Alt-Tech Social Forensics: Forensic Analysis of Alternative Social Networking Applications

By:
Hailey Johnson (University of New Haven), Karl Volk (University of New Haven), Robert Serafin (University of New Haven), Cinthya Grajeda-Mendez and Ibrahim Baggili (University of New Haven)

From the proceedings of
The Digital Forensic Research Conference
DFRWS USA 2022
July 11-14, 2022

Forensic Science International: Digital Investigation 42 (2022) 301406

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS 2022 USA - Proceedings of the Twenty-Second Annual DFRWS USA

Alt-tech social forensics: Forensic analysis of alternative social networking applications

Hailey Johnson*, Karl Volk, Robert Serafin, Cinthya Grajeda, Ibrahim Baggili

Cyber Forensics Research and Education Group (UNHcFREG), Samuel S. Bergami Jr. Cybersecurity Center, Connecticut Institute of Technology, University of New Haven, 300 Boston Post Rd., West Haven, CT, 06516, USA

A R T I C L E I N F O

Article history:

Keywords:
Digital forensics
Mobile
Network
Artifacts
Alternative social media
Parler
MeWe
CloutHub
Minds
SafeChat
GETTR
Wimkin
2nd1st

A B S T R A C T

Mainstream social platforms boast billions of users worldwide. In recent years, popular social platforms have seen a decline in their users that are choosing to migrate to alternative-tech social applications reinforced by frustrations of mainstream social platforms over alleged censorship of free speech and banning of predominant public figures such as the former president of the United States (U.S.). As such, group effect of similar minded users on alternative-tech social platforms may lead to fostering events such as the U.S. Capitol attack on January 6th, 2021, where the spreading of false information and extremist ideologies through alt-tech applications such as Parler and MeWe took place. These cases demonstrate the immense forensic need to understand how alternative-tech social applications operate and what they store about their users' personal information and activities. We present the primary account for the digital forensic study of ($n = 9$) alternative-tech social applications used on Android and iOS devices. Our analysis includes Parler, MeWe, CloutHub, Wimkin, Minds (Minds Mobile and Minds Chat), SafeChat, 2nd1st, and GETTR. Results revealed that some applications do store unencrypted user information on the devices, such as usernames, phone numbers, email addresses, posts and comments, and private chat messages. Furthermore, some security vulnerabilities were discovered that allow users to download data that should have been private (such as sent private images) without authentication and authorization by other users. Finally, to aid in the analysis and automatic extraction of relevant evidence, we share Alternative Social Networking Applications Analysis Tool (ASNAAT), that automatically aggregates forensically relevant data from the alt-tech social networking applications when presented with a mobile device's forensic image.

Published by Elsevier Ltd.

1. Introduction

Alternative social media platforms have been on the rise, coinciding with the growing distrust in the United States mainstream media. Fast growth in alternative social media interest has also been linked to mainstream social media platforms policing themselves more heavily during and after the 2020 presidential election. This was one of the direct causes for some users, including extremists to flock to other platforms, which ultimately led to the expansion of “alt-tech”. Alt-tech is described as clone technology created by extremists in order to combat “deplatforming” (Mak, 2021). Examples of popular alt-tech social media platforms include applications such as Parler, MeWe, CloutHub, SafeChat, and Minds. In November 2020 alone, Parler doubled its active members from five million to ten million (Newhouse, 2020).

Disturbances and concerns can ensue regarding smaller and less regulated platforms. When these types of applications are being used by like minded individuals, there is a heightened potential for users to become radicalized. The spreading of certain ideologies and an increased amount of misinformation on these platforms can lead to negative consequences (Dickson, 2020). This is not only a problem on alt-tech platforms, but also in more popular ones like Twitter, Facebook, and Reddit etc. Our focus in this work was on applications such as Parler, MeWe, CloutHub, and SafeChat, because of their growing popularity and existing links to extremist individuals and recent threats to the United States government. Not only have some of these applications been discussed on national news and forums, but the sheer number of their downloads (See

* Corresponding author.
E-mail addresses: hjohn5@unh.newhaven.edu (H. Johnson), kvolk1@unh.newhaven.edu (K. Volk), rsera1@unh.newhaven.edu (R. Serafin), cgrajedamendez@newhaven.edu (C. Grajeda), ibaggili@newhaven.edu (I. Baggili).

https://doi.org/10.1016/j.fsidi.2022.301406
2666-2817/Published by Elsevier Ltd.

Table 1
Application capabilities, tests performed, and retrieved data.

Application	Downloads	Capabilities	Performed tests	Retrieved artifacts
MeWe (2.18.30)	*5M+	Post, comment, chat	Post on feed Comment on posts Sent/received private messages Voice/video call	Message, post, contact and account information, cached images, device fingerprint/ID
CloutHub (1.19.7-Android) & (2.0.41-iOS)	*100k+	Post, comment, chat, and groups	Post on feed Comment on posts Post and comment in group Sent/received private messages	Message, post, and user information, and cached images
Minds (Mobile) (4.14.2-Android) & (4.20.0-iOS)	*500k+	Post & react to posts	Post on feed Comment on posts	Comment and post and cached images
Minds (Chat) (1.1.7-dev-Android) & (1.6.6-iOS)	*5k+	Chat and groups	Sent/received private messages	Videos, images, and emojis related to chat
SafeChat (0.9.46-Android) & (0.9.66-iOS)	*100k+	Post, comment, chat, voice/video call	Post on feed Commented on posts Sent/received private messages on feed	Message, post, and user information
GETTR (1.0.7-Android) & (1.2.7-iOS)	*5M+	Post & comment	Post on feed Commented on posts on feed	User, post, and comment information
Wimkin (2.1)	*5k+	Post, comment, chat, and groups	Post on feed Commend on posts Post and comment in group Sent/received private messages	Message, post, comment, user, and account information, and cached images
2nd1st (1.2.21)	**N/A	Post, comment, chat	Post on feed Comment on posts Sent/received private messages	Image posted
Parler (3.0.1-Android) & (2.50-iOS)	***16M	Post & comment	Post on feed Comment on posts	Image posted and account email

Where information on downloads was retrieved: *Google Play Store, **Download information not found, ***TechCrunch article (Silberling, 2022).

Table 1), was also a determining factor in selecting them for our investigation. Active user statistics for these applications were searched for, however, there was not a definitive number of active users obtained, and there were some varying results based on different articles and sites. Therefore, we turned to user download information gathered from credible sources to depict the popularity of these applications.

Parler, one of the first popular alt-tech apps, was at the center of attention during and after the 2020 presidential election in the United States. After the January 6th insurrection at the Capitol building, Apple and Google both banned Parler from their app stores for being connected with these events. Soon after that, Amazon announced that they would stop hosting the application on their servers. This caused a temporary shut down for the application (Brown, 2021). The application returned to the Apple App Store on May 2021. However, to this date it has not returned to the Google Play store. The Android application package (APK) can be downloaded from Parler's website. Following the U.S. Capitol invasion, a group of hackers downloaded and uploaded terabytes of Parler data by exploiting a security vulnerability. The application had an insecure direct object reference, and hackers were able to create accounts with administrator privileges and scrape the site's data (Greenberg, 2021).

Wimkin, another atl-tech application we examined was also banned from the Apple App Store due to the lack of censorship, and shortly after, it was hit with a Distributed Denial of Service (DDoS) attack (Shalvey, 2021). Not only are some of these newer applications growing in user popularity, but their lack of security makes them attractive targets to hackers.

Mobile devices have become crime-fighting tools and are often obtained to gather evidence for criminal investigations (University, 2016). Understanding and identifying radicalization through social media is important in combating online extremism (Ferrara, 2017). To date, and the best of our knowledge, there has not been a formal forensics analysis on the following applications, Parler, MeWe, SafeChat, Clouthub, Minds, 2nd1st, Wimkin, GETTR. Therefore, our work contributions include the following:

- A primary mobile and network forensic analysis of the Parler, MeWe, SafeChat, Clouthub, Minds, 2nd1st, Wimkin, and GETTR applications.
- A collection of discovered digital forensic artifacts shared on the Artifact Genome Project.[1]
- An upgraded Python tool originally created in our lab and presented at the 2021 12th EAI International Conference on Digital Forensics & Cyber Crime that will be published in the conference proceedings. New enhancements include the ability to directly analyze and extract data from the examined applications (Table 1). The tool can be used by digital forensic investigators to extract relevant data from the applications, and can be found in this GitHub Repository: https://github.com/unhcfreg/ASNAAT.git.

This paper is organized as follows: Section 2 presents related work. Section 3 lists the tools used to conduct this research. Section 4 discusses the methodology and setup, and section 5 discusses our analysis and results. Section 6 discusses the tool developed based on the results discovered. In section 7, an overview of the results is discussed and conclusions are made. Lastly, section 8 proposes limitations and future work.

[1] https://agp.newhaven.edu.

2. Related work

To the best of our knowledge, our methodical analysis of the nine alternative social media applications is the first of its kind. The current literature related to the forensic analysis of mobile applications is vast and includes a host of different applications and operating systems. Research has been conducted on popular instant messaging applications, social networking applications, vault applications, and more, in order to identify forensically relevant artifacts (E. Salamh et al., 2021). However, there has not been an in-depth forensic investigation of related alternative social media applications, which have become popular for individuals who want to veer from mainstream social media. The next subsections highlight some related research conducted in the last ten years.

2.1. Social media applications

As modern living has become increasingly entwined in the virtual world, social media application usage has increased. Research shows that social media applications can store data of forensic value. Not only can the applications store data useful to investigators, but social media applications can be an avenue for law enforcement personnel to investigate and acquire intelligence from criminals (Richard Jones, 2017).

Popular social media applications include Instagram, Facebook, and Twitter. Global Social Media Stats reported Facebook has 2.895 billion active users monthly, Instagram has around 1.4 billion users as of October 2021, and Twitter has around 211 million daily active users (Datareportal, 2021). Al Mutawa et al. (2012) were motivated by the increased use of social media applications to analyze Facebook, Twitter, and MySpace on BlackBerrys, iPhones, and Android smartphones. Manual forensic analyses were conducted, as well as the acquisition of logical images. It was determined that while multiple valuable data such as timestamps, photos, usernames, and contacts could be recovered from the iPhone and Android, nothing was recovered from the BlackBerry.

Walnycky et al. (2015) performed network and device forensic analyses of 20 Android social messaging applications, including Instagram, Facebook, and Viber. The device-stored data and network traffic were analyzed. Results showed that in most cases, reconstruction or interception of passwords, screenshots, pictures, videos, audio, messages, and more were successful. Alisabeth and Pramadi (2020) also performed a forensic analysis on Instagram and was able to recover user account information and activity information, such as uploads and private message traces. Finally, a forensic analysis of Instagram and other applications such as LINE, Whisper, WeChat and Wickr were performed on an Android device, resulting in the recovery of relevant artifacts (Menahil et al., 2021).

2.2. Instant messaging applications

Similar to mainstream social media, instant messaging applications have also become more prevalent and commonly used on smartphones. This is due to their simple direct and private messaging features. Mahajan et al. (2013) noted the increased use of instant messaging applications, and that the evidence left on mobile devices by these applications would be helpful for digital forensic experts. They analyzed the WhatsApp and Viber applications on several Android devices, and found sent and received chat messages, timestamps, profile pictures, and more. It is important to note that WhatsApp implemented end-to-end encryption in 2016 (Barrett, 2021).

Karpisek et al. (2015) analyzed WhatsApp's calling feature which had been added in 2015. This research provided the tools and methods used to decrypt WhatsApp network traffic, and developed a tool to display WhatsApp protocol messages. On the other hand, Arista Yuliani and Riadi (2019) extracted WhatsApp databases saved on disk storing encrypted messages. Oxygen forensics was used to decrypt the database.

Consequently, WhatsApp is not the only application that has been found to encrypt messages. M. Ovens et al. (2021) analyzed Wickr and the Private Text Messaging application, and were able to detect cryptographic processes through static and dynamic analysis. This led to the development of a decryption methodology for relevant artifacts.

2.3. Extremist propaganda in social media

As alternative social media platforms have become increasingly popular for individuals with extremist's views, it is important to not only understand the evidence stored in devices but also the motives leading to any radical behavior of such individuals. Aliaoulios et al. (2021) analyzed Parler and presented an extensive dataset. They noted the increase in right-wing applications and how the emergence of these ideologies and spreading of misinformation can lead to harmful and even dangerous consequences. Ferrara (2017) and Erbschloe (2019) examined how the spread of propaganda in social media has propelled radicalization. They recognized that studying extremist propaganda and its effects are important in mitigating security threats. Lastly, Longhi (2021) investigated the use of digital humanities and linguistics to assist with terrorism investigations. Although linguistic tools alone cannot solve a case, the analysis of the style, grammar, and contents that make up texts can provide investigators with clues that they otherwise might not have discovered.

2.4. Other related applications

Similar research has been conducted on many other applications and devices yielding to the discovery of critical digital forensic artifacts, this includes research conducted on video conferencing applications such as in (Mahr et al., 2021), virtual reality (Casey et al., 2019; Casey et al., 2019; Yarramreddy et al., 2018), small scale and IoT devices including smart watches (Baggili et al., 2015), drones (Clark et al., 2017), Amazon Alexa (Dorai et al., 2018; Chung et al., 2017), Google Home (Yıldırım et al., 2019), and more.

3. Apparatus

The hardware and software used to conduct this research are presented on Table A3, Appendix A.

4. Methodology

The forensic analysis of the nine alternative social media applications (with Minds Mobile and Minds Chat being separate but linked applications) consisted of four phases: 1) Scenario creation and setup, 2) data acquisition, 3) data analysis, and 4) tool enhancement. The hardware and software used to conduct our research are presented in Table A3, Appendix A.

4.1. Setup & scenario creation

This phase consisted of testing the nine applications' features with a user account. The mobile devices were factory reset and rooted or jail-broken prior to use. The applications were downloaded from the Google Play store[2] and Apple App store,[3] with the exception of Parler which was downloaded as an APK file from Parler's official website.[4]

To save time, a scenario was created that not only aimed to test the various functionalities within each application, but to also imitate realistic user activity. The specifics of the scenarios and tests differed based on the features provided by each application. The tests were conducted using accounts with fictitious credentials on a rooted Android and jail-broken iPhone. The accounts on the two devices interacted with each other to generate data based on the scenario created. This scenario was created and messages were sent manually in order to mimic a lifelike scenario with realistic artifacts. Most applications tested were available for Android and iOS, however, 2nd1st was only available for iOS and Wimkin was only available for Android. A summary of the tests performed based on each application's capabilities that resulted in the creation of relevant data are shown in Table 1.

4.2. Data acquisition

In this phase, in order to capture and analyze important artifacts in each application, data acquisition from mobile devices through device imaging was performed on the Android and iOS devices. Magnet Acquire[5] was used to acquire physical images of both devices after testing each application.

Moreover, network traffic was captured with Wireshark[6] while testing each application and NetworkMiner[7] and Fiddler[8] were utilized to analyze the traffic. This was a preliminary analysis as it was noted that most of the traffic was encrypted or encoded and no essential artifacts were found that required an in-depth analysis that fit in the scope of this research.

5. Analysis & experimental results

To extract and analyze relevant artifacts from our data acquisition, the tools shown in Table A3 in Appendix A were utilized along with some manual analysis. In this section, major artifacts found across all devices are summarized in their own subsections related to mobile disk forensics. Moreover, these artifacts are referenced in Tables A.4 and A.5. The tables highlight artifacts' details and path locations within the tested devices. Table 2 presents information found within artifacts in a more granular manner, prioritizing those that could be important in a forensics examination. Finally, note that the iPhone's file system directory naming convention uses a Universally Unique Identifier (UUID) to identify the application. This is used as a placeholder in the artifacts' tables.

5.1. Major artifacts found in mobile devices

Results indicated that most applications store data about users and their activities across devices in a similar manner. They rely on the device's storage to save this information. Due to the vast amount of artifacts discovered and the similarities between them, this section presents a summary of the critical information that was recovered across all applications. For a more in-depth analysis of each artifact, refer to Tables A.4, A.5, and 2.

5.1.1. Account/user information

User account and other related information is important in investigations as it links a user to their account. Several applications stored artifacts that contained this type of information. Out of the nine applications tested, six (55.6%) stored information about their users' accounts. Twelve of these were databases containing the username of the device/account owner, and ten of these also saved the user ID, full name, email or phone number used during login, and timestamps related to events. Thirteen of the databases also stored user information related to the contact the account owner was interacting with, including usernames, user IDs, friend status, and more.

Android applications also use Extensible Markup Language (XML) files to record user information. Four of those files were identified as containing user information such as username, full name, email address and more. Two of them contained user ID and authentication session tokens. One of them, *Clouthub.xml* (File ID 2.2 Tables A4 and 2) also contained the unsalted MD5 hashed password to log into the user's account (See Fig. 1). While only one of these files *io.invertase.firebase.xml*, (File ID 7.6 Tables A4 and 2), stored notification information. This included private chat information such as plain text messages and any file attachment, and the user ID, full name, and username of the contact the account owner was interacting with.

5.1.2. Posts and comment information

Posting information and commenting on posts on a user's social media timeline whether the profile is private or public are important features of social media applications. These artifacts display the user's activity and interactions with others. Out of the nine applications examined, eight (88.9%) were found to locally store some of this information.

Of the databases discovered, seven contained post and/or comment information. Five of these stored information identifying the user who posted, such as user IDs, full names, usernames, timestamps of the post or comment, post and comment IDs, contents of the post and/or comment, and more. It should be noted that for the *sgrouplesdb.sqlite* database (File ID 1.1 tables A.5 and 2), the only comments stored are those made by the account owner, and not comments made by others on such posts. Besides databases, only one XML file (File ID 7.6 Tables A.4 and 2) found pertained to application notifications, such as posts on the feed or comments on the account owner's page.

Furthermore, out of these files, only one database from the Gettr application in the Android device, *libCachedImageData.db* (File ID 6.1, tables A4 and 2) stored metadata related to GIFs posted by the account owner. This included a URL to access the GIF over the browser. It is important to note that this link is still active even though the validation date has already passed according to the stored timestamp. Moreover, these GIFs were also cached and downloaded to both devices (File IDs 6.4, 6.8, and 6.8 respectively in tables A.4, A.5 and 2). Complementing the utilization of GIFs in this application, two XML files were found. One stored the ID of recent GIFs posted by the account owner and the other contained plain text keywords of GIFs searched by the account owner.

Conversely, the *SafeChat.db* (File ID 5.3 table A4, File ID 5.4 table A.5, and File ID 5.3 and 5.4 Table 2) database found in both devices contained the post ID and timestamp of other posts clicked on and recently viewed by the user.

Consequently, not only were artifacts discovered that recorded posts and other information from the feed, but several applications downloaded media posted by the account owner to the device. Of the thirty-four directories found storing media, eighteen contained posted videos and images.

5.1.3. Private chat information

Social network applications containing private chat

[2] https://play.google.com.
[3] https://www.apple.com/app-store/.
[4] https://parler.com/android.html.
[5] https://www.magnetforensics.com/resources/magnet-acquire/.
[6] https://www.wireshark.org.
[7] https://www.netresec.com.
[8] https://www.telerik.com/fiddler.

Table 2
Important artifacts extracted across all forensic acquisitions.

Application, File ID & Artifact Name	Important Data Found in Disk										
	User ID	Name	Email	Phone #	Username	Timestamps	Posts/ comments	Media posted	Chats	Files sent/ received	Cached data
MeWe											
1.1 app_database, sgrouplesdb.sqlite	●, ○	●, ○			●, ○	●, ○	●, ○		●, ○		
1.2 Cache.db	○	○		○	○	○	○		○		
1.2 SGSession.xml	●	●		●	●	●					
1.3 tmp								○		○	
1.3 image_manager_disk_cache								●		●	●
1.4 default								○		○	○
Clouthub											
2.1 Clouthub.xml	●	●	●	●	●	●					
2.2 compressor								●		●	
2.1 Cache.db	○	○	○	○	○	○			○		
2.3 image_manager_disk_cache								●		●	●
2.2 tmp										○	
2.3 default											○
Minds Mobile											
3.1 minds1.db	●, ○				●, ○	●, ○	●, ○				
3.2 RKStorage	●				●	●					
3.3, 3.2 react-native-image-crop-picker								●, ○			
3.3 fsCachedData											○
3.4 image_manager_disk_cache								●			●
3.4 default											○
Minds Chat											
4.1 D										●	
4.1 Caches										○	
4.2 fsCachedData											○
4.2 emoji-recent-manager.xml						●			●		
4.3 image_manager_disk_cache										●	●
SafeChat											
5.1 SafeChat								●		●	
5.1.video								○		○	
5.2 cache										●	
5.2 tmp										○	
5.3, 5.4 SafeChat.db	●, ○	●, ○	●, ○	●, ○	●, ○	●, ○	●, ○		●, ○		
5.3 imagecache											○
5.4 download_tasks.db						●				●	
GETTR											
6.1 libCachedImageData.db								●			
6.2, 6.1 private_hughhen123.db, private_jmanny3.db	●, ○				●, ○	●, ○					
6.3, 6.2 g db	●, ○				●, ○	●, ○					
6.3 Cache.db	○				○	○	○				

(continued on next page)

Table 2 (*continued*)

Application, File ID & Artifact Name	Important Data Found in Disk										
	User ID	Name	Email	Phone #	Username	Timestamps	Posts/ comments	Media posted	Chats	Files sent/ received	Cached data
6.4, 6.8 libCachedImageData								●, ○			
6.4 flutter-images								○			
6.5 giphy_recents_file.xml								●			
6.5.image								○			
6.6 giphy_searches_file.xml								●			
6.6.video								○			
6.7 image_manager_disk_cache								●			●
6.7 fsCachedData											○
6.9 default								○			○
Wimkin											
7.1 RKStorage	●	●	●		●	●					
7.2 v2.ols100.1								●		●	●
7.3 react-native-image-crop-picker										●	
7.4 cache										●	
7.5 rocketUser.xml	●										
7.6 io.invertase.firebase.xml	●	●			●	●	●		●		
7.7 chatplus-chat.wimkin.com.db.db	●	●			●	●			●		
2nd1st											
7.1 tmp								○			
Parler											
8.1 file_0.localstorage			○								
8.2 https_parler.com_0.localstorage			○								
8.3 tmp								○			

Key: ●: Android Mobile, ○: iOS Mobile.

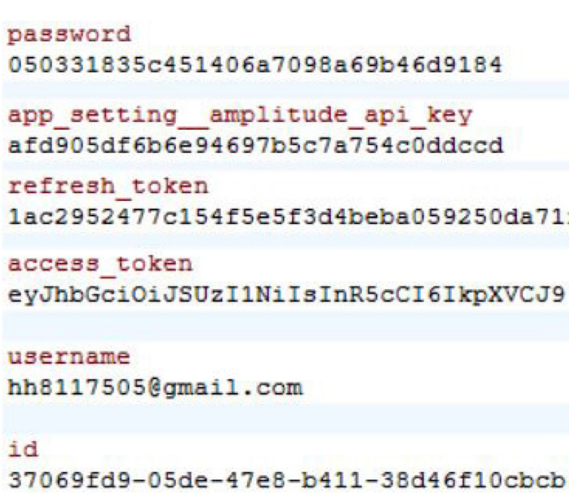
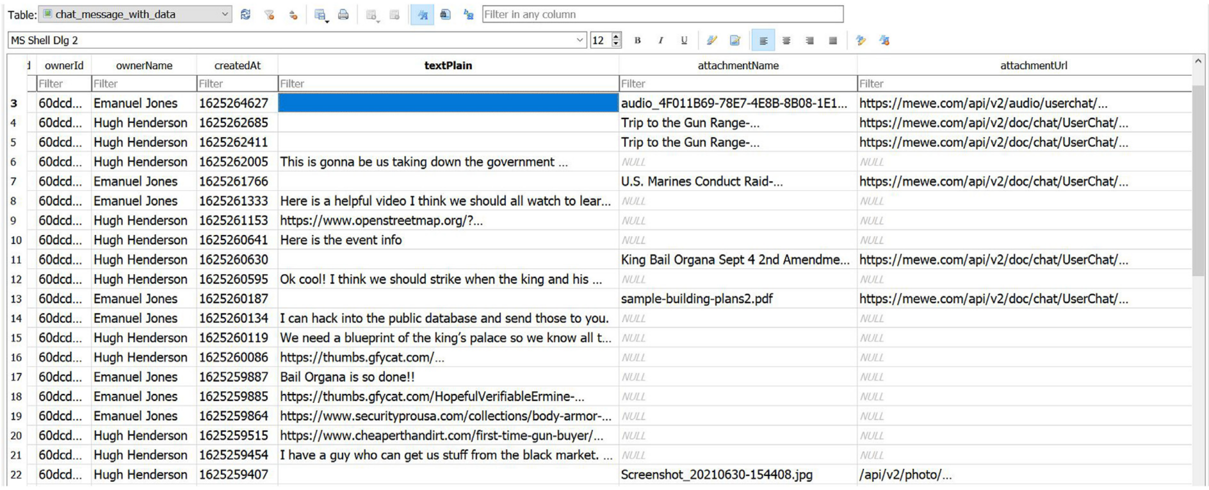

password
050331835c451406a7098a69b46d9184
app_setting__amplitude_api_key
afd905df6b6e94697b5c7a754c0ddccd
refresh_token
1ac2952477c154f5e5f3d4beba059250da71
access_token
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9
username
hh8117505@gmail.com
id
37069fd9-05de-47e8-b411-38d46f10cbcb

Fig. 1. Sample Clouthub.xml with PII.

functionalities add an important source of evidence to an investigation, especially when these messages are stored in the device in plain text. These artifacts could assist in the reconstruction of events and provide insight into who the user was communicating with. Out of the nine applications investigated, five (55.6%) stored some trace of private messages in the local disk. It is important to note that not all applications tested had chat messaging functionality.

Nine artifacts were found to store chat information, this included user IDs, names, timestamps, and plain text chat messages. Some of these messages also included links and attachment information of other media sent and/or received within the chat (See Fig. 2). Additionally, six of these artifacts contained usernames of the chat participants. It should be noted that the database *Cache.db* (File ID 1.2 tables A.5 and 2) stores links that were sent in private messages and the last message sent or received, not the entire conversation between users. Moreover, *io.inverse.firebase.xml* file (File ID 7.6 tables A4 and 2) stored notification information, thus only messages received along with other data were stored.

Similar to how the media posted in an application's feed was automatically stored in the device, some media and other type of files sent and received through private messaging were also downloaded to the device. Of the thirty-four folders discovered containing media and other types of files, seventeen contained files sent or received through private messaging. Twelve of these folders contained media sent, such as videos or images, four contained files sent, such as documents or audio files, and one contained media and files that had been received through private messages. A database from the SafeChat application, *download_tasks.db* (File ID 5.4 tables A4 and 2), was also discovered to store information on the file downloaded from private messages, including the ID, URL, filename, timestamp of the download, and more.



	ownerId	ownerName	createdAt	textPlain	attachmentName	attachmentUrl
3	60dcd...	Emanuel Jones	1625264627		audio_4F011B69-78E7-4E8B-8B08-1E1...	https://mewe.com/api/v2/audio/userchat/...
4	60dcd...	Hugh Henderson	1625262685		Trip to the Gun Range-...	https://mewe.com/api/v2/doc/chat/UserChat/...
5	60dcd...	Hugh Henderson	1625262411		Trip to the Gun Range-...	https://mewe.com/api/v2/doc/chat/UserChat/...
6	60dcd...	Hugh Henderson	1625262005	This is gonna be us taking down the government ...	NULL	NULL
7	60dcd...	Emanuel Jones	1625261766		U.S. Marines Conduct Raid-...	https://mewe.com/api/v2/doc/chat/UserChat/...
8	60dcd...	Emanuel Jones	1625261333	Here is a helpful video I think we should all watch to lear...	NULL	NULL
9	60dcd...	Hugh Henderson	1625261153	https://www.openstreetmap.org/?...	NULL	NULL
10	60dcd...	Hugh Henderson	1625260641	Here is the event info	NULL	NULL
11	60dcd...	Hugh Henderson	1625260630		King Bail Organa Sept 4 2nd Amendme...	https://mewe.com/api/v2/doc/chat/UserChat/...
12	60dcd...	Hugh Henderson	1625260595	Ok cool! I think we should strike when the king and his ...	NULL	NULL
13	60dcd...	Emanuel Jones	1625260187		sample-building-plans2.pdf	https://mewe.com/api/v2/doc/chat/UserChat/...
14	60dcd...	Emanuel Jones	1625260134	I can hack into the public database and send those to you.	NULL	NULL
15	60dcd...	Hugh Henderson	1625260119	We need a blueprint of the king's palace so we know all t...	NULL	NULL
16	60dcd...	Hugh Henderson	1625260086	https://thumbs.gfycat.com/...	NULL	NULL
17	60dcd...	Emanuel Jones	1625259887	Bail Organa is so done!!	NULL	NULL
18	60dcd...	Emanuel Jones	1625259885	https://thumbs.gfycat.com/HopefulVerifiableErmine-...	NULL	NULL
19	60dcd...	Emanuel Jones	1625259864	https://www.securityprousa.com/collections/body-armor-...	NULL	NULL
20	60dcd...	Hugh Henderson	1625259515	https://www.cheaperthandirt.com/first-time-gun-buyer/...	NULL	NULL
21	60dcd...	Hugh Henderson	1625259454	I have a guy who can get us stuff from the black market. ...	NULL	NULL
22	60dcd...	Hugh Henderson	1625259407		Screenshot_20210630-154408.jpg	/api/v2/photo/...

Fig. 2. Sample MeWe stored chat.

5.1.4. *Cached data*

Another prevalent category of data retrieved from most applications in both devices was cached data. As commonly observed in mobile applications and other software, they cache or save pieces of data commonly used by the application in a storage location in the device to aid the application retrieve that information faster and provide the user with an efficient experience. This means the application does not have to download information such as media and other types of files commonly used. Cached data is supposed to be temporary and eventually will automatically clear itself or allow users to do it manually as needed (Johnson, 2020).

What is critical to realize about cached memory is that it could contain a plethora of structured and unstructured data that could be powerful in an investigation as long as the data remains intact. As noted in our findings, we were able to locate critical artifacts that could be categorized as evidence in an investigation. The cache could also be used as a second location to identify where evidence might be stored. Out of the nine applications examined, eight (88.9%) contain some type of cached data stored in the device.

Of the thirty-four folders identified, fourteen contained cached images. Most of the cached images are believed to have come from the application's feed from public user's posts and other locations the user interacted with in the application, however, some contained images/videos posted by the account owner, images/videos sent/received through private messages, profile pictures of public users, and emojis. Due to applications' request to access user's photos, images in the account owner's gallery that had not been posted or sent could also be seen in some of these folders. Of these folders, seven not only contained cached images from the feed from random public users, but also contained media that had been posted by the account owner. Four of these seven folders also contained media that had been sent/received through private messages. Two contained photos and emojis from the account user's gallery. Refer to tables A.4, A.5, and 2 for directory names and paths of these folders containing cached data.

In addition to media files being cached and stored by the applications, three of the nine applications tested contained cache databases populated with other cache data. This included posts and comments from the feed and made by the account owner, as well as private message information. User IDs, names, phone numbers, and usernames of the account owner and other users interacted with were stored, as well as timestamps of user activities such as posts or comments made on the feed.

5.1.5. *Access to personal data without authentication/authorization*

Apart from discovering relevant artifacts stored on the disk, vulnerabilities were found in three of the nine applications tested (33.3%), in which personal data was able to be accessed without expected authentication or authorization. Unencrypted links to media posted and sent through direct messages were stored in databases and were able to be accessed without proper authorization. Some were able to be accessed simply by copying and pasting the URL on the browser, while others required an account on the application to access the link. However, the account used to send and/or post the content was not required, any user with an account on the application could view the link. These vulnerabilities could be exploited, and certain URL fuzzing attacks could be used to target servers hosting these data, causing an attacker to download personal files that could be very damaging to users.

Moreover, one of the databases stored private information from channels that had been created. This included phone numbers, addresses, and emails that had been entered during the account creation. Although the accounts were public, this was not information that could be seen from looking at the public profile on the application. Due to the magnitude of these flaws in the applications and the consequences of accessing unauthorized data, we are unable to provide a more detailed description of them. Moreover, it is essential for us to inform the companies about these issues prior to publishing, in order to provide developers the opportunity to remedy these flaws before potential malicious actors are informed. By the time this paper is published, the companies will have already been informed of these flaws and been given time to resolve them.

6. Tool development

The purpose of Alternative Social Networking Applications Analysis Tool (ASNAAT) is to automatically aggregate the forensically relevant data from the researched alt-tech social applications. Extracting and presenting critical pieces of information in a report will assist investigators in triaging evidence found in forensic images acquired from Apple and Android smartphones. ASNAAT is the second generation of the original Python tool created in past research which was presented at the 2021 12th EAI International Conference on Digital Forensics & Cyber Crime and it will be published in the conference proceedings.

To continue improving this open source tool to meet the demands of forensics investigations and triaging, new enhancements added to the report include hyperlinks directing the user to the actual artifact files. In this case, the user will not need to go to the artifact directory to view the actual file. Moreover, each artifact file is automatically hashed with SHA256 to uniquely identify it. Finally, the tool is now more efficient in presenting evidence from different applications as it separates the data in different reports for each application.

Algorithm 1. High-level Automation Algorithm

```
Algorithm 1 High-level Automation Algorithm
Requirements: Python3
Input: TAR image of a device.
Output: HTML Report

Select files to compare:
if "Help Option" then
 | show_manual();
else if "Apple Tar" then
 | A();
else if "Android Tar" then
 | B();

Select from installed apps (For terminal output):
if "All" then
 | AnalyzeApps = Installed;
else if "Specified" then
 | AnalyzeApps = Specified;

All apps analysis:
Initial_hash();
Search_archive();
extract_found();
analyze_files();
check_hashing();
generate_report();
```

6.1. Overview

The ASNAAT tool was developed to take in a tar archived forensic image of either an Apple or Android smartphone. As of this paper, tar images are the only supported format. In future enhancements of this tool, other types will be added. The reason for this limited capability is due to our preference of using open source tools, such as Magnet Acquire to image devices. By default, Magnet Acquire only outputs iPhone images in a tar archive.

Consequently, the tool was constructed in a way where relevant artifacts are extracted from the forensic images via wordlist files for each of the applications. The default provided wordlists contain filenames and file paths found though manual analysis and are presented in Tables A.5 and 2. In the event multiple locations in the images have the same file name, the paths allow for more control in the extraction. It is important to note that since artifact paths are different for both Apple and Android, separate folders split the wordlists. ASNAAT targets artifacts types such as SQLite databases, XML files and files found within caches identified as important. The high-level algorithm is shown in Algorithm 1.

6.2. Usage and output

ASNAAT is designed to execute on the command line terminal. A help menu is provided to show the types of image formats and flags allowed (Listing 1). The user can select either the -a flag to analyze an Apple image or -b for the Android. After the tool identifies which applications are installed from the image provided, the tool presents another menu to the user where they can select to view a quick report on the terminal for all applications or a specific one. A HyperText Markup Language (HTML) report is generated at the end of the analysis with the results from all installed applications regardless of what was selected in the terminal (See Figures B4, B6 and Appendix B).

Listing 1. Tool Terminal Menu

```
Usage:    ASNAAT.py [options] <input_file>
Example:  ASNAAT.py -a Apple.tar
Options:
          -h, --help
          -a              apple image tar
          -b              android image tar
```

7. Conclusion/discussion

As alternative social media applications appealing to certain groups of the population has increased, so has the risk of the spread of false information and radical propaganda. Applications such as Parler and MeWe have gained a substantial number of users since the 2020 United States Presidential election, and become top applications in app stores. With more concentrated groups communicating on platforms that are less regulated, the risk of propagating extremist ideologies has also grown Not only are these applications used by individuals with more conservative views, but they have also become home to anti-government extremists and white supremacists (Yurrieff et al., 2021).

The Capitol attack that occurred January 6, 2021, demonstrated how these applications were being used to communicate threatening beliefs and encourage acting upon them. Due to the popularity of these applications, it is important to conduct research and identify important artifacts that could be essential in a digital forensics investigation.

Our findings revealed that a vast amount of information can be extracted from these applications. User information such as usernames, emails, full names, phone numbers, profile pictures, and more could be found, along with posts and comments made, and private messages. Not all of the information listed was discovered for all applications, and the findings depended on the functionality provided by the applications. Not all applications had much relevant data stored, creating the opportunity to spread radical ideas. Not only was user interactions and information extracted, but vulnerabilities were discovered related to user privacy, and data was able to be accessed without proper authorization. While data was found through mobile forensics and acquisition techniques, it was noted that secure methods were used when transferring information over the network.

Finally, our work contribution consists of forensic analyses, as well as a tool written in Python that aims to automatically aggregate data based on our analysis and our artifact identification.

8. Limitations/future work

Due to the vast number of data recovered from these applications, not all of the artifacts discovered are discussed in this paper. This paper discusses the artifacts that were identified as most important across applications and devices. Moreover, many of the resulting data was redundant. To view all digital artifacts that were collected, refer to the Artifact Genome Project (https://agp.newhaven.edu/).

During the testing phase, there were updates being made. Although most updates did not alter the features, significant changes were made and compared from the preliminary testing phase for Parler. During preliminary testing, media posted by the account owner, cached media, and other user generated data including a cache.json file containing user activities and account information were discovered. However, little to no data was recovered from the newer version of the application which was updated when the application returned to the Apple store. This demonstrates how versions can have drastic differences in the potential evidence that can be uncovered, and the impact this can have on forensic investigations. Moreover, new applications are being developed more rapidly in order to fulfil the demand for counter-cultural applications. Future work should explore updated versions of the applications investigated, as well as new applications similar to the ones investigated in this paper. The tool can also be improved by adding new applications, accounting for varying versions, and more. In order to continue maintaining the tool, other future work would include continual testing and improvements based on new and updated applications.

Acknowledgements

This material is based upon work supported by National The Science Foundation under Grants No. 1900210 and 1921813. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Appendix A. Apparatus & Important Artifacts Tables

Table A.3
Apparatus

Hardware/Software	Use	Company	Software Version
Galaxy S6	Application accounts (excluding 2nd1st)	Samsung	Nougat 7.0
Android ZTE	Application accounts (excluding 2nd1st)	Samsung	Nougat 7.1.1
iPhone 6s	Application accounts (excluding Wimkin)	Apple	iOS 14.4.2
iPhone 8	Application accounts (excluding Wimkin)	Apple	iOS 14.4.2
Thinkpad X1	Acquisition and analysis	Lenovo	Windows 10
Macbook Pro	Acquisition and analysis	Apple	macOS Big Sur 11.6
Ryzen Desktop PC	Acquisition and analysis	MSI	Windows 10 Education
Ubuntu Virtual Machine	Testing and analysis	Ubuntu	Ubuntu 20.04
Windows Virtual Machine	Testing and analysis	Windows	Windows 10
VirtualBox	Host VMs for testing and analysis	Oracle	6.1
Parler	Android and iOS Parler accounts	Parler	3.0.1 (Android) & 2.50 (iOS)
MeWe	Android and iOS MeWe accounts	MeWe	2.18.30
Clouthub	Android and iOS Clouthub accounts	Clouthub	1.19.7 (Android) & 2.0.41 (iOS)
Minds Mobile	Android and iOS Minds Mobile accounts	Minds Mobile	4.14.2 (Android) & 4.20.0 (iOS)
Minds Chat	Android and iOS Minds Chat accounts	Minds Chat	1.1.7-dev (Android) & 1.6.6 (iOS)
Safechat	Android and iOS Safechat accounts	Safechat	0.9.46 (Android) & 0.9.66 (iOS)
GETTR	Android and iOS GETTR accounts	GETTR	1.0.7 (Android) & 1.2.7 (iOS)
Wimkin	Android and iOS Wimkin accounts	Wimkin	2.1
2nd1st	Android and iOS 2nd1st accounts	2nd1st	1.2.21
Android Debug Bridge (ADB)	Communicate with tool and extract application data		1.0.41
Filza File Manager	File system manager	TIGI Software	3.8
DB Browser for SQLite	View databases	DB	3.35.5
iBackup Viewer	View iOS plists	iMacTools	4.22.1
Magnet Acquire	Physical acquisition for Android and iOS	Magnet Forensics	2.46.0.28200
Autopsy	Image viewer used for analysis	The Sleuth Kit	4.19.1
Wireshark	Capture and analyze network traffic	Wireshark	3.4.8
Fiddler	Analayze network traffic	Progress Software Corporation	3.0.1
Network Miner	Analyze network traffic	Netresec	2.7.1

Table A.4
Important Data Path Directories and Files Found in Android Device

File ID	Path	Application	Description
1.1	vol20/data/com.mewe/databases/app_database	MeWe	Contains message, post, and contact information
1.2	vol20/data/com.mewe/shared_prefs/SGSession.xml	MeWe	Tokens, keys, timestamps, primary phone number
1.3	vol20/data/com.mewe/cache/image_manager_disk_cache	MeWe	Downloads videos & images posted & sent in private messages, profile pics, emojis, and cached images
2.1	vol20/data/com.clouthub.clouthub/shared_prefs/Clouthub.xml	Clouthub	User account information
2.2	vol20/data/com.clouthub.clouthub/cache/compressor	Clouthub	Downloads images posted by account owner
2.3	vol20/data/com.clouthub.clouthub/cache/image_manager_disk_cache	Clouthub	Downloads videos & images posted & sent in private messages, profile pics, emojis, and cached images
3.1	vol20/data/com.minds.mobile/databases/minds1.db	Minds Mobile	Comment and post information from feed
3.2	vol20/data/com.minds.mobile/databases/RKStorage	Minds Mobile	Contains account owner user information

(continued on next page)

Table A.4 (*continued*)

File ID	Path	Application	Description
3.3	vol20/data/com.minds.mobile/cache/react-native-image-crop-picker	Minds Mobile	Downloads images & videos posted by account owner
3.4	vol20/data/com.minds.mobile/cache/image_manager_disk_cache	Minds Mobile	Downloads videos & images from feed and cached images
4.1	vol20/data/com.minds.chat/cache/downloads/83e36ef56991d5c448f05d365f2212c9/F/D/	Minds Chat	Stores folders that contain images downloaded from chat
4.2	vol20/data/com.minds.chat/shared_prefs/emoji-recent-manager.xml	Minds Chat	Recent emoji used
4.3	vol20/data/com.minds.chat/image_manager_disk_cache	Minds Chat	Downloads videos & images sent in private messages and cached images
5.1	vol20/data/net.safechat.app/cache/SafeChat	Safechat	Downloads videos posted and sent in private messages
5.2	vol20/data/net.safechat.app/cache	Safechat	Downloads audio sent in private messages
5.3	vol20/data/net.safechat.app/databases/SafeChat.db	Safechat	Private Messages and user information
5.4	vol20/data/net.safechat.app/databases/download_tasks.db	Safechat	Downloaded media from messages
6.1	vol20/data/com.gettr.gettr/files/libCachedImageData.db	GETTR	GIFs posted by account owner
6.2	vol20/data/com.gettr.gettr/databases/private_hughhen123.db	GETTR	Followed list, username, searches
6.3	vol20/data/com.gettr.gettr/databases/g.db	GETTR	User information for account owner and ‘friend’
6.4	vol20/data/com.gettr.gettr/cache/libCachedImageData	GETTR	Downloads GIFs posted by account owner
6.5	vol20/data/com.gettr.gettr/shared_prefs/giphy_recents_file.xml	GETTR	Account owner's recently posted GIF ID
6.6	vol20/data/com.gettr.gettr/shared_prefs/giphy_searches_file.xml	GETTR	Account owner's GIF searches
6.7	vol20/data/com.gettr.gettr/cache/image_manager_disk_cache	GETTR	Downloads images posted, profile pics, emojis, phone gallery images, cached images
7.1	vol20/data/com.wimkin.android/databases/RKStorage	Wimkin	Contains account owner's user information
7.2	vol20/data/com.wimkin.android/cache/image-cache/v2.ols100.1	Wimkin	Stores folders containing cached images and images and document sent in private messages
7.3	vol20/data/com.wimkin.android/cache/react-native-image-crop-picker	Wimkin	Downloads videos sent in private messages
7.4	vol20/data/com.wimkin.android/cache	Wimkin	Downloads audio message sent in private messages
7.5	vol20/data/com.wimkin.android/shared_prefs/rocketUser.xml	Wimkin	User ID and authentication token
7.6	vol20/data/com.wimkin.android/shared_prefs/io.invertase.firebase.xml	Wimkin	Notification information
7.7	vol20/data/com.wimkin.android/chatplus-chat.wimkin.com.db.db	Wimkin	Chat message data

Table A.5
Important Data Path Directories and Files Found in iPhone Device

File ID	Path	Application	Description
1.1	private/var/mobile/Containers/Data/Application“UUID”/Documents/sgrouplesdb.sqlite	MeWe	User, chat, post and contact information
1.2	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.mewe/Cache.db	MeWe	Information from account owner and ‘friend’, post and comment information from feed, private messages
1.3	private/var/mobile/Containers/Data/Application/“UUID”/tmp	MeWe	Downloads image posted and sent in private messages by account owner
1.4	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.hackemist.SDImageCache/default	MeWe	Downloads image posted by account owner, GIF sent in private messages, and cached GIFs and images
2.1	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.clouthub.clouthubapp.Cache.db	Clouthub	Information regarding account owner and private messages
2.2	private/var/mobile/Containers/Data/Application/“UUID”/tmp	Clouthub	Downloads images posted and sent in private messages
2.3	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.hackemist.SDImageCache/default	Clouthub	Downloads profile pictures, and cached images
3.1	private/var/mobile/Containers/Data/Application/“UUID”/Library/LocalDatabase/minds1.db	Minds Mobile	Comment and post information from account owner and feed
3.2	private/var/mobile/Containers/Data/Application/“UUID”/tmp/react-native-image-crop-picker	Minds Mobile	Downloads image and video posted by account owner
3.3	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.minds.mobile/fsCachedData	Minds Mobile	Downloads cached images
3.4	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.hackemist.SDImageCache/default	Minds Mobile	Downloads cached images
4.1	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches	Minds Chat	Downloads videos sent in private messages
4.2	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/com.minds.chat/fsCachedData	Minds Chat	Downloads cached images
5.1	private/var/mobile/Containers/Data/Application/“UUID”/tmp/.video	Safechat	Downloads videos posted and sent in private messages
5.2	private/var/mobile/Containers/Data/Application/“UUID”/tmp	Safechat	Downloads audio Message sent in private messages
5.3	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/imagecache	Safechat	Downloads cached images
5.4	private/var/mobile/Containers/Shared/AppGroup/“UUID”/SafeChat.db	Safechat	Private Messages and user information
6.1		GETTR	Followed list, username, searches

Table A.5 (*continued*)

File ID	Path	Application	Description
	private/var/moible/Containers/Data/Application/“UUID”/Documents/ private_jmanny3.db		
6.2	private/var/moible/Containers/Data/Application/“UUID”/Documents/g.db	GETTR	User information for account owner and ‘friend’
6.3	private/var/mobile/Containers/Data/Application/“UUID”/Library/Caches/ com.gettr.gettr/Cache.db	GETTR	Post and comment data from feed
6.4	private/var/moible/Containers/Data/Application/“UUID”/tmp/flutter-images	GETTR	Downloads image posted by account owner
6.5	private/var/moible/Containers/Data/Application/“UUID”/tmp/.image	GETTR	Downloads image posted by account owner
6.6	private/var/moible/Containers/Data/Application/“UUID”/tmp/.video	GETTR	Download Video posted by account owner
6.7	private/var/moible/Containers/Data/Application/“UUID”/Library/Caches/ com.gettr.gettr/fsCachedData	GETTR	Downloads cached images
6.8	private/var/moible/Containers/Data/Application/“UUID”/Library/Caches/ libCachedImageData	GETTR	Downloads GIFs posted by account owner
6.9	private/var/moible/Containers/Data/Application/“UUID”/Library/Caches/ com.hackemist.SDImageCache/default	GETTR	Downloads cached images and images posted by account owner
7.1	private/var/mobile/Containers/Data/Application/“UUID”/tmp	2nd1st	Downloads image posted by account owner
8.1	private/var/moible/Containers/Data/Application/“UUID”/Library/WebKit/ WebsiteData/LocalStorage/file_0.localstorage	Parler	Contains email and hashes
8.2	private/var/moible/Containers/Data/Application/“UUID”/Library/WebKit/ WebsiteData/LocalStorage/https_parler.com_0.localstorage	Parler	Contains email, tokens, and hashes
8.3	private/var/moible/Containers/Data/Application/“UUID”/tmp	Parler	Downloads image posted by account owner

Appendix B. ASNAAT Tool Output

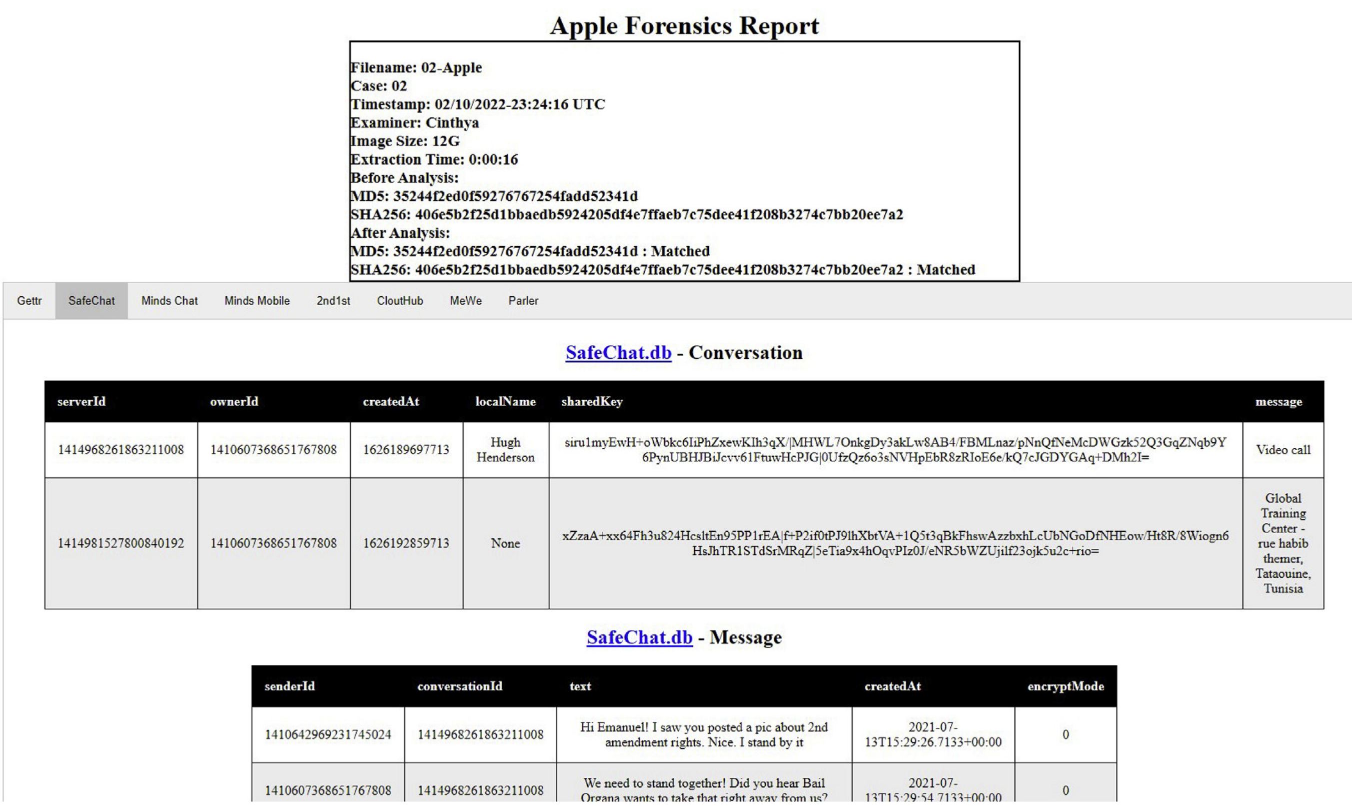
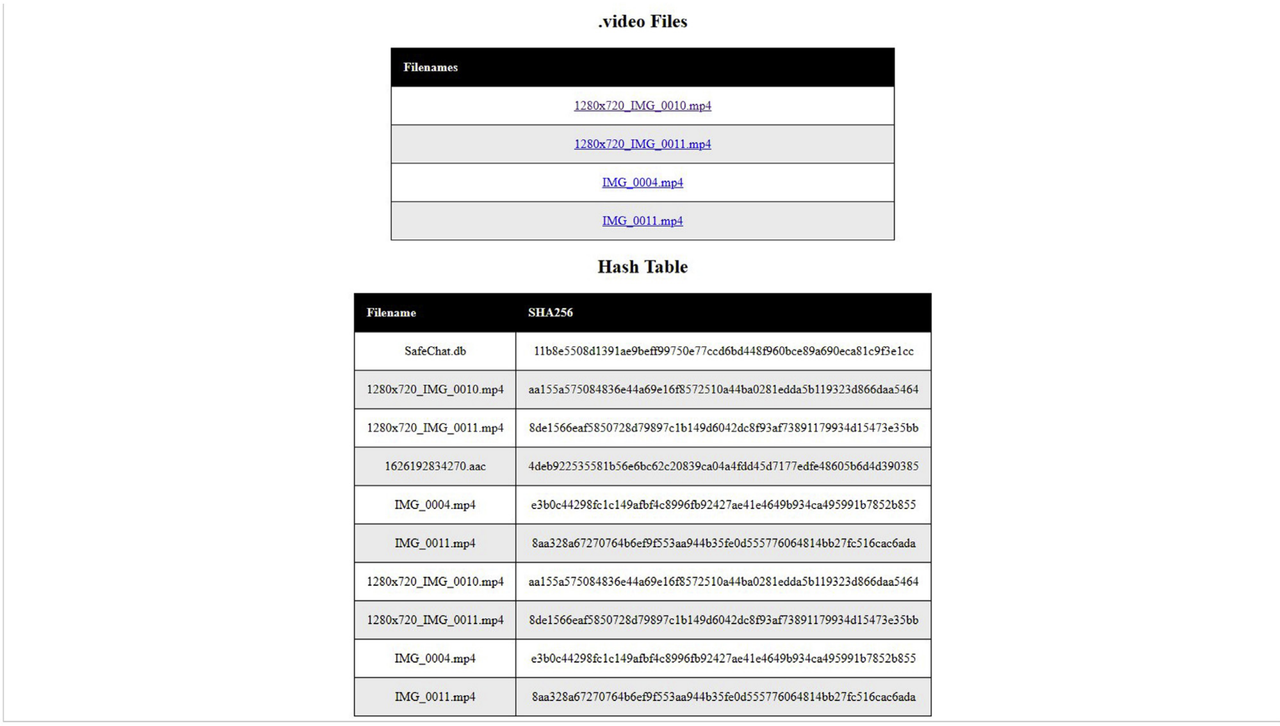


Fig. B.3. iPhone report output.



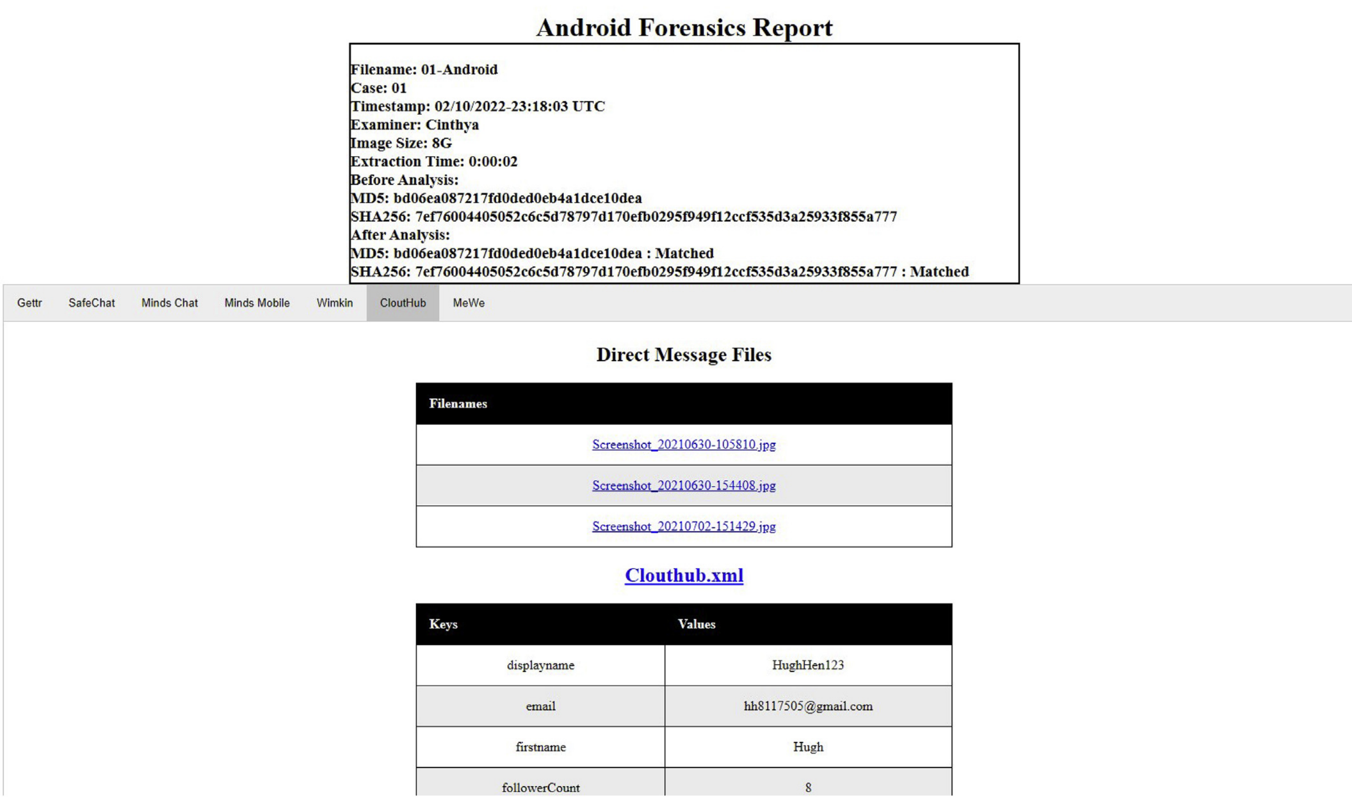
.video Files

Filenames
1280x720_IMG_0010.mp4
1280x720_IMG_0011.mp4
IMG_0004.mp4
IMG_0011.mp4

Hash Table

Filename	SHA256
SafeChat.db	11b8e5508d1391ae9beff99750e77ccd6bd448f960bce89a690eca81c9f3e1cc
1280x720_IMG_0010.mp4	aa155a575084836e44a69e16f8572510a44ba0281edda5b119323d866daa5464
1280x720_IMG_0011.mp4	8de1566eaf5850728d79897c1b149d6042dc8f93af73891179934d15473e35bb
1626192834270.aac	4deb922535581b56e6bc62c20839ca04a4fdd45d7177edfe48605b6d4d390385
IMG_0004.mp4	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
IMG_0011.mp4	8aa328a67270764b6ef9f553aa944b35fe0d555776064814bb27fc516cac6ada
1280x720_IMG_0010.mp4	aa155a575084836e44a69e16f8572510a44ba0281edda5b119323d866daa5464
1280x720_IMG_0011.mp4	8de1566eaf5850728d79897c1b149d6042dc8f93af73891179934d15473e35bb
IMG_0004.mp4	e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
IMG_0011.mp4	8aa328a67270764b6ef9f553aa944b35fe0d555776064814bb27fc516cac6ada

Fig. B.4. iPhone report output continued.



Android Forensics Report

Filename: 01-Android
Case: 01
Timestamp: 02/10/2022-23:18:03 UTC
Examiner: Cinthya
Image Size: 8G
Extraction Time: 0:00:02
Before Analysis:
MD5: bd06ea087217fd0ded0eb4a1dce10dea
SHA256: 7ef76004405052c6c5d78797d170efb0295f949f12ccf535d3a25933f855a777
After Analysis:
MD5: bd06ea087217fd0ded0eb4a1dce10dea : Matched
SHA256: 7ef76004405052c6c5d78797d170efb0295f949f12ccf535d3a25933f855a777 : Matched

Gettr SafeChat Minds Chat Minds Mobile Wimkin CloutHub MeWe

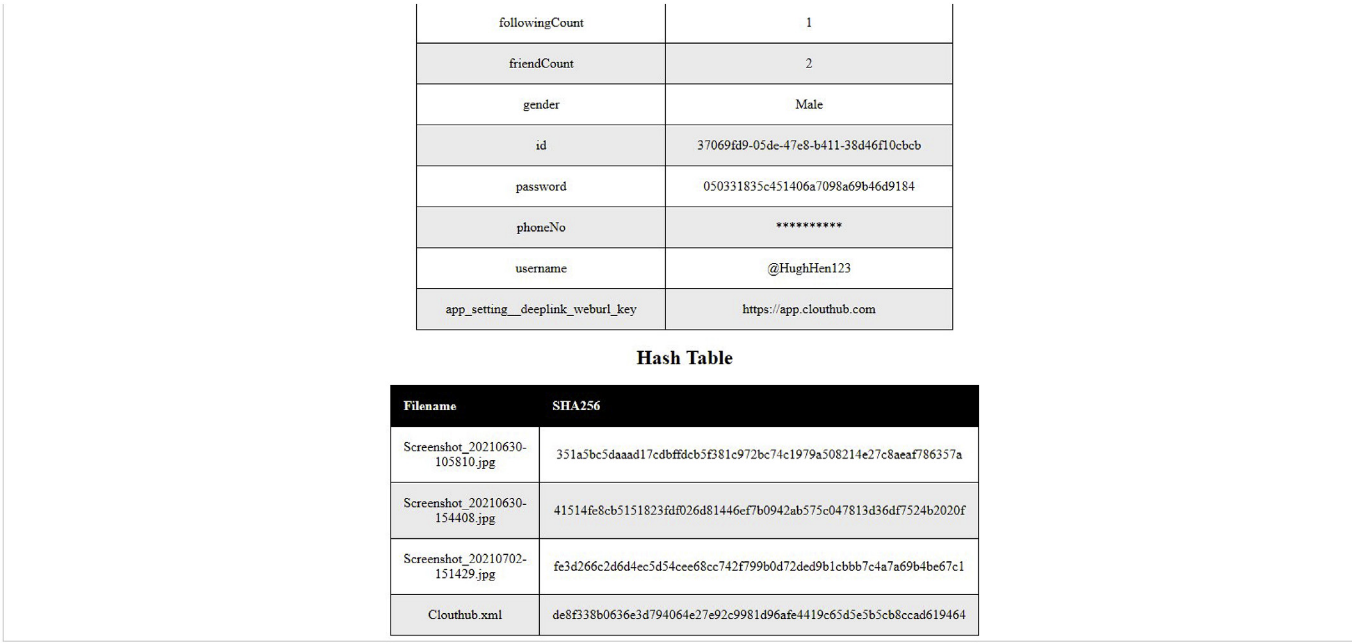
Direct Message Files

Filenames
Screenshot_20210630-105810.jpg
Screenshot_20210630-154408.jpg
Screenshot_20210702-151429.jpg

Clouthub.xml

Keys	Values
displayname	HughHen123
email	hh8117505@gmail.com
firstname	Hugh
followerCount	8

Fig. B.5. Android report output.



followingCount	1
friendCount	2
gender	Male
id	37069fd9-05de-47e8-b411-38d46f10cbcb
password	050331835c451406a7098a69b46d9184
phoneNo	**********
username	@HughHen123
app_setting__deeplink_weburl_key	https://app.clouthub.com

Hash Table

Filename	SHA256
Screenshot_20210630-105810.jpg	351a5bc5daaad17cdbffdcb5f381c972bc74c1979a508214e27c8aeaf786357a
Screenshot_20210630-154408.jpg	41514fe8cb5151823fdf026d81446ef7b0942ab575c047813d36df7524b2020f
Screenshot_20210702-151429.jpg	fe3d266c2d6d4ec5d54cee68cc742f799b0d72ded9b1cbbb7c4a7a69b4be67c1
Clouthub.xml	de8f338b0636e3d794064e27e92c9981d96afe4419c65d5e5b5cb8ccad619464

Fig. B.6. Android report output continued.

References

Al Mutawa, N., Ibrahim, B., Marrington, A., 2012. Forensic analysis of social networking applications on mobile devices. Digit. Invest. 9, 742–2876.

Aliaoulios, M., Bevensee, E., Blackburn, J., Bradlyn, B., De Cristofaro, E., Stringhini, G., Zannettou, S., 2021. A large open dataset from the parler social network. In: Proceedings of the International AAAI Conference on Web and Social Media (ICWSM2021), pp. 943–951.

Alisabeth, C., Pramadi, Y.R., 2020. Forensic analysis of instagram on android. In: IOP Conference Series: Materials Science and Engineering. IOPscience.

Arista Yuliani, V., Riadi, I., 2019. Forensic analysis whatsapp mobile application on android-based smartphones using national institute of standard and technology (nist) framework. Int. J. Cyber Secur. Digit. Forensic. Soc. Digital 8.

Baggili, I., Oduro, J., Anthony, K., Breitinger, F., McGee, G., 2015. Watch what you wear: preliminary forensic analysis of smart watches. In: Availability, Reliability and Security (ARES), 2015 10th International Conference on. IEEE, pp. 303–311.

Barrett, B., 2021. Whatsapp fixes its biggest encryption loophole. https://www.wired.com/story/whatsapp-end-to-end-encrypted-backups/.

Brown, A., 2021. Conservative social media app parler is pretty much dead. https://www.forbes.com/sites/abrambrown/2021/01/10/conservative-social-media-app-parler-is-pretty-much-dead/?sh=16396faf6a53.

Casey, P., Baggili, I., Yarramreddy, A., 2019a. Immersive virtual reality attacks and the human joystick. IEEE Trans. Dependable Secure Comput. 1, 1.

Casey, P., Lindsay-Decusati, R., Baggili, I., Breitinger, F., 2019b. Inception: virtual space in memory space in real space–memory forensics of immersive virtual reality with the htc vive. Digit. Invest. 29, S13–S21.

Chung, H., Park, J., Lee, S., 2017. Digital forensic approaches for amazon alexa ecosystem. Digit. Invest. 22.

Clark, D.R., Meffert, C., Baggili, I., Breitinger, F., 2017. Drop (drone open source parser) your drone: forensic analysis of the dji phantom iii. Digit. Invest. 22, S3–S14.

Datareportal, 2021. Global social media stats. https://datareportal.com/social-media-users.

Dickson, E., 2020. Free speech' social-media apps see enormous growth after the election. https://www.rollingstone.com/culture/culture-features/trump-election-facebook-twitter-mewe-parler-1088427/.

Dorai, G., Houshmand, S., Baggili, I., 2018. I know what you did last summer: your smart home internet of things and your iphone forensically ratting you out. In: 'Proceedings of the 13th International Conference on Availability, Reliability and Security', ARES 2018. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3230833.3232814.

Erbschloe, M., 2019. Extremist Propaganda in Social Media. Taylor & Francis Group.

Ferrara, E., 2017. Contagion dynamic of extremist propaganda in social networks. Inf. Sci. 418–419, 1–12.

Greenberg, A., 2021. An absurdly basic bug let anyone grab all of parler's data. https://www.wired.com/story/parler-hack-data-public-posts-images-video/.

Johnson, D., 2020. What is a cache? a complete guide to caches and their important uses on your computer, phone, and other devices. https://www.businessinsider.com/what-is-cache.

Karpisek, F., Baggili, I., Breitinger, F., 2015. Whatsapp network forensics: decrypting and understanding the whatsapp call signaling messages. Digit. Invest. 15, 110–118.

Longhi, J., 2021. Using digital humanities and linguistics to help with terrorism investigations. Forensic Sci. Int. 318.

Mahajan, A., Dahiya, M., Sanghvi, H., 2013. Forensic analysis of instant messenger applications on android devices. Int. J. Comput. Appl. 68.

Mahr, A., Cichon, M., Grajeda, C., Baggili, I., 2021. Zooming into the pandemic! a forensic analysis of the zoom application. Forensic Sci. Int.: Digit. Invest. 36.

Mak, T., 2021. Across the internet, a game of whac-a-mole is underway to root out extremism. https://www.npr.org/2021/03/16/972519460/across-the-internet-a-game-of-whac-a-mole-is-underway-to-root-out-extremism.

Menahil, A., Iqbal, W., Iftikhar, M., Bin Shahid, W., Mansoor, K., Rubab1, S., 2021. Forensic analysis of social networking applications on an android smartphone. Wireless Commun. Mobile Comput. 2021.

Newhouse, A., 2020. What to know about parler, the social-media platform that is now attracting millions of trump supporters. https://www.marketwatch.com/story/what-to-know-about-parler-the-social-media-platform-that-is-attracting-millions-of-trump-supporters-11606497149.

Ovens, K.M., Morison, G., 2021. Forensic analysis of instant messaging apps: decrypting wickr and private text messaging data. Digit. Invest. 17, 40–50.

Richard Jones, K., 2017. Law enforcement use of social media as a crime fighting tool. https://core.ac.uk/reader/84755700.

Salamh, E., Mirza, Meraj, Hutchinson, S., Han Yoon, Y., Karabiyik, U., 2021. What's on the horizon? an in-depth forensic analysis of android and ios applications. IEEE Access 99, 1–1.

Shalvey, K., 2021. Wimkin, a free speech network, says it was hit with a 'massive' ddos attack after being banned from apple's app store. https://www.businessinsider.com/apple-google-removed-wimkin-app-founder-reports-ddos-attack2021-1.

Silberling, A., 2022. Right-wing social app parler raises $20 million in funding. https://techcrunch.com/2022/01/07/right-wing-social-app-parler-raises-20m-in-funding/.

University, S., 2016. Fighting crime with mobile technology. https://www.southuniversity.edu/news-and-blogs/2016/08/fighting-crime-with-mobile-technology-137309.

Walnycky, D., Baggili, I., Marrington, A., Moore, J., Breitinger, F., 2015. Network and device forensic analysis of android social-messaging applications. Digit. Invest. https://www.sciencedirect.com/science/article/pii/S1742-287615000547.

Yarramreddy, A., Gromkowski, P., Baggili, I., 2018. Forensic analysis of immersive virtual reality social applications: a primary account. In: '2018 IEEE Security and Privacy Workshops (SPW)'. IEEE, pp. 186–196.

Yıldırım, I., Bostancı, E., Güzel, M.S., 2019. Forensic analysis with anti-forensic case studies on amazon alexa and google assistant build-in smart home speakers. In: '2019 4th International Conference on Computer Science and Engineering. UBMK)', pp. 1–3.

Yurrieff, K., Fung, B., O'Sullivan, D., 2021. Parler: everything you need to know about the banned conservative social media platform. https://www.cnn.com/2021/01/10/tech/what-is-parler/index.html.