, execute data analysis noting results and then do a comparative analysis of tools and techniques utilized towards provisioning alternative solutions.***
***Keywords: Social Media Application, Android Forensics, Open Source Tools, Data Acquisition, Application Analysis***

## I. INTRODUCTION

Android devices with embedded Multi-Media Card (eMMC) employ trim support, erasing the content of unallocated space every time the device is shut down. eMMC storage integrates flash memory and controller onto a single chip. The controller among other things perform trimming of unused data blocks, remapping secure data erasure when requested. This is another reason we need not shutdown an Android device but put it on charger and place it on Faraday bag [1]. As per the study by [2] various applications in general do store and access data the user is not aware of.

He reveals that this is facilitated by the fact that many applications by design seek permission during the installation process to access various hardware, services and data on the mobile device camera, GPS navigation service and photos respectively. This data with no doubt can be a primary source of evidence before the jury.

According to [1], every activity a user does on Android device interacts with its associated application. Some applications come preinstalled by Original Equipment Manufacturer (OEM), while others are third party Apps downloaded and installed by the user like Twitter and Clubhouse among others. Routine functions such as contacts, calls and SMS are performed through these applications.

Android application analysis is crucial during the course of an investigation given their increased use in the modern world of web based social networking.

Most of these Apps store sensitive information on the device's internal memory or SD card such that, performing forensics of social media Apps and many more other Apps may provide valuable information such as user location, communication details and many more. However, forensic examiners need to develop necessary skills to convert available data into meaningful insights by having a comprehensive understanding of how android applications handle user data.

The fact that Android applications are not bound by Certificate Authority and use self-signed certificates, this has prompted development of unprecedented number of various applications for which some are designed for social interaction.

According to [2], mobile applications generate and store large data sets on mobile devices which they are hosted and this data can be useful in the reconstruction of events. Because of this, forensics of android social applications play a crucial role during an investigation.

©IJRASET: All Rights are Reserved 

In their study, [3] mentioned that it is inevitable for Law enforcement agents not to find either popular or less popular social media applications with interfaces of different languages and functionalities in which investigators would not be having sufficient expert knowledge about. In most cases, investigators are not aware of potential evidentiary artifacts that can be discovered from such applications. It is from this background that the researchers were triggered to explore more on the behavior of Twitter with spaces and Clubhouse so as to try and discover the potential traces they may leave behind in emerging Android devices hosting them.

Since android mobile devices have become essential for communication and socialization, privacy concerns have grown, so most smartphone vendors have implemented multiple security protection measures like encryption to protect user data on their products, making forensics harder. Inevitably, smartphone producers are always behind digital forensic professionals in developing forensically sound methods that yield valuable insights admissible in court. [4] proposes that mobile forensic research should find intrusive approaches like bypassing security features by exploiting hardware and social media application vulnerabilities to fill this gap. This study also aligned itself with this similar technique in doing forensics of the aforementioned social media applications to establish a sound procedure for forensic investigations on android-based social media platforms.

Twitter is a social networking service on which users post and interact with messages known as "tweets" [5]. Only registered users can tweet, like tweets, and retweet tweets, yet unregistered users are limited to only reading tweets which are public. Apart from the former functionalities, twitter plartform also allows for direct messages between two accounts and can further enable users to host or participate in live audio virtual environment called spaces which allows for real time group conversations which can be moderated by the Host or co-hosts. This platform accommodates an unlimited number of listeners, 1 host, 2 co-hosts and 10 speakers. The host has a privilege of recording the whole meeting for future reference. Any Twitter user can create a Space from an Android or iOS based device, however this study only focuses on Android based Twitter.

Just like Twitter spaces, Clubhouse is also a virtual place for casual, drop-in audio chats. According to [6], This platform provides for numerous virtual rooms which can accommodate a lot of people to explore on various conversations. This plartform provides a space for friends and new people to meet and tell stories, ask questions, debate, learn and have impromptu conversations on thousands of different topics. Though the center of this platform is all about audio charts, it also provides for full multimedia interactions through text, picture and video clip attachments if need be. Whenever people get together, it is beautiful, but it can also get messy. Clubhouse Community norms prohibit nudity, terrorism, harassment, intellectual property violations, suicide, and other illegal actions. Certain regulations have been put in place to reduce wrongdoing, yet some will still be found wanting.

This study investigated Twitter and Clubhouse artifacts on developing android technology using sound forensics. Source SMG20+ and SMGTA7. After installing the social networking apps and creating test accounts on both devices, test data was input by performing various user behaviors. Open source and enterprise methods extracted relevant application data. Analyzing retrieved logical images with open source and business methods yielded results. Finally, all strategies were compared and recommendations offered. The study examined the behavior of the social networking apps to find the traces they leave on android devices and analyze how well these traces may be used for forensic purposes.

## II. LITERATURE REVIEW

In their research, [10] performed forensics on Tik Tok social media application. The approach taken was to perform common activities using different tik tok accounts and find out how this impacts discovery of meaningful insights. The research concluded that each logged in account activity is recorded in the *aweme_user.xml* but accounts would be combined with the order of the entry code of each ID such that information about each application user can be traced.

Challenges concerning Android Forensics were revealed by [11] in which they delve into how Proactive and Reactive forensic techniques can be utilized as solutions. They also looked into the application, permission and extraction based challenges faced by current forensic tools. recommendation was made that proactive techniques be adopted to monitor suspects particularly on cases involving national security, while reactive forensics can be used on petty cases.

The unsend message feature in Android Social Media Applications can be taken advantage of by criminals as an antiforensic technique. In view to this, [12] investigated whether it’s possible to recover from such predicament. The approach taken was to simulate attacker and victim on phones used in the research as suspect devices. Messages were exchanged on all applications in question and an usend function was invoked on selected messages in view to recover them in a forensically sound manner.

In this study, MOBILedit failed to produce any results whilst UFED techniques yielded good results on all applications except for Line and Snapchat.

Forensic of Android Kik Messenger was done by [13]. The main objective of the study was to identify, recover and analyze forensic artefacts of Kik Messenger in order to come up with meaningful insights. The researcher recommended further forensics of improved versions of Kik Messenger preferably on other OS platforms in the market.

In their study, [14] provided a comprehensive overview on the assessment of techniques of Android forensics and Android Antiforensics. They started by expounding on android operating system and its architecture in detail. Thereafter they used the general stages in mobile forensics to assess various android forensic techniques. This was achieved by reviewing literature related to android forensic techniques. Furthermore, the study discussed on the four basic classes of android antiforensics which are, *Destroying Evidence, Trail obfuscation, Data Wiping, Counterfeiting evidence and Attacks against forensics processes or tools*. The research concluded by stating that Android Technology is dynamic and fast evolving and lacks proper standardization.

Android forensic analysis of private chat and normal chat on social messenger was done by [15]. Their study worked on the acquisition, analysis, and interpretation of private chat's metadata which are obtained from Telegram, Line, and KakaoTalk. In this study, the researchers demonstrate how the artifacts are related to one another between analyzed results from normal and private chats. They then presented a guide on how to go about conducting a cybercrime investigation on social messenger applications. The approach used was to first take note of all present directories contained in the package folder of each application in question before any activity is performed. Hash functions of all these directories were taken note of and thereafter various activities were performed and the same package folders were checked for consistence through the hash functions. Only directories with changed hash functions were investigated. For future works, the researchers recommended further studies on the investigation of deleted chat recovery, decryption of encrypted chat, and memory forensics in smartphones.

In their extensive investigation of establishing a novel model that can extract data from encrypted mobile devices, [16] suggested that future research should focus on more invasive strategies including circumventing security features and exploiting known weaknesses. This study lists five Conventional Mobile Forensic Extraction Techniques: Manual, Logical, Hex Dumping/JTAG, Chip-Off, and Micro Read. They examined how enhanced encryption affects these strategies and android phone data privacy and confidentiality. Manufacturers cannot access data at rest or the hard-coded unique passwords and keys used for decryption. Trusted Execution Environment (TEE) protects user data and OEMs' unique data and technologies. Mobile device users and forensic examiners have minimal control over mobile devices. Root of Trust is another OEM anti-forensic method (RoT). This technique examines all hardware and software in the boot-chain to guarantee that only approved components are run during boot. This renders all typical acquisition forensic approaches involving unsigned third-party software ineffective. The paper suggests more research on mobile forensic data extraction standardization and validation.

Overally, the review of literature reflected that both Android Social Media Applications and the associated devices hosting them continue to evolve and continuous research needs to be done to improve and ensure relevance of existing forensic techniques hence the justification for this study.

## III. PROPOSED WORK AND METHODOLOGY

The approach taken was to set a case study involving two emerging android phones as mentioned in the Hardware and Software Requirements. In these two phones, Twitter and Clubhouse were installed on both devices. Accounts for each application in both phones were setup and known test data was fed through random communication between the two phones. During the experiments, the file systems of the device storage were actively monitored, so that the data created or modified by each actions was located and correlated with that action. Thereafter, extraction of data for each application on both phones was done using open source and enterprise tools. Analysis of the extracted data was then done using open source and commercial techniques. A comparative analysis of various selected techniques was conducted and thereafter a report was made.

### *A. Hardware and Software Requirements*

The forensic workstation was an Acer Aspire A515-56 (11th Gen Intel® Core i5-1135G7 @ 2.4GHz, 2419Mhz with 8 Logical Processors, Memory 20GB DDR4, Graphics Full HD 1080p and Storage 512GB SSD). The physical Android devices used in the research are, SMGS20+ (model SM-G986B with Exynos Processor 990 running Android version 12) and SMGTA7 (model SM-T505N with Qualcomm Snapdragon Processor 665 running Android version 11.).

### *B. Workflow Diagram*

Figure 2 shows the logical flow of how the proposed work was to be carried out in chronological order.

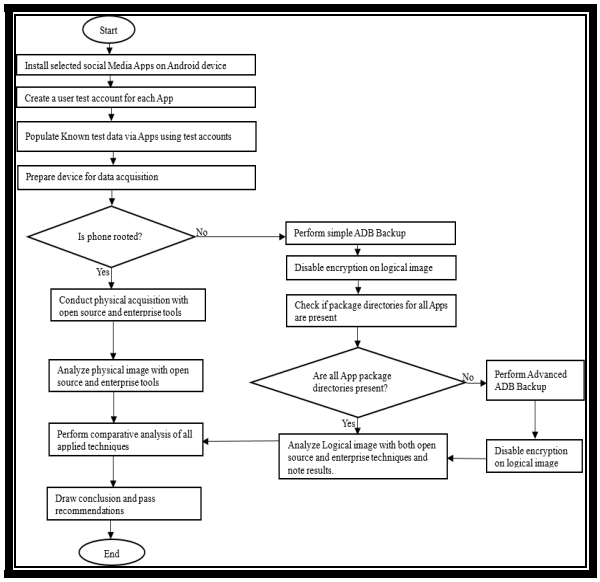



Figure 2: Flow chart of the proposed work

### *C. Preliminary Work*

The preliminary work covered the selection of open source and enterprise techniques used in the extraction and analysis of Twitter and Clubhouse sitting in SMGS20+ and SMGTA7. Table 1 shows the open source tools considered while Table 2 shows enterprise tools. The open source tools were freely acquired whilst 30-day trial versions of selected enterprise tools were prudently solicited for by the researchers from respective vendors.

Table 1: Relevance of Selected Open Source Techniques

| | Tool | Purpose and Description |
|---|---|---|
| 1 | JDK v18.0.1 | The forensic workstation requires the Java Development Kit (JDK) installed, because the ASDK is dependent on it. |
| 2 | ASDK | The ASDK provides for adb tool which connects the forensic workstation to suspect device for data extraction. |
| 3 | Abe.jar | Android-backup-extractor (ABE) is a Java library typically used in ADB Backup Recovery |
| 5 | DB4S v3.12.2 | SQLite is a popular database format present in many mobile systems. DB Browser for SQLite (DB4S) is in this study to facilitate easy view of application databases that would have been extracted. |
| 6 | Autopsy v4.19.3 | Autopsy is a graphical interface to the command line digital investigation analysis tools in The Sleuth Kit used to analyse various disks and filesystems. |
| 7 | CPU-Z v1.41 | It is a freeware that gathers information on some of the main devices of your system |
| 8 | Root Check v6.5.0 | Free Android App to Verify Proper Root Installation on an android device it has been installed. |
| 9 | Device Drivers | Without the necessary drivers, the computer may not be able to identify and work with the connected device. Each manufacturer has their own proprietary drivers and distributes them along with the phone. |
| 10 | Samsung Backup | Emerging Samsung devices have an inbuilt Backup Data to USB Drive or SD card. This facility can be capitalized to extract various logical files. |

### *D. Implementation*

Guided by the scope, the implementation of the project covered Preparation Phase, Processing Phase and Documentation and Reporting. The preparation phase involved research about SM G20+ and SM G TA7 which were the phones to be examined. This also covered a further study of the acquisition and examination techniques selected during the preliminary work. In this phase, Identification of social media applications to be worked with was done. The Processing phase comprised these major tasks amongst many, logical image acquisition, decrypting logical backup and analyzing the image to get meaningful insights. Lastly, Documentation and Reporting of every activity was performed.

### *E. Preparation Phase*

The implementation started with the acquisition of a forensic workstation and two emerging android devices stated in the Hardware and Software requirements provided earlier. Twitter and Clubhouse were installed in both SMGS20+ and SMGTA7. Pertinent application accounts were created and data entry though various application activities over a period of time was carried out. Thereafter a forensic workstation was setup and all selected open source and enterprise tools were installed. Figure 3 shows a screenshot of a setup forensic workstation with MOBILedit, Belkasoft, Magnet Axiom Process & Examine, UFED 4PC, Andriller and Autopsy.



Figure 3: Screenshot of installed forensic tools

### *F. Processing Phase*

The processing phase started with gathering requisite information about SM G20+ and SM G TA7 needed for preparation of extraction of possible images. This was then followed by the extraction of logical images using open source tools and lastly by enterprise tools. Table 3 shows a comparative analysis of Information gathered using CPU-Z, Root Checker and Manual check.

Table 3: Info Gathered about SMG20+ & SMGTA7

| Device Name | Information Gathered from CPU-Z, Root Checker & Manual Method | | |
|---|---|---|---|
| | **CPU-Z** | **Root Checker** | **Manual Check** |
| SMGS20+ | Model of phone:*SM-G986B*<br>Manufacturer: *Samsung*<br>Processor type: *Exynos 990*<br>Processor cores: *8*<br>Processor speed:*2.73 GHz*<br>Total Ram: *10901MB*<br>Internal Storage: *108.27GB*<br>Android Version: *12*<br>API: *31*<br>Bootloader ver: *G986BXXUEFVDB*<br>Root Access: *No* | Model of Phone: *SM-G986B*<br>Android Version: *12*<br>Root Access: *Not properly Installed* | Go to ***Settings →About Phone.***<br>Model of phone:*SM-G986B*<br>Manufacturer: *Samsung*<br>Root Access: *Not found*<br>Bootloader status: *Locked*<br>***NB**: Root status was checked manually as shown in the screenshot below.*<br>C:\Users\Austin Makate\AppData\Local\Android\Sdk\platform-tools>adb shell su /system/bin/sh: su: inaccessible or not found ✓ |
| SMGTA7 | Model of phone:*SM-T505N*<br>Manufacturer: *Samsung*<br>Processor type: *Qualcomm Snapdragon 665*<br>Processor cores: *8*<br>Processor speed:*2.02 GHz*<br>Total Ram: *2703MB*<br>Internal Storage: *21.87GB*<br>Android Version: *11*<br>API: *30*<br>Bootloader ver: *T505NDXU3BVE1*<br>Root Access: *No* | Model of Phone: *SM-T505N*<br>Android Version: *11*<br>Root Access: *Not properly Installed* | Model of Phone: *SM-T505N*<br>Android Version: *11*<br>Root Access: *Not found*<br>Bootloader status: *Locked* |

### *G. Image Extraction Using Open Source tools*

Three open source techniques were considered for data extraction on SMGS20+ and SMGTA7 and these are ADB tool, Extraction by device driver installation and the Default backup method. Figure 4 shows how the adb backup was done on SM G20+ using the *adb backup –shared –all* command. The unlock password was entered on the phone and the backup immediately begun. This same process was repeated on SMGTA7 and again a logical image was successfully extracted.

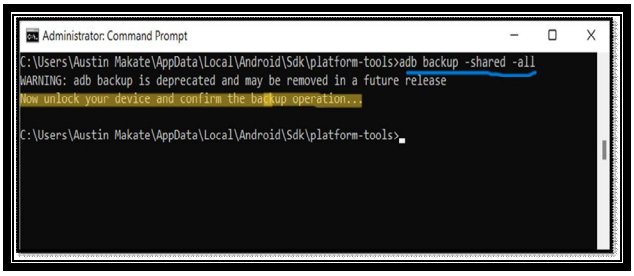



Figure 4: adb backup on SM G20+

After extracting the logical image using the adb tool, the image was then decompressed using the abe.jar tool. The Backup.ab was first uniquely renamed to *bac.ab* and was saved in *C:\Program Files\Java\jdk-18.0.1\bin* where the abe.jar tool was installed. The command used to decrypt the *bac.ab* file into bac.tar file is: *Java.exe –jar abe.jar unpack bac.ab bac.tar*. The backup password which is the phone password was entered and the decryption began as shown in Figure 5.




Figure 5: Decryption of SM G20+ bac image with abe.jar tool

*H. Image Extraction Using Enterprise Tools*

As revealed in Table 3 five enterprise techniques were considered for extraction of logical images on SMGS20+ and SMGTA7. All techniques used managed to extract images and Figure 6 shows the advanced logical backup using the downgrade technique by MOBILedit. It was observed that Belkasoft also uses this same technique to extract user data on applications that have restricted data extraction in the android manifest file.

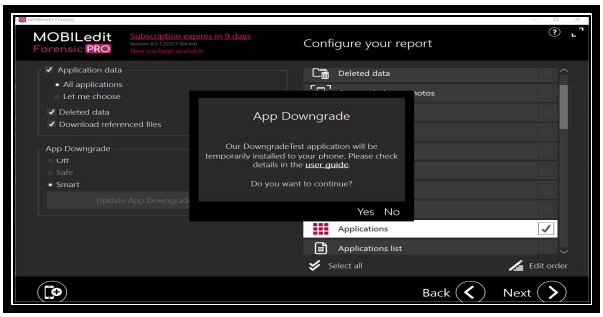



Figure 6: Downgrade Extraction Technique by MOBILedit

*I. Image Analysis using Open Source Tools*

In this Study, manual checking of extracted application packages was done and results of this process are shown it Table 4. Autopsy and DB4S were then used to analyze all extracted images from SMGS20+ and SMGTA7. Figure 7 shows analysis by Autopsy of an image extracted through driver installation method on SMGS20+. In this screenshot is an exhibit of a twitter audio sent by Austin to OleTropics. All found SQL lite databases were analyzed with DB4S, no databases were found for twitter save for clubhouse.

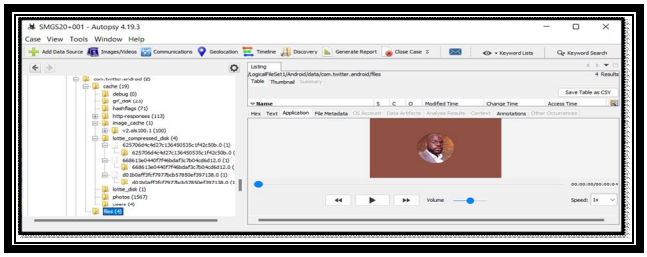



Figure 7: Analysis of SM G20+001 Image by Autopsy

*J. Image Analysis using Enterprise Tools*

Enterprise techniques considered in analyzing various logical images extracted in the study are MOBILedit, Belkasoft and Axiom Examine. Figure 8 shows the overall result of analysis by Axiom Examine of an image that was acquired from SMGS20+ by Axiom Process. This technique was also used to analyze images acquired by other techniques and summary statistics are given in Table 5.

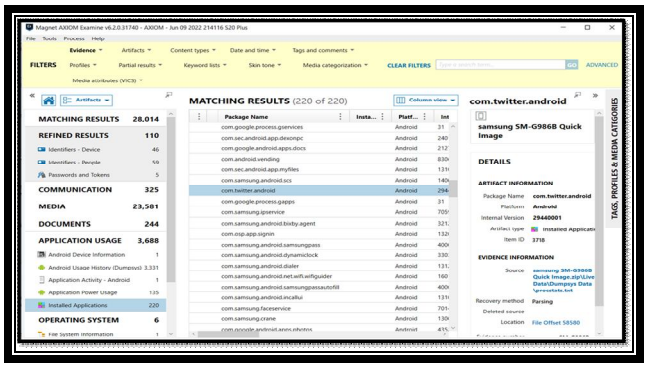


Figure 8: Analysis of SMGS20+ image by Axiom Examine

## IV. RESULTS

This section covered extraction and analysis results by open and enterprise techniques. The results are published in a comparative manner in an effort to reveal the performance of each applied technique.

### *A. Extraction Results*

Only simple and advanced logical extractions were considered in this study because the phones were not rooted. Rooting the phones would risk data loss as the bootloaders of the devices under study were locked. It was investigated and noted that the devices under investigation were all associated with data wiping antiforensics mechanism whenever one tries to unlock the bootloader hence no attempt was made to try and root the phones. Table 4 shows the overall results of extraction by each technique taking cognizance of how they performed regarding the pulling of pertinent application package folders i.e. *com.clubhouse.app* and *com.twitter.android* for both SMGS20+ and SMGTA7. The packages were manually checked for through windows explorer as the images were all in a logical format.

Table 4: Summary Result on Image Extraction

| | Acquisition Technique | | Application Packages Found | |
|---|---|---|---|---|
| | Open Source | Enterprise | com.clubhouse.app | com.twitter.android |
| 1 | ADB Tool | | Found | Not Found |
| 2 | Device Driver | | Found | Found |
| 3 | Default Backup | | Found | Found |
| 4 | | Andriller | Found | Not Found |
| 5 | | MOBILedit | Found | Found |
| 6 | | Magnet Axiom Process | Found | Not Found |
| 7 | | Belkasoft | Found | Found |
| 8 | | UFED 4PC | Found | Found |

### *B. Examination & Analysis Results*

As can be seen from Table 5, eight techniques were considered for logical extraction on both SMGS20+ and SMGTA7 phones. Thereafter, five techniques were used to analyze the eight logical images from each phone whilst noting artefacts found for Twitter and Clubhouse applications. The red X shows that no artefacts were discovered whilst the green tick shows that artefacts were found. An interesting coincidence is that a general summary outcome was similar for both the phones though artifacts found in each case were different. In both cases, when Autopsy analyzed the image from ADB tool and Andriller, only artefacts from Clubhouse were found, this is so because from the extraction process, the adb tool failed to extract the Twitter package file. Autopsy found artefacts for both Twitter and Clubhouse from images extracted by Device Driver, MOBILedit and Belkasoft techniques. It failed to analyze images from Default Backup, Axiom Process and UFED 4PC as they had formats that were not compatible with it.

Table 5: Comparative Analysis of Tools based on Results

| | Techniques used to Acquire Images: | Techniques used to Analyse Acquired Images and their Associated Results: | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Autopsy | | Axiom Examine | | MOBILedit | | Belkasoft | | DB4S | |
| | | twitter | clubhouse | twitter | clubhouse | twitter | clubhouse | twitter | clubhouse | twitter | clubhouse |
| 1 | ADB Tool | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 2 | Device Driver | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 3 | Default Backup | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ |
| 4 | Andriller | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 5 | MOBILedit | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 6 | Axiom Process | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ |
| 7 | Belkasoft | ✓ | ✓ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✓ |
| 8 | UFED 4 PC | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ |

As can be seen from Table 5, Magnet Axiom Examine was only able to analyze the image extracted by Axiom Process. By default, Axiom Examine can only analyze files with *(.mfdb)* extension hence why it was not able to analyze files extracted from other extraction techniques.

MOBILedit is compatible with many file formats including the *(.ufdx)* extension from UFED 4PC, hence it was able to analyze most images except those from Axiom Process and Default Backup. Since Andriller and ADB tool technique failed to pull Twitter package as indicated in Table 4, MOBILedit had no artifacts found the same in both phones.

As can be seen in Table 5 again, Belkasoft failed to analyze any image simply because the trial version came by default without a decryption module. The decryption module is offered by a third party company and does not come along with a trial version package.

DB4S was only able to analyze SQL lite databases found on images extracted by ADB tool, Device Driver, Andriller, MOBILedit, Axiom Process and Belkasoft. Since no SQL Lite database was found for twitter on all images extracted, DB4S was only able to find meaningful insights for Clubhouse.

It is significant to note that none of the tools were able to analyze data extracted by the Default Backup method. It seems this Samsung backup technique encrypts data in a format that cannot be handled by any of the explored techniques.

Whilst Table 5 is just a general summary on artifacts found by each analytical technique applied on each extracted image, Table 6 and 7 reflects summary of type of artifacts found by each analytical technique for Twitter and Clubhouse respectively on both SMGS20+ and SMGTA7. We have same indication of results in tables 5, 6 and 7 for both the two phones simply because same activities were performed on the two Samsung phones and similar forensic techniques were also applied.

Table 9: Twitter Artifacts discovery by each Technique

| *Twitter Artifacts* | *Autopsy* | *Axiom Examine* | *MOBILedit* | *Belkasoft* | *DB4S* |
|---|---|---|---|---|---|
| *User Handle* | *Yes* | *Yes* | *No* | *No* | *No* |
| *User Location* | *No* | *No* | *No* | *No* | *No* |
| *Direct Messages* | *No* | *No* | *No* | *No* | *No* |
| *Followers* | *No* | *No* | *No* | *No* | *No* |
| *Followings* | *No* | *No* | *No* | *No* | *No* |
| *Tweets* | *No* | *No* | *No* | *No* | *No* |
| *Pictures & Videos* | *Yes* | *Yes* | *No* | *No* | *No* |
| *Voice Notes* | *Yes* | *Yes* | *No* | *No* | *No* |
| *Profile Picture* | *Yes* | *Yes* | *No* | *No* | *No* |
| *Date App Installed* | *No* | *Yes* | *Yes* | *No* | *No* |
| *Twitter Space Logs* | *No* | *Yes* | *No* | *No* | *No* |

Table 10: Clubhouse Artifacts discovery by each Technique

| *Clubhouse Artifacts* | *Autopsy* | *Axiom Examine* | *MOBILedit* | *Belkasoft* | *DB4S* |
|---|---|---|---|---|---|
| *User account* | *Yes* | *Yes* | *No* | *No* | *No* |
| *User Location* | *No* | *Yes* | *No* | *No* | *No* |
| *Direct Messages* | *No* | *No* | *No* | *No* | *No* |
| *Virtual Call logs* | *No* | *No* | *No* | *No* | *No* |
| *Voice Notes* | *Yes* | *Yes* | *No* | *No* | *No* |
| *Profile Picture* | *Yes* | *Yes* | *No* | *No* | *No* |
| *Contacts* | *No* | *Yes* | *No* | *No* | *Yes* |
| *Date App Installed* | *No* | *Yes* | *Yes* | *No* | *No* |
| *Activity Logs* | *No* | *Yes* | *No* | *No* | *No* |

## V. LIMITATIONS

The major limitation is that both phones were not rooted and rooting them was a challenge as their bootloader were locked. This limited the researcher to only consider simple and advanced logical extractions. The other limitation is that the enterprise tools secured were 30-days' trial versions, the researcher had to learn and use the tools in a limited time. This indeed put the researcher under pressure and some features of other tools were not fully exhausted. In trying to overcome this challenge, the researcher improvised by seeking startup training and tutorials form the technical support teams of the sought tools and by working overtime. Belkasoft had a limitation that it came along without a decryption module, hence all images extracted by it were encrypted. However, this challenge was overcome by decrypting these images through mobile edit and abe.jar tool. This was almost a similar case with Andriller as its decryption module failed handle the compression format of the emerging android phones used in the study. This challenge was overcome by using the abe.jar tool.

## VI. CONCLUSION AND FUTURE WORK

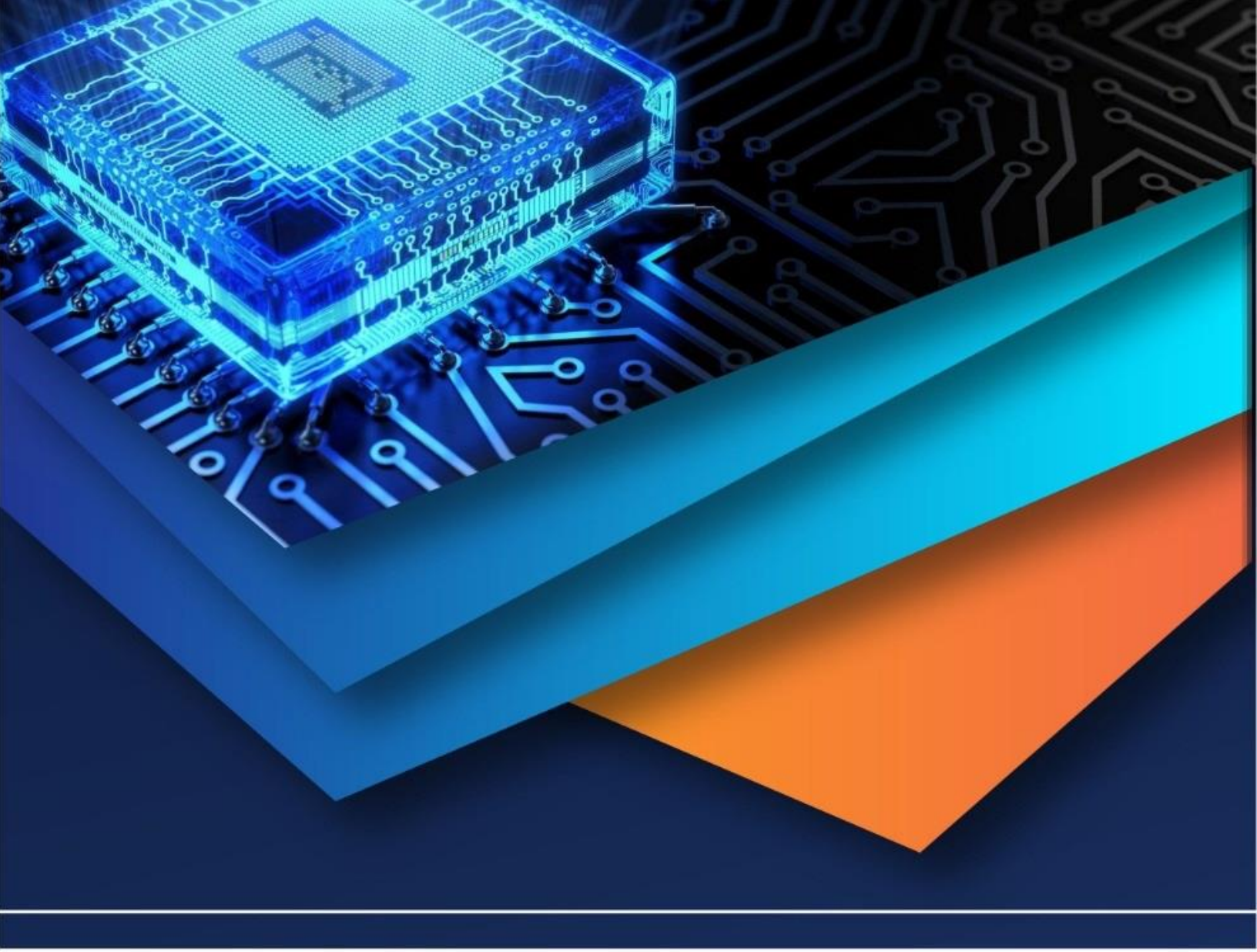
From the results of this study, it can easily be deduced that open source tools can be a good alternative of expensive enterprise tools if properly utilized. Almost every technique utilized in this study exhibited some degree of relevance one way or the other in finding meaningful evidential artifacts on Twitter and Clubhouse. The findings of this research are in line with the statement made by research predecessors that not one tool does it all, a good combination of tools can be much more relevant in providing solutions than depending on one tool. Regarding the downgrade technique used by Belkasoft and MOBILedit, it has been noted with a warning that this technique is not forensically sound as it tempers with source evidence. The technique first downgrades the secure application in order to extract user data. Thereafter, the technique then tries to restore the application as it where but in some cases this is not attainable and would result in data loss. It has also been noted that enterprise tool uses some of the open source tools in the backend like ADB tool. The study revealed that even enterprise tools are not able to extract a physical image on android devices which are not rooted and in view to this, the researcher recommends more research to be carried out on how to root android devices.

From the results in table 5, it is clear that none of the analytical tool was able to analyze logical files extracted through the Samsung default backup method. More research can be conducted in this regard to try and find analytical techniques that can easily parse such data into meaningful insights. Whilst the researcher established that MOBILedit and Belkasoft used the downgrade technique to extract user data on applications that have restricted adb backup in the android manifest file, the study failed to establish the technique utilized by UFED 4PC to easily extract data from such applications. Twitter user data was easily extracted with UFED 4PC. In view to this, the researcher recommends further research to be carried out to establish the technique being utilized by UFED 4PC in extracting user data from complex application such as Twitter and establish whether the technique is forensically sound or not. If the technique is discovered, it may go a long way in contributing positively in mobile forensics using open source tools. Enterprise tools are expensive and out of reach to many, hence more work needs to be done to find more effective open source techniques to grow the mobile forensic fraternity in a nearly cost free approach.

## REFERENCES

[1] Tamma Rohit, Oleg Skulkin, Heather Mahalik and Satish Bommisetty 2020 Practical Mobile Forensics Fourth Edition.
[2] Anglano, Cosimo, Canonico, Massimo, Guazzone, Marco, 2020. The Android Forensics Automator (AnForA): A tool for the Automated Forensic Analysis of Android Applications. Computers & Security 88 101650.
[3] Z. Xu, C. Shi, C. Cheng, N. Z. Gong and Y. Guan, "A Dynamic Taint Analysis Tool for Android App Forensics," 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 160-169, doi: 10.1109/SPW.2018.00031.
[4] Fukami, Aya, Radina Stoykova, and Zeno Geradts. 2021. A New Model for Forensic Data Extraction from Encrypted Mobile Devices. Forensic Science International: Digital Investigation 38: 1–10.
[5] https://en.wikipedia.org/wiki/Twitter#Tweets as@26 April 2022.
[6] https://community.clubhouse.com/ as@ 25 April 2022.
[7] https://www.online-tech-tips.com/smartphones/why-its-so-hard-to-recover-deleted-data-on-android-and-what-to-do-about-it/
[8] Oleg Afonin, Vladimir Katalov 2016 Mobile Forensics – Advanced Investigative Strategies pages 54,56
[9] https://resources.infosecinstitute.com/topic/common-mobile-forensics-tools-techniques/
[10] Nasution, Muhammad & Luthfi, Ahmad & Prayudi, Yudi. (2022). Investigating Social Media User Activity on Android Smartphone. International Journal of Computer Applications. 183. 46-52. 10.5120/ijca2022921890.
[11] Hazra, Sudip & Mateti, Prabhaker. (2017). Challenges in Android Forensics. 286-299. 10.1007/978-981-10-6898-0_24.

[12] Hermawan, Tofan & Suryanto, Yohan & Alief, Fahdiaz & Roselina, Linda. (2020). Android Forensic Tools Analysis for Unsend Chat on Social Media. 233-238. 10.1109/ISRITI51436.2020.9315364.
[13] Adebayo, Olawale & Sulaiman, Salamatu & Osho, Oluwafemi & Alhassan, John & Abdulhamid, Shafi'i. (2017). Forensic Analysis of Kik Messenger on Android Devices.
[14] Maček, Nemanja and Štrbac, Perica and Čoko, Dušan and Franc, Igor and Bogdanoski, Mitko (2016) Android Forensic and Anti-Forensic Techniques – A Survey. In: 8th International Conference on Business Information Security (BISEC'2016), 15 Oct 2016, Belgrade, Serbia.
[15] Satrya, Gandeva & Daely, Philip & Shin, Soo. (2016). Android Forensics Analysis: Private Chat on Social Messenger. 10.1109/ICUFN.2016.7537064.
[16] Fukami, Aya, Radina Stoykova, and Zeno Geradts. 2021. A New Model for Forensic Data Extraction from Encrypted Mobile Devices. Forensic Science International: Digital Investigation 38: 1–10.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)