Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Forensic Science International: Digital Investigation 36 (2021) 301107

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

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

Zooming into the pandemic! A forensic analysis of the Zoom Application

Andrew Mahr, Meghan Cichon, Sophia Mateo, Cinthya Grajeda, Ibrahim Baggili Elder Family Endowed Chair *

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

ARTICLE INFO

Article history:
Received 24 July 2020
Accepted 5 January 2021
Available online 23 January 2021

Keywords:
Network
Disk
Memory forensics
Artifacts
Zoom video conferencing

ABSTRACT

The global pandemic of COVID-19 has turned the spotlight on video conferencing applications like never before. In this critical time, applications such as Zoom have experienced a surge in its user base jump over the 300 million daily mark (ZoomBlog, 2020). The increase in use has led malicious actors to exploit the application, and in many cases perform *Zoom Bombings*. Therefore forensically examining Zoom is inevitable. Our work details the primary disk, network, and memory forensic analysis of the Zoom video conferencing application. Results demonstrate it is possible to find users' critical information in plain text and/or encrypted/encoded, such as chat messages, names, email addresses, passwords, and much more through network captures, forensic imaging of digital devices, and memory forensics. Furthermore we elaborate on interesting anti-forensics techniques employed by the Zoom application when contacts are deleted from the Zoom application's contact list.

1. Introduction

Digital evidence acquired from video conferencing applications may prove useful in investigations and are used by individuals in all sectors. Applications like Skype, Google Video/Messaging series, and Microsoft Teams have been more commonly used in recent years (Abbott, 2020). Due to the COVID-19 pandemic,[1] many schools, businesses, and people have turned to the Zoom video conferencing application to communicate with one another. This rapid increase of user traffic has led to scrutiny and suspicion regarding the cybersecurity practices of the company after major exploits were found within its protocols. These security issues have led to privacy breaches committed through *Zoom Bombings* (O'Flaherty, 2020; Lorenz and Alba, 2020) and the exploitation of basic protocols. *Zoom Bombings* involve unwanted conference disruptions of any kind, including, but not limited to, the projection of illicit images and the use of verbal profanity, which could be a form of criminal harassment (Office, 2020; Setera, 2020).

The most notable security issues come in Common Vulnerabilities and Exposures (CVE) reports. Zoom published a security report in 2018 detailing two major CVEs. CVE-2018-15715[2] showed how malicious actors could take control of users' screens, spoof chat messages, and control other aspects of the meeting. CVE-2020-11443[3] detailed how the Windows Zoom IT Installer, which deletes files and data before reinstalling Zoom, could be exploited to delete files a user would not normally be allowed to delete. Additional vulnerabilities were found in the Zoom application and Zoom has responded with patches for these issues (Zoom, 2020*c*).

As video conferencing applications continue to be the main communication method during events such as this pandemic, it is important that we understand the forensic artifacts produced by these systems. Our work aims to investigate the digital evidence produced by the Zoom application and provides an analysis of the critical data that can be found. The devices investigated in our work include a Samsung Galaxy S6, an iPhone 5s, a Windows 10 Virtual Machine (VM), and an Apple MacBook Pro.

To date, and to the best of our knowledge, there has not been a formal forensic analysis of the Zoom application and therefore our work contributes as follows:

* Corresponding author.
E-mail address: ibaggili@newhaven.edu (I. Baggili).

[1] https://www.cdc.gov/coronavirus/2019-ncov/index.html.

[2] https://support.zoom.us/hc/en-us/articles/360020436071-Security-CVE-2018-15715.

[3] https://support.zoom.us/hc/en-us/articles/360043036451-Security-CVE-2020-11443.

https://doi.org/10.1016/j.fsidi.2021.301107
2666-2817/

- A primary disk, memory, and network forensic analysis of the Zoom platform.
- A collection of Zoom application digital forensic artifacts shared on the Artifact Genome Project[4] (Grajeda et al., 2018).
- A collection of SQL queries that can be used by digital forensic investigators to extract relevant data from the application databases.

This paper is organized as follows. Section 2 presents previous research and other related work. Section 3, outlines the tools used to conduct our research. Section 4, discusses the applied methodology, while Section 5, discusses our analysis and results. Section 6, provides SQL queries which aim to speed up relevant data acquisition during investigations. Lastly, Section 7, concludes our work while Section 8 presents future work.

2. Related work

To the best of our knowledge, our methodical analysis of the Zoom application is the first of its kind involving multiple device platforms. Existing research on related applications is also limited to the Skype application, even though there are multiple types of applications used by millions to communicate with others.

The next subsections highlight related research conducted on similar applications.

2.1. Video conferencing applications

Research shows that for the last ten years, Skype has emerged from the rest as being one of the most forensically examined video conferencing application. Skype was created sixteen years ago (Whent, 2012).

For instance, Simon and Slay (2010) examined the process used to acquire the physical memory locations and application data of Skype within Android and Windows devices. Al-Saleh and Forihat (2013) explored the flash memory Skype artifacts on Android concluding that there is a persistence pattern used by the Skype application. They found evidence of Skype calls, chats, and meeting IDs in NAND and RAM many hours after the calls and chats took place.

On the other hand, Azab et al. (2012) characterized network traffic from the Skype application and demonstrated the difficulties forensic experts face when trying to intercept or analyze this traffic. The work also identified and discussed the differences discovered in the traffic between older and different versions of the Skype application. Subsequently, Majeed (2016) explored the behavior of Skype, Facebook, and Twitter within the Windows 10 environment. It was discovered that Skype stored plaintext chat messages as well as other information pertaining to a user on disk.

In the last ten years, research related to the forensics of video conferencing applications heavily focused on Skype (Levinson et al., 2011; Chang et al., 2013; Teng and Lin, 2012; Al Barghuthi and Said, 2013). To elaborate on that literature is beyond the scope of our work.

2.2. Messaging & social media applications

As mobile adoption increased, forensics research followed that trend and focused on social messaging mobile applications. Similar to Skype, research has shown that these types of applications also store important user information on the device.

For instance, Walnycky et al. (2015) investigated the security and forensics of social messaging applications such as WhatsApp, Viber, Tango, and ooVoo. Their work concluded it was possible to find user information within the application data folders. Additionally, it demonstrated some of these applications stored publicly accessible user data on their servers and transmitted plaintext information on the network.

Primary work was also conducted on the network forensics of WhatsApp, and focused on decrypting the WhatsApp call signaling protocol (Karpisek et al., 2015). The researchers described how to decrypt the network traffic and obtain forensic artifacts that relate to: a) WhatsApp phone numbers, b) WhatsApp server IPs, c) WhatsApp audio codec (Opus), d) WhatsApp call duration, and e) WhatsApp's call termination.

Similarly, Anglano et al. (2017) investigated the Telegram application and showed that message history, contacts and other user information may be reconstructed by forensic examiners.

Lastly, Al Mutawa et al. (2012) conducted a primary analysis of social networking applications on mobile devices. Their work demonstrated that user information such as user settings, chat messages, and timestamps could be found in plaintext stored in two of the examined devices, except for the Blackberry device.

2.3. Other related applications

During the last decade, similar research has also been conducted on other types of devices that may be used for communication. Some examples include, Android vault applications (Zhang et al., 2017), smartwatch devices (Baggili et al., 2015), portable web browsers (Marrington et al., 2012), drones (Clark et al., 2017), Amazon Kindle (Iqbal et al., 2014), health and fitness applications (Hassenfeldt et al., 2019), home IoT devices (Dorai et al., 2018), Amazon's Echo Dot (Chung et al., 2017), virtual reality (Casey, Baggili and Yarramreddy, 2019; Casey, Lindsay-Decusati, Baggili and Breitinger, 2019; Yarramreddy et al., 2018) and more.

3. Apparatus

The hardware and software used to conduct this research are presented on Table A.4, Appendix A.

4. Methodology

Forensic research of the Zoom application was conducted in four phases: scenario creation and setup, data acquisition, data analysis, and SQLite database query creation. Due to experiments being conducted at a time when Zoom was constantly updating and patching the application's latest vulnerabilities in all OS platforms,[5] not all tests were conducted on the same version of the software. In fact, it was decided not to update at all to the latest version, until all tests were finalized. Nonetheless, even after declining to update the Zoom application after each use, Zoom forced an update to the newest version. Surprisingly, this happened only in the Android and Windows VM, and not the Apple devices. The newest version tested at the time was 5.1.2, and was tested across all devices to note any differences between Zoom versions. We note that only limited tests that yielded important results from previous versions were conducted with the latest version of Zoom.

Table A.4 in Appendix A demonstrates all tools used to conduct this research. The devices tested were used to simulate various use cases of the Zoom platform. Details of these four phases and results are found in the next two subsections and Sections 5 and 6.

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

[5] https://support.zoom.us/hc/en-us/articles/201362233-Where-Do-I-Download-The-Latest-Version.

4.1. Setup & scenario creation

This phase consisted of testing the Zoom application's features on all devices by mimicking free Basic and Licensed account usage. To acquire a complete dataset, all mobile devices were first reset and rooted. Moreover, to test the desktop applications, a macOS laptop was used and a clean Windows 10 Virtual Machine was downloaded from the Microsoft Developer's website.[6] Additionally, all Zoom applications were downloaded from the Zoom website and respective mobile stores.[7]

Creating each scenario stemmed from testing common user actions to more advanced features that Basic Zoom accounts did not include. Thus, Basic accounts were created and tested. Then, all accounts were switched to Licensed University accounts that all students at the University *of* New Haven[8] possess. These Licensed accounts are now used by the University to conduct remote online learning activities.

Within these test environments, different application features and settings were examined. Tests were conducted with the devices communicating as a group, one-on-one, and individually to allow for an understanding of the interactions between different device platforms. All of the tests were conducted by creating meetings that used a mix of each device's Personal Meeting IDs and General Meeting IDs generated by Zoom. These meetings were created through the Zoom application, scheduled using the Outlook Calendar plugin, and started through the contacts page of Zoom. The following tested features yielded the most important results:

- Added contacts
- Deleted contacts
- Searched for keywords using the application's Search feature
- Chatted through the Chat feature only
  - Exchanged text files and other types of files, such as pictures
  - Exchanged screenshots taken in the chat
  - Exchanged URLs
- Conducted a Zoom video meeting and sent and received chat messages and files
- Saved in video meetings locally and to the cloud
- Installed the Zoom Outlook plugin to schedule meetings
- Implemented the Twitter application from the Zoom Marketplace and tested the following:
  - Sent tweets
  - Started meeting through Twitter chat bot
- Attended a webinar as an attendee and panelist

4.2. Data acquisition

In this phase, network and disk forensics were performed on all devices with some limitations, while memory dumps were captured only on the Windows Virtual Machine.

To acquire network traffic from exchanged Zoom communications, a unique wireless hotspot was created to isolate each device's network. To confirm all Zoom's network traffic was encrypted, Wireshark was used to capture the packets while each test took place. We used Fiddler[9] to also capture, decrypt, and decode HTTPS network traffic. Fiddler decrypts HTTPS traffic by generating a root certificate that the user is required to trust on the device under analysis. For example, when using Windows, it imports "the generated root certificate into the current user's Trusted Root Certification Authorities store" (Lawrence, 2019). At the time of testing Fiddler, the latest version of Zoom was 5.0.2, and it was only successfully tested on desktop applications. Unfortunately, Fiddler and Zoom did not work well together through the Fiddler proxy when using the mobile applications, thus, the mobile traffic captured turned out to be unfruitful for this research. Nevertheless, network traffic packets containing critical data may be similar regardless of the type of device used as it was confirmed using two different operating systems, macOS and Windows.

Subsequently, FTK Imager was used at the end of each major round of testing in the Windows VM to capture its physical disk image. FTK and Comae DumpIt tools were also used to acquire the VM's memory when the application was active and terminated. Finally, Magnet Acquire was used to collect a physical image of the Android and a logical image of the iOS device. It is important to note that even though the iPhone was jailbroken, Magnet Acquire only offered support to acquire a logical image of the device (Magnet Forensics, 2020). After conducting preliminary tests, it was concluded that the macOS and iOS application data were similar and therefore the decision was made not to physically image the macOS device. The macOS data directory was then acquired logically using the file system.

5. Analysis & experimental results

In order to analyze and extract relevant artifacts from all the forensic acquisitions, different tools shown in Table A.4 in Appendix A were utilized along with some manual analysis. In this section, details on artifacts found across all devices are summarized in their own subsections related to disk, network, and memory. It is important to note that most of the artifacts found were similar across tested devices. We will elaborate on any artifacts that were deemed unique to a specific device. All major artifacts and their file paths found within the tested devices are highlighted in Tables 1 and 2.

Table 1 contains details regarding the location of important artifacts found on the disk of their corresponding device. Table 2 lists important data found within the files stored on disk, memory dumps, and network traffic.

5.1. Zoom data directory structure

To identify major artifacts and the location they were stored in all devices, it is critical to understand how the Zoom application organizes this data. In each device's respective Zoom data directory, there were numerous folders created containing different types of files. It appears the application names main directories, some database files and some of its tables after the account's Jabber ID (JID), such as "9z4z2l54qbswpudnk0r_ba@xmpp.zoom.us"; . This JID uniquely identifies individual users, as well as user chat groups within the stored Zoom data. JIDs are the user's Extensible Messaging and Presence Protocol (XMPP) chat addresses. JID values are constructed first with the "localpart", which in this case would be "9z4z2l54qbswpudnk0r_ba", the domain part, and resource part followed after the "@" character (Saint-Andre, 2011). It is uncertain what type of encoding or encryption Zoom uses to create the JID's local values.

Analysis of the Zoom directory on each device confirmed that Zoom creates separate data folders for each account that was logged into the device. Since two types of user accounts were tested, a Basic and a Licensed school account, separate file folders were found for both accounts. Note, some of these actions do not occur unless the user is logged into the Zoom application. If no account is logged-in, then Zoom uses its default zoomus.db and

[6] f.
[7] https://zoom.us/download#client_4meeting.
[8] https://www.newhaven.edu/.
[9] https://www.telerik.com/fiddler.

Table 1
Important data path directories and files found in disk across device.

File ID	Path	Account	Device	Description
1.1	vol_vol20/data/us.zoom.videomeetings/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.asyn.db	Any	Android	Chats
1.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/“USER JID”.xmpp.zoom.us/ “USER JID”@zoom.us.asyn.db	Any	iOS	…
1.3	Library/Application Support/zoom.us/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.asyn.db	Any	macOS	…
1.4	/AppData/Roaming/Zoom/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.asyn.db	Any	Windows	…
2.1	vol_vol20/data/us.zoom.videomeetings/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.db	Any	Android	Contacts
2.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/“USER JID”.xmpp.zoom.us/ “USER JID”@xmpp.zoom.us.db	Any	iOS	…
2.3	Library/Application Support/zoom.us/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.db	Any	macOS	…
2.4	/AppData/Roaming/Zoom/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.db	Any	Windows	…
3.1	vol_vol20/data/us.zoom.videomeetings/data/“USER JID”@xmpp.zoom.us.idx.db	Any	Android	Index Information and Cached Data
3.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/“USER JID”.xmpp.zoom.us/ “USER JID”@xmpp.zoom.idx.db	Any	iOS	…
3.3	Library/Application Support/zoom.us/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.idx.db	Any	macOS	…
3.4	/AppData/Roaming/Zoom/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.idx.db	Any	Windows	…
4.1	vol_vol20/data/us.zoom.videomeetings/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.sync.db	Any	Android	Msg Invitations and Contact Requests
4.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/“USER JID”.xmpp.zoom.us/ “USER JID”@xmpp.zoom.us.sync.db	Any	iOS	…
4.3	Library/Application Support/zoom.us/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.sync.db	Any	macOS	…
4.4	/AppData/Roaming/Zoom/data/“USER JID”@xmpp.zoom.us/”; USER JID”@xmpp.zoom.us.sync.db	Any	Windows	…
5.1	vol_vol/data/us.zoom.videomeetings/data/zoommeeting.db	Any	Android	In-Meeting Encoded or Encrypted Chats
5.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/zoommeeting.db	Any	iOS	…
5.3	Library/Application Support/zoom.us/data/zoommeeting.db	Any	macOS	…
5.4	/AppData/Roaming/Zoom/data/zoommeeting.db	Any	Windows	…
6.1	vol_vol20/data/us.zoom.videomeetings/data/zoomus.db	Any	Android	User Account Information
6.2	private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/zoomus.db	Any	iOS	…
6.3	Library/Application Support/zoom.us/data/zoomus.db	Any	macOS	…
6.4	/AppData/Roaming/Zoom/data/zoomus.db	Any	Windows	…
7.1	/vol_vol20/data/data/us.zoom.videomeetings/data/“Hashed File Name”.db	Any	Android	Temporary Webinar Database
7.2	/private/var/mobile/Containers/Data/Application/“Container ID”/Documents/data/“Hashed File Name”	Any	iOS	…
7.3	Library/Application Support/zoom.us/data/“Hashed File Name”.db	Any	macOS	…
7.4	/AppData/Roaming/Zoom/data/“User JID”@xmpp.zoom.us/“; Hashed File Name”.db	Any	Windows	…
8.1	vol_vol20/data/us.zoom.videomeetings/files/data/SSBAvatarCacheIndex.ini	Any	Android	Avatar URL Cache Index
8.2	private/var/mobile/Containers/Data/Application/“Application ID”/Library/Preferences/Avatar Cache Index.plist	Any	iOS	…
8.3	Library/Preferences/Avatar Cache Index.plist/	Any	macOS	…
8.4	/AppData/Roaming/Zoom/data/SSBAvatarCacheIndex.ini	Any	Windows	…
9	/private/var/mobile/Containers/Data/Application/“Container ID”/Library/Preferences/ us.zoom.videomeetings.plist	Any	iOS	Recent Meeting Settings and Actions
10	/private/var/moible/Containers/Shared/AppGroup/“App Group ID”/Library/Caches/contacts.db	Any	iOS	bpList File of Contact Names & JIDS
11	/AppData/Roaming/Zoom Plugin/ex2smtp.json	Licensed	Windows	Outlook Plugin JSON
12	/AppData/Roaming/Zoom Plugin/userSetting.json	Licensed	Windows	Outlook Plugin JSON
13	/AppData/Roaming/Zoom Plugin/alternateHosts.json	Licensed	Windows	Outlook Plugin JSON

zoommeeting.db databases to store information. Details about these files are discussed on Section 5.2.

Consequently, in the latest Zoom application (5.1.2) and previous ones tested, it was discovered that Zoom created one folder for each account identified by its JID. For example, one folder named “1i-y1fdkqskijzvp3uidhq@xmpp.zoom.us,” which contained databases of interest, the user's profile avatar picture, and other contacts' avatars the user has communicated with directly or indirectly as being part of a Zoom session. Moreover, this directory also contains a folder that stores any media files that are exchanged in Zoom.

On the other hand, Zoom application versions tested prior to the latest used to create another folder “1i-y1fdkqsijzv-p3uidhq@xmpp.zoom.us_sip,” which contained a possible encrypted database file named zoom.sip.enc.db. According to Zoom, any VoIP media is encrypted with AES-128 encryption (Zoom Video Communications, 2020). This file is possibly related to Zoom's H.323 and Session Initiation Protocol (SIP) device support for Zoom Rooms (Zoom, 2020*a*). Note, no SIP devices were tested in this investigation. To speculate this type of file was encrypted, the Shannon Entropy was calculated for each file found across all devices. The average entropy was determined to be 7.9[10] (rounded), which suggests file encryption.

5.2. Major artifacts found in disk

In this subsection, all of our main artifacts are discussed. Note, a place holder such as “USER JID” was used in Tables 1 and 2, and this results section to identify path location names and Jabber ID (JID) values as they are unique to an individual device and Zoom user account.

5.2.1. “USER JID”@xmpp.zoom.us.asyn.db - Zoom Chat feature

This database (Tables 1 and 2, File ID 1) stores numerous tables with information pertaining to devices associated with the Zoom account, as well as chat session information such as messages, files, emojis exchanged through the Zoom Chat feature, devices status, some information about other contacts, in Chat feature calls, and more.

[10] https://github.com/mattnotmax/entropy.

Table 2
Important artifacts extracted across all forensic acquisition Type.

File ID & Name	Important Data Found in Disk, Memory & Network Traffic																		
	JID		Email		Name		Password		Timestamps	Chats		Files Sent/ Received	Buddy JID	Buddy Email	Buddy Name		Avatar URL/Path	File Web URLs	Questions & Answers
	E	U	E	U	E	U	E	U	-	E	U	-	-	-	E	U	-	E	
1. USERJID@xmpp.zoom.us.asyn.db	🖵, 🖵	🖵, 🖵				🖵, 🖵	🖵, 🖵		🖵, 🖵		🖵, 🖵	🖵, 🖵	🖵, 🖵			🖵, 🖵		🖵, 🖵	
2. USERJID@xmpp.zoom.us.db		🖵, 🖵				🖵, 🖵							🖵, 🖵	🖵, 🖵		🖵, 🖵	🖵, 🖵		
3. USERJID@xmpp.zoom.us.idx.db		🖵, 🖵				🖵, 🖵			🖵, 🖵		🖵, 🖵	🖵, 🖵	🖵, 🖵	🖵, 🖵		🖵, 🖵			
4. USERJID@xmpp.zoom.us.sync.db									🖵, 🖵				🖵, 🖵						
5. zoommeeting.db					🖵				🖵	🖵					🖵				
6. zoomus.db									🖵, 🖵								🖵, 🖵		
7. "Temporary Webinar Database"		🖵, 🖵				🖵, 🖵			🖵, 🖵				🖵, 🖵			🖵, 🖵			🖵, 🖵
8. Avatar Cache Index (pList/ini)									🖵, 🖵								🖵, 🖵		
9. us.zoom.videomeetings.plist		🍎/📞		🍎/📞															
10. contacts.db													🍎/📞			🍎/📞			
11. ex2smtp.json				🪟										🪟					
12. userSettings.json		🪟		🪟		🪟													
13. alternateHosts.json														🪟		🪟			
MEM		🪟		🪟		🪟			🪟		🪟		🪟	🪟		🪟	🪟		
NET ***		🖵, 🖵		🖵, 🖵		🖵, 🖵		🖵, 🖵	🖵, 🖵		🖵, 🖵	🖵, 🖵	🖵, 🖵	🖵, 🖵		🖵, 🖵	🖵, 🖵		

Key: E = Encrypted/Encoded, U = Unencrypted or Plain text, 🖵: Desktops (Windows + macOS), 🖵: Mobiles (Android + iOS), 🪟: Windows, 🍎/📞: iOS, ***Note: Fiddler was used to decrypt network traffic

The first data of interest was stored in the "mmkv" table, which contains configuration settings about the Zoom session. For example, this table stores the end-to-end encryption public certificate and a private Privacy Enhanced Mail (PEM) text block which we assume could contain the private key. Additionally, this table stores the PEM password. The text was found to be encoded or encrypted with an algorithm we were unable to decipher. Nevertheless, according to Zoom, they use Advanced Encryption Standard (AES) 256 GCM algorithm at the application layer to encrypt presentation content (Zoom Video Communications, 2020). Similar to the previous table, the "msg_active_devices" also stores an encoded/encrypted certificate, PEM, and password for each active device the account was logged into.

Other tables of interest involve chat messages exchanged through the Chat feature (see Appendix B; Figure B.1). A new table is created in this database named after either a group ID, if more than two people are messaging, or the JID of the user the chats are exchanged with. This also includes messages exchanged when using the Twitter bot feature. The messages are stored in plain text along with timestamps and names of the users in the chat, among other things. More importantly, these messages are stored in all of the devices users utilize in the chat. Thus, providing extra venues of interest when looking for this type of evidence.

Subsequently, details on the different types of files exchanged through the chat, such as images and screenshots, are stored in the "zoom_mm_file" and "zoom_mm_file_download_table" tables (see Appendix B; Figure B.2). It is important to note that these tables also contain the local path names where the files are stored in the device and the "zoom_mm_file", specifically, contains partially Base64 encoded URLs of where those files are stored in a Zoom server. Decoding those revealed data to be partially a combination of the sender's JID, Zoom's web file ID, and something else that it is believed to be Zoom's domain name. Moreover, this database stores emojis and any HTTPS URLs that were exchanged through the Chat feature. Finally, Zoom stores a call history with user names and a Zoom assigned number in this database along with information of any type of searches conducted within the application using the Search feature.

5.2.2. "USER *JID*"@xmpp.zoom.us.db & Contacts.db - contacts

The Zoom application stores a user's contacts in this database (Tables 1 and 2, File ID 2). The main table of interest is the "zoom_mm_buddy" table which contains the names, JIDs, emails, phone numbers, profile picture active URLs and the path where those were stored locally in the device, work departments, job positions, and other private user information in plain text. When analyzing the same database found in the Licensed University account, a large directory of students, staff, and faculty was found. The contact information of these users was found in the database despite having no direct communication with most of them. Moreover, the amount of users found in the database depends on the type of account a user possesses, i.e., student, faculty, etc. This may potentially be a security risk as anyone who gains illegal access to a domain account could acquire information about users without their knowledge. This also includes alumni who have no active access to the University, however, by using their student email address they could still login into Zoom and access this information. During the analysis of the iPhone image, a binary pList file titled "contacts.db" was also found masquerading as a database and contains a simplified list of the names and JIDs of a Zoom user's contacts.

5.2.3. "User *JID*"@xmpp.zoom.us.idx.db - cache

This database (Tables 1 and 2, File ID 3) combines cached data about the two previously mentioned databases above. The two main tables of interest include the "mm_buddy_index_cache_table" which includes a more simplified list of information for a user's contacts such as the contacts JIDs, names, nicknames, and emails. The "mm_msg_cache_table" also contains a recent history of messages that were exchanged on Zoom's Chat feature. Consequently, partial fragments of chat messages, file names of files that were exchanged through the Chat feature, and users' contact information were found in some tables stored in blobs.

5.2.4. "USER *JID*"@xmpp.zoom.us.sync.db - contact requests

In order to make and maintain connections inside and outside of a Zoom's user account domain, a user must make a request to become a contact. Zoom stores these contact requests in this database (Tables 1 and 2, File ID 4). When a user sends or receives a contact request, this database stores information about the JID of the requesting user or target contact, the timestamp associated with the request, and a request message. This message may contain an email address, the display name of the user, and other contact information. Note, this database does not delete the contact request when deleting a contact from the Zoom application. This is a way to verify that at some point a user that was deleted had contact with such user.

5.2.5. zoommeeting.db - zoom video meetings

This database (Tables 1 and 2, File ID 5) stores important encrypted and plain text information about the most recent video meeting conducted through Zoom desktop applications only. The important data was stored in two tables. Table "zoom_conf_cc_gen2" contains information about recorded meetings saved locally on the device and any closed captioned plain text that has been provided during the meeting. Data of interest includes timestamps, plain text

closed captioning, and a 1/0 code to denote when the recording started and ended respectively. If there were any messages exchanged within the video meeting chat while it was recording, the application saved this recording in the local disk along with a plain text transcript of the messages exchanged and the closed captioning text. The "zoom_conf_chat_gen2" table stores the encrypted in-meeting chat messages exchanged from the most recent meeting session. Data of interest includes, encrypted messages, plain text timestamps, encrypted sender and receiver names, and entries denoting whether the meeting started recording and when it ended.

On the other hand, Zoom mobile applications are not capable of saving recorded meetings locally to the device, but only to the cloud with a Licenced account. Therefore, it is believed that this may be one of the reasons why this database was found with no data when testing the mobile devices.

5.2.6. zoomus.db & avatar cache - user, device configurations & more

This database (Tables 1 and 2, File ID 6), stores important data pertaining to user account and Zoom account configurations. For instance, the "z_cert_info" table stores certificate data from certificates that have been trusted by the Zoom application.

The "zoom_conf_avatar_image_cache" table stores cached active profile pictures' URLs, their path location on the device, and timestamps. This table stores this information only when users conduct in video Zoom meetings and the information that is saved belongs to all of the users that have taken part of the meeting. It is essential to note that this information also appears in the "USER JID"@xmpp.zoom.us.db - Contacts database previously mentioned above; however, that information updates every time a user changes their profile picture. Moreover, there is another avatar cache index file that Zoom creates in all devices (Tables 1 and 2, File ID 8). This file updates every time a user changes their profile picture as well, and it only includes the URL and timestamp when the picture was downloaded to the device. Consequently, our results so far indicate these URLs do not expire and one can easily access them on a web browser. However, if a user does change the profile picture, then the link previously stored on the mentioned files would become invalid.

The "zoom_kv" table contains important account configurations such as Zoom application version, the last time the client was connected, IPs, ports, URLs Zoom uses to connect on each session (Zoom Video Communications, 2020), the token refresh URL Zoom uses every time it needs to update the session token, encoded/encrypted Security Assertion Markup Language (SAML) single sign-on (SSO) login with password, meeting ID, and more.

The "zoom_meet_history" table stores information about meeting sessions that were recorded to the device using the desktop application. Important artifacts include the host ID, the path location where the recording was saved, the name of the meeting, the time the recording started, and its duration.

The final table of interest in this database is the "zoom_-user_account_enc" table which stores encrypted user information about the account that is logged-in. This includes username, Zoom refresh token, email, profile picture URL, first name and last name of the account owner, and more.

5.2.7. Zoom webinars - attendees and Q & A

To obtain a better perspective of the features that Zoom offers, attending a webinar as a normal attendee and as a panelist was necessary. However, being able to achieve these tasks was one of the hardest tests to conduct in this research. Webinars are a paid feature of Zoom and most of the time one has to be invited or registered to an event in order to attend one.

Nevertheless, the outcomes from these tests yielded different results depending on the type of user attending the webinar. Moreover, the main database of interest, with a name that is always encoded/encrypted such as "cxPKzMaNQUWBFd9HwEr3Ig = = .db", and that we believe is the meeting ID, is no longer stored permanently in the Zoom data directory. The last Zoom version tested where the database remained in the directory was 4.6.2; however, the latest version we have tested (5.1.2) does not store this database anymore and it actually removes it from the Zoom data directory once the webinar ends. Unfortunately, our attempts to recover this database from a the Windows VM forensic acquired image were not successful, as it was not found using the Autopsy tool. Nevertheless, as an attendee or panelist of a webinar, one has always the chance to acquire the live database while the webinar is taking place.

Consequently, this database contains two main tables of interest, the table "zoom_qa_buddies" which stores a list of all the people who attended the meeting to include panelists and normal attendees. The table stores the name of the user, whether it is the original user name or one the user assigned themselves for the webinar, and a unique JID generated for the webinar such as "wu_92104247635_zo0i6r1uqgqntrp0cyef6g#159228971773 433@xmpp.zoom.us"; . This JID includes three unique strings of interest assigned by Zoom, such as the webinar's meeting ID, the user's webinar's JID, and a timestamp of the time the user joined the webinar. It is important to note that users who join the webinar without providing a name or signing in are still identified, but only by their unique webinar JID.

The fields of interest in this table are viewable depending on the type of attendee and Zoom account (Licensed or Basic). As a normal attendee, users never see any other attendee's names on the interface while conducting a webinar, however, panelists do. Thus, attending the webinar as a panelist, this table stores all of the users' names in the database. As a normal user, the only names stored in the database are the ones from the panelists that the user can see on the application. Nevertheless, the latter does not a apply to Licensed accounts belonging to the same organization. As a normal user, one is able to see all of the names of attendees stored in the database even when they were not viewable in the Zoom webinar interface.

The final table of interest, "zoom_qa_messages", stores a list of all the questions and answers in the webinar, their timestamps, a unique sender JID, and sender name of the person who asked or answered the question. The table also stores flags pertaining to whether the question was answered live, read, dismissed, or deleted, and whether the question was marked by a user to be asked as "anonymous" or sent in private. It is interesting to note that even when users opt to ask a question as "anonymous" in the webinar, the names of the users are still stored in the database providing no anonymity. Subsequently, the database stored in a panelist device would contain all of the questions submitted, while the normal attendee would only contain the questions that had been answered by panelists.

5.2.8. Zoom Outlook plugin - scheduled meetings

The Zoom plugin for Microsoft Outlook was tested on the Windows VM with a Zoom Licensed account as part of this research. This plugin is part of the tools Zoom provides to implement it with the Zoom desktop application. This plugin allows users to schedule meetings through the Outlook application with one click (Zoom, 2020*b*).

Important artifacts discovered through experiments revealed that JSON files are created when meetings are scheduled depending on the settings of the meeting. Three of those files were deemed important and discussed in this section (Tables 1 and 2, File IDs 11–13). For instance, the "ex2smtp.json" stores Outlook Simple Mail Transfer Protocol (SMTP) data and meeting participants' email

addresses. This file updates every time a meeting is scheduled, updated, or canceled. The “alternateHosts.json” file stores the display names and email addresses of users that have been added as co-hosts when scheduling meetings. Lastly, the “userSetting.json” file contains user information, meeting invitation details, and settings pertaining to the user's personal default meeting. Critical artifacts found in this file include, the account owner's name, first and last name, email address, personal meeting ID, personal JID, and the local path of the device where the user's profile avatar picture was stored along with the active avatar URL. Additionally, this file contains more important data already found in plain text, but was also encoded in Base 64. It appears this encoded data contains invitation information that could be sent to users who may not have Zoom installed.

5.3. Major artifacts found in network traffic

According to Zoom, they secure network traffic by using Hypertext Transfer Protocol Secure (HTTPS) and encrypting it with 256-bit Transport Layer Security (TLS) encryption standard (Zoom Video Communications, 2020). Our research proved this to be correct when capturing network packets using the Wireshark tool. Nevertheless, our investigation went a step further to discover the types of encrypted artifacts Zoom transfers over the network. As stated in Section 4.2, the Fiddler tool was used to capture and decrypt this traffic when conducting tests on the desktop applications.

Our results were successful for the most part as the Fiddler tool was able to decrypt most traffic (Table 2, File ID NET). Results include login credentials (username and password) that were transferred in the network when attempting to login to the application using a Basic account and a Licensed account (Appendix B; Figure B.3). The only difference between these accounts is the fact that Zoom uses SAML single-sign-on (SSO) through the browser when logging into a Licensed account. This is a less secure way to sign in compared to using the Zoom application because the password is transferred through HTTPS on the network as well which allowed the Fiddler tool to decrypt it. Moreover, other important artifacts that are fetched by the Zoom application while logging in include account email, JIDs (Jabber IDs), cookies, session access tokens, device ID, MAC address, profile picture, personal meeting room invitation containing the personal meeting ID and meeting password, a list of recorded meetings saved on the cloud, any Outlook plugin data and calendar implementation, any chat history that took place using the Zoom Chat feature, and more.

Consequently, other tests performed during video meetings and in-meeting chats revealed that no messages were found in the network traffic. However, file names of files that were sent through the chat did appear in the network traffic. Moreover, this was also true when testing the Zoom Chat feature only. However, the Zoom Chat feature has more capabilities than the in-video meeting chat, thus, additional artifacts were discovered in the traffic. This includes, any HTTPS links that were sent in the chat and were activated through Zoom's link preview feature, Graphics Interchange Format (GIFs), and any other type of files that were received. Lastly, scheduled meeting information, recorded meeting information, keywords searched through the Search feature in Zoom, and file history could be viewed in the network traffic when using these features in the Zoom application.

5.4. Artifacts found in memory

This section discusses a preliminary memory forensic analysis performed on the Windows Virtual Machine while testing different features of the Zoom application. This analysis was limited as our main intent was to investigate the difference between memory captures taken when the Zoom application was actively open and after the application had completely exited the system. The goal was to search for critical data (i.e., chat messages) that we had already found on disk and network traffic and how much of that data would be removed from memory when exiting the application. Major tools used in this analysis are Volatility and Strings (Appendix A; Table A.4).

Results from this analysis demonstrated that system Random Access Memory (RAM) stores a plethora of information that could be very useful for investigators, especially when conducting investigations in the field based on triage. Important data found in memory before and after the Zoom application was terminated includes user and contacts' information such as plain text and encrypted names, email addresses, and JIDs, profile avatar's URLs, and encrypted and plain text chat messages, webinar information, and more. Moreover, it is believed that since Zoom does fetch account history when first connecting to the application, a lot more information is passed through memory that is already stored in the databases in the disk; this includes end-to-end encryption certificates, PEM key and passwords, chat history and call history, file names that have been exchanged during chat sessions, scheduled meeting information such as meeting ID's and passwords, keywords searched in the Zoom application, and much more. It is important to note that the encrypted messages exchanged in an in-video meeting could also be found in plain text in memory if the meeting is being recorded. This is due to Zoom storing a transcript of the video recording with the messages. Moreover, if Closed Captioning (CC) is enabled, a transcript in plain text is also stored in disk and could be found in memory.

Nevertheless, our results differed based on the type of tool used to analyze the memory. As in the case of Volatility, all major artifacts were found in the memory acquired when the application was opened; this makes sense since the process was active (see Appendix B; Figure B.4). However, when analyzing the memory acquired after the application was terminated, most of the information could not be located using the “yarascan” plugin. Additionally, the Strings tool was run on the memory captures and surprisingly, Strings proved to be a powerful tool as it extracted the artifacts Volatility could not find (see Appendix B; Figure B.5). Thus, it is important to note that there is still a difference in terms of the amount of data that is collected when a process is running as opposed to when is closed. Nevertheless, even when terminating the process a lot of evidence could still be found and help immensely in an investigation.

5.5. Anti-forensic techniques

This section highlights interesting anti-forensic techniques discovered when two people communicate through the Zoom application interface, and one person deletes a contact, causing an effect in both devices. These tests were conducted in all devices using different versions[11] of the Zoom application at the time. Table 3, shows more details of these results. The first four devices in the table belong to the contact that was deleted, while the two at the bottom belong to the user who deleted the contact from the Zoom interface. Results to the right of the table show that in the case of the Android and macOS devices, the chat history and contacts were removed from the Zoom application interface, while in the Windows and iOS devices, only the contact was removed.

[11] Windows, macOS & Android (5.1.2, 5.0.2) & iOS (5.1.1).

Moreover, in the case of the Android, some of the critical data was also removed from important databases and data directory, such as chat messages and exchanged media files. Nevertheless, it was noted that the Android device only experiences this momentarily as the server pulls all of the chat history back to the application's interface when exiting and reopening the Zoom application. All other data remained in the devices as normally expected. It is important to note that in Zoom version 5.0.2, the Windows device had a similar effect as the Android device in removing the information from the interface.

On the other hand, the two devices shown at the bottom of the table belonging to the user who deleted the other contact were affected mostly as expected in the Zoom interface, databases, and media directory. However, there were a lot of traces of data left behind about interactions between both contacts, such as contact information, traces of files and chat messages that were exchanged and more. This could still be useful to identify who the user was communicating with and some of the interactions between them.

As noted, this is an alarming breach of trust as critical information could be removed without the user's permission, even if it is momentarily as in the case of the Android device. No information should ever be deleted from the application and device of the user who was being removed from someone else's contact list.

6. Creation of SQLite database queries

Due to relevant data being stored mostly in SQLite databases, a helpful way to identify this data is through the use of database queries which can be found in Table A.5 of Appendix A.

All of the queries aim to simplify the acquisition of information that can be used during forensic examinations. The following queries will provide examiners a brief overview of the chat interaction between Zoom users. Queries 1, 2, and 3 deal with simplifying the acquisition of the most recent cached message bodies, user information, and timestamps from the chat cache table of the "USER JID@zoom.us.idx.db"; file. However, this table does not list the files or images that may have been sent in chats. Additional queries have been developed, specifically Query 7, which can be modified to search the above mentioned by utilizing the "USER JID@xmpp.zoom.us.asyn.db"; database's tables for the images and files sent within the target chat session identified by its "JID".

Queries 4 and 5 deal with acquiring information about the contacts a user account has and what group chats they may belong to from the "User JID@xmpp.zoom.us.db"; database file. Query 4 selects all of the information pertaining to the entire contact base of the user. Query 5 identifies any group chats the user belongs to or hosts as well as the contact information for the chat owner. Queries 6 through 9 provide investigators a list of the chats within a "Target Chat Session" found in the "Chats" database. Query 6 provides a list of the relevant information for a Chat Session such as the name of the sender, the body of the message and the message timestamp. Query 7 selects the sender information as well as the name of the multimedia files sent and their timestamps. Query 8 selects the messages that were commented on by using emojis while Query 9 selects the messages where files were sent and had been commented on. Query 10 provides investigators with the start date and the "messageID" for the last message sent for each non-meeting chat session a user device may have.

7. Conclusion/discussion

Zooming through the pandemic was something most of us never imagined to happen in our lifetime. Even at this moment, Zoom still is the primary application people use to communicate and conduct businesses through a screen while having to maintain social distance. We believe the COVID-19 pandemic makes our work even more relevant as utilizing this application has become a necessity to society. Therefore, to the best of our knowledge, this is the primary forensic analysis of the Zoom Video Conferencing application. This was accomplished by conducting tests on different devices centered around disk, memory, and network forensic acquisitions. The goal of this research was to measure Zoom application's level of security and privacy granted to protect users' data and whether any findings would be beneficial for forensic investigators and adversaries alike.

Our findings demonstrate that even when the Zoom organization has been continuously patching their application to fix and prevent security risks as presented in their blog (Zoom, 2020*c*), a plethora of user information could still be found in different parts of a system. This includes, plain text user information, such as chat messages, profile pictures, files exchanged, user contact information, and much more. Additionally, some of this data was still found to be stored in the system even when a user had opted to delete a contact from their application. Notwithstanding, Zoom did use secure methods when storing some information in disk and when transferring user account information through the network, such as encrypted passwords and in-video meeting chat messages.

In the case of the network traffic however, it was proven that HTTPS could be decrypted using the tool Fiddler, this could be rare, but in certain cases could still pose a threat to user privacy if access to the device falls in the wrong hands. Furthermore, in terms of the memory analysis, it was concluded that plenty of the evidence already discussed could be found in memory even after the application had completely exited the system. This information could be useful to investigators on the field needing to prioritize collection and analysis of evidence.

Consequently, this research demonstrated some techniques carried out through the Zoom application that could be possibly flagged as anti-forensic. While these techniques were not true for all tested devices, knowing that there are certain Zoom application versions that could possibly cause a user to lose their chat and contact history due to someone deleting them from their contacts list without their permission is problematic. Only the account owner should be able to delete any information in their Zoom application and device.

Finally, our work contributed a series of SQLite Queries aimed at assisting investigators to triage the Zoom databases for all valuable information that may be useful in a case. Moreover, all digital artifacts collected in this investigation can be found in the Artifact Genome Project[12] repository.

8. Future work

Future work should be conducted in this rapidly changing field. As noted research, it was difficult to forensically examine Zoom while trying to keep up with constant software updates. This shows that data changes constantly, and while our results may be valid now, they may become outdated. Furthermore, Zoom is not the only video conferencing application that needs forensic analysis. Future work should explore other applications such as Google Meet, CISCO Webex Meetings, Bluejeans, and Microsoft Teams[13,14].

[12] agp.newhaven.edu.

[13] https://www.howtogeek.com/661906/the-6-best-free-video-conferencing-apps/.

[14] https://www.techradar.com/best/best-video-conferencing-software.

Table 3
Anti-forensic findings.

	Zoom Interface		Chats Db					Contact & Requests Dbs	Cached Db			Files in Directory	
Affected Data - Device Belonging to the Contact that Got Deleted													
	Deleted Contacts	*Deleted Chats*	*Chat Messages*	*Emojis/ GIFs*	*Exchanged Files*	*Last Session*	*Call History*	*Contact History*	*Contact History*	*File History*	*Chat History/ Segments*	*Avatars*	*Exchanged Media*
[b]**Android**	✓	✓	✓	✖/✖	[a]✓/✖	✖	✖	✖	✖	✖	✓/✖	✖	[a]✓
[b]**macOS**	✓	✓	✖	✖/✖	✖	✖	✖	✖	✖	✖	✖/✖	✖	✖
Windows	✓	✖	✖	✖/✖	✖	✖	✖	✖	✖	✖	✖/✖	✖	✖
[b]**iOS**	✓	✖	✖	✖/✖	✖	✖	✖	✖	✖	✖	✖/✖	✖	✖
Affected Data - Device Belonging to the User Deleting the Contact													
Windows	✓	✓	✓	✖	[a]✓/✖	✓	✖	✖	✖	✖	✓/[c]✓	✖	✓
iOS	✓	✓	✓	✖	[a]✓/✖	✓	✖	✖	✖	✖	✓/✖	✖	✓

Key: ✓: Yes Deleted ✖: Not Deleted.
[a] Note: Files were deleted from one table or folder but remain in one or more tables/folder.
[b] Note: The Windows VM was the one used to delete the contacts in these devices while the iOS was used to delete the Windows.
[c] Note: The results in this field apply to Windows when interacting with the Android device.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number 1900210. 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 C. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.fsidi.2021.301107.

Appendix A. Apparatus & SQLite Queries

Table A.4
Apparatus

Hardware/Software	Use	Company	Software Version
Galaxy S6	Zoom Account	Samsung	Nougat 7.0
iPhone 5s	Zoom Account	Apple	iOS 12.4.5, 12.4.6
Windows Virtual Machine	Zoom Account	Windows	Windows 10
MacBook Pro	Zoom Account	Apple	Catalina 10.15.4, 10.5.5
VirtualBox	Hosted Windows Virtual Machine	Oracle VM VirtualBox	6.1.4
Zoom Mobile Application	Android Zoom Account	Zoom Video Communications	4.6.9, 4.6.10, 4.6.11, 5.0.2 (25692.0524), 5.1.2 (28652.0706)
Zoom Mobile Application	iOS Zoom Account	Zoom Video Communications	4.6.9 (19213.0327), 5.0.2 (24042.05.09), 5.1.1 (28562.0629)
Zoom Desktop Application	Windows Zoom Account	Zoom Video Communications	4.6.9 (19213.0327), 4.6.10 (20033.0407), 5.0.2 (24046.0510), 5.1.2 (28642.0705)
Zoom Desktop Application	macOS Zoom Account	Zoom Video Communications	4.6.11(20561.0413), 5.0.2 (24030.0508), 5.1.2 (28648.0705)
Zoom Outlook Plugin	Schedule Meetings in Windows Desktop	Zoom Video Communications	5.1.2 (27830.0612)
Wireshark	Observe Live Network Traffic (all devices)	Wireshark	3.2.3
Magnet Acquire	Full Image Creator (Android & iOS)	Magnet Forensics	2.25.0.20236
DumpIt	Memory Acquisition	Comae	3.0.20200224.1
FTK Imager	Full Image Creator and Memory Capture	AccessData	4.3.0.18
Autopsy	Full Image Viewer	The Sleuth Kit	4.14.0
Android Debug Bridge (ADB)	Android Data Extraction Tool	Android Studio Developers	1.0.41, Version 29.0.6–6198805
DB Browser for SQLite	View Sqlite/DB files	DB	3.11.2
checkra1n	iOS Jailbreak Tool	checkra1n	0.9.7 BETA
SuperSU	Android Jailbreak Tool	Senior Recognized XDA Developer Chainfire	V2.82
Volatility	Desktop Memory Analysis	Volatility Foundation	Volatility 2.6.1 & Volatility 3 1.0.0-beta.1
GNU Strings	String Finder	Free Software Foundation, Inc.	2.33.1
Fiddler 4	Decrypt Network Traffic	Progress Software Corporation	5.0.20202.18177
Base64 Encoder/ Decoder	Decryption Tool	Base64	Online
Entropy	File Entropy Calculator	GitHub user: mattnotmax	N/A
Filza File Manager	File System Manager	TIGI Software	3.7 Build 7

Table A.5
SQLite Queries

Query ID	Query	Database	Result
1	Select senderName, groupID, buddyID, body, strftime('%Y-%m-%d %H:%M:%S',messageTimeStamp/1000, 'unixepoch', 'localtime') as timeStamp from mm_msg_cache_table ORDER by timeStamp asc;	USERJID@xmpp.zoom.us.idx.db	Lists Recent Cached Chat Messages by Timestamp
2	Select senderName, groupID, buddyID, body, strftime('%Y-%m-%d %H:%M:%S',messageTimeStamp/1000, 'unixepoch', 'localtime') as timeStamp from mm_msg_cache_table where sentByMe = 1 ORDER by timeStamp asc;	USERJID@xmpp.zoom.us.idx.db	Lists Recent Cached Messages ONLY sent BY Account User
3	Select senderName, groupID, buddyID, body, strftime('%Y-%m-%d %H:%M:%S',messageTimeStamp/1000, 'unixepoch', 'localtime') as timeStamp from mm_msg_cache_table where sentByMe = 0 ORDER by timeStamp asc;	USERJID@xmpp.zoom.us.idx.db	Lists Recent Cached Messages ONLY sent TO Account User
4	Select jid, firstName, lastName, phoneNo, phoneNumber email, picPath, avatarUrl, meetingNumber from zoom_mm_buddy order by firstName;	USERJID@xmpp.zoom.us.db	Selects ALL Relevant Contact Information from Contacts Table
5	Select groupID, name as participants, firstName as ownerFirstName, lastName as ownerLastName, email as ownerEmail from zoom_mm_buddy, zoom_mm_group where zoom_mm_buddy.jid = zoom_mm_group.ownerID;	USERJID@xmpp.zoom.us.db	Lists Group Chat Participants and Group Chat Owner Contact Information
6	Select messageID, senderName, body, strftime('%Y-%m-%d %H:%M:%S', messageTimeStamp/1000, 'unixepoch', 'localtime') as messageTimestamp from "TARGET CHAT TABLE" order by messageTimestamp;	USERJID@xmpp.zoom.us.asyn.db	Lists All Chat Messages from Target Chat Table
7	Select A.messageID, A.senderName, A.body, strftime('%Y-%m-%d %H:%M:%S', A.messageTimeStamp/1000, 'unixepoch', 'localtime') as messageTimestamp, B.name, B.localpath from "TARGET CHAT TABLE" as A, zoom_mm_file as B WHERE A.messageID = B.messageID order by messageTimestamp;	USERJID@xmpp.zoom.us.asyn.db	Lists All Files Sent and Local Paths for Target Chat Table
8	Select A.messageID, senderName, body, strftime('%Y-%m-%d %H:%M:%S', messageTimeStamp/1000, 'unixepoch', 'localtime') as messageTimestamp, emoji as emojiComment, strftime('%Y-%m-%d %H:%M:%S', first_emoji_t/1000, 'unixepoch', 'localtime') as commentTimestamp from "TARGET CHAT TABLE" as A, emoji_comment_table as B where A.messageID = *B.msg_id*	USERJID@xmpp.zoom.us.asyn.db	Selects ALL Chats from Target Thread with Emoji Comments
9	Select A.messageID, A.senderName, A.body, strftime('%Y-%m-%d %H:%M:%S', A.messageTimeStamp/1000, 'unixepoch', 'localtime') as messageTimestamp, B.emoji, B.contain_mine as emojiSentByMe, C.name as fileName, C.localPath from "TARGET CHAT TABLE" as A, emoji_comment_table as B, zoom_mm_file as C WHERE A.messageID = B.msg_id AND A.messageID = C.messageID order by messageTimestamp;	USERJID@xmpp.zoom.us.asyn.db	Selects Messages with Files AND Emoji Comments
10	Select sessionID, lastMsgID, strftime('%Y-%m-%d %H:%M:%S',readedMsgTime/1000, 'unixepoch', 'localtime') as chatStartDate from zoom_mm_session;	USERJID@xmpp.zoom.us.asyn.db	Selects Chat Session Start Date and Last MessageID

Appendix B. Artifact Figures

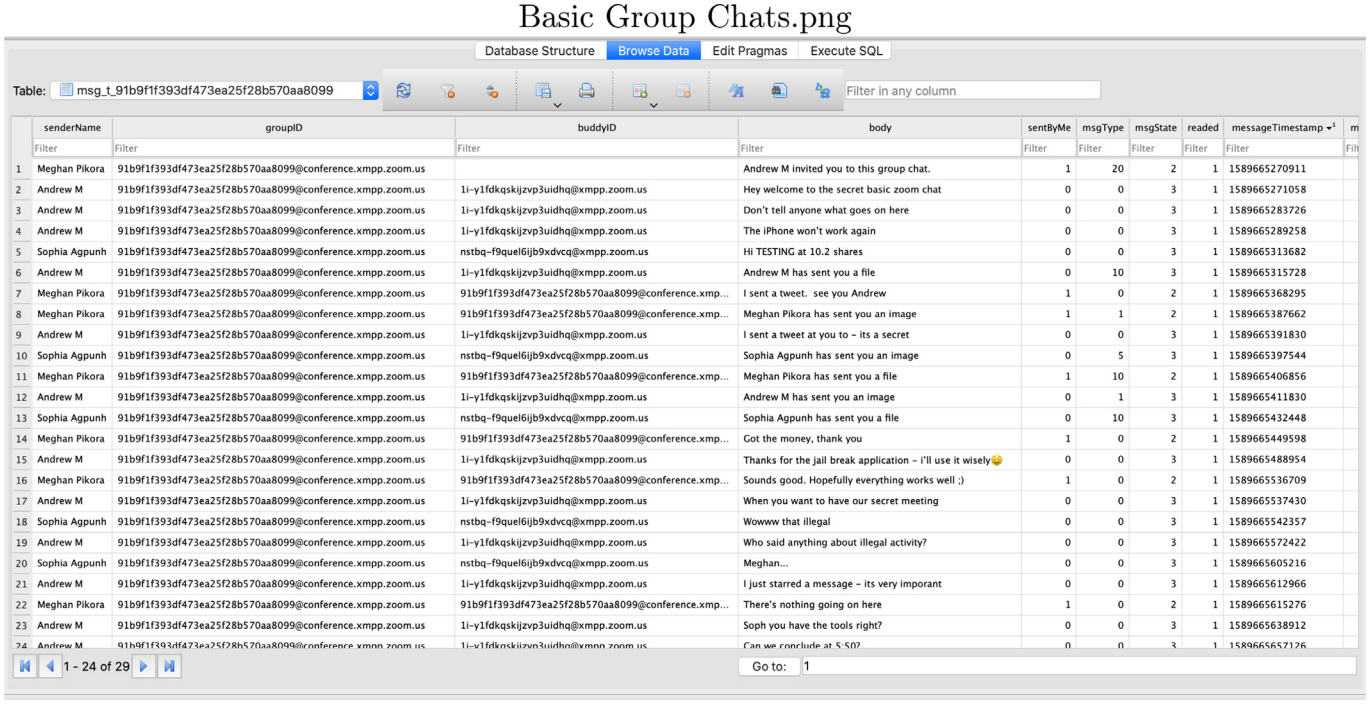

Fig. B.1. Windows VM "USER JID"@zoom.us.asyn.db Database - Displays Zoom Chat Feature Group Messages

VM Files Picture.png

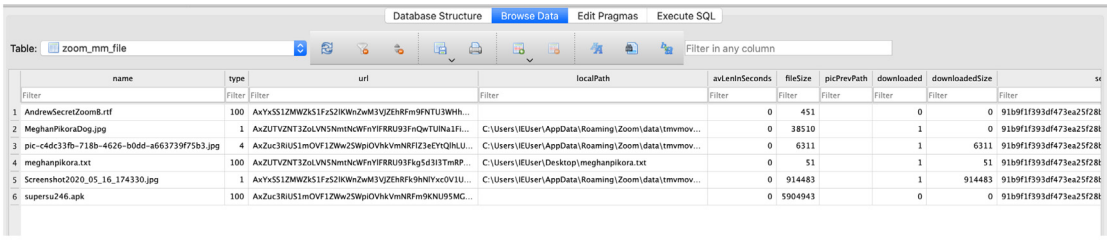

Fig. B.2. Windows VM "USER JID"@zoom.us.asyn.db Database - Displays Files Exchanged Within the Zoom Chat Feature

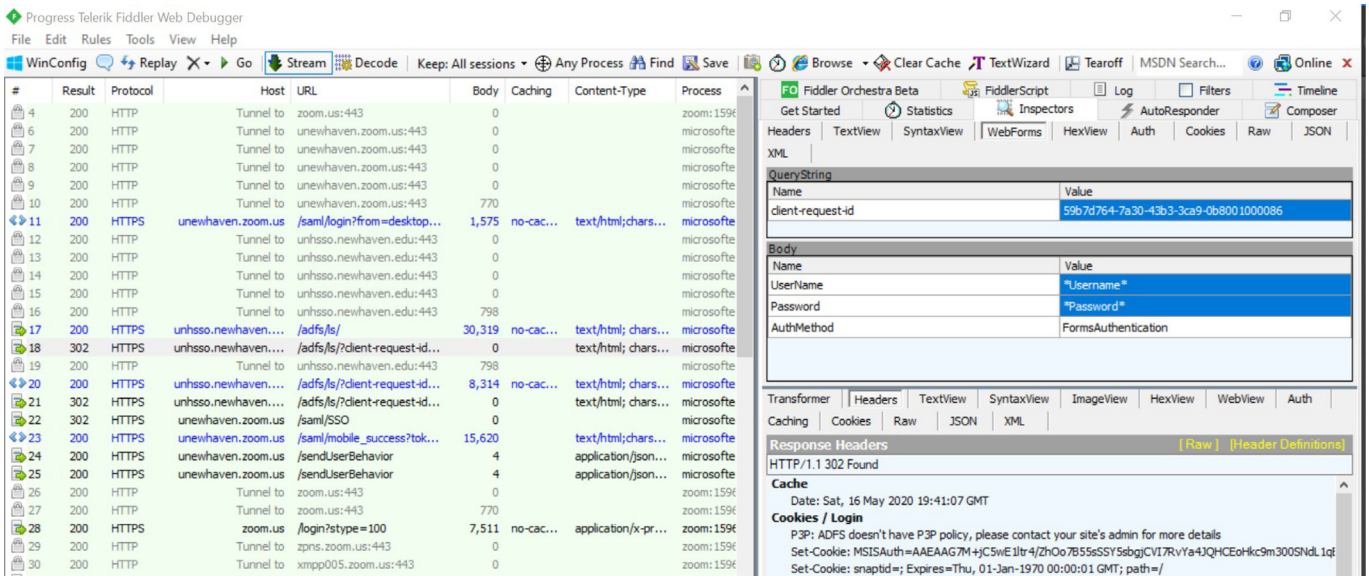

Fig. B.3. Zoom Account User Credentials Found when Decrypting Network Traffic

```
Owner: Process Zoom.exe Pid 2984
0x0e6e3995  54 65 73 74 20 32 20 75 73 69 6e 67 20 63 68 61   Test.2.using.cha
0x0e6e39a5  74 20 66 65 61 74 75 72 65 20 6f 6e 6c 79 20 2d   t.feature.only.-
0x0e6e39b5  20 74 65 78 74 69 6e 67 20 4d 61 72 69 79 20 4c   .texting.Mariy.L
0x0e6e39c5  61 62 62 3c 2f 62 6f 64 79 3e 3c 74 68 72 65 61   abb</body><threa
0x0e6e39d5  64 3e 67 6c 6f 6f 78 7b 38 35 45 32 36 43 42 41   d>gloox{85E26CBA
0x0e6e39e5  2d 46 37 34 36 2d 34 34 34 41 2d 41 36 35 30 2d   -F746-444A-A650-
0x0e6e39f5  30 46 33 39 31 32 35 37 39 44 46 38 7d 3c 2f 74   0F3912579DF8}</t
0x0e6e3a05  68 72 65 61 64 3e 3c 61 63 74 69 76 65 20 78 6d   hread><active.xm
0x0e6e3a15  6c 6e 73 3d 27 68 74 74 70 3a 2f 2f 6a 61 62 62   lns='http://jabb
0x0e6e3a25  65 72 2e 6f 72 67 2f 70 72 6f 74 6f 63 6f 6c 2f   er.org/protocol/
0x0e6e3a35  63 68 61 74 73 74 61 74 65 73 27 2f 3e 3c 7a 6d   chatstates'/><zm
0x0e6e3a45  65 78 74 3e 3c 66 72 6f 6d 20 6e 3d 27 43 69 6e   ext><from.n='Cin
```

Fig. B.4. Chat Message, Sender and Receiver Names Found in Memory through Volatility

```
8004bcaa Mariy Labbpragt0fqr4smrupggve_mq@xmpp.zoom.usYes. Texting back. Is this in memory?
8004bd0f ZoomChat_mobile
8004bd2e A#faee65bc-94ee-4c9e-bebe-01edeb868f4d
8004bd81 {BCE39DBF-1D7A-40D2-AF78-5BDE3C085BC7}_
8004bdab Cin Last Namepragt0fqr4smrupggve_mq@xmpp.zoom.usTest 2 using chat feature only - texting Mariy Labb
8004be32 {BCE39DBF-1D7A-40D2-AF78-5BDE3C085BC7}
```

Fig. B.5. Chat Messages, Sender/Receiver Names and JIDs Found in Memory Acquired After the Zoom Application had Exited the System through Strings Tool

References

Abbott, T., 2020. Best video conferencing apps: the best platforms for video calls. https://www.reviews.org/internet-service/best-video-conferencing-apps/.

Al Barghuthi, N.B., Said, H., 2013. Social networks im forensics: encryption analysis. J. Commun. 8 (11), 708–715.

Al Mutawa, N., Baggili, I., Marrington, A., 2012. Forensic analysis of social networking applications on mobile devices. Digit. Invest. 9, S24–S33.

Al-Saleh, M.I., Forihat, Y.A., 2013. Skype forensics in android devices. Int. J. Comput. Appl. 78 (7).

Anglano, C., Canonico, M., Guazzone, M., 2017. Forensic analysis of telegram messenger on android smartphones. https://www.sciencedirect.com/science/article/abs/pii/S-1742287617301767.

Azab, A., Watters, P., Layton, R., 2012. Characterising network traffic for skype forensics. In: 2012 Third Cybercrime and Trustworthy Computing Workshop, pp. 19–27.

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.

Casey, P., Baggili, I., Yarramreddy, A., 2019a. Immersive virtual reality attacks and the human joystick. IEEE Trans. Dependable Secure Comput. 1–1 https://ieeexplore.ieee.org/document/8675340.

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.

Chang, Y.-T., Chung, M.-J., Lee, C.-F., Huang, C.-T., Wang, S.-J., 2013. Memory forensics for key evidence investigations in case illustrations. In: 2013 Eighth Asia Joint Conference on Information Security. IEEE, pp. 96–101.

Chung, H., Park, J., Lee, S., 2017. Digital forensic approaches for amazon alexa ecosystem. Digit. Invest. 22, S15–S25.

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.

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.

Grajeda, C., Sanchez, L., Baggili, I., Clark, D., Breitinger, F., 2018. Experience constructing the artifact genome project (agp): managing the domain's knowledge one artifact at a time. Digit. Invest. 26, S47–S58.

Hassenfeldt, C., Baig, S., Baggili, I., Zhang, X., 2019. Map my murder. In: Proceedings of the 14th International Conference on Availability, Reliability and Security - ARES '19.

Iqbal, A., Alobaidli, H., Baggili, I., Marrington, A., 2014. Amazon kindle fire hd forensics, 132, 39–50.

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

Lawrence, E., 2019. Faq - certificates in fiddler. https://www.telerik.com/blogs/faq-certificates-in-fiddler.

Levinson, A., Stackpole, B., Johnson, D., 2011. Third party application forensics on apple mobile devices. In: 2011 44th Hawaii International Conference on System Sciences. IEEE, pp. 1–9.

Lorenz, T., Alba, D., 2020. ''zoombombing' becomes a dangerous organized effort'. https://www.nytimes.com/2020/04/03/technology/zoom-harassment-abuse-racism-fbi-warning.html.

Magnet Forensics, 2020. Magnet acquire. https://www.magnetforensics.com/resources/magnet-acquire/.

Majeed, A., 2016. Forensic analysis of social media apps in windows 10. NUST Journal of Engineering Sciences 10.

Marrington, A., Baggili, I., Al Ismail, T., Al Kaf, A., 2012. Portable web browser forensics: a forensic examination of the privacy benefits of portable web browsers. In: Computer Systems and Industrial Informatics (ICCSII), 2012 International Conference on. IEEE, pp. 1–6.

Office, F.N.P., 2020. Fbi warns of child sexual abuse material being displayed during zoom meetings. https://www.fbi.gov/news/pressrel/press-releases/fbi-warns-of-child-sexual-abuse-material-being-displayed-during-zoom-meetings.

O'Flaherty, K., 2020. Beware Zoom Users: Here's How People Can 'zoom-Bomb' Your Chat. https://www.forbes.com/sites/kateoflahertyuk/2020/03/2-7/beware-zoom-users-heres-how-people-can-zoom-bomb-your-chat/#27016316618e.

Saint-Andre, P., 2011. Extensible Messaging and Presence Protocol (Xmpp): Address Format, Technical Report, RFC 6122, March.

Setera, K., 2020. Fbi warns of teleconferencing and online classroom hijacking during covid-19 pandemic. https://www.fbi.gov/contact-us/field-offices/boston/news/press-releases/fbi-warns-of-teleconferencing-and-online-classroom-hijacking-during-covid-19-pandemic.

Simon, M., Slay, J., 2010. Recovery of skype application activity data from physical memory. In: 2010 International Conference on Availability, Reliability and Security. IEEE, pp. 283–288.

Teng, S.-Y., Lin, Y.-L., 2012. Skype chat data forgery detection. In: International Conference on Future Generation Communication and Networking. Springer, pp. 108–114.

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

Whent, R., 2012. A Brief History of Skype. https://www.itbusiness.ca/blog/a-brief-history-of-skype/20750.

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.

Zhang, X., Baggili, I., Breitinger, F., 2017. Breaking into the vault: privacy, security and forensic analysis of android vault applications. https://www.sciencedirect.com/science/article/pii/S01674-04817301529.

Zoom, 2020a. Getting started with h.323/sip room connector. https://support.zoom.us/hc/en-us/articles/201363273-Getting-Started-With-H-323-SIP-Room-Connector.

Zoom, 2020b. Microsoft outlook plugin (desktop). https://support.zoom.us/hc/en-us/articles/200881399-Microsoft-Outlook-plugin-desktop-.

Zoom, 2020c. Security. https://support.zoom.us/hc/en-us/sections/201728933-Security.

Zoom Video Communications, I., 2020. Zoom security guide. https://zoom.us/docs/doc/Zoom-Security-White-Paper.pdf.

ZoomBlog, 2020. 90-day security plan progress report: april 22. https://blog.zoom.us/wordpress/2020/04/22/90-day-security-plan-progress-report-April-22/.