A FORENSIC ANALYSIS OF MICROSOFT TEAMS

by

Herschel Riley Bowling

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the Degree of

Master of Science

Department of Computer and Information Technology

West Lafayette, Indiana

August 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL

STATEMENT OF COMMITTEE APPROVAL

Dr. Kathryn C. Seigfried-Spellar, Chair

Department of Computer and Information Technology

Dr. Umit Karabiyik

Department of Computer and Information Technology

Dr. Marcus K. Rogers

Department of Computer and Information Technology

Approved by:

Dr. John A. Springer

Head of the CIT Graduate Program

Dedicated to my parents David and Debbie,
for inspiring my success, and helping me set my compass

ACKNOWLEDGMENTS

I wish to gratefully acknowledge my thesis committee for their insightful comments and guidance. Without their help and expertise in the field of digital forensics, my studies and acheivments would not be possible.

TABLE OF CONTENTS

LIST OF TABLES 8

LIST OF FIGURES 9

ABSTRACT 10

CHAPTER 1. INTRODUCTION 11

1.1 Background 11

1.2 Scope 12

1.3 Purpose 12

1.4 Research Question 13

1.5 Aims 13

1.6 Hypotheses 13

1.7 Assumptions 14

1.8 Limitations 14

1.9 Summary 15

CHAPTER 2. REVIEW OF RELEVANT LITERATURE 16

2.1 Skype 16

2.2 Chrome 18

2.3 WhatsApp 19

2.4 Slack 20

2.5 Discord 21

2.6 Mobile Forensics 22

CHAPTER 3. METHODOLOGY 25

3.1 Populated Artifacts 25

3.2 Research Environments 27

3.3 Hardware and Software Specifications 27

3.3.1 Windows Admin Machine 27

3.3.2 Windows Host Machine 28

3.3.3 IOS Mobile Phone 28

3.3.4 Android Mobile Phone 28

3.4 Population of Data . . . 29
3.4.1 Device Preparation . . . 29
3.4.1.1 Windows . . . 29
3.4.1.2 iOS . . . 30
3.4.1.3 Android . . . 30
3.4.1.4 Administrative Virtual Machine . . . 31
3.4.2 Organization Creation . . . 31
3.4.3 Microsoft Account Creation . . . 31
3.4.4 Messages . . . 32
3.4.5 Calls . . . 33
3.4.6 Calendar Events . . . 34
3.5 Acquisition . . . 34
3.5.1 Windows . . . 34
3.5.2 iOS and Android . . . 35
3.6 Analysis . . . 36
3.6.1 Forensic Tools . . . 36
3.6.2 Manual . . . 37
CHAPTER 4. RESULTS . . . 39
4.1 General Artifact Locations . . . 39
4.1.1 Windows . . . 39
4.1.2 iOS . . . 40
4.1.3 Android . . . 42
4.2 Account Information . . . 43
4.2.1 Windows . . . 43
4.2.2 iOS . . . 44
4.2.3 Android . . . 46
4.3 Messages . . . 48
4.3.1 Windows . . . 48
4.3.2 iOS . . . 49
4.3.3 Android . . . 50

4.4 Calls . . . 52
4.4.1 Windows . . . 53
4.4.2 iOS . . . 54
4.4.3 Android . . . 55
4.5 Channel File Share / File Messages . . . 57
4.5.1 Windows . . . 58
4.5.2 iOS . . . 59
4.5.3 Android . . . 60
4.6 Calendar . . . 60
4.6.1 Windows . . . 61
4.6.2 iOS . . . 63
4.6.3 Android . . . 64
4.7 Hypotheses . . . 65
CHAPTER 5. DISCUSSION . . . 67
5.1 Artifacts Trends and Observations . . . 67
5.2 Recommendations for Investigators . . . 70
5.3 Impact to Future Research . . . 71
5.4 Limitations . . . 72
5.5 Future Work . . . 72
5.6 Conclusion . . . 73
REFERENCES . . . 75
APPENDIX A. LIBRARY OF CONGRESS FILES . . . 81
APPENDIX B. APPROVAL OF RESEARCH . . . 84

LIST OF TABLES

2.1 Skype artifact paths 17
2.2 WhatsApp artifact paths 20
2.3 Slack artifact paths 21
3.1 Account Details 32
4.1 iOS Teams Directories 42
4.2 Profile Picture and Account Information Recovery 47
4.3 Message Recovery by Forum 52
4.4 Message Recovery by Condition 53
4.5 Call Details Recovery 57
4.6 File Recovery 62
4.7 Calendar Recovery 65
4.8 Total Artifact Recovery 66
A.1 File Locations and Names 81
A.2 File Hashes 82
A.3 File Online Locations 83

LIST OF FIGURES

3.1 Laptop storage drive with USB write blocker 35
4.1 Chromium cache extracted from the Windows machine 40
4.2 The Teams cache explored with ChromeCacheView 40
4.3 The Teams levelDB structure 41
4.4 The iOS SQLite database file 42
4.5 The SkypeTeams.db file 43
4.6 The profile pictures of the iOS, Windows, and Android user respectively 44
4.7 The profile pictures of the Android user stored in the cache 44
4.8 Windy's Profile image carved from the chromium cache by AXIOM 45
4.9 Part of the ZUSER table 46
4.10 Part of the User table 47
4.11 An iOS group message found in the levelDB files 49
4.12 Messages recovered by AXIOM from the Windows machine 49
4.13 Messages deleted on iOS and found in the ZSMESSAGE table of the cache database 51
4.14 Record of edited messages in the Message table of the Android teams cache database 52
4.15 Potential call log fragments found in Window's levelDB storage 54
4.16 Teams calls in iOS Native call log displayed by Cellebrite 55
4.17 iOS native call log database in SQLite viewer 55
4.18 Call beginning and end records in the Android database 56
4.19 Several files named 1.jpg in the chromium cache that correspond to sent file images . 58
4.20 An excerpt from the ZFILELISTING table 59
4.21 An excerpt from the FileInfo table 61
4.22 A deleted event in the levelDB storage 63
4.23 A binary property list being converted to XML using plistutil 63
4.24 An XML dictionary with the details of a scheduled event 64

ABSTRACT

Digital forensic investigators have a duty to understand the relevant components of the cases that they work. However, with the constant evolution of technologies, and the release of new platforms and programs, it is impossible for an investigator to be familiar with every application they encounter. It can also be difficult to know how forensic tools handle certain applications. This is why forensic researchers study and document new and emerging technologies, platforms, and applications, so that investigators have resources to utilize whenever they encounter an unfamiliar element in a case.

In 2017, Microsoft released a new communication platform, Microsoft Teams (Koenigsbauer, 2017). Due to the application's relatively young age, there has not been any significant forensic research relating to Microsoft Teams. This platform as of April 2021 had 145 million daily active users (Wright, 2021), nearly double the number of daily users at the same time in 2020 (Zaveri, 2020). This rapid growth is attributed in part to the need to work from home due to the COVID-19 virus (Zaveri, 2020). Given the size of its user base, it seems likely that forensic investigators will encounter cases where Microsoft Teams is a relevant component but may not have the knowledge required to efficiently investigate the platform.

To help fill this gap, an analysis of data stored at rest by Microsoft Teams was conducted, both on the Windows 10 operating system as well as on mobile operating systems, such as IOS and Android has been conducted. Basic functionality such as messaging, sharing files, participating in video conferences, and other functionalities that Teams provides were performed in an isolated testing environment. These devices were analyzed with both automated forensic tools, and non automated investigation. Specifically, Cellebrite UFED for the mobile devices, and Magnet AXIOM for the Windows device were used. Manual or non-automated investigation recovered, at least partially, the majority of artifacts across all three devices. In this study, the forensic tools used did not recover many of the artifacts that were found with manual investigation. These discovered artifacts, and the results of the tools, are documented in the hopes of aiding future investigations.

CHAPTER 1. INTRODUCTION

This chapter outlines the current study. Topics such as the background of Microsoft Teams, the scope of this study, its purpose, as well as research questions, assumptions, and limitations.

1.1 Background

Microsoft Teams is a relatively young platform, being officially released in March of 2017 (Koenigsbauer, 2017). Teams had millions of users shortly after launch, but Business Insider reported that in March 2020 Teams jumped from 12 million daily active users to 44 million in just a week (Zaveri, 2020). By April 2021, Microsoft announced 145 million users were utilizing the platform daily across the globe (Wright, 2021). This sharp increase can be attributed to the COVID-19 pandemic, as need for services that enable businesses to operate remotely has increased, with other platforms such as Zoom and Slack seeing similar rises in use (Zaveri, 2020). Teams is an application aimed at businesses, as it is specifically a replacement of Skype for Business, with the platform having been retired in July of 2021 (Chin, 2020). For comparison Skype for Business saw 10 million active users as of October 2019, which is 5 years after its release (Kieller, 2019).

Due to its more corporate nature, forensic investigations involving Teams may involve more white collar crimes than other platforms. To date there are no known or publicized incidents of investigation involving Teams, corporate, criminal, or otherwise. This lack of incident gives forensic researchers the opportunity to get ahead and lay the groundwork for response. This platform can be utilized on many different devices, including both desktop and mobile devices. It is important to understand how Teams interacts with those devices and operating systems, particularly those most utilized. Of similar importance is understanding how our existing forensic tools interact with and gather artifacts from Teams on these devices. These two ideas are the main focus of this study.

1.2 Scope

This study includes artifact analysis of The Microsoft Teams Windows 10 desktop client, the IOS mobile operating system, and Android operating system. Teams can run on other operating systems, but these were chosen due to their popularity. As of September 2020 (StatCounter, 2020b) IOS and Android account for a shared 99.42% of global mobile OS market share. Windows was chosen as the desktop OS for two reasons. First, similar to the mobile platforms Windows accounts for a majority 77.12% of global desktop OS market share as of September 2020 (StatCounter, 2020a). Second is that both the Windows operating system and Teams were developed by Microsoft, making it reasonable to believe Teams artifacts can be recoverable from a Windows system. Windows mobile OS was considered for this study, but with a global mobile OS market share of 0.03% (StatCounter, 2020b) efforts were focused on more widely utilized systems.

Limiting the scope of this study to the most utilized devices is a practical necessity, and similarly the tools used to analyze artifacts have been limited to those most used by the forensic community. For desktop Magnet AXIOM was chosen. AXIOM is utilized by law enforcement and endorsed by Police1, an online resource for police officers (Police1, 2018). Magnet Axiom was the receiver of the 2020 4:cast Digital Forensics Commercial Tool of the Year award, continuing a 7-year streak of Magnet winning 4:cast awards (4:cast, 2020; Magnet, 2020). For the mobile devices, Cellebrite UFED was chosen. Cellebrite specializes in mobile devices, has deployed more than 60,000 UFED licenses globally in over 150 different countries, (Cellebrite, 2020) and is listed as part of the Infosec Institute's list of top forensics tools (Shankdhar, 2019). These tools are in active use by law enforcement making them good candidates for testing in this study. There are other tools that could be tested with similar procedures in order to expand this work.

1.3 Purpose

The intent of this study is to aid in future criminal investigations that involve Microsoft Teams. At the time of writing, there is no known source of significant forensics research on the

platform, so it would be difficult for an investigator to isolate even basic artifacts that would be helpful in an investigation. There are also no known criminal cases involving Microsoft Teams at the time of writing, but one possible explanation for this, other than the platform's youth, would be the inability of investigators to efficiently and in a forensically sound manner, analyze the evidence found on Teams. The purpose of this study is to eliminate that gap and provide a framework for investigators to use for cases involving the Teams platform.

1.4 Research Question

The core aim of this study is to document all findings for future investigation, by answering the following research question:

- What artifacts of investigative significance can be recovered using forensic techniques from the Microsoft Teams Windows 10, iOS, and Android clients?

1.5 Aims

Specifically, the research question is answered with the following aims:

- Determine if the type of operating system (i.e., Windows 10, IOS, and Android) has an impact on what can be recovered for the Microsoft Teams desktop client / mobile apps.
- Determine the capability of AXIOM to recover artifacts from the Microsoft Teams desktop client.
- Determine the capability of UFED to recover artifacts from Microsoft Teams mobile apps.

1.6 Hypotheses

There are two Hypotheses for this study:

- H_1 More than 50% of populated artifacts will be fully or partially discovered through manual investigation.

- H_2 More than 50% of populated artifacts will be fully or partially discovered by the forensic tools.

1.7 Assumptions

The assumptions for this study are:

- Each of the three devices have one unique user
- Each user only uses one device for all communications
- No encryption was used on the devices, other than encryption provided by operating system default behavior
- The applications were not deleted or removed from the devices prior to extraction
- The users were logged in to their Teams account at the time of extraction
- All desired artifacts were viewed by the user prior to extraction (e.g., read every message, looked at the calendar, etc.) except those intended to be left unseen (unread messages)

1.8 Limitations

- The current study only considered Windows 10 desktop operating system. Other types of desktop operating systems and versions of Windows were not considered.
- The current study only considered IOS and Android mobile operating systems. Other types of mobile operating systems were not considered.
- The current study did not consider the Microsoft Teams web client, or any resulting browser related artifacts.
- All operating system versions were kept consistent throughout the study, and not allowed to change / upgrade.

- All application versions were kept consistent throughout the study, and not allowed to change / upgrade.
- Live memory forensics was not considered for the scope of this study.

1.9 Summary

This chapter provides background on the Microsoft Teams platform, establishing that it is young but widely used. The lack of forensic research on this platform makes it an excellent subject for this study. Windows 10, IOS, and Android implementations of Teams were studied using UFED, and AXIOM. While other operating systems and tools would be appropriate to research, these OSs were chosen due to their relative high percentage of market share, and the tools due to their acceptability and use in the forensics and law enforcement communities. This study has identified which artifacts can be discovered, and which ones are currently capable of being discovered by the selected tools.

CHAPTER 2. REVIEW OF RELEVANT LITERATURE

As of July 2021, there appears to be no significant forensic research relating to Microsoft Teams. The only source that could be found was a poster from SANS published in March of 2021, that acts as a quick reference for iOS applications (Epifani). Teams was one of 75 applications included on this poster and included file locations potentially useful for iOS analysis. This lack of published research is likely because Teams is a relatively young platform, having been released in 2017 (Koenigsbauer, 2017). For this reason, it is not possible to review past works on this specific problem. Instead recent forensics works relating to Microsoft's Skype, the Google Chrome web browser, WhatsApp, Slack and Discord are being considered. It is beneficial to look at known forensics of Skype as it is a Microsoft product that does many of the same things Teams does, including voice and text chat. It is reasonable to believe that some of the same artifacts found by Skype may be discoverable for Teams as well. The Chrome web browser is not a communications platform, but preliminary analysis of the Team's caching structure on a Windows 10 computer suggests that it is very similar to the known caching structure of Chrome on PC (Suma, Dija, & Pillai, 2017). WhatsApp, Slack, and Discord forensics research is also of interest as they are previously studied communications platforms available on both mobile and desktop.

2.1 Skype

Past research has shown that significant artifacts can be recovered from Skype on IOS (Sgaras, Kechadi, Le-Khac, et al., 2016) Android (Al-Saleh & Forihat, 2013; Sgaras et al., 2016) and Windows desktop (Yang, Dehghantanha, Choo, & Muda, 2016) as well as Skype for Business on Windows desktop (Nicoletti & Bernaschi, 2019) Microsoft's enterprise version of Skype. While Skype, Skype for Business, and Microsoft Teams are independent programs, seeing where and how artifacts were stored in these past communication platforms may grant insight on research into Teams.

Research was conducted on Skype running on IOS 6.1.3 non jail-broken (Sgaras et al., 2016), Android 2.3.5 non rooted (Sgaras et al., 2016), and Android 4.0.3 rooted (Al-Saleh & Forihat, 2013). The devices that were neither jailbroken nor rooted (Sgaras et al., 2016)

Table 2.1. *Skype artifact paths*

Item	Path
1	%AppData%\Local\Packages\Microsoft.SkypeApp_kzf8qxf38zg5c\LocalState\
2	%LOCALAPPDATA%\Microsoft\Office\16.0\Lync\
3	HKEY_USERS\<SID>\Software\Classes\LocalSettings\Software \Microsoft\Windows\CurrentVersion\AppModel\Repository \Families\Microsoft.SkypeApp_kzf8qxf38zg5c
4	HKEY_CURRENT_USER\Software\Microsoft\Office\16.0\Lync

potentially recovered fewer artifacts than theoretically possible as these processes yield at least the same amount of information, if not more (Sgaras et al., 2016). Sgaras et al. (2016) used devices that had no passcode locks to enable focused investigation, but Al-Saleh and Forihat (2013) did not indicate the lock state of the devices used. Several target artifacts were used by Sgaras et al. (2016) including installation data, content data, user profile data, contact database, and attached/exchanged files among others. However Sgaras et al. (2016) did not provide details on how the devices were populated with data, where Al-Saleh and Forihat (2013) provided explicit detail about the data that was used. This included 6 calls of varying lengths, and 6 chat messages (3 sent and 3 received). Al-Saleh and Forihat (2013) used 3 separate scenarios that varied whether the user signed out or not, and whether they deleted the call and chat history. Artifacts were analyzed and compared following each scenario (Al-Saleh & Forihat, 2013).

For Skype on Windows desktop Yang et al. (2016) found several sources of artifacts left behind by the program. A key recurring file path discovered by Yang et al. (2016) can be seen in Table 2.1 as item 1. In this path %AppData% is a Windows shorthand that point to the current user's roaming application data (Pointlogic, 2018). Similarly item 2, a noteworthy path found for Skype for Business (Nicoletti & Bernaschi, 2019) utilizes %LOCALAPPDATA% which is a Windows shorthand for where a user's local application data is stored (Pointlogic, 2018). Items in the Windows registry can be just as useful for forensic investigation as file paths. Artifacts related to Skype can be found at item 3 in Table 2.1 (Yang et al., 2016) and for Skype for Business at Item 4 (Nicoletti & Bernaschi, 2019). These artifact locations of previously designed Microsoft communication platforms could prove informative in discovering artifact locations in Microsoft Teams.

2.2 Chrome

Microsoft has confirmed that Teams is built on Electron, which uses chromium as an engine, giving Teams the chromium cache structure (Maguire, Martinez, Payne, & Borys, 2020). As the Google Chrome web browser is also built on chromium, it uses the same cache structure (Suma et al., 2017), (Shafqat, 2016). According to Suma et al. (2017) the index file contains references to data stored in the data files or f files. The size of the data determines which data file it will be stored in, or if the data exceeds 16 kilobytes it is stored in its own f file (Suma et al., 2017). If Microsoft Teams does indeed adhere to this format, tools that parse this format for Chrome, such as ChromeCacheView by NirSoft (2021) can be safely used on Teams cache data. Enabling investigators to reliably use these tools for cache analysis is important, as "the most relevant file in [web browser] forensic investigation is the cache file" (Suma et al., 2017, p. 1) and this may be true for Teams as well.

Chrome uses this cache format to store data useful for a web browser, such as the HTML of visited pages, loaded images, JavaScript files, visit time, search terms, and downloads (Suma et al., 2017), (Shafqat, 2016) but this structure could theoretically be used to store diverse types of content, which seems likely in the case of Teams, as it presumably has less need to store things like HTML and JavaScript files. Google Chrome also makes extensive use of SQLite databases to store information (Rathod, 2017). Data stored in these databases includes history, cookies, login data, search terms, download information (Rathod, 2017).

Like many modern web browsers, Chrome has a private browsing mode which can have significant impacts on what artifacts can be found (Shafqat, 2016). As well as a private mode, chrome also has a portable version which can be placed on a USB drive or other portable medium, which also changes what a forensic investigator can find (Shafqat, 2016). In private mode browsed websites can be seen in RAM but cannot be found stored in their normal location (Shafqat, 2016). Items such as cookies, top sites, and bookmarks are also not able to be found following a private browsing session (Nelson, Shukla, & Smith, 2020). For a portable installation, all the normal artifacts can be found on the portable drive where Chrome portable is installed, but interestingly there are several stored on the host computer as well (Shafqat, 2016). It was found that "browsing history, cookies, cached websites, saved passwords etc.", also got stored in the

../LocalSettings/Temp/GoogleChromePortable folder in the C drive, and remained there even after the USB, containing Portable Chrome application, was detached (Shafqat, 2016, p. 129). This means data can be found if Chrome was executed on a machine, even if it was not installed on that machine. Even using private mode on Chrome portable leaves traces in the Windows page file (Shafqat, 2016) and extensions like the History Clear add on will leave some trace of its presence (Morris & Moses, 2018).

2.3 WhatsApp

WhatsApp is not a Microsoft product, nor are there any known structural similarities to Microsoft Teams, but it is a popular messaging platform (Onovakpuri, 2018) with many of the same functionalities as Teams. Both Teams and WhatsApp allow users to exchange text and multimedia messages, make Voice over IP (VoIP) calls, and delete messages after they have been sent (Vukadinovic, 2019), (Yadav, Prakash, Dayal, & Singh, 2020). This makes the forensics of the WhatsApp platform potentially relevant for guiding forensic discovery in Microsoft Teams.

Vukadinovic (2019) conducted a systematic analysis of the WhatsApp desktop application, as well as an analysis of browser artifacts in multiple web browsers after using the WhatsApp web client, in both cases testing on Windows and Mac operating systems. In addition to desktop forensics the WhatsApp Android application was analyzed by Vukadinovic (2019) on a Google Pixel XL. Nearly all desired artifacts were discovered in the Chrome client on both Windows and Mac (Vukadinovic, 2019) making it the most reliable and fruitful platform for WhatsApp artifact recovery. This included artifacts such as text messages, deleted messages, files sent, and others (Vukadinovic, 2019). It was also found that the forensic tools FTK, AXIOM, and Autopsy were all able to recover the same number of artifacts.

Among other applications Onovakpuri (2018) analyzed WhatsApp on an android device. Several artifacts were discovered in the key locations listed in Table 2.2. Similar to Vukadinovic (2019) various artifacts could be found such as text messages, contact information, contact photo, sent and received video, and others. Each of these artifacts also had associated timestamps (Onovakpuri, 2018) which is very beneficial to investigators.

Table 2.2. *WhatsApp artifact paths*

Item	Path
1	`/data/com.whatsapp/files/Avatars/60xxxx@s.whatsapp.net`
2	`/data/com.whatsapp/databases/wa.db`
3	`/data/com.whatsapp/databases/msgstore.db`

WhatsApp allows users to delete messages within one hour of the message being sent (Yadav et al., 2020) which poses clear challenges for forensic investigators. It is important to answer questions like, can the deleted message be recovered, or if not, can it be shown that a message was sent? It has been found that with some manual effort by an investigator a deleted message can be recovered from the sender's device (Yadav et al., 2020). This is suggested to be a time consuming process involving extraction and use of cryptographic keys (Yadav et al., 2020) but the fact that it is possible is a great boon to investigators that need to know the contents of deleted messages in order to make a case.

2.4 Slack

Like Teams, Slack does much of its business with corporations. The platform has been adopted by over 750,000 companies and 12 million daily active users as of 2019 (Curry, 2020). Its features include messaging in channels both publicly available to all members of an organization, and private channels available to specific users. Direct and group messaging is also supported, as are direct calls. Features like screen sharing and group calls, and video calls are available but not on all plans and platforms (Slack, 2020c). Slack has many of the same offerings as Teams, notable utilization, and a similar audience, which make it a natural inclusion in this literature review.

Applications that are hosted by remote servers, like Slack and Teams, can often have artifacts extracted directly from the service as well as from the end user's device. Slack offers three ways of retrieving information directly from the service (Pochron, 2018). Slack allows administrative users to export all public messages to JSON files with its standard export feature (Slack, 2020a). Standard exports do not include direct, group, or private channel messages (Pochron, 2018). Slack Plus customers can apply for the ability to perform corporate exports, enabling the export of direct, group, and private channel messages (Pochron, 2019). For either of

Table 2.3. *Slack artifact paths*

System	Path
Mac	`\Users\<username>\Library\ApplicationSupport\Slack`
Windows	`\Users\<username>\AppData\Roaming\Slack`

these options, attachments are not included but rather links to attachments, which can add extra steps to the legal admissibility of that evidence (Pochron, 2018). The third export option supported by Slack is the Discovery API to export content to a third party eDiscovery or data loss prevention service (Slack, 2020b). Data collected by eDiscovery services like Onna or Druva can be used in a court of law, or for auditing purposes.

If available, retrieving data directly from the service can be a very efficient way for an investigator to get what they need. Unfortunately, it is often necessary to retrieve the data directly from the end user's mobile or desktop device. The Leahy Center for Digital investigations (LCDI) discovered several different artifacts of interest for the Slack desktop client on both Windows and Mac systems (LCDI, 2017). Theses artifacts can be found in a folder whose location varies slightly between the operating systems which can be seen in Table 2.3. Within this folder, other folders exist such as `cache` that contains "gifs that were used as well as images" (LCDI, 2017, p. 6), the `GPUCache` folder which contains profile pictures, and inside of data files were users that were previously members of the slack group. These data files were named `Data_0`, `Data_1`, `Data_2`, and `Data_3` (LCDI, 2017). This appears to be the same as the Google Chrome caching format that Teams appears to use. Similar results were found in a 2016 taxonomy of 30 different productivity apps (Azfar, Choo, & Liu, 2017). This study showed that for Slack on Android artifacts such as username, profile image, and sent/received messages/images (Azfar et al., 2017).

2.5 Discord

Discord is a popular communication platform, utilized heavily by people playing online video games. As a result of the Discord Windows client being built on Electron (Electron, 2016), which uses the chromium engine, discord also uses the chromium cache structure. This makes

Discord a valuable application for analysis, as it serves a similar role to Teams, and is built on some of the same technologies.

Discord is also a relatively young program, though older than Team, so there has not been much research on the platform until recently. Recently Shin, Park, Kim, and Kim (2020) studied Discord on an Android phone and were able to discover artifact locations for "received/sent messages, shared files, chat rooms and user account information" (Shin et al., 2020, p. 799). Many artifacts were located in the discord package directory located at `\data\data\com.discord\` and in various sub directories. In addition to the Android application, the Windows client was analyzed. It was confirmed that Discord does indeed use the chromium cache format, and useful artifacts were found there (Shin et al., 2020).

The process of extracting Discord artifacts has been automated by Motyliński, MacDermott, Iqbal, Hussain, and Aleem (2020) with the development of the DiscFor tool. This tool is capable of extracting artifacts from the Windows and MacOS Discord clients, which use the Chromium caching format, as well as the Linux Discord app which uses Simple Cache (Motyliński et al., 2020). This is possible as even though different storage formats are used, the files that are being stored are similar across these platforms as the same developers are producing the different clients, and the same Discord API is being used.

Recent research has highlighted not only the forensics involved in analyzing Discord, but the security and privacy implications. Discord not only caches messages and media, but also logs information users might have assumed Discord forgot such as drafts of typed messages, or login tokens that could be used to hijack their account (Moffitt, Karabiyik, Hutchinson, & Yoon, 2021). This arguably overeager logging could be a serious privacy concern for some users, and behavior like this should be noted. Due to the lack of research on Teams, it is not known if the platform is over-logging in the same way. While not the main purpose of this study, noting such behavior could make a difference to privacy conscious users, and future privacy research.

2.6 Mobile Forensics

Understanding the process of digital forensics is important for this study as two of the three devices under study are mobile devices. There is much variety in the devices mobile

forensic analysts encounter. With "Different models, hardware, memory structure, operating systems... available" (Dogan & Akbal, 2017, p. 1242) it can be difficult for investigators to know exactly what to do with a given device. While every device may be different, there is a general process when investigating any mobile device. Jones and Winster (2017) identify four steps in the basic framework of digital forensics as seizure, acquisition, examination / analysis, and report generating. While Sathe and Dongre (2018) identify six steps to mobile forensics, Goel and Kumar (2019) propose a seven layer framework, and Dogan and Akbal (2017) lists nine steps, there is significant overlap in these systems, and they largely follow the same path. A step at the end of the investigative process not covered by Jones and Winster (2017) but detailed by the other included frameworks is the presentation of evidence (Dogan & Akbal, 2017; Goel & Kumar, 2019; Sathe & Dongre, 2018).

When a crime occurs, special care must be taken to preserve the digital evidence during seizure (Jones & Winster, 2017). The main way this is done for mobile devices is isolating them "from the outside world in order to avoid contamination" (Sathe & Dongre, 2018, p. 280). One way to achieve this is to put the device into airplane mode, isolating it from outside networks (Goel & Kumar, 2019). A similar effect can be achieved by putting the device in a Faraday bag, which will block any incoming or outgoing signals such as cellular connections (Goel & Kumar, 2019).

Once the device is seized acquisition takes place. Acquisition is the process of obtaining a mirror image of the data in a device (Goel & Kumar, 2019; Sathe & Dongre, 2018). Unlike computer forensics where it is often possible to copy stored data while the device is powered down, "most mobile acquisition is performed live because it is not possible to acquire the data when the power is down" (Jones & Winster, 2017, p. 1862). By acquiring live the state of the device can be changed, which is why it is important to use reliable tools that minimize the risk of altering artifacts like MOBILedit, Bulk extractor, and Cellebrite UFED (Goel & Kumar, 2019).

Following the forensic copy of a device, it can be analyzed to gain meaningful forensic insights (Sathe & Dongre, 2018). This is often done with the aid of forensic tools, and ideally multiple are used to ensure the most possible information is extracted (Jones & Winster, 2017). Goel and Kumar (2019) consider examination and analysis the same phase of the framework but distinguish between the two terms. Examination is the more technical process of identifying and

retrieving the evidence from the acquisition, while analysis takes the pieces of evidence and uses them to make determinations about the larger case (Goel & Kumar, 2019).

At the conclusion of an investigation, a report is made so that others can see and understand the artifacts and investigative conclusions that were uncovered (Sathe & Dongre, 2018). Reports are often non-technical (Jones & Winster, 2017) as they need to be understood by others involved in the case who are not trained in digital forensics. These reports will also likely be used by used to "facilitate further legal proceedings" (Goel & Kumar, 2019, p. 560) and so the findings should be clear to lawyers, judges, jury members, and possibly others.

This is because at the end of the investigative process is the presentation of findings, often in a court proceeding (Sathe & Dongre, 2018). Not all cases an investigator works on will go to court, but this is always a possibility (Goel & Kumar, 2019), and so investigators must be prepared to present their findings in a clear, understandable, and professional manor. Dogan and Akbal (2017) also include a final step following presentation, which is archiving. This is necessary, so that if the evidence in a case is required in the future, it is accessible.

It is important that investigators understand the entire mobile forensics lifecycle, from the crime scene to the court room. A recent report found that in mobile forensic trainings, acquisition and analysis are covered heavily, while the other stages are not covered in sufficient depth (Humphries, Nordvik, Manifavas, Cobley, & Sorell, 2021). Other key kill shortages of mobile forensic investigators included "the lack of basic knowledge, generic skills in forensics and investigation, lack of skilled practitioners, and necessary mindsets to critically think, investigate and avoid dependency on Digital Forensic software" (Humphries et al., 2021, p. 1). Of particular interest to this study is the dependence on forensic software. The authors argue that the field is becoming more automated, making it more necessary to validate the tools we use by "setting up experiments to define and test hypotheses" (Humphries et al., 2021, p. 10). The current study aims to be such an experiment and may assist the understanding of future investigators.

CHAPTER 3. METHODOLOGY

This chapter describes techniques that were used in this study. Microsoft Teams is a young platform which operates on many different technologies. To date there is no significant forensic research on the platform; thus, it is important to start mapping how Teams interacts with these technologies. This chapter outlines how these technologies were populated, and how the artifacts were assessed.

3.1 Populated Artifacts

For this study an artifact is any piece of information discovered on a system during forensic analysis that corresponds to a user action or provides an investigative insight. It is important to populate many different types of artifacts, as it is not possible to know which ones will be important for future investigations, as it can vary widely from case to case. For this research, there are 20 types of artifacts, broken into six categories, that are thought to be potentially recoverable. These artifacts and categories are:

Account Information per device

- 1 sets of local account details
- 1 local account pictures
- 2 sets of non-local account details: each device should be able to see the account information of the other two devices
- 2 non-local account pictures: each device should be able to see the account picture of the other two devices

60 Messages per forum (direct, group, and channel). 180 Messages in total

- 5 unaltered messages per device per forum
- 5 deleted messages per device per forum

- 5 edited messages per device per forum
- 5 unread messages per device per forum

9 files per forum (direct, group, and channel). 27 Files in total

- 1 video per device per forum
- 1 image per device per forum
- 1 PDFs per device per forum

Calls per forum. 30 calls, 300 minutes in total

- Channel (No distinction between audio and video calls)

    2 calls per device (5 and 15 minutes)

- Direct

    2 Audio calls per device (5 and 15 minutes)

    2 Video calls per device (5 and 15 minutes)

- Group

    2 Audio calls per device (5 and 15 minutes)

    2 Video calls per device (5 and 15 minutes)

Channel File Share. 9 Files in total across all 3 devices

- 3 PDFs: one sent by each device
- 3 Microsoft Word Documents: one created by each device
- 3 Microsoft Excel Documents: one created by each device

Calendar. 15 items per device, 45 items in total across all 3 devices

- 5 unmodified calendar items per calendar
- 5 deleted calendar items per calendar

- 5 edited calendar items per calendar

3.2 Research Environments

Some applications like Skype or WhatsApp, allow any user to potentially contact any other user. Microsoft Teams is more like Skype for Business where users primarily contact others within their organization. Teams allows anyone to set up their own "organization" using only an email address, as well as add other users to this organization. In order to create a controlled environment a new organization was set up for this study, using an email address created for this study. To further control the environment, the organization was created and administrated from a machine who's only purpose is to control the organization. This machine is referred to as the admin machine, and was not forensically analyzed, or involved in any messaging, to better isolate the artifacts of interest.

A total of four machines were used. The admin machine was a virtual machine as it was not of forensic interest. This machine was hosted in a VMware vSphere version 7 environment. The Windows 10 device was a laptop. The other two machines were physical mobile devices, one running IOS and one running Android. All devices were wiped and either jail broken or rooted to allow the best possible chance of obtaining reliable results.

3.3 Hardware and Software Specifications

3.3.1 Windows Admin Machine

- CPU: 1 virtualized dual core CPU
- RAM: 8 GB virtualized
- Hard drive: 64 GB, not pre-allocated (thin provisioned)
- OS: Windows 10 Enterprise, build 19042

3.3.2 Windows Host Machine

- Make: Lenovo
- Model: IdeaPad Yoga 13 (20175)
- Capacity: 256GB
- OS: Windows 10 Pro Education, 19041
- Teams Client Version: 1.4.00.4167

3.3.3 IOS Mobile Phone

- Make: Apple
- Model: iPhone X
- Capacity: 64 GB
- OS: IOS 13.7
- Teams Client Version: 2.5.0

3.3.4 Android Mobile Phone

- Make: Motorola
- Model: moto g^7 plus
- Capacity: 64 GB
- OS: Android 10

- Teams Client Version: 1416

3.4 Population of Data

Care was taken with population, so that the desired artifacts could be acquired. The National Institute (NIST) sets a standard for populating data onto mobile phones. Of particular interest to this study are sections 5, 6, and 7 which pertain to text messages, multimedia messages, and stand-alone files, respectively (NIST, 2016).

3.4.1 Device Preparation

3.4.1.1 Windows

The Lenovo laptop and its storage drive had been used for previous forensic research. In order to ensure all artifacts found were related to this study, Windows was reinstalled, and the storage drive was wiped, having all of the data sectors replaced with all 0s. This was achieved by booting to a Windows 10 installation USB, and following the Windows 10 installation wizard, with one key exception.

On the first screen of the wizard, a command prompt was accessed by pressing shift+F10. In the command prompt the command "diskpart" was used to start the Windows disk partitioning utility (Gerend et al., 2020). Within this utility the "list disk" command was used to identify the 256GB storage drive, and then that drive was selected with the command "select disk #" where # corresponded to the appropriate device number given in the list. The command "clean all" was used to not only deallocate the drive, but replace the contents with 0s (Gerend et al., 2017). This is different than the "clean" command which only deallocates and does not override the data (Gerend et al., 2017). This process took most of an hour to complete, at which point the command prompt was closed, and the wizard was followed.

After installation, during initial setup the device was connected to the internet via WiFi, and the account of the Windows user created for this study was used to sign into the device. The creation of this account is covered in section 3.4.3. On the privacy screen of the setup, all options

were set to "no" to avoid additional tasks unrelated to Teams taking place on the device. After setup the Teams Windows client was downloaded from microsoft.com.

3.4.1.2 iOS

The iPhone X used had previously been used as a personal device. In order to ensure artifacts found were related to this study, the device was reset using the native reset procedure. Specifically the feature accessed was at Settings > General > Reset > Erase All Content and Settings. The phone was setup as a new device and was connected to the internet via WiFi. Location services were enabled for the device, and a passcode was set. Because an account is necessary to download the Teams app from the App Store, the outlook email made for the iOS device in section 3.4.3 was used to create an Apple ID, which was signed into on the device. This account is not the same as the account that was used to interact with Teams.

After initial setup, the device was jailbroken with the checkra1n jailbreak (Panhuyzen, 2021). Specifically the graphical version of checkra1n 0.12.2 was used on an Ubuntu 20.04.2 device. The tool was able to detect the iPhone automatically and the instructions provided were followed to jailbreak the device. Once the device was jailbroken, Cydia, an alternative app store for jail broken iPhones (Freeman, n.d.), was used to enable SSH functionality and the default root user password was changed. Following the jail break, the Teams app was downloaded from the Apple App Store.

3.4.1.3 Android

The Android device was new and had not been used before this study, so it was not wiped. Similar to how the iOS device was jailbroken, the Android phone was rooted to increase the potential for artifact recovery. In order to unlock the bootloader of this Motorola phone, the unlock data was retrieved using Android SDK platform-tools (Android Developers, 2021). Specifically the command "fastboot oem get_unlock_data" was used while the device was in bootloader mode. This code was provided to Motorola's bootloader unlocking portal, and Motorola emailed back an unlock key. This key was used with the command "fastboot oem unlock <bootloader-unlock-key>" to unlock the bootloader.

With the bootloader unlocked TWRP recovery was installed. TWRP is a recovery tool that allows for the installation of third party firmware (TeamWin, 2021). In order to do this, the TWRP 3.4.0 version 2.1 image was transferred to the device and booted to using Android SDK platform-tools. Using TWRP, Magisk version 21.2 was installed, which is a tool that allows the device to actually be rooted (Magisk Manager, 2021). After the device was rooted, the Teams app was installed from the Google Play Store. Because the Play Store requires a Google account, one was created for signed into. This account is not the same as the account that was used to interact with Teams.

3.4.1.4 Administrative Virtual Machine

This machine was created in a VMware vCenter Server environment and accessed remotely with VMware Workstation. A Windows 10 installation ISO file was connected to the virtual disk drive. The installation wizard was used to partition the virtual disk and install Windows 10. During setup a local account was created, not an online Microsoft account. Using the preinstalled Edge browser, three other web browsers were downloaded (Chrome, Firefox, and Opera), one corresponding to each Microsoft account created. The Teams client was also installed on this machine.

3.4.2 Organization Creation

From the admin machine https://teams.microsoft.com was accessed. This led to a login page, from which the administrator's Microsoft account was created, which is further described in Section 3.4.3. After account creation the option to sign up for free was selected. At this time Teams is free if other Microsoft Office 365 apps are not used with it. After providing a name, company name, and region, the organization was created with the newly created admin account in control. Once the other user accounts were created, they were invited to the organization by this user through the Teams client.

3.4.3 Microsoft Account Creation

User accounts were created for each of the three user's populating artifacts, as well as the admin user. Each user was created from the admin machine on separate web browsers to avoid potential issues involving cookies and sessions. Each account was created by going to https://teams.microsoft.com where the option to create a new account was selected. The option to get a new email address was selected, and an @outlook.com email was created for each user. The only information required to create these accounts was a birthday, but after creation other details such as name and profile photos were added by logging in as the user and updating account their information. These details can be seen in Table 3.1. Phone numbers were also used to verify accounts but are not included in this table as they are privately owned. After the four user accounts and the organization were created, the admin user invited the other accounts via email. Each of these user's opened this email on their respective devices and accepted the invite.

Table 3.1. *Account Details*

Device	First	Last	Birthday	Email
Windows	Windy	Whale	1/1/2000	Purdue.CNIT.Sp21.HB.Windy@outlook.com
iOS	Ian	Iguana	1/1/2000	Purdue.CNIT.Sp21.HB.Ian@outlook.com
Android	Andrew	Alpaca	1/1/2000	Purdue.CNIT.Sp21.HB.Andrew@outlook.com
Admin	Terry	Turtle	1/1/2000	Purdue.CNIT.Sp21.HB.TeamMaker@outlook.com

3.4.4 Messages

The text messages sent are each distinct as duplicate messages could cause confusion and would make artifact analysis more difficult. The form "sender:receiver:type:alterations:number" was used for the text messages. While not representative of a natural conversation, this allows for efficient and accurate artifact identification, and is practical given the number of messages that had to be sent. Each element of this format is explained below.

1. Sender: The OS of the sending device [Windows, iOS, or Android]
2. Receiver: The OS of the receiving device [Windows, iOS, Android, or multiple]
3. Type: Describes how the discussion is taking place [direct, group, or channel]
4. Alterations: What type of alteration was used, if any [none, delete, edit, edited, unread]

5. Number: An incrementing number to differentiate messages with the same above properties

As an example, the first direct message, intended for editing, sent from the iOS device to the Android device, was "iOS:Android:direct:edit:1" when the message was initially sent, and then became "iOS:Android:direct:edited:1" after the message had been edited. Similar care was used with the images, videos, and PDF files sent. These files were not constructed in a formulaic way like the text messages, they were instead chosen from document publicly available from the Library of Congress. Links to the chosen document pages on the Library of Congress website (loc.gov) are included, along with the file hashes, in Appendix A.

Most channel messages were sent in the General channel, and viewed by each user prior to extraction, with the only exception being unread and file messages. Each user used a separate channel to send their unread channel messages, so that they could be left truly unread. File channel messages from all devices were sent in a separate file channel.

Group messages were all sent in the same group, as creating multiple groups with the same member's was not possible. Because of this, unread messages could not remain unread to all users, as whichever user sent their message second would see the first messages sent, and whoever sent last would see all messages. For this reason the second and third members selected each message meant to be unread and chose to mark them as unread. The Windows user sent their messages first, leaving the unread messages truly unread, followed by the iOS user and then the Android user.

There were similar issues leaving direct messages unread, as whoever sent the messages last would see the earlier messages. In order to avoid this, each user only messaged one other user. This was done in the same sending order as the group messages. This means the Windows user messaged the iOS user who messaged the Android user, who finally messaged the Windows user.

3.4.5 Calls

For the channel calls, each user called the general channel for five minutes and then fifteen minutes. When starting a channel call there is no option for video or audio call. For group calls each user called the group created earlier for messages. When starting a group call there is an option to start as either an audio or video call. Each user started two of each, one being five

minutes and the other 15 minutes. For direct calls, the same order was used as the direct messages. This means the Windows user called the iOS user who called the Android user, who finally called the Windows user. Similar to group calls, each user made both audio and video calls, as well as five and fifteen minute calls, for a total of four direct calls per user.

3.4.6 Calendar Events

Each user created fifteen calendar events for their calendar. Five of the events were unmodified, five were edited, and five were deleted. These events were named with the same pattern as the text messages, except type is always "calendar." So for example an event named "Windows:multiple:calendar:edited:2" would be the second event created by the windows device that had been edited. All calendar events were mistakenly marked as "multiple" as they were only present on the user's calendar who created them.

3.5 Acquisition

Following population of artifacts across all three devices and ensuring the appropriate items had been viewed by the clients, forensic acquisitions of the devices were made. These acquisitions create copies of the devices as they were at the time so they can be analyzed later. Care is taken to minimize or eliminate any change to the data on the devices. The techniques used vary between devices and are described below.

3.5.1 Windows

The storage drive was removed from the Lenovo laptop so that a forensic copy could be made. The mSATA drive was connected to an mSATA to USB adapter, which was connected to a USB write blocker. Specifically a Tableau Forensic USB 3.0 Bridge (T8u) running firmware version 2.1.0.3 was used. This device prevents any new data from being written to the drive while allowing the data present to be read (Casey, 2009). The write blocker was connected to a computer and using FTK Imager 4.5.0.3 a segmented bit by bit copy of the drive was created.

Magnet AXIOM Process version 4.11.0.24063 was then used to create a case from the forensic copy of the physical device.

Figure 3.1. Laptop storage drive with USB write blocker

3.5.2 iOS and Android

For both iOS and Android, Cellebrite UFED 4PC version 7.42.0.82 was used to perform the extractions. In both cases, a Cellebrite case was created as well. For the iOS device, a full file system extraction was performed. A physical extraction would have been preferred, but this was not an option provided by UFED 4PC. During extraction, the device passcode was provided as well as the root user password. For the Android device, a physical extraction was initially performed. This extraction provided various partitions for the device and few artifacts, none of which were related to Teams. Attempts were made to extract relevant information from these

partitions, but when exported from the file system view of Cellebrite, the sum of all partitions once uncompressed was 3.75GB, far less than the storage capacity of the phone. This suggests that there may have been an error during extraction. Subsequent attempts at physical extraction led to the same result. Because attempts to extract useful information from these partitions were not successful, other extraction modes were used. Full file system ADB was used for this study as relevant Teams artifacts were provided, and the file system could be traversed without issue. Physical extractions are usually preferred as they carry more information, but it is unclear why that is not the case here.

3.6 Analysis

This section describes the methods used to account for the populated artifacts. This includes what the tools were able to find as well as what was discovered by manual investigation. In digital forensics, the term manual often means finding artifacts by directly interacting with a device, for example scrolling through a user's text message history on a phone. For this study, the term manual refers to the non-automated work of an investigator searching for the populated artifacts. While the devices were directly interacted with after extraction to help correlate and confirm artifact details, the term manual in this study only refers to non-automated investigation.

The scope of analysis is limited to the explicitly populated artifacts described in this chapter. While other useful items may be present, such as application usage logs, or device locations, these were not included unless they directly related to an artifact that was intentionally populated.

3.6.1 Forensic Tools

The available features of the forensic tools were used to find any of the populated artifacts. This primarily included the "Analyzed Data" section within Cellebrite and the "Artifacts" section of AXIOM. Both tools also provide timeline analysis functionalities. Because it is known on what days the artifacts were populated, the timeline can be used to find artifacts from a specific time period that could have been potentially overlooked in the analyzed data sections. Other ways the

tools present artifacts were considered, such as the connections and registry sections of AXIOM. Artifacts found in the file system section were not considered to have been discovered by the tools, as having to search through the file system without addition context or help from the tools is seen as a manual task.

A useful feature provided by both tools is keyword searching. Keywords can be used to search all artifacts for the matching term. The keyword searches used were; Teams, team, Microsoft, Skype, Windows, Windy, Whale, iOS, Ian, Iguana, Android, Andrew, Alpaca, channel, group, direct, calendar, none, edit, edited, delete, unread, and multiple. These terms all relate to either Teams, or data that would be expected to be found in the populated artifacts such as messages pieces. Windy Whale, Ian Iguana, and Andrew Alpaca were searched for as these were the names chosen for the users of the Windows, iOS, and Android devices, respectively. The full content of every message is represented in these search terms. While the contents of messages might not always be known to investigators ahead of time, educated guesses are often possible, making keyword searches very useful. In this case, all contents are known, so they can be easily searched. These terms were also used in conjunction with one another at times to narrow the search results. For example a keyword search for all unread direct messages from Windows would look like ":Windows:direct:unread:".

3.6.2 Manual

Manual investigation largely involved exploration of the device file systems, and the data storage structures discovered. While the file system views of the tools were used to initially traverse the devices and identify areas of interest, these areas were exported to be parsed with other techniques. One such technique was to use the Linux terminal tool grep (Free Software Foundation, 2020) to search the exported directories for the same keywords mentioned in the previous section. Many of the artifacts were discovered in SQLite databases. These databases were explored with the help of DB Browser for SQLite version 3.11.1 (*DB Browser for SQLite - About*, 2021). The tool ChromeCacheView version 2.25 by Nirsoft was used for extracting files from a chromium cache structure (NirSoft, 2021). When necessary, property lists and binary property list files had to be converted to XML using a Debian tool called plistutil ("C", Szulecki,

& Bassen, 2020). If a potential artifact source could not be read or parsed any other way, the hex editor HxD version 2.4 was used (Hörz, 2020) or the text editor Notepad++ version 8.1.2 (Ho, 2021).

CHAPTER 4. RESULTS

This chapter describes the results of this investigation. Several data points were populated including account information, text messages, media messages such as image and video, file messages, audio calls, file shares, and calendar events. Each platform has key locations where artifacts were found, these locations are described below. Each artifact type is covered per platform, with specific locations for those artifacts.

4.1 General Artifact Locations

4.1.1 Windows

Windows applications commonly store data in their own folder beneath the AppData directory (Ail, 2020). The AppData folder is commonly located at `C:\Users\<username>\AppData\Roaming` where <username> represents the user whose data is stored there. The AppData path can be changed, but is pointed to by the windows variable %AppData%. The path to the Teams AppData folder was found to be `C:\Users\<username>\AppData\Roaming\Microsoft\Teams`. In the Teams AppData folder, a folder named Cache contains files in the Chromium cache format. This is likely because "the Teams desktop client was developed on Electron, which uses Chromium for rendering" (Maguire et al., 2020, p. 1). The format used is the same as the Chrome web browser, so the program ChromeCacheView version 2.25 by NirSoft (NirSoft, 2021) was able to parse the Teams cache and display the files from the cache. The file structure of the cache can be seen in Figure 4.1 and the contents as parsed by ChromeCacheView can be seen in Figure 4.2.

Also within the Teams AppData folder, a folder named indexedDB that contained a levelDB database structure. LevelDB "is an on-disk key-value store where the keys and values are both arbitrary blobs of data" (Caithness, 2020, p. 5). The file structure of this database can be seen in Figure 4.3. Attempts were made to manually parse this data structure, though none were successful. AXIOM was able to parse some of this structure, so the results of the tool will be more heavily utilized for the Windows platform than the other devices. Despite this, pieces of

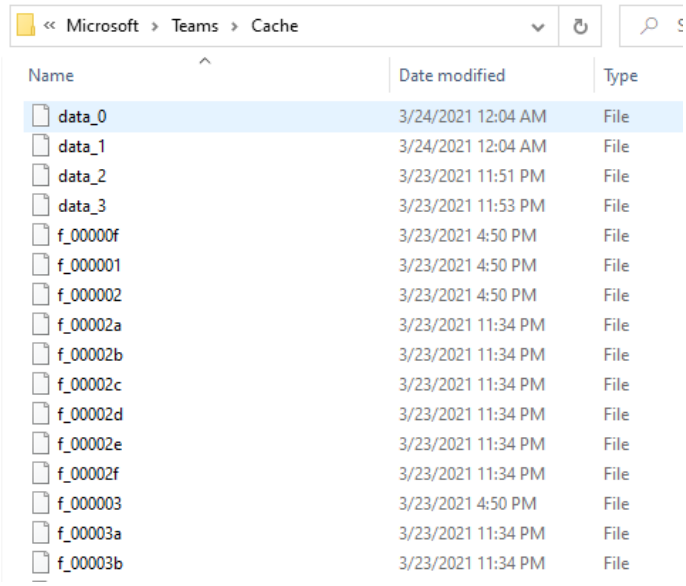


Figure 4.1. Chromium cache extracted from the Windows machine

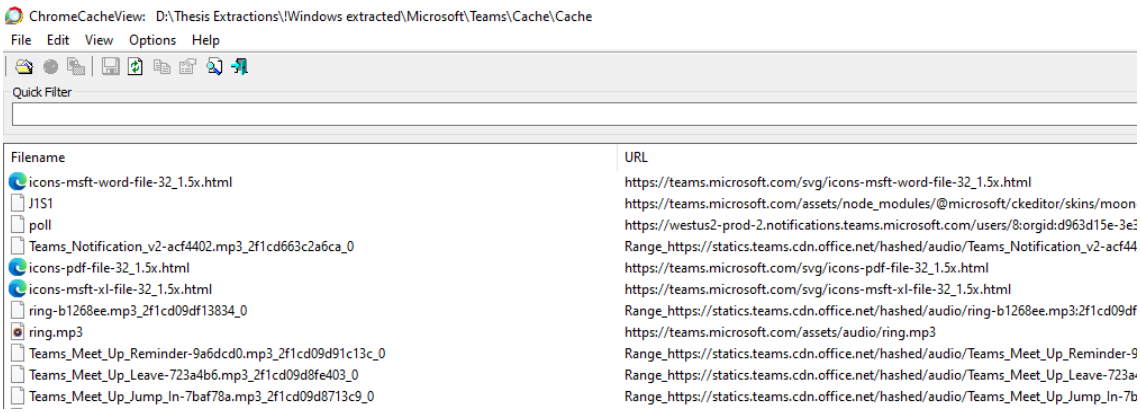


Figure 4.2. The Teams cache explored with ChromeCacheView

message and calendar artifacts can still be seen when reviewing the data with tools like hex editors and regular expression matching.

4.1.2 iOS

A useful location for identifying installed applications on an iOS device is the applicationState.db file located under `\private\var\mobile\Library\FrontBoard\` (Brignoni, 2018a). This database indicated that the application identifier for the Teams app is

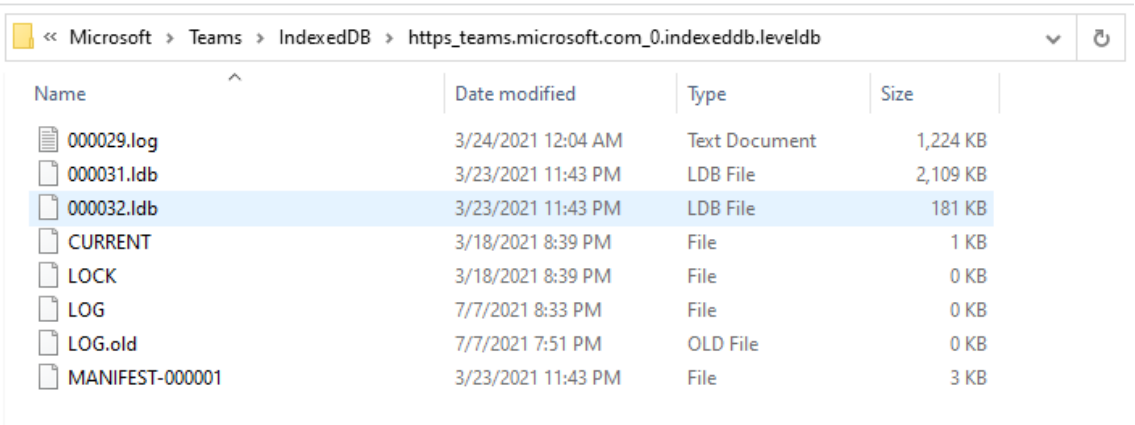


Figure 4.3. The Teams levelDB structure

"com.microsoft.skype.teams" and when following the procedure described by Brignoni (2018a), the first and third paths in Table 4.1 were discovered. The 32 digit long hexadecimal at the end of each path in Table 4.1 represents a Global Unique Identifier (GUID) and is likely, though not guaranteed to be, the same for installations of Teams on other iOS devices. In addition to the two paths indicated in the applicationState.db file, four others were discovered that relate to the "com.microsoft.skype.teams" identifier. All GUID directories encountered contained a file named ".com.apple.mobile_container_manager.metadata.plist" which is a binary property list that contains information about its current directory. If one of these files has a value like "com.microsoft.skype.teams" for the MCMMetadataIdentifier field, then its parent directory was considered related to the Microsoft Teams app and included in Table 4.1.

The directory with the most artifacts is the one that begins with A0426D52. Below that directory is an SQLite database containing the majority of artifacts that were populated onto the iPhone. The full path to this database is `\private\var\mobile\Containers\Shared\AppGroup\A0426D52-2902-4962-B563-D66C3E4A2325\SkypeSpacesDogfood\<GUID>\SkypeSpacesDogfood-<GUID>.sqlite` where <GUID> in both cases is a GUID, but not the same one. This database is shown in the Cellebrite databases section but is not marked as a database related to Teams, nor are the contents of the database contextualized or presented to the user as artifacts. This can be seen in Figure 4.4. A SANS poster from March 2021 suggests that the directory directly containing the SQLite database should be the same as the GUID of the Team, but that was not the case here (Epifani).

This database uses an all caps and Z prefix naming convention, meaning the tables and columns are like ZTABLENAME or ZCOLUMNNAME. In addition to containing the SkypeSpacesDogfood-<GUID>.sqlite database, the A0426D52 directory contains files and images that were present in the Teams chats and file shares.

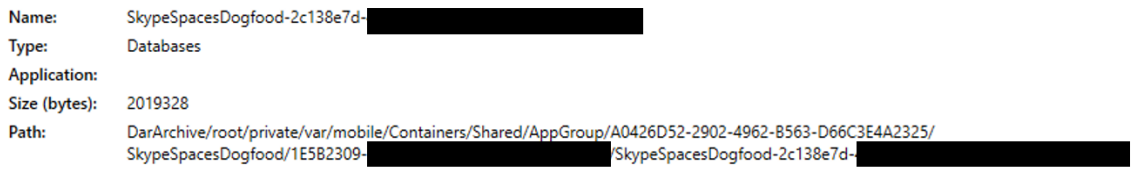


Figure 4.4. The iOS SQLite database file

Table 4.1. *iOS Teams Directories*

\private\var\containers\Bundle\Application\EB91DCF5-F586-44C1-9806-7F22559DF170
\private\var\mobile\Containers\Shared\AppGroup\A0426D52-2902-4962-B563-D66C3E4A2325
\private\var\mobile\Containers\Data\Application\40D88DF7-0DB7-4728-8D29-E04F54F2673D
\private\var\mobile\Containers\Data\PluginKitPlugin\9039DB36-64CB-4C2C-BEEE-56842BBE87A4
\private\var\mobile\Containers\Data\PluginKitPlugin\E860FAF4-C242-46DA-BCDB-F9D7285BAF83
\private\var\mobile\Containers\Data\PluginKitPlugin\02ECFEED-B4F6-4EE6-82F1-01646F6C29D1

4.1.3 Android

All Android artifacts related to Teams were found under the path `\data\data\com.microsoft.teams\` which appears to be the Teams application folder. The package name "com.microsoft.teams" is consistent with the package name for Microsoft Teams on the Google Play Store. However this is similar but different from the name used for iOS which is "com.microsoft.skype.teams" which references Teams predecessor Skype. Within the Teams app directory there are three locations of particular interest. First is `...\databases\SkypeTeams.db` which is an SQLite database counting the majority of populated artifacts, including messages and file details. This database is shown in the Cellebrite databases section as a database related to Teams, but the contents of the database are not contextualized or presented to the user as artifacts. This can be seen in Figure 4.5. The directory cache directly beneath the Teams app directory contains cached images both directly in the directory, and in sub directories. Lastly the directory `...\files\fileCache\` contains files that

were sent in chats and in team file shares. Each file has its original name and is in a parent directory with a random but identifying set of base64 characters. This directory name corresponds with the ID column of the FileCache table in the SkypeTeams.db database.

Application	Row count	Name	Path
Microsoft Teams	3494	SkypeTeams.db	Motorola GSM_XT1965-3_...

Figure 4.5. The SkypeTeams.db file

4.2 Account Information

In addition to the user details populated during account creation, each user was given a unique profile picture that corresponded with their animal theme. Unfortunately these images did not automatically migrate to Teams when set in their Microsoft Account settings, so the users all have default profile images with their initials on a mono colored background. These can be seen in Figure 4.6.

A synopsis of the account information retrievable cab be found in 4.2. If the profile pictures can be found, the data is considered recovered. If there is a way of telling which profile picture belongs to which user, then the context of the artifact is considered recovered. The tools are able to find the profile images, do not show the important context of which user an image belongs to. Information like name, email, and user ID considered account data. Knowing which user is the logged into the Teams client is the context in this case.

4.2.1 Windows

When the levelDB is parsed by AXIOM, both the unique ID and the display name can be found for users who have sent messages. This source seems to only contain users who have sent messages and not those who the local user has seen but has not posted anything. Within the cache there are jfif files named like "displayname=<display%20name>&size=HR<pixels>x<pixels>" where <display%20name> is the user's display name with special characters like spaces escaped,

and <pixels> is the dimensions in pixels of the image. This provides the display name and profile picture for the users the local user has seen, even if they have not posted anything, but does not offer other details about them.

The profile images are stored with the display name of the user they belong to, as seen in Figure 4.7, so both the image data and its context are found. Axiom was able to carve the profile images from the Chromium style cache, but because it was carved instead of parsed from the cache, the file name is not included with the result. An example of this can be seen in Figure 4.8. This means there is no way of knowing which user a picture belongs to, or if it is a profile picture at all, only the image data is recovered by AXIOM.

Figure 4.6. The profile pictures of the iOS, Windows, and Android user respectively

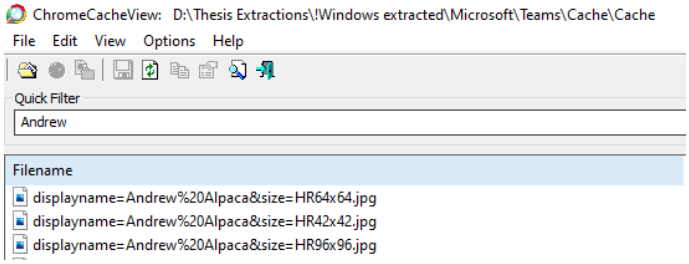


Figure 4.7. The profile pictures of the Android user stored in the cache

4.2.2 iOS

The table ZUSER within the SkypeSpacesDogfood-<GUID>.sqlite database contains a row for each user. Some key columns are ZTSID, ZDISPLAYNAME, ZGIVENNAME, and

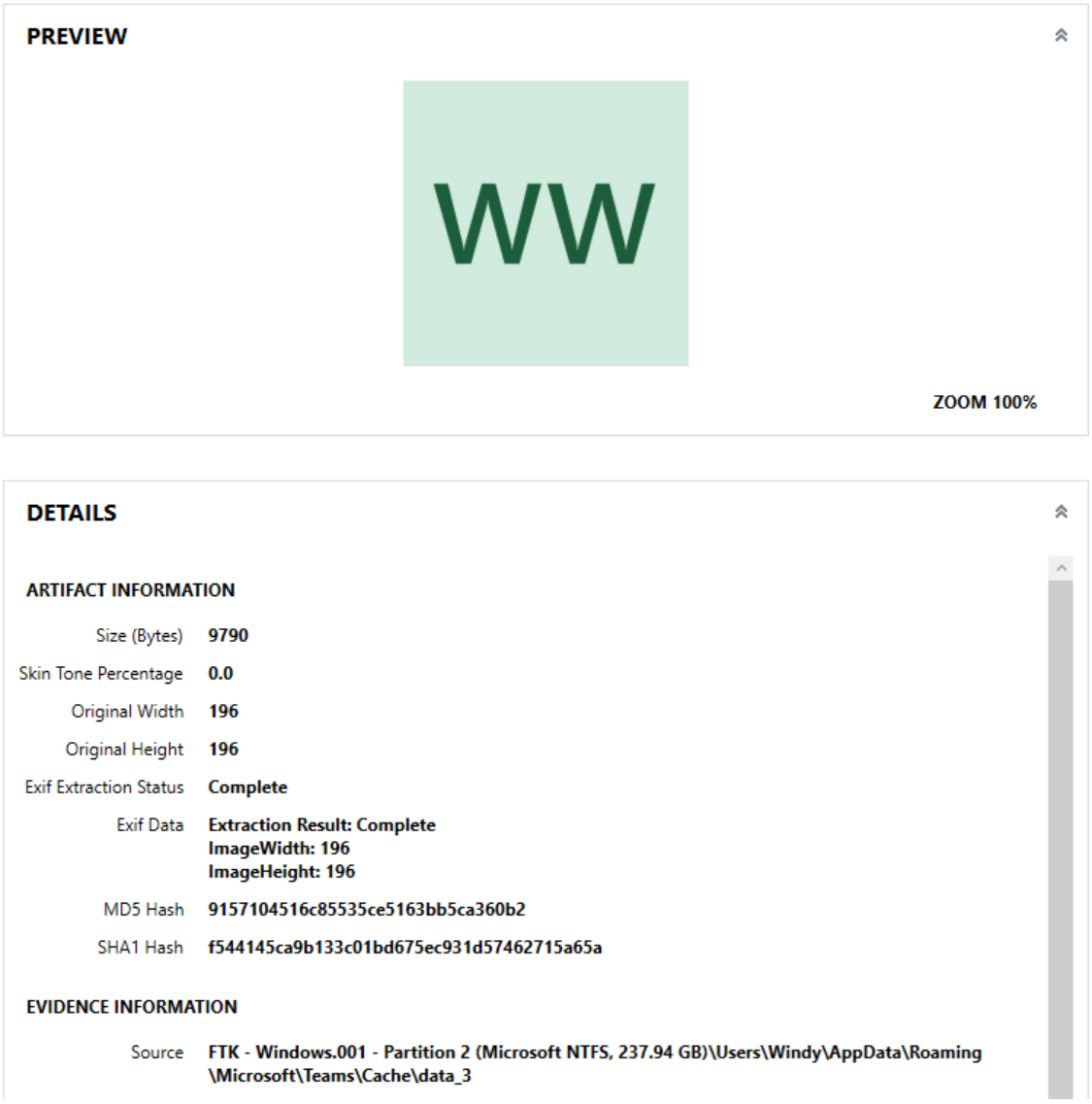


Figure 4.8. Windy's Profile image carved from the chromium cache by AXIOM

ZSURNAME which contains the user's unique ID, their name as it is seen in Teams, their first name, and their last name respectively. The unique user id for each user is found in the ZADID column of this table. The user's email can be found in the ZMAIL column. There are other seemingly useful columns such as ZHOMENUMBER and ZJOBTITLE, but these data attributes were not populated. An excerpt of this table can be seen in Figure 4.9

Under the directory beginning with 40D88DF7 in Table 4.1 there is a directory at `...\Library\Caches\ProfilePhotos\`. This directory contains the photos of the local user

ZUSER

:PARTMENT	ZDISPLAYNAME	ZGIVENNAME	ZHOMENUMBER	ZIMAGEURI	ZJOBTITLE	ZMAIL
r	Filter	Filter	Filter	Filter	Filter	Filter
L	Ian Iguana	Ian Iguana	NULL	NULL	NULL	purdue.cnit.sp...
L	Andrew Alpaca	Andrew Alpaca	NULL	NULL	NULL	purdue.cnit.sp...
L	Terry Turtle	Terry	NULL	NULL	NULL	purdue.cnit.sp...
L	Windy Whale	Windy Whale	NULL	NULL	NULL	purdue.cnit.sp...

Figure 4.9. Part of the ZUSER table

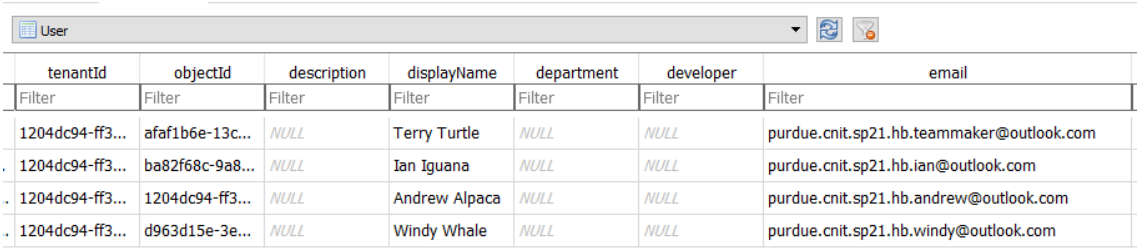
and the other users, as well as group chat icons and the team icon. Unfortunately there is no clear way to tie a profile image to a specific user.

The profile images were found by both manual investigation and Cellebrite, but they are not able to be attributed to specific users, so the context for the images is not found. Account details for foreign users have been fully discovered, but there is not a method for identifying which user is logged into the device, so there is only a partial recovery for the local user. Cellebrite did not recover any account details for the iOS device.

4.2.3 Android

Account information for both the local user and other users that have been interacted with can be found in the SkypeTeams.db database under the User table. The unique GUID of the logged in user was found in every row of the tenantId column. An example of this is seen in Figure 4.10 The unique GUID of both the local user and the foreign users were found in the objectId column. Basic information such as the name of the user is provided in the givenName, surname, and displayName columns and the email of the users is listed twice, once in the email column and once in the mail column. There are several columns that could be useful for quickly determining which users have been contacted most, like chatCount, mentionCount, and callCount.

Profile images for these users can be found in the cache directory, along with other images that have been cached. Unfortunately there is no clear way to tie a profile image to a specific user, or tell which images are profile images and which are images shared in chats or file shares.



User

tenantId	objectId	description	displayName	department	developer	email
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1204dc94-ff3...	afaf1b6e-13c...	NULL	Terry Turtle	NULL	NULL	purdue.cnit.sp21.hb.teammaker@outlook.com
1204dc94-ff3...	ba82f68c-9a8...	NULL	Ian Iguana	NULL	NULL	purdue.cnit.sp21.hb.ian@outlook.com
1204dc94-ff3...	1204dc94-ff3...	NULL	Andrew Alpaca	NULL	NULL	purdue.cnit.sp21.hb.andrew@outlook.com
1204dc94-ff3...	d963d15e-3e...	NULL	Windy Whale	NULL	NULL	purdue.cnit.sp21.hb.windy@outlook.com

Figure 4.10. Part of the User table

Because the profile images were found by both manual investigation and Cellebrite, but not able to be attributed to specific users, only the image data is recovered. Account details for both the local user, and foreign users have been fully discovered. There is a method for identifying which user is logged into the device. Cellebrite did not recover any account details for the Android device.

Table 4.2. *Profile Picture and Account Information Recovery*

			Profile Pictures			Account Details		
		Recovered	Windows	iOS	Android	Windows	iOS	Android
Windows	Manual	Data	✓	✓	✓	✓	✓	✓
		Context	✓	✓	✓	-	-	-
	AXIOM	Data	✓	✓	✓	✓	✓	✓
		Context	-	-	-	-	-	-
iOS	Manual	Data	✓	✓	✓	✓	✓	✓
		Context	-	-	-	✓	✓	✓
	Cellebrite	Data	✓	✓	✓	-	-	-
		Context	-	-	-	-	-	-
Android	Manual	Data	✓	✓	✓	✓	✓	✓
		Context	-	-	-	✓	✓	✓
	Cellebrite	Data	✓	✓	✓	-	-	-
		Context	-	-	-	-	-	-
Total	Manual	Data	3	3	3	3	3	3
		Context	1	1	1	2	2	2
	Tool	Data	3	3	3	1	1	1
		Context	0	0	0	0	0	0

4.3 Messages

There are three forums messages were sent in (team, group, and direct messages), and four conditions each message could have (edited, deleted, unread, or none of these "normal"). The data for messages is the content of the message that was sent. When this is found, the data for that message is considered found. For edited messages, the data is found even if the original form of the message cannot be located. This was the case for all edited messages, the original message was not discovered on any device or forum. The context of a messages includes information about the message other than its content, such as who sent the message and when. For deleted messages, if there is a record of the message, but the contents are missing, the context can still be found, but the data cannot be. Tables 4.3 and 4.4 provide the recovery frequencies for all messages sent, grouped by forum and condition respectively.

4.3.1 Windows

The messages are stored in the levelDB structure. These messages were found by opening the .log and .ldb files in Notepad++ version 8.1.2 and HxD version 2.4. The same keyword searches used in Section 3.6.1 were used with these tools to discover artifacts or pieces of them. Messages can be found from all three forums, including edited, read, unread, and normal messages. The only type of message not found were deleted messages. This suggests, though does not confirm, messages are truly deleted on the Windows platform. The original form of edited messages could not be found. The manual investigation did not find a difference between read and unread messages that had been received. Figure 4.11 shows the contents of a message that has been recovered. While other useful information might be present, the context of the message is not found because the structure cannot be parsed.

AXIOM parsed 29 items, 25 of which were populated messages from the levelDB structure. All of these messages are unread, no direct messages are present. It seems likely that the reason unread messages can be discovered is because they were populated most recently. AXIOM only considers the .log file of the levelDB structure which has only the most recent information (Caithness, 2020). Figure 4.12 shows some of the messages provided by AXIOM.

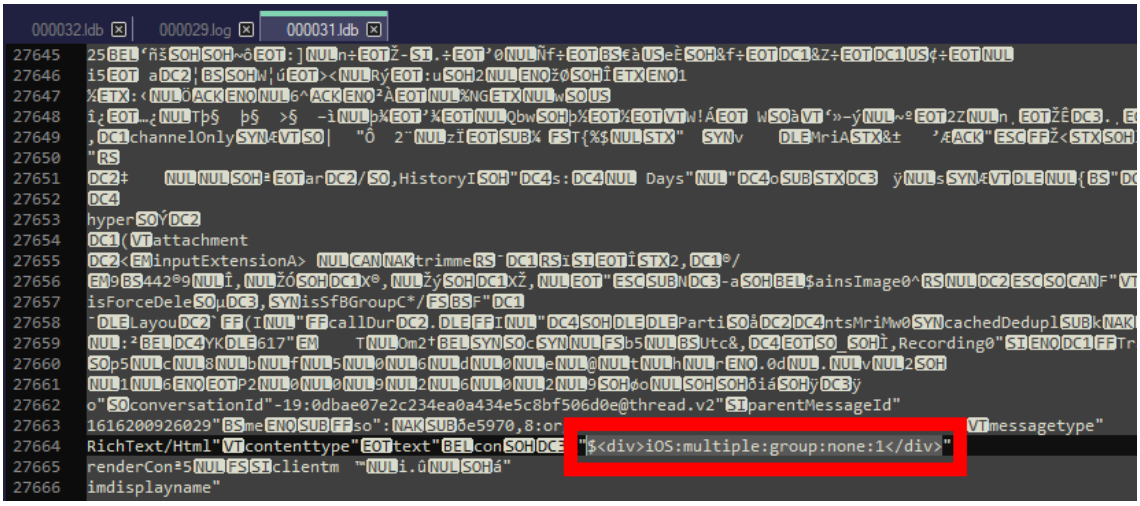


Figure 4.11. An iOS group message found in the levelDB files

EVIDENCE (29)

Sender Display Name	Message Type	Content
Andrew Alpaca	RichText/Html	Android:Windows:direct:unread:1
Andrew Alpaca	RichText/Html	Android:Windows:direct:unread:2
Andrew Alpaca	RichText/Html	Android:Windows:direct:unread:3
Andrew Alpaca	RichText/Html	Android:Windows:direct:unread:4
Andrew Alpaca	RichText/Html	Android:Windows:direct:unread:5
Ian Iguana	RichText/Html	iOS:multiple:group:unread:1

Figure 4.12. Messages recovered by AXIOM from the Windows machine

4.3.2 iOS

The ZTHREAD table in the SkypeSpacesDogfood-<GUID>.sqlite database has five columns of particular interest to investigators. ZTHREADTYPE show what type of thread this row is. The types are chat for either a private or group chat, topic for a channel, meeting for a scheduled meeting, and space is an entire team. ZTHREADTOPIC has the name of the channel, meeting, or group message but is empty for private chats and spaces (teams). ZTSID contains the identifier for the thread and appears to be unique across all of Microsoft Teams. ZMEMBERS contains a binary property list (bplist) per row, that when decoded contains the unique user IDs of those in the thread.

The ZSMESSAGE table has many columns relevant to an investigator trying to understand the messages that were sent. ZFROM and ZIMDISPLAYNAME show who sent the message as the unique user ID and the users name as it is seen in Teams respectively. ZTHREADID indicates which thread the message was sent in. ZCONTENT is the actual message that was sent. Some of the "messages" are not text messages but instead user actions such as adding users or call log entries. Many of the messages representing actions do not have a user in ZIMDISPLAYNAME and have a channel ID instead of a user ID in ZFROM.

ZCOMPOSETIME, ZARRIVALTIME, and ZTS_NUMERICARRIVALTIME contain a UTC timestamp, an Apple Cocoa Core Data timestamp, and as a Unix timestamp for the message respectively. Despite the column names suggesting they might represent different events at different times, all three of the timestamps correspond to the same time, to the precision of one millisecond, for all messages. It is unclear if these timestamps represent the time the message arrived on the client, or the server, and if the client being offline when a message is sent would alter this timestamp. The ZFILES column contains a bplist with details about the file that was sent with a message if a file was sent. While the actual file data is not contained, decoding the bplist can show information about it such as the path where it is located within Teams, and the file name.

There are several columns in the ZSMESSAGE table related to the edit, delete, and read status of messages. ZEDITTIME and ZDELETETIME are Unix timestamps regarding when a message was last edited or deleted respectively. An example of messages that have been deleted but can still be viewed can be seen in Figure 4.13. If the message has not been edited or deleted, there is a null value instead. The content of the deleted messages can be seen as they were before they were deleted, but the original content of edited messages was not found. This shows deleted messages can be recovered from Teams on iOS. Cellebrite did not discover any text messages, or message metadata for the iOS device.

4.3.3 Android

Messages from channels, group messages, and direct messages can be found in the Message table of the SkypeTeams.db database. This database was explored with DB Browser for SQLite version 3.11.1. Events that would not be thought of as messages are present in this table

ZCOMPOSETIME	ZCONTENT	ZDELETETIME
Filter	Filter	<>NULL
2021-03-19T06:10:00.4340000Z	iOS:multiple:channel:deleted:1	1616194672041
2021-03-19T06:10:05.1420000Z	iOS:multiple:channel:deleted:2	1616194676409
2021-03-19T06:10:08.8930000Z	iOS:multiple:channel:deleted:3	1616194681474
2021-03-19T06:10:13.2670000Z	iOS:multiple:channel:deleted:4	1616194686088
2021-03-19T06:10:19.0270000Z	iOS:multiple:channel:deleted:5	1616194691586

Figure 4.13. Messages deleted on iOS and found in the ZSMESSAGE table of the cache database

as well. This includes things like call records, and events like adding a user to a channel. All messages have a messageType of either Text or RichText/Html, though some other events also fall into this category. The content column has the message body while the from and userDisplayName columns have the sending user's unique id and name display name respectively. Messages are in an HTML format and are nearly all wrapped in an HTML div tag.

Messages that have been edited show the final form of the message, but also have a value in the editTime column indicating when in Unix milliseconds the message was edited. Messages that were not edited have a null value in the editTime column. Messages that were deleted have an empty div tag for message content, meaning the message was actually deleted. In addition there is a value in the deleteTime column indicating when the message was deleted, similar to the editTime column. Unlike the editTime column, messages that have not been deleted have a value of 0 instead of null. While there is an isRead column, the value is 0 for all messages and there does not seem to be a distinction between read and unread messages. Record of edited messages can be seen in Figure 4.14 File messages have a hasFileAttachment value of 1, and those that do not, have a value of 0. Cellebrite did not discover any text messages, or message metadata for the Android device.

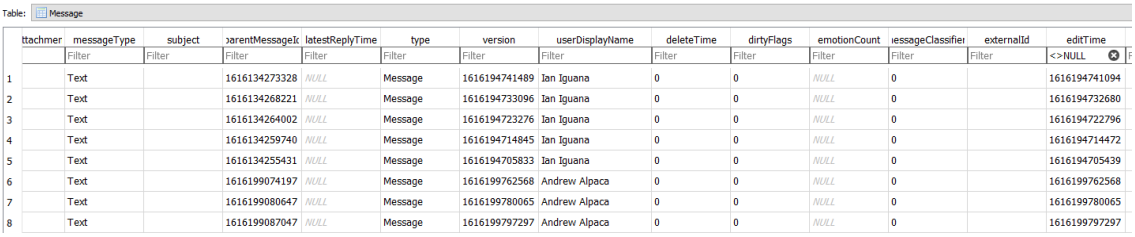


Table: Message

	ttachmer	messageType	subject	parentMessageId	latestReplyTime	type	version	userDisplayName	deleteTime	dirtyFlags	emotionCount	iessageClassifier	externalId	editTime
		Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	<>NULL
1		Text		1616134273328	NULL	Message	1616194741489	Ian Iguana	0	0	NULL	0		1616194741094
2		Text		1616134268221	NULL	Message	1616194733096	Ian Iguana	0	0	NULL	0		1616194732680
3		Text		1616134264002	NULL	Message	1616194723276	Ian Iguana	0	0	NULL	0		1616194722796
4		Text		1616134259740	NULL	Message	1616194714845	Ian Iguana	0	0	NULL	0		1616194714472
5		Text		1616134255431	NULL	Message	1616194705833	Ian Iguana	0	0	NULL	0		1616194705439
6		Text		1616199074197	NULL	Message	1616199762568	Andrew Alpaca	0	0	NULL	0		1616199762568
7		Text		1616199080647	NULL	Message	1616199780065	Andrew Alpaca	0	0	NULL	0		1616199780065
8		Text		1616199087047	NULL	Message	1616199797297	Andrew Alpaca	0	0	NULL	0		1616199797297

Figure 4.14. Record of edited messages in the Message table of the Android teams cache database

Table 4.3. *Message Recovery by Forum*

			Forum							
			Channel		Group		Direct			
		Recoverable	$n = 60$	%	$n = 60$	%	$n = 40$	%	$N = 160$	%
Windows	Manual	Not	39	65.0	32	53.3	19	47.5	90	56.3
		Data Only	21	35.0	28	46.7	21	52.5	70	43.8
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0
	AXIOM	Not	60	100.0	46	76.7	30	75.0	136	85.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	14	23.3	10	25.0	24	15.0
iOS	Manual	Not	0	0.0	1	1.7	0	0.0	1	0.6
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	60	100.0	59	98.3	40	100.0	159	99.4
	Cellebrite	Not	60	100.0	60	100.0	40	100.0	160	100.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0
Android	Manual	Not	0	0.0	0	0.0	0	0.0	0	0.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	15	25.0	15	25.0	10	25.0	40	25.0
		Fully	45	75.0	45	75.0	30	75.0	120	75.0
	Cellebrite	Not	60	100.0	60	100.0	40	100.0	160	100.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0

4.4 Calls

Similar to messages, calls were made in three different forums (team, group, and direct calls). For Group and Direct calls, both audio and video calls were made. For team calls there is no option for audio or video, just meet, so this distinction is not present. While there are many data points relevant to a call, to have the actual call contents an audio recording, video recording,

Table 4.4. *Message Recovery by Condition*

			Condition									
			None		Edited		Deleted		Unread			
		Recoverable	$n = 40$	%	$n = 40$	%	$n = 40$	%	$n = 40$	%	$N = 160$	%
Windows	Manual	Not	19	47.5	19	47.5	40	100.0	12	30.0	90	56.3
		Data Only	21	52.5	21	52.5	0	0.0	28	70.0	70	43.8
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
	AXIOM	Not	40	100.0	40	100.0	40	100.0	16	40.0	136	85.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	24	60.0	24	15.0
iOS	Manual	Not	0	0.0	0	0.0	1	2.5	0	0.0	1	0.6
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	40	100.0	40	100.0	39	97.5	40	100.0	159	99.4
	Cellebrite	Not	40	100.0	40	100.0	40	100.0	40	100.0	160	100.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
Android	Manual	Not	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	40	100.0	0	0.0	40	25.0
		Fully	40	100.0	40	100.0	0	0.0	40	100.0	120	75.0
	Cellebrite	Not	40	100.0	40	100.0	40	100.0	40	100.0	160	100.0
		Data Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Context Only	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0
		Fully	0	0.0	0	0.0	0	0.0	0	0.0	0	0.0

or transcript of the call would be necessary. Nothing of this nature was discovered, but important context information about the calls can be found. This includes information like when the call was and who it was with. Table 4.5 accounts for the call records that have been discovered. Calls of both five and fifteen minutes were made, but there was not a case where a call of one length was found, and the corresponding call (same forum, same caller) was not also found. For this reason calls that had the same properties except for length were group in Table 4.5. This table also contains intersections of artifact conditions that were not populated. As previously mentioned, the Windows user called the iOS user, who called the android user, who called the Windows user. So for example there is no call from the iOS user to the Windows user. The intersection where this type of artifact would be on the table is marked with an asterisk.

4.4.1 Windows

Unfortunately no full call records could be found in the cache, levelDB structure, or anywhere else on the Windows device. It seems likely that there are call records similar to those described in the iOS and Android section that mark the beginning and end of a call. This is

believed because the terms "event/call" and "partlist" are seen in the levelDB files. These terms were seen in the call records of both the iOS and Android devices. Figure 4.15 is an example of both of these terms being present in the levelDB files. These partial records were unable to attributed to specific calls.

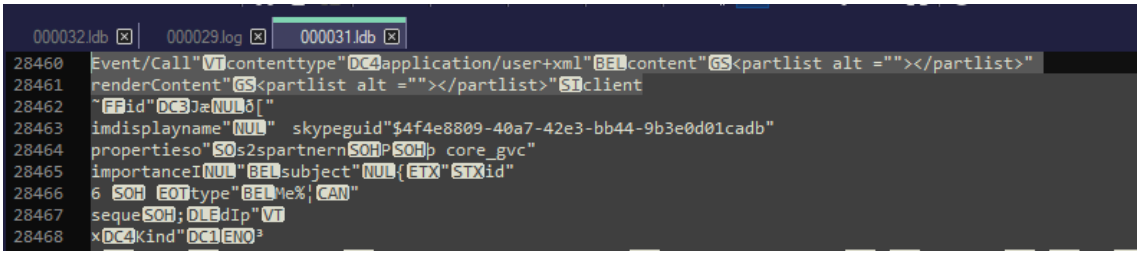


Figure 4.15. Potential call log fragments found in Window's levelDB storage

4.4.2 iOS

Data about Teams calls on the iOS device can be seen in the native call log. The data was collected from the native iOS call log located at `\private\var\mobile\Library\CallHistoryDB\CallHistory.storedata`. The native iOS call log is likely the best source of information for these calls, as it has information like the name of the call or who it was with, the time, whether it was incoming or outgoing, if it was answered or not, and the duration. Figure 4.16 shows some of these call logs as shown by Cellebrite. Cellebrite's results were verified by extracting the call history database and opening it with DB Browser for SQLite. The ZCALLRECORD table had the same artifacts displayed by Cellebrite. This is shown in Figure 4.17.

There are also call records in the ZMESSAGE table of the SkypeSpacesDogfood-<GUID>.sqlite database. Rows with a value of Event/Call for ZMESSAGETYPE are calls that occurred. Calls have two rows in this table, one for when the call started and one for when the call ended. In the ZCONTENT column, rows that correspond to the start of a call have a single XML tag called partlist that is empty. Rows that represent the end of a call have a self-closing tag called ended and a partlist tag with child objects called part that

Parties	Timestamp	Duration	Status
From: Windy Whale Windy Whale	3/23/2021 9:26:43 PM(UTC-4)	00:15:13	Answered
To: Andrew Alpaca Andrew Alpaca	3/23/2021 8:39:39 PM(UTC-4)	00:15:08	Answered
From: Windy Whale Windy Whale	3/23/2021 7:55:03 PM(UTC-4)	00:05:06	Answered
To: Andrew Alpaca Andrew Alpaca	3/23/2021 7:43:42 PM(UTC-4)	00:05:02	Answered
From: Windy Whale Windy Whale	3/23/2021 7:22:25 PM(UTC-4)	00:15:03	Answered
To: Andrew Alpaca Andrew Alpaca	3/23/2021 6:14:01 PM(UTC-4)	00:15:17	Answered
From: Windy Whale Windy Whale	3/23/2021 6:07:49 PM(UTC-4)	00:05:14	Answered
To: Andrew Alpaca Andrew Alpaca	3/23/2021 5:55:36 PM(UTC-4)	00:05:02	Answered

Figure 4.16. Teams calls in iOS Native call log displayed by Cellebrite

ZCALLRECORD

RIFICATIONSTA	ZDATE	ZDURATION	)UNTF	)CATI	ZNAME
Filter	Filter	Filter			Filter
0	638242003.539413	913.038884997368	NULL	<<...	Windy Whale
0	638239179.702515	908.675238966942	NULL	<<...	Andrew Alpaca
0	638236503.400349	306.893555045128	NULL	<<...	Windy Whale
0	638235822.148394	302.520467996597	NULL	<<...	Andrew Alpaca
0	638234545.594697	903.171146035194	NULL	<<...	Windy Whale
0	638230441.377246	917.025946974754	NULL	<<...	Andrew Alpaca
0	638230069.84787	314.565643906593	NULL	<<...	Windy Whale

Figure 4.17. iOS native call log database in SQLite viewer

represents a user and has information about that user. This would appear to be a participant list, with the participants of the call, and how long they were a part of the call.

Cellebrite displays the native call log data as coming from an unspecified app. Despite not indicating which app the calls came from, there is sufficient information to consider all calls recovered by Cellebrite. The manual investigation has also recovered call records for all calls both in the native call log and the SQLite database.

4.4.3 Android

Information about calls can be found in the SkypeTeams.db file under the Message table and the SkypeCall table. In the Message table, rows with a value of Event/Call for messageType are calls that occurred. Calls have two rows in this table, one for when the call started and one for

when the call ended. Rows with the same skypeGuid are for the same call. In the content column, rows that correspond to the start of a call have a single XML tag called partlist that is empty. Rows that represent the end of a call have a self-closing tag called ended and a partlist tag with child objects called part that represents a user and has information about that user. This would appear to be a participant list, with the participants of the call, and how long they were a part of the call. The approximate length of the call in milliseconds can be calculated by subtracting the arrivalTime or composeTime of the row representing the start of a call from the arrivalTime of the row representing the end of the call. For example in Figure 4.18 rows one and two are from the same call, because they have the same skypeGuid (not pictured). Subtracting the composeTime of the earlier record with an empty part list from the later record with an ended tag and a complete participant list will leave 34735 milliseconds or 347 seconds. This is roughly the same time listed in the filled part list. This table only has calls made in team channels, or group messages, and does not have records of direct calls. Records of direct calls were not able to be located.

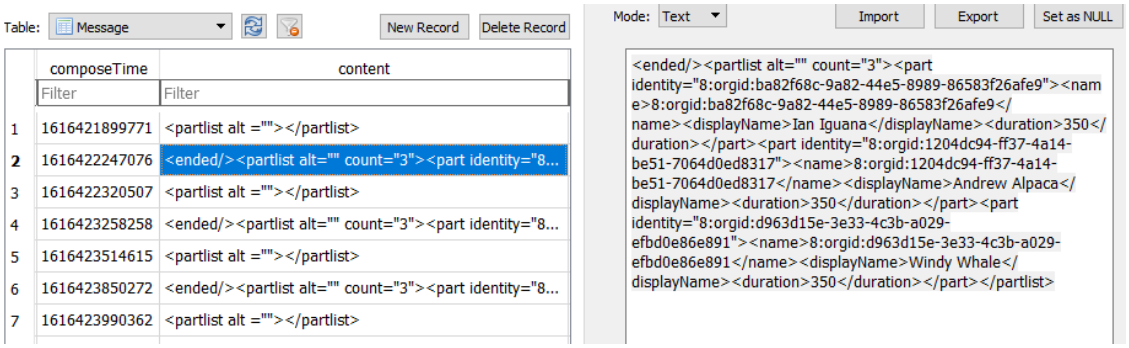


Figure 4.18. Call beginning and end records in the Android database

There are some columns that are most useful when cross referenced with other tables. The From column provides the unique ID of the user who initiated the call, and their name can be found in the user table. What channel a call took place in can be found by comparing the conversationID column to the threadId column of the Thread table and looking at the corresponding displayName. If the displayName is blank, then the call was not in a channel. For channel meetings, the name of a meeting can be found by correlating the skypeGuid in the Message table with the skypeGuid in the SkypeCall table.

Cellebrite did not recover any categories of call artifacts. The manual investigation was able to find sufficient details for all team and group calls to be considered fully found. Unfortunately direct call artifacts were not found.

Table 4.5. *Call Details Recovery*

			Windows		iOS		Android	
Caller	Forum		Manual	AXIOM	Manual	Cellebrite	Manual	Cellebrite
Windows	Channel		0	0	2	2	2	0
	Group		0	0	4	4	4	0
	Direct		0	0	4	4	*	*
iOS	Channel		0	0	2	2	2	0
	Group		0	0	4	4	4	0
	Direct		*	*	4	4	0	0
Android	Channel		0	0	2	2	2	0
	Group		0	0	4	4	4	0
	Direct		0	0	*	*	0	0
Total	Channel	$n = 6$	0	0	6	6	6	0
		%	0	0	100	100	100	0
	Group	$n = 12$	0	0	12	12	12	0
		%	0	0	100	100	100	0
	Direct	$n = 8$	0	0	8	8	0	0
		%	0	0	100	100	0	0

Note. Cells with a * are omitted as no direct call was populated that would correspond with the cell.

4.5 Channel File Share / File Messages

Video, image, and PDF files were shared across three forums (team, group, and direct chat). In addition a PDF was sent by each device to the team file share, and each device created both a word and excel document in the same file share. The data for a file is considered recovered if the actual file can be found. It is possible to find a file but not have any context for it, such as who sent it, where was it downloaded from, and when. In this case the data is found but not the context. The reverse is also possible where records for a file are found but not the file itself. Table 4.6 accounts for both of these types of recoveries. The symbol ✓in a D (Data) column means the file was found, and a ✓in the C (context) column means record for that file were found. A dash in a column indicates the file or other data about it could not be recovered. Just as before, due to the

way direct messages were handled, there are some intersections that have no valid artifact that was populated. These table cells are noted with an asterisk.

4.5.1 Windows

Several JPG files were discovered in the cache that were sent as file messages. Specifically all three images sent by the Windows user, as well as the Android channel image. Within the cache all four files had the same name of 1.jpg. This is seen in figure 4.19 as ChromeCacheView is used to find and extract the files from the cache. AXIOM also discovered these images via carving. In addition all three PDFs shared in the file system were found in the chrome cache, both by manual investigation and carved by AXIOM. Using the known file names, the levelDB files were searched for references to the populated files. file names for two videos, two shared PDFs, and the Android file share excel document were found. Unfortunately other information about these files could not pulled from the levelDB

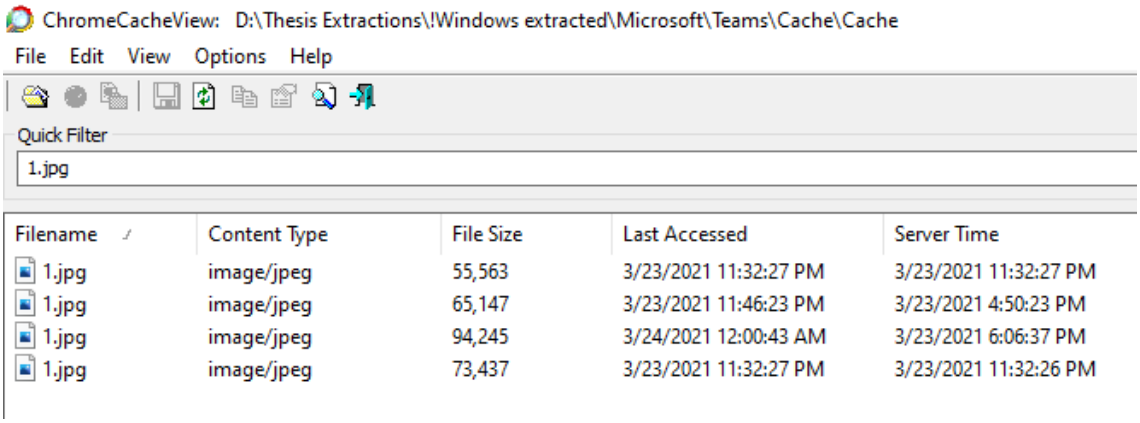


Figure 4.19. Several files named 1.jpg in the chromium cache that correspond to sent file images

AXIOM carved the same files that were found in the cache, listed above. AXIOM recovered no other file message files, and no metadata about them. All three PDFs from the file share were carved from the cache by AXIOM, and these files were recovered manually from the cache as well. AXIOM did not find any other file share files or metadata about them. Manual investigation found a partial file name that corresponds to the Android Excel sheet, as well as the Windows group and direct PDFs but did not find any other file share or file message files.

4.5.2 iOS

Details about files found in channel file shares can be found in the ZFILELISTING table of the SkypeSpacesDogfood-<GUID>.sqlite database. This table contains information about files that are in a file share, which includes those that were sent via channel message, as those files are automatically added to the channel file share. Information such as the name of the file, the ID of the thread it belongs to, and the last modified time as a UTC time stamp can be found in ZTITLE, ZTHREADID, and ZLASTMODIFIEDTIME respectively. Figure 4.20 is an excerpt from the ZFILELISTING table in the SQLite database. More information can be found about each file in the ZSFILE table, which also has rows for files from direct and group messages. Many of the same columns from ZFILELISTING are present as well as other useful information like who created the file and when, as well as who last modified the file.

Table: ZFILELISTING

	Z_PK	Z_ENT	Z_OPT	ZISFOLDER	ASTMODIFIEDTI	RENTREFERENC	ZTHREADID	ZTITLE
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	13	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Windows.xlsx
2	14	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Windows.docx
3	15	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Windows.pdf
4	12	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Android.xlsx
5	16	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Android.docx
6	17	12	1	0	2021-03-24T0...	root	19:11626a74...	fs_Android.pdf
7	11	12	2	0	2021-03-24T0...	root	19:11626a74...	fs_iOS.xlsx
8	10	12	3	0	2021-03-24T0...	root	19:11626a74...	fs_iOS.docx
9	9	12	6	0	2021-03-24T0...	root	19:11626a74...	fs_iOS.pdf
10	8	12	7	0	2021-03-20T0...	root	19:11626a74...	Windows_channel_pdf.pdf
11	5	12	7	0	2021-03-20T0...	root	19:11626a74...	Windows_channel_vid.mpeg
12	3	12	7	0	2021-03-20T0...	root	19:11626a74...	Windows_channel_img.jpg
13	7	12	7	0	2021-03-20T0...	root	19:11626a74...	Android_channel_pdf.pdf
14	4	12	7	0	2021-03-20T0...	root	19:11626a74...	Android_channel_vid.mp4
15	2	12	7	0	2021-03-20T0...	root	19:11626a74...	iOS_channel_pdf 1.pdf
16	6	12	7	0	2021-03-20T0...	root	19:11626a74...	iOS_channel_vid.mp4
17	1	12	7	0	2021-03-20T0...	root	19:11626a74...	iOS_channel_pdf.pdf

Figure 4.20. An excerpt from the ZFILELISTING table

The actual files can be found near the SQLite database under the following downloads directory `...\A0426D52-...\SkypeSpacesDogfood\Downloads`. In this directory are .tmp files that can be viewed as images. While .pdf and .xlsx files can be found in directories that only contain one file, no video or .docx files were discovered.

Cellebrite recovered all image but two files. Specifically the iOS group and direct message images were not found. No videos, PDFs, or other file metadata was recovered by Cellebrite. Word files were the only file share files not found by Cellebrite, or by manual investigation. Manual investigation recovered 22 file messages, and file share files. No video files were found. Metadata about all three channel videos, as well as metadata for the iOS group and direct videos was discovered in the SQLite database.

4.5.3 Android

The FileCache and FileInfo tables within the SkypeTeams.db table have information about files in group messages, direct messages, channel messages, and those put directly in a channel file share. The FileCache table specifically has information about the files that are in the `...\files\fileCache\` directory within the Teams app directory. The id column of this table corresponds to the names of the directories that contain the cached file. This table only has PDF files listed, but this does not mean other types of files are never stored here. The FileInfo table has information about varied types of files, including PDFs. The FileUploadTask table is useful as it has records of only the files uploaded by the local user. Twelve of the populated files are able to be found, and information on thirteen other files was found in the FileInfo or FileUploadTask tables. An example of this file data can be seen in Figure 4.21 Cellebrite recovered the same twelve files recovered during manual investigation. Cellebrite did not provide any file context for these twelve files or any others.

4.6 Calendar

A free instance of Microsoft Teams was used for this study, which requires no other Office 365 products be connected to the instance. As a result the users did not receive outlook calendars connected to the team, and events were schedule using the "Meetings" feature instead. This is similar in functionality to "Calendar" but not the same. For this reason, these results may not apply to enterprise users with an Outlook calendar. Each user created 15 schedule events, 5 unmodified, 5 edited, 5 deleted. If the title of a populated event can be found, then the data for that

Table: FileInfo

	fileName	type	lastModifiedTime	sFolde	lastModifiedBy
	Filter	Filter	Filter		Filter
1	iOS_channel_vid.mp4	mp4	2021-03-20T02:4...	0	Ian Iguana
2	iOS_channel_pdf.pdf	pdf	2021-03-20T01:1...	0	Ian Iguana
3	iOS_channel_pdf 1.pdf	pdf	2021-03-20T02:4...	0	Ian Iguana
4	fs_iOS.xlsx	xlsx	2021-03-24T01:2...	0	Ian Iguana
5	fs_iOS.pdf	pdf	2021-03-24T01:0...	0	Ian Iguana
6	fs_iOS.docx	docx	2021-03-24T01:2...	0	Ian Iguana
7	fs_Windows.xlsx	xlsx	2021-03-24T03:3...	0	Windy Whale
8	fs_Windows.pdf	pdf	2021-03-24T03:3...	0	Windy Whale
9	fs_Windows.docx	docx	2021-03-24T03:3...	0	Windy Whale
10	fs_Android.xlsx	xlsx	2021-03-24T01:3...	0	Andrew Alpaca
11	fs_Android.pdf	pdf	2021-03-24T01:3...	0	Andrew Alpaca
12	fs_Android.docx	docx	2021-03-24T01:3...	0	Andrew Alpaca
13	Windows_channel_vid.mpeg	mpeg	2021-03-20T05:0...	0	Windy Whale
14	Windows_channel_pdf.pdf	pdf	2021-03-20T05:0...	0	Windy Whale
15	Windows_channel_img.jpg	jpg	2021-03-20T05:0...	0	Windy Whale
16	Android_channel_vid.mp4	mp4	2021-03-20T02:5...	0	Andrew Alpaca
17	Android_channel_pdf.pdf	pdf	2021-03-20T03:0...	0	Andrew Alpaca

Figure 4.21. An excerpt from the FileInfo table

event is considered to be found. If other information can be found such as the time and location of a calendar event, then the context is considered found. This is reflected in Table 4.7. Deleted events are considered to have their data recovered only if the original name of the event is found.

4.6.1 Windows

Similar to the call records, there were no full calendar records discovered. Notepad++ was used to open the .log and .ldb files of the levelDB structure and search for the keywords in Section 3.6.1. This can be seen in Figure 4.22. This reveals that there are calendar records present. The full extent of what is present is not currently known because this structure has not been

Table 4.6. *File Recovery*

			Windows				iOS				Android			
			Manual		AXIOM		Manual		Cellebrite		Manual		Cellebrite	
Sender	Forum	Type	D	C	D	C	D	C	D	C	D	C	D	C
Windows	Channel	Video	-	-	-	-	-	✓	-	-	-	✓	-	-
		Image	✓	-	✓	-	✓	✓	✓	-	✓	✓	✓	-
		PDF	-	-	-	-	✓	✓	✓	-	✓	✓	✓	-
	Group	Video	-	-	-	-	-	-	-	-	-	-	-	-
		Image	✓	-	✓	-	✓	-	✓	-	✓	-	✓	-
		PDF	-	✓	-	-	✓	-	✓	-	✓	✓	✓	-
	Direct	Video	-	-	-	-	-	-	-	-	*	*	*	*
		Image	✓	-	✓	-	✓	-	✓	-	*	*	*	*
		PDF	-	✓	-	-	✓	-	✓	-	*	*	*	*
	File Share	PDF	✓	-	✓	-	✓	✓	✓	-	✓	✓	✓	-
		Word	-	-	-	-	-	✓	-	-	-	✓	-	-
		Excel	-	-	-	-	✓	✓	✓	-	-	✓	-	-
iOS	Channel	Video	-	-	-	-	-	✓	-	-	-	✓	-	-
		Image	-	-	-	-	✓	-	✓	-	-	-	-	-
		PDF	-	-	-	-	✓	✓	✓	-	✓	✓	✓	-
	Group	Video	-	✓	-	-	-	✓	-	-	-	-	-	-
		Image	-	-	-	-	✓	✓	-	-	-	-	-	-
		PDF	-	-	-	-	✓	✓	-	-	-	-	-	-
	Direct	Video	*	*	*	*	-	✓	-	-	-	-	-	-
		Image	*	*	*	*	✓	✓	-	-	-	-	-	-
		PDF	*	*	*	*	✓	✓	-	-	✓	✓	✓	-
	File Share	PDF	✓	-	✓	-	✓	✓	✓	-	✓	✓	✓	-
		Word	-	-	-	-	-	✓	-	-	-	✓	-	-
		Excel	-	-	-	-	✓	✓	✓	-	-	✓	-	-
Android	Channel	Video	-	✓	-	-	-	✓	-	-	-	✓	-	-
		Image	✓	-	✓	-	✓	-	✓	-	-	-	-	-
		PDF	-	-	-	-	✓	✓	✓	-	✓	✓	✓	-
	Group	Video	-	-	-	-	-	-	-	-	-	✓	-	-
		Image	-	-	-	-	✓	-	✓	-	-	✓	-	-
		PDF	-	-	-	-	✓	-	✓	-	✓	✓	✓	-
	Direct	Video	-	-	-	-	*	*	*	*	-	✓	-	-
		Image	-	-	-	-	*	*	*	*	-	✓	-	-
		PDF	-	-	-	-	*	*	*	*	✓	✓	✓	-
	File Share	PDF	✓	-	✓	-	✓	✓	✓	-	✓	✓	✓	-
		Word	-	-	-	-	-	✓	-	-	-	✓	-	-
		Excel	-	✓	-	-	✓	✓	✓	-	-	✓	-	-

Note. D = Data, C = Context

Cells with a * are omitted as no direct file message was populated that would correspond with the cell.

successfully parsed, and AXIOM is not displaying these records. Information related to eleven of the fifteen events is present. Events that have been edited have their new edited values. Records of events that were deleted are still present, suggesting that events are not truly deleted, or at least not right away, by the Windows Teams client. AXIOM did not recover any calendar event details in any form.

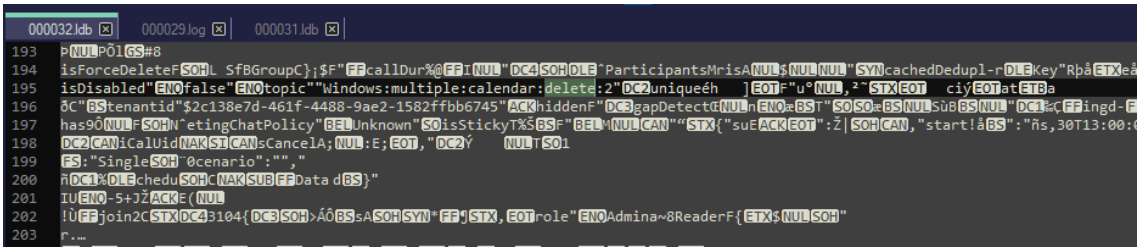


Figure 4.22. A deleted event in the levelDB storage

4.6.2 iOS

The ZCALEVENTDETAILS table within the SkypeSpacesDogfood-<GUID>.sqlite database is where calendar events are stored. The start time and end time are provided as an Apple Cocoa Core Data timestamp in ZSTARTTIME and ZENDTIME, and the event title can be found in the ZSUBJECT column. There are only two events in this table, but fortunately the tables ZTHREAD and ZCONVERSATION also have all of the meetings with the correct names listed under the ZTOPIC column, including the meetings that were deleted. The meetings that were edited have only the new edited details. In the ZTHREAD table the ZMEETING column contains a bplist that when decoded contains the subject, start time, end time, and organizer ID for the meeting. Figures 4.23 and 4.24 show a bplist from this column being decoded with plistutil. The event details can be found in the resulting XML.

Cellebrite did not recover any calendar event details. Manual investigation fully recovered both normal and deleted event details for all 15 events. The original data from edited events could not be found, but all 5 edited events were found.

```
cflstudent@CFLWIN12:/mnt/d/Thesis Extractions$ plistutil --infile iOS_multiple_calendar_none_5.bplist --outfile iOS_multiple_calendar_none_5.xml
cflstudent@CFLWIN12:/mnt/d/Thesis Extractions$
cflstudent@CFLWIN12:/mnt/d/Thesis Extractions$ cat iOS_multiple_calendar_none_5.xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>$version</key>
        <integer>100000</integer>
        <key>$archiver</key>
        <string>NSKeyedArchiver</string>
        <key>$top</key>
        <dict>
                <key>root</key>
                <dict>
                        <key>CF$UID</key>
                        <integer>1</integer>
                </dict>
        </dict>
        <key>$objects</key>
        <array>
```

Figure 4.23. A binary property list being converted to XML using plistutil

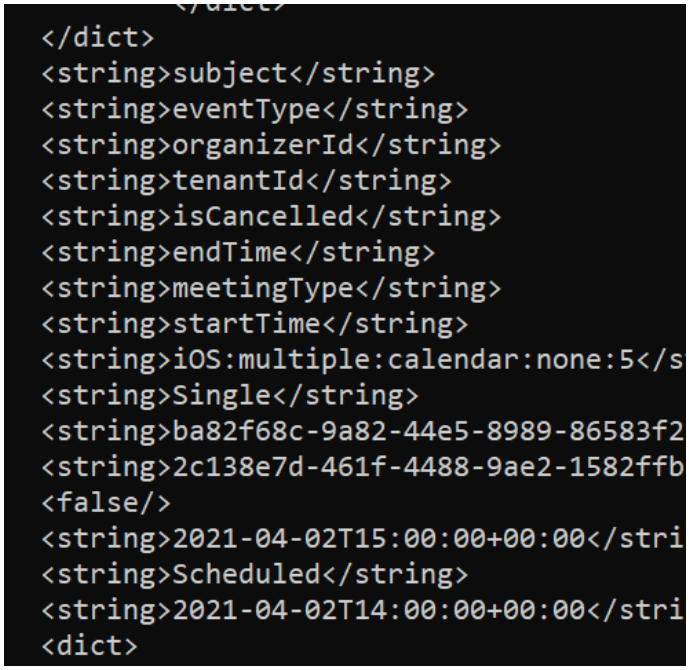


Figure 4.24. An XML dictionary with the details of a scheduled event

4.6.3 Android

Information for calendar events is the CalendarEventsDetails table in the SkypeTeams.db database. The start and end times in Unix millisecond is provided in the startTime and endTime columns. Other useful columns include subject, location, and information about the organizer. Just like with the iOS device, only two events show up in this table. In this case it is one edited and one unmodified event. The edited events show their edited values The table ChatConversation lists all 15 calendar events, including the deleted events, but does not include any event details. Because there aren't any bplists like on the iOS device, or similar secondary sources of information, only the two events could be fully recovered. Cellebrite did not recover any calendar event details.

Table 4.7. *Calendar Recovery*

						Total	
			None	Edited	Deleted	N = 15	%
Windows	Manual	Data	3	3	5	11	73.3
		Context	0	0	0	0	0.0
	AXIOM	Data	0	0	0	0	0.0
		Context	0	0	0	0	0.0
iOS	Manual	Data	5	5	5	15	100.0
		Context	5	5	5	15	100.0
	Cellebrite	Data	0	0	0	0	0.0
		Context	0	0	0	0	0.0
Android	Manual	Data	5	5	5	15	100.0
		Context	1	1	0	2	13.3
	Cellebrite	Data	0	0	0	0	0.0
		Context	0	0	0	0	0.0

4.7 Hypotheses

There are two hypotheses for this study. Both hypotheses favor finding artifacts over not finding them. Specifically the hypotheses are:

- H_1 More than 50% of populated artifacts will be fully or partially discovered through manual investigation.
- H_2 More than 50% of populated artifacts will be fully or partially discovered by the forensic tools.

In total 297 artifacts were populated, 99 from each device. Not all 297 artifacts are on each device. Direct messages and direct calls are only potentially recoverable on the devices involved. For example a direct message from the iOS user to the Android user would not be present on the Windows device. In addition the calendar events were only on the user's calendar who made the event. When these artifacts are accounted for, each device can have a maximum 240 populated artifacts.

Table 4.8 shows the number and percentage of artifacts discovered per device by both manual and forensic tool investigation. Overall there is evidence to support H_1 as a total of 77.6% of artifacts were partially or fully recoverable in the manual investigation. Only 13.8% of artifacts

were partially or fully recovered by the forensic tools across all three devices, meaning H_2 is not supported.

Table 4.8. *Total Artifact Recovery*

		Fully		Partially		Not	
		n	%	*n*	%	*n*	%
Windows	Manual	3	1.3	96	40.0	141	58.8
	AXIOM	24	10.0	13	5.4	203	84.6
iOS	Manual	191	79.6	45	18.8	4	1.7
	Cellebrite	0	0.0	47	19.6	193	80.4
Android	Manual	136	56.7	88	36.7	16	6.7
	Cellebrite	0	0.0	15	6.3	225	93.8
Total	Manual	330	45.8	229	31.8	161	22.4
	Tool	24	3.3	75	10.4	621	86.3

CHAPTER 5. DISCUSSION

This study is intended to aid future investigations of Microsoft Teams by documenting what artifacts can be recovered, and where those artifacts can be found. In addition, the results of the widely used forensics tools Cellebrite and AXIOM are analyzed to better understand how these tools handle data from the Teams clients. Specially, Windows 10, iOS 13.7, and Android 10 devices were used in this study.

5.1 Artifacts Trends and Observations

For the Windows 10 laptop, the majority of artifacts are found in either the Chromium cache structure, or the levelDB structure, both found within the user's Teams AppData folder. It is unknown if the Teams desktop client is just a version of the Teams web app in an Electron wrapper, or if the client is substantially different. If the client is simply an Electron version of the web app, then these results could be similar to those of Teams on a web browser. While methods to parse Chromium style caches are known (Brignoni, 2018b), a method to parse the levelDB structure has not been found. This is unfortunate as review of the files that make up this structure show indications that nearly all kinds of message data can be found, as well as call logs, meeting details, and file metadata. Because this structure is a key value storage system (Caithness, 2020), it does not seem likely that additional files are stored here, but just records.

AXIOM is able to parse the information in the levelDB structure and displays some of the messages. The data displayed is limited and seems to only be from one of the files that makes up the levelDB structure. The file AXIOM uses is the .log file, which for this data structure contains the most recent changes to the database (Caithness, 2020). Older records get saved to more permanent .ldb files in the same directory (Caithness, 2020), and are not included in AXIOM's results. This is likely why only unread messages are found. For this research, unread messages were populated last out of practicality, ensuring messages were not accidentally read during other population steps. The fact that only unread messages are recovered by AXIOM seems to have more to do with the fact they are the most recent messages, than anything special about unread messages.

AXIOM is able to parse some of the levelDB database, and manual investigation is not able to parse any. This means the Windows group unread messages and Windows direct unread message are the only artifact groups where the forensic tool performs better than manual investigation. The only artifacts not at all recoverable for the Windows device are the deleted messages and call logs. There are some patterns in the levelDB files that appear similar to the call logs present in both the iOS and Android devices, but these are not confirmed as calendar events. No deleted messages on the Windows device are found, but it could be that if the levelDB is properly parsed, metadata about the deleted messages could be found, with the content of those messages likely not present.

The iOS device is the only device where all artifact types are at least partially recoverable. Only four artifacts were not at all recoverable from the iOS device: one message and three video file messages. For file messages, only video messages are not fully recovered, as the files could not be found, but context about sent videos could. Video files are not found on any other device, so it seems likely that video files are not cached by Teams.

Cellebrite finds call logs for the iOS device because records of the calls are stored with the native iPhone call log. Despite not identifying these calls as being from Teams, details about when the call took place, the duration, and who the call was with are present. Investigators should consider not only sources within the Teams-related directories, but native sources that are well documented and easy for forensic tools to analyze.

The Android phone is the only device where a method for determining which user is logged into Teams has been discovered. This is because the user's objectId is repeated in all rows of the tenantId column of many tables, including the user table. This behavior is not seen in the comparable iOS table. The worst device for forensic tool reversibility is the android device. Only image file messages and file share messages are in the file cache. Similarly profile photos are missing important context as they can be found, but not attributed to a specific user.

Deleted messages results vary across all three devices. The iOS device keeps the messages in full with an indication that the messages were deleted. The Android device keeps records of a message, but the actual content is removed. This suggests that on Android the messages are actually deleted, and not on iOS. For Windows, no records of deleted messages are found though they may still be present in the levelDB database.

Edited messages results are mostly consistent across the three devices. The new content of the message is present, and the old content could not be found. For iOS and Android, there is an indication the message is edited, in the form of a timestamp for the edit. This type of timestamp is not found on the Windows machine, though it may be part of the same record as the message, but this cannot be shown at this time.

In terms of forensic recovery, there does not seem to be a distinction between read and unread messages. Both types of messages are equally recoverable by manual investigation in all cases, on all clients. This suggests that once a message is received by the client, it is cached even if the user does not open it. In this case, AXIOM recovers only unread messages, but as previously mentioned, this is likely because unread messages were populated last, and not because the message is unread.

Similarly there is no distinction found between team channel, group, or direct messages or file messages in terms of what is manually recoverable. These messages are stored in the same location in all cases and have similar attributes. For the file messages, the files and file metadata are also stored in the same locations. In this case, AXIOM recovers some group and direct messages, but no team channel messages. This is again believed to be more about the order of message population, as unread group and unread direct messages were populated after unread channel messages.

In all cases, across all devices, only metadata is found for video files. Videos files are likely not recoverable due to their larger size. Unfortunately, the largest PDF included is smaller than the smallest video, so it is not possible to know if size is the only factor. The largest PDF is 2.17MB and the smallest video is 2.9MB. It could be that the cutoff for files to be cached is somewhere between those two numbers, like 2.5MB. It seems likely given the relatively similar sizes that the type of file may play a roll. It could be that video formats like .mp4 and .mpg are not cached to save on disk space. At this times, it is not possible to know for certain which explanation accounts for the lack of cached video files.

Unlike videos, image files are found one all devices from both manual and tool investigations. The forensic tools do not identify these images as relating to the Teams app, but the images do appear as media artifacts, so they are considered recovered. This is true for images regardless of what forum they were sent.

5.2 Recommendations for Investigators

Microsoft Teams is a young, but quickly growing platform. In the first three years after its 2017 launch, the platform grew to an active daily user base of 12 million users (Zaveri, 2020). In the following 13 months Teams swelled to 145 million users (Wright, 2021). This growth can likely be attributed to multiple factors such as the impending retirement of Teams predecessor Skype for Business in July 2021 (Chin, 2020), and the COVID-19 pandemic requiring many to work from home and utilize Teams to communicate with their teams (Zaveri, 2020). With such wide use, it seems inevitable that investigations involving the platform will be necessary. Despite this, there is currently no known research on the computer or mobile forensics of Microsoft Teams. This study helps to fill this gap in the present research.

Humphries et al. (2021) identify that fundamental knowledge regarding forensic tools is needed by all investigators. In order to facilitate this "the need for setting up experiments to define and test hypotheses" is present (Humphries et al., 2021, p. 10). This study is one such experiment focused on Microsoft Teams, documenting the locations of artifacts, how they should be interpreted, as well as validating commonly used forensic tools.

At this point in time if an investigator encounters Teams in a case, manual investigation should be primarily used. While the forensic tools can be helpful in identifying some artifacts of Microsoft Teams, those used in this study are not at the point where their results can be consistently relied on. The only time in this study a forensic tool specifically presents artifacts labeled as Teams-related is the messages discovered by AXIOM on the Windows platform. But with this result, only a selection of recent messages are included, even though other messages are available from the same source. These tools do provide other valuable insights such as Cellebrite parsing the iOS native call log, but for any investigations centered on Teams, what is provided is not comprehensive, making manual discovery required.

In this study there are many artifact groups that are not discoverable by manual investigation, either because the artifacts are not present, or methods for discovery have not been found. To retrieve artifacts that are not found on a device, investigators must go through Microsoft. Teams is a cloud only platform, and therefore companies do not have any on premise servers hosting the service. This means unlike some other Microsoft products where a company

can give their own or external investigators the ability to analyze the server data, investigators must request the data from Microsoft. This request must come from a court as Microsoft requires "a warrant or court order for content, or a subpoena for subscriber information or other noncontent data" (Microsoft, 2021, p. 1). Cloud data likely goes back further in time than what is found on a device, as most of the artifacts found were in cache structures, which are temporary in nature. It is recommended that investigators retrieve artifacts from the cloud if possible, as it will compliment, if not encompass, the findings from a device.

5.3 Impact to Future Research

While one of the main objectives of this study is to document Teams and aid investigators, future forensic research can also benefit from this work. While the approach to analyzing every application is different, the methodology described for population and analysis could be adapted to other applications, especially communication platforms. Using the same or similar methodologies across applications enables the possibility for a direct comparison of those applications.

It is not unreasonable to believe there will be commonalities in software designed by the same people or same cooperation. The specific results of this research could serve as a starting point for investigating future Microsoft applications. Even if Teams were to be retired, remnants of its structure could exist in subsequent or related Microsoft projects. More likely, as Teams is updated, parts of this research will become outdated, but because the current structure is known, these changes can be discovered more easily.

While application security is not a concern of this study, security and privacy researchers may be able to build from this work. This study shows that Teams stores cached data at rest in plain text non encrypted. While this is not necessarily uncommon for applications, knowing this when developing new research can be useful. As Teams is updated, how data is handled could change which could impact the platform’s data privacy for better or worse. As much of the current state of Teams data handling is now known, significant changes to this structure will be easier to spot.

5.4 Limitations

In an effort to isolate this study, an independent "organization" is used to host the team. While this helps keep the data from being inadvertently altered, this is not a common use case. As a result, the calendar function does not behave as it does for users with calendars connected to the organization. There may be other differences between the kind of organization that was used in this study, and a real cooperate organization, or academic institution for example. While the clients likely behave in a similar way, this difference could impact how data is handled and stored, which would impact how an investigator or tool should handle a case.

Another limitation is that other types of forensics such as cloud and memory forensics were not considered. While live memory acquisitions would have been possible, this was considered out of scope for this study. Acquisition of cloud data is not possible without a court document (Microsoft, 2021) or consent by those involved to have the data released. In this case, even with consent cloud data cannot be recovered, as this option is not made available to free Teams instances. Just as with memory forensics, a comparison of cloud data could yield many interesting results, but this is considered out of scope for this study.

Only the results of the AXIOM and Cellebrite's primary processing are considered in this study. This includes the analyzed data, artifacts, timeline, and other artifact display methods provided by the tools. Any secondary processing, such as Cellebrite's App Genie or AXIOM's Dynamic App Finder were not used. Only artifacts that were explicitly populated were discussed. Some teams-related artifacts were not included, such as application usage logs or other sources that did not provide information related to a populated artifact type.

5.5 Future Work

One way to expand on this research would be to use different devices, operating systems, or forensic tools. Windows 10, iOS 13.7, and Android 10 were used in this study, but there are many others that could be of interest. Discord researchers Motyliński et al. (2020) report that while Discord uses the same caching system on Windows and Mac, on Linux a separate system is used. This could be the case for Teams as well, and Teams is available for both Mac and Linux,

making them excellent candidates for future Teams research. Understanding how forensic tools, other than AXIOM and Cellebrite, handle Teams clients would also be of value to investigators.

Team is not only available as a client, but also available through a web interface. Understanding how Teams creates browser artifacts could be useful to investigators. Many OS and browser combinations could be considered, such as Mozilla Firefox, Google Chrome, Microsoft Edge, with the operating systems used in this study as well as Mac and Linux, or others.

At the time of writing, Microsoft's next operating system Windows 11 has recently been announced, and along with it news that Teams will be directly integrated into the new OS (Welch, 2021). According to announcements, not only is the software going to be preinstalled, but a Teams chat feature will be embedded as a native operating system function (Novet, 2021). Developers have confirmed that with this change, the architecture of "Teams 2.0" Windows client will be largely changed and no longer use Electron (Arbuthnot, 2021, p. 3). This admittedly may make the Windows results of this study quickly outdated. Hopefully these results can be used as a starting point to explore the new Teams client on the new Windows 11 operating system.

The new Teams client for Windows 11 may or may not use the same or similar levelDB structure seen in Windows 10. Understanding how to effectively parse this structure could be very useful to investigators. Even if the new Teams client does not use the same storage systems, the ability to parse this kind of structure could be useful when investigating other applications that use it.

5.6 Conclusion

This study provides useful artifact locations and information for the Microsoft Teams client on Windows desktop, iOS, and Android. For the mobile devices, the populated artifacts can largely be found in SQLite databases, or in file caches. For Windows, the file artifacts can be found in a Chromium cache structure, and the others appear to be in a levelDB key value pair structure. Artifacts are largely manually recoverable from the mobile phones, but due to the inability to parse the levelDB structure, many Windows artifacts that are likely recoverable, are not able to be accounted for in this study.

Cellebrite and AXIOM are largely unable to recover data from the Teams client. Cellebrite's largest success is identifying files, but these files are not displayed as associated with Team to the user. On the iOS device, Cellebrite recovers the call logs not by analyzing the Teams data structure, but the native iPhone call log. AXIOM similarly is able to recover files from Windows, with little indication that they are teams- related. AXIOM is able to recover full message details for some messages, but only the most recent ones. This appears to be because not all files that make up the levelDB database are considered. These results are not at all indications that these are poor tools, just that they do not currently handle the artifacts from the Microsoft Teams client as well as is possible.

REFERENCES

4:cast. (2020, Jul). *4:cast 2020 awards.* Retrieved from https://forensic4cast.com/forensic-4cast-awards/2020-forensic-4cast-awards/

Ail, V. (2020, Jul). *Appdata – where to find the appdata folder in windows 10.* Retrieved from https://www.freecodecamp.org/news/appdata-where-to-find-the-appdata-folder-in-windows-10/

Al-Saleh, M. I., & Forihat, Y. A. (2013). Skype forensics in android devices. *International Journal of Computer Applications*, *78*(7).

Android Developers. (2021, Jul). *Sdk platform tools release notes.* Retrieved from https://developer.android.com/studio/releases/platform-tools

Arbuthnot, T. (2021, Jun). *Microsoft teams 2.0 will use half the memory, dropping electron for edge webview2.* Retrieved from https://tomtalks.blog/2021/06/microsoft-teams-2-0-will-use-half-the-memory-dropping-electron-for-edge-webview2/

Azfar, A., Choo, K.-K. R., & Liu, L. (2017). Forensic taxonomy of android productivity apps. *Multimedia Tools and Applications*, *76*(3), 3313–3341.

Brignoni, A. (2018a, Dec). *Identifying installed and uninstalled apps in ios.* Blogspot. Retrieved from https://abrignoni.blogspot.com/2018/12/identifying-installed-and-uninstalled.html

Brignoni, A. (2018b, Mar). *Identifying installed and uninstalled apps in ios.* Blogspot. Retrieved from https://abrignoni.blogspot.com/2018/03/finding-discord-app-chats-in-windows.html

Caithness, A. (2020, Sep). *Hang on! that's not sqlite! chrome, electron and leveldb.* Retrieved from https://www.cclsolutionsgroup.com/post/hang-on-thats-not-sqlite-chrome-electron-and-leveldb

Casey, E. (2009). *Handbook of digital forensics and investigation*. Academic Press.

Cellebrite. (2020). *Cellebrite company profile.* Retrieved from https://www.cellebrite.com/en/about/company/

Chin, L. (2020, Nov). *Skype for business online retirement - microsoft teams.* Retrieved from https://docs.microsoft.com/en-us/microsoftteams/skype-for-business-online-retirement

Curry, D. (2020, Oct). *Slack revenue and usage statistics (2020).* Retrieved from https://www.businessofapps.com/data/slack-statistics/

Db browser for sqlite - about. (2021). Retrieved from https://sqlitebrowser.org/about/

Dogan, S., & Akbal, E. (2017). Analysis of mobile phones in digital forensics. In *2017 40th international convention on information and communication technology, electronics and microelectronics (mipro)* (pp. 1241–1244).

Electron. (2016). *Discord: Apps.* Retrieved from https://www.electronjs.org/apps/discord

Epifani, M. (2021, March). *ios third-party apps forensics.* SANS.

Free Software Foundation. (2020, Sep). *Gnu grep.* Retrieved from https://www.gnu.org/software/grep/manual/

Freeman, J. (n.d.). *Cydia.* Retrieved from https://cydia.saurik.com/

Gerend, J., Downie, K., Ross, E., Parente, J., Coulter, D., Jacobs, M., ... Plett, C. (2017, Oct). *clean.* Microsoft. Retrieved from https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/clean

Gerend, J., Ross, E., Coulter, D., Schonning, N., Prittie, I., & Moore, G. (2020, Dec). *diskpart.* Microsoft. Retrieved from https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/diskpart

Goel, M., & Kumar, V. (2019). Layered framework for mobile forensics analysis. In *Proceedings of 2nd international conference on advanced computing and software engineering (icacse)* (pp. 557–561).

Ho, D. (2021). *What is notepad++.* Retrieved from https://notepad-plus-plus.org/

Humphries, G., Nordvik, R., Manifavas, H., Cobley, P., & Sorell, M. (2021). Law enforcement educational challenges for mobile forensics. *Digital Investigation*, 2–12.

Hörz, M. (2020). *Hxd - freeware hex editor and disk editor.* mh-nexus. Retrieved from https://mh-nexus.de/en/hxd/

Jones, G. M., & Winster, S. G. (2017). Forensics analysis on smart phones using mobile forensics tools. *International Journal of Computational Intelligence Research*, *13*(8), 1859–1869.

Kieller, K. (2019, Oct). *Skype for business – death exaggerated, but on life support.* Retrieved from https://www.nojitter.com/team-collaboration-tools-workspaces/skype-business-death-exaggerated-life-support

Koenigsbauer, K. (2017, Mar). *Microsoft teams rolls out to office 365 customers worldwide.* Retrieved from https://www.microsoft.com/en-us/microsoft-365/blog/2017/03/14/microsoft-teams-rolls-out-to-office-365-customers-worldwide/

Leahy Center for Digital Investigation. (2017, May). *Application analysis.* Retrieved from https://www.champlain.edu/Documents/LCDI/ApplicationAnalysis_S17.pdf

Magisk Manager. (2021). *Download magisk manager.* Retrieved from https://magiskmanager.com/

Magnet. (2020, Jul). *Magnet forensics wins leading industry awards for second year in a row.* Retrieved from https://www.magnetforensics.com/news/magnet-forensics-wins-leading-industry-awards-for-second-year-in-a-row/

Maguire, D., Martinez, J., Payne, H., & Borys, A. (2020, Nov). *How microsoft teams uses memory - microsoft teams.* Microsoft. Retrieved from https://docs.microsoft.com/en-us/microsoftteams/teams-memory-usage-perf

Microsoft. (2021). *About our practices and your data.* Retrieved from https://blogs.microsoft.com/datalaw/our-practices/

Moffitt, K., Karabiyik, U., Hutchinson, S., & Yoon, Y. H. (2021). Discord forensics: The logs keep growing. In *2021 ieee 11th annual computing and communication workshop and conference (ccwc)* (pp. 0993–0999).

Morris, N., & Moses, A. (2018). Investigating google chrome 66.0. 3359 artefact: Internet forensics approach. *International Journal of Computer Science and Mobile Computing*, *7*, 112–122.

Motyliński, M., MacDermott, Á., Iqbal, F., Hussain, M., & Aleem, S. (2020). Digital forensic acquisition and analysis of discord applications. In *2020 international conference on communications, computing, cybersecurity, and informatics (ccci)* (pp. 1–7).

Nelson, R., Shukla, A., & Smith, C. (2020). Web browser forensics in google chrome, mozilla firefox, and the tor browser bundle. In *Digital forensic education* (pp. 219–241). Springer.

Nicoletti, M., & Bernaschi, M. (2019). Forensic analysis of microsoft skype for business. *Digital Investigation*, *29*, 159–179.

NirSoft. (2021). *Cache viewer for google chrome web browser.* Retrieved from https://www.nirsoft.net/utils/chrome_cache_view.html

NIST. (2016, Mar). *Mobile device data populations setup guide, version 2.0.*

Novet, J. (2021, Jul). *Microsoft rolls out windows 11 update with teams chat built in.* CNBC. Retrieved from https://www.cnbc.com/2021/07/20/microsoft-starts-rolling-out-windows-11-update-with-teams-chat.html

Onovakpuri, P. E. (2018). Forensics analysis of skype viber and whatsapp messenger on android platform. *IJCSDF*.

Panhuyzen, A. (2021). *checkra1n.* Retrieved from https://checkra.in/

Pochron, J. (2018, Mar). *Need to collect data from slack? read this first.* Retrieved from https://www.transperfect.com/blog/need-to-collect-data-from-slack--read-this-first

Pochron, J. (2019, May). Cutting us some slack. *Digital Forensics*(39), 42–47. Retrieved from https://onna.com/wp-content/uploads/2019/05/Digital-Forensics-Mag-Onna.pdf

Pointlogic. (2018, Jan). *Accessing the application data folder.* Nielsen. Retrieved from https://support.pointlogic.com/faq/troubleshooting/accessing-the-appdata-folder

Police1. (2018, Mar). *Analyze and share digital evidence faster with a tool developed by and for police.* Retrieved from https://www.police1.com/police-products/investigation/computer-digital-forensics/articles/analyze-and-share-digital-evidence-faster-with-a-tool-developed-by-and-for-police-3Z5uUyAYZ1MQwXWJ/

Rathod, D. M. (2017). Web browser forensics: google chrome. *International Journal of Advanced Research in Computer Science*, *8*(7).

Sathe, S. C., & Dongre, N. M. (2018). Data acquisition techniques in mobile forensics. In *2018 2nd international conference on inventive systems and control (icisc)* (pp. 280–286).

Sgaras, C., Kechadi, M., Le-Khac, N.-A., et al. (2016). Forensics acquisition and analysis of instant messaging and voip applications. *arXiv preprint arXiv:1612.00204*.

Shafqat, N. (2016). Forensic investigation of user's web activity on google chrome using various forensic tools. *IJCSNS Int. J. Comput. Sci. Netw. Secur*, *16*(9), 123–132.

Shankdhar, P. (2019, Feb). *Popular computer forensics top 21 tools.* Retrieved from https://resources.infosecinstitute.com/computer-forensics-tools/

Shin, S., Park, E., Kim, S., & Kim, J. (2020). Artifacts analysis of slack and discord messenger in digital forensic. *(J. DCS)*, *21*(4), 799–809.

Slack. (2020a). *Guide to slack import and export tools.* Retrieved from https://slack.com/help/articles/204897248-Guide-to-Slack-import-and-export-tools

Slack. (2020b). *A guide to slack's discovery apis.* Retrieved from https://slack.com/help/articles/360002079527-A-guide-to-Slacks-Discovery-APIs

Slack. (2020c). *Make calls in slack.* Retrieved from https://slack.com/help/articles/216771908-Make-calls-in-Slack

StatCounter. (2020a, Sep). *Desktop operating system market share worldwide.* Retrieved from https://gs.statcounter.com/os-market-share/desktop/worldwide

StatCounter. (2020b, Sep). *Mobile operating system market share worldwide.* Retrieved from https://gs.statcounter.com/os-market-share/mobile/worldwide

Suma, G., Dija, S., & Pillai, A. T. (2017). Forensic analysis of google chrome cache files. In *2017 ieee international conference on computational intelligence and computing research (iccic)* (pp. 1–5).

TeamWin. (2021). *Twrp - about.* Retrieved from https://twrp.me/about/

Vukadinovic, N. V. (2019). *Whatsapp forensics: Locating artifacts in web and desktop clients* (Unpublished doctoral dissertation). Purdue University Graduate School.

Welch, C. (2021, Jun). *Microsoft teams will be directly integrated as part of windows 11.* The Verge. Retrieved from https://www.theverge.com/2021/6/24/22548738/microsoft-teams-windows-11-integration-announcement-features

Wright, T. (2021, Apr). *Teams hits 145 million daily active users.* Retrieved from https://www.uctoday.com/collaboration/teams-hits-145-million-daily-active-users/

Yadav, S., Prakash, S., Dayal, N., & Singh, V. (2020). Forensics analysis of whatsapp in android mobile phone. *Available at SSRN 3576379*.

Yang, T. Y., Dehghantanha, A., Choo, K.-K. R., & Muda, Z. (2016). Windows instant messaging app forensics: Facebook and skype as case studies. *PloS one*, *11*(3), e0150300.

Zaveri, P. (2020, Apr). *Microsoft teams now has 75 million daily active users, adding 31 million in just over a month.* Retrieved from https://www.businessinsider.com/microsoft-teams-hits-75-million-daily-active-users-2020-4

APPENDIX A. LIBRARY OF CONGRESS FILES

Table A.1. *File Locations and Names*

#	OS	Type	Forum	Filename
1	iOS	Image	Channel	iOS_channel_img.jpg
2	iOS	Image	Group	iOS_group_img.jpg
3	iOS	Image	DM	iOS_dm_img.jpg
4	iOS	Video	Channel	iOS_channel_vid.mp4
5	iOS	Video	Group	iOS_group_vid.mp4
6	iOS	Video	DM	iOS_dm_vid.mp4
7	iOS	PDF	Channel	iOS_channel_pdf.pdf
8	iOS	PDF	Group	iOS_group_pdf.pdf
9	iOS	PDF	DM	iOS_dm_pdf.pdf
10	Android	Image	Channel	Android_channel_img.jpg
11	Android	Image	Group	Android_group_img.jpg
12	Android	Image	DM	Android_dm_img.jpg
13	Android	Video	Channel	Android_channel_vid.mp4
14	Android	Video	Group	Android_group_vid.mp4
15	Android	Video	DM	Android_dm_vid.mp4
16	Android	PDF	Channel	Android_channel_pdf.pdf
17	Android	PDF	Group	Android_group_pdf.pdf
18	Android	PDF	DM	Android_dm_pdf.pdf
19	Windows	Image	Channel	Windows_channel_img.jpg
20	Windows	Image	Group	Windows_group_img.jpg
21	Windows	Image	DM	Windows_dm_img.jpg
22	Windows	Video	Channel	Windows_channel_vid.mpg
23	Windows	Video	Group	Windows_group_vid.mpg
24	Windows	Video	DM	Windows_dm_vid.mp4
25	Windows	PDF	Channel	Windows_channel_pdf.pdf
26	Windows	PDF	Group	Windows_group_pdf.pdf
27	Windows	PDF	DM	Windows_dm_pdf.pdf
28	iOS	PDF	Fileshare	fs_iOS.pdf
29	Android	PDF	Fileshare	fs_Android.pdf
30	Windows	PDF	Fileshare	fs_Windows.pdf

Table A.2. *File Hashes*

#	MD5	SHA1
1	c8ad2975bd399bc874d55d11b240ab5a	14583daba0c2874d9ec8be1ad045f8226a3e5d7c
2	f6c7eecf6f69e39a7d61efd6a983aa75	af2dda329c179dfb472b34d025e9e7d4b098b6e1
3	45709c32624093206ec446d15232e29b	750b55ca0d222ef4b49712f4618c052403789038
4	032899397322a56e9b077a8429f693e0	0046b9834df8320aae52285ce2a35668b72f3915
5	af722617ff086a639c3ab96d608c8bfd	65b706ef85af84c2c2378fe1aa8752e0889974ce
6	5234e7fab1dc796374b4aa604d7122fb	417d757411197f9a22979c7d770c23b0bac779a1
7	2d005e3074c32f2daef8c280539e9242	7cdfcf83b30d5d9c8ff6cc98a1c102892ff5211e
8	9b8207972819a374b244aff5b880319e	dd5d61f7f70ce9ef53739e06f1ddddaee896f53b
9	05f4b3912f41aeadc1d8440014f32af6	4ee37e785b852ee679e1b7c4531912d80402ea0c
10	865982cc40faec997611b8126f977f46	0c0f7d139983d0a7838b4887b8290855c1a15278
11	fa5344eb1e4bf122761e6a4270f0aec6	3fd68dbcfc18e0a866a6433dc3cb0f6bee6b37ba
12	3d8367d87b9b44fa56bc0b39b3a44beb	064938f44bb3d547a7f31999c8f8ee8066e6e0c7
13	96914044ebaa1327669ec2d0dd5631a7	8f0b41e90777b9a28bb2e96ea1d7ef31fa434bce
14	0ada77462bf1018f16fefdc764cf82bc	0dfa6bb91758e606223b108c27a5052b75bf9b02
15	983a4ab9ea6c92da57371a665ea6b4c4	3a35f9a2242d001ec46e4c7e325b3206f743a873
16	ee4b0c1f637af0e4596fcdf58731abc7	8c777dfab16b3828e2eb85fd0738cb1d338508e2
17	d298f33c6525a02050975adc01fcd672	05c9813919143f85623a916ae55b0d177cbf98b0
18	90de2d2dd4d9199bf4b9c4e7a51e6f57	25322bf0bede5a7faf86b97a54ca4b8aa410ead2
19	e9109c45a3bfe1a1adc3a2f39ceaf060	c760811ee32a9eef32fdf22de8aeffd40793250e
20	3f031c3f7f476fd18ab09c17660092cd	24e6c96c26f7ad9bc940c66646e450bbd0fcb62c
21	a6b6d94cc19695d6267c8421f578c5f8	f1adafac4d83158b11425ba41147103ed8a2c65e
22	3231b1078118c507fa80699a7f1a6b57	d917f6bcb75eb435e2b6ea8dc13a7b3d7608b38c
23	d3d7ccb660d5ed6c878180ee54d4563d	0e38a8f5ec8cdae77e567b2ea5c311943ce8523d
24	5ad1e018f5a7461b34352d3dc8d8a77c	f24f276f89cab29a0fe9e80e046a26437ae376da
25	2e491281c183e9d45cb0a6369d42384b	3af739f95f74ab90b64f35345577afc4b02a4ce6
26	4399abc48fc733756bb6b42a23e1253f	503ad315a68efc8f19e156f8342726866bd0bca5
27	f5720129b38df56d8dd6468befb0f05a	1db080be5fbde2aa9f124aca6544d3dfe8ac8626
28	fa6f5a11bf08b404622823a9f448f202	f3cd89de92c56610d3079a36aa65e04ec5cdd483
29	36f9a0bcd3523a1a558f76d9e4a29fce	e7986fd834978fe20ffd4dd173324050684a2a60
30	5cc524ce41913e6ba82d9fe5253a0cf7	282cda3daf4b478a0d211780eb48ede45bc7b42c

Table A.3. *File Online Locations*

#	LOC Item Link
1	https://www.loc.gov/item/2006683836/
2	https://www.loc.gov/item/92520321/
3	https://www.loc.gov/item/2017677132/
4	https://www.loc.gov/item/webcast-9665/
5	https://www.loc.gov/item/webcast-6002/
6	https://www.loc.gov/item/webcast-6000/
7	https://www.loc.gov/resource/sn84026897/1898-10-05/ed-1/?sp=1
8	https://www.loc.gov/resource/sn84026897/1898-10-05/ed-1/?sp=2
9	https://www.loc.gov/resource/sn84026897/1898-10-05/ed-1/?sp=3
10	https://www.loc.gov/item/2001705522/
11	https://www.loc.gov/item/2001705523/
12	https://www.loc.gov/item/2001705524/
13	https://www.loc.gov/item/webcast-6735/
14	https://www.loc.gov/item/webcast-6738/
15	https://www.loc.gov/item/webcast-6736/
16	https://www.loc.gov/item/magbell.03710304/
17	https://www.loc.gov/item/rbpe.12800300/
18	https://www.loc.gov/item/rbpe.15905300/
19	https://www.loc.gov/item/2002698808/
20	https://www.loc.gov/item/94507935/
21	https://www.loc.gov/item/2014650133/
22	https://www.loc.gov/item/96521838/
23	https://www.loc.gov/item/96521901/
24	https://www.loc.gov/item/96515575/
25	https://www.loc.gov/item/wpalh002764/
26	https://www.loc.gov/item/ihas.100002958/
27	https://www.loc.gov/item/ihas.100002684/
28	https://www.loc.gov/item/sn85029856/1916-09-01/ed-1/
29	https://www.loc.gov/item/rbpe.07204800/
30	https://www.loc.gov/item/cosmos000094/

APPENDIX B. APPROVAL OF RESEARCH

PURDUE
UNIVERSITY

This Memo is Generated From the Purdue University Human Research Protection Program System, Cayuse IRB.

Date: January 6, 2021
PI: KATHRYN SEIGFRIED-SPELLAR
Re: Initial - IRB-2020-1854
Forensic Analysis of Microsoft Teams (Master's Thesis)

Through the answers you provided in response to questions in the Cayuse IRB system, Purdue's HRPP has determined that the research does not qualify as Human Subjects Research under federal human subjects research regulations (e.g., 45 CFR 46).

Decision: No Human Subjects Research
Findings:
Research Notes:

The answers provided in your Cayuse IRB application indicate:

- You will not collect data from human subjects for the purpose of research intended to create generalizable knowledge. Reasons that are not considered research include purposes such as internal programmatic evaluation, quality improvement, or business analysis.
- You will not involve human subjects by collecting data from a living individual through intervention of interaction with the individual and/or identifiable private information.

What are your responsibilities now, as you move forward?

- If you have further questions about this determination, you must contact the Purdue HRPP/IRB.
- You and the members of your research team acknowledge that this study is subject to review at any time by Purdue's HRPP staff, Institutional Review Board, and/or Research Quality Assurance unit. At any time, this project may be subject to monitoring by these Purdue entities to confirm the applicability of this determination. The Purdue IRB has final authority in determining if an activity is Human Subjects Research requiring IRB review.
- This determination is the Purdue HRPP assessment of regulations related only to human subjects research protections. This determination does not constitute approval from any other Purdue campus department or outside agency. The Principal Investigator and all researchers are required to affirm that the research meets all applicable local/state/federal laws and university policies that may apply.
- Finally, if any changes occur with respect to this project, recognize that such changes could change the need for review by HRPP/IRB. Should you change the intent of the activity to involve publication, presentation, or any different application of this work, it is likely that IRB review will be required. Therefore, it is important that you again complete Cayuse IRB to ensure that the IRB review requirements remain the same.

If you need assistance with the submission revisions, please contact irb@purdue.edu for assistance or an appointment. We are here to help!

Sincerely,

Purdue University Human Research Protection Program/ Institutional Review Board