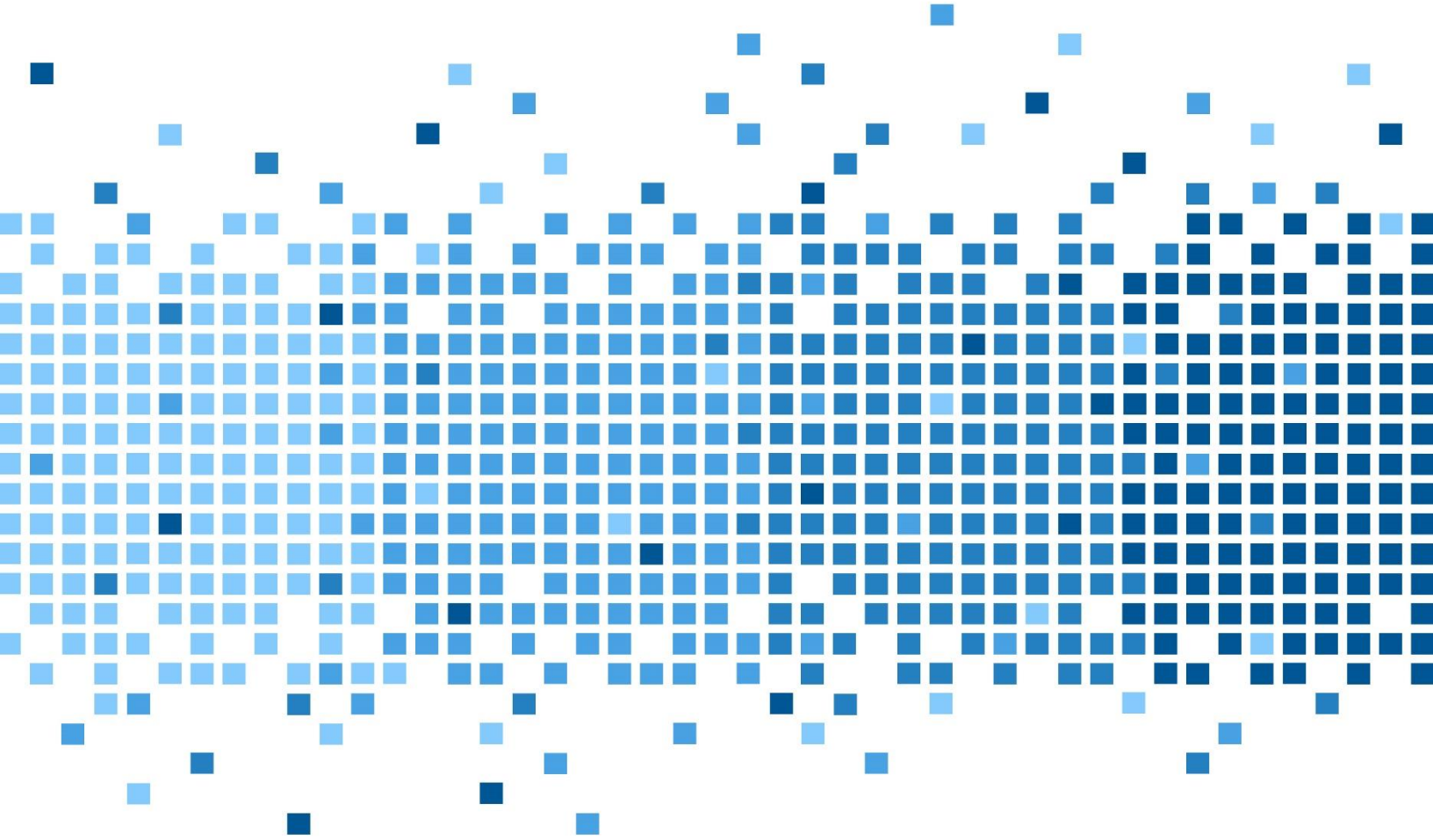
CHAMPLAIN COLLEGE | LCDI Leahy Center for Digital Investigation

Application Analysis



05/03/2017
http://lcdiblog.champlain.edu/

175 Lakeside Ave, Room 300A
Phone: (802) 865-5744
Fax: (802) 865-6446

Disclaimer:

This document contains information based on research that has been gathered by employee(s) of The Senator Patrick Leahy Center for Digital Investigation (LCDI). The data contained in this project is submitted voluntarily and is unaudited. Every effort has been made by LCDI to assure the accuracy and reliability of the data contained in this report. However, LCDI nor any of our employees make no representation, warranty or guarantee in connection with this report and hereby expressly disclaims any liability or responsibility for loss or damage resulting from use of this data. Information in this report can be downloaded and redistributed by any person or persons. Any redistribution must maintain the LCDI logo and any references from this report must be properly annotated.

Contents

Introduction 2

Background: 2

Purpose and Scope: 2

Research Questions: 2

Terminology: 3

Methodology and Methods 4

Equipment Used 5

Analysis 5

Results 5

Slack 5

Dropbox - Windows 7 7

Discord 9

Discord for Mac 10

Conclusion 11

Further Work 11

Appendix 12

Appendix A - Data Gen 12

References 16

Introduction

Web Applications are applications that run off internet resources, where all needed resources are downloaded at start time. With the rise of internet technology integration within business and personal life, these apps have become a prominent factor in everyday life. These apps are being relied on to transmit personal information with data such as documents, photos, and general messages. In concept, downloading everything at run time should not leave as much data behind. However, in practice, downloading everything at runtime is memory and network intensive, especially with media files stored in cache. With increased popularity using cache memory more data is being generated as time progresses. Based on this, programs like Twitter, Discord, Slack, and Dropbox will be analyzed.

Background:

While a similar project was conducted by the LCDI in the autumn of 2013 under the name "Cloud Forensics," it focused more broadly on Web Applications and investigated them from a storage perspective, along with their relation to cloud services. The Application Analysis Project looks at the client side of things, investigating what identifiable artifacts are left behind and if they can give details on activities and profiles of the person using the service. We based our methodology off of the techniques from Windows Instant Messaging App Forensics. We applied the simulation of logins and messages using the app, accompanied with file uploads. These actions are meant to simulate the everyday actions of consumers that use the app. For this purpose we narrowed it down to two big kinds of Web Apps: Business and Social. The Business category is focused on apps that utilize web apps pertaining to communication (Slack) and File Sharing (Dropbox). Social based apps are more in communication (Discord) and sharing of personal thoughts and ideas (Twitter). These two categories were chosen as they are widespread and most people have either come face to face with them or use them daily. With each of these apps having different standards and policies when it comes to security, they will be compared separately and together to see how the individual web apps stack up.Variations between business and social apps will also be considered.

Purpose and Scope:

The purpose of our research is to see if we can find variations of artifacts left behind by different desktop applications. No matter the data or artifact, our group will investigate with the utmost consideration. Even if an app has been deleted, there is always a chance important information has been left behind. The research will provide a glimpse into the inner functions of web apps and their security.

Research Questions:

1. What data is recoverable in each application from both Mac and Windows Operating Systems?
2. How have these apps evolved over time?
3. What data is recovered after deleting the application?

Terminology:

010 - a program, called a hex editor, which displays every character in a file in hexadecimal notation. Also called a "binary editor," hex editors are used extensively in program development to view the actual content of a file, as well as to view files in old formats that are not recognized by today's applications.

Artifacts - any data generated by user interaction that can be collected and examined. Any user data retrieved from the browser is considered an artifact, including cookies, caches, geolocation, search history, etc.

Cache - a component that stores data, to allow future requests for the data to be processed faster. Data stored in a cache could be the results of a computation, or data that has been duplicated and is stored elsewhere.

Digital Evidence - is “information of probative value that is stored or transmitted in a binary form” (NCFS, 2012). Digital evidence not only includes computers in the traditional sense, but also digital audio, video, and pictures.

Digital Forensics - a division of forensic science which focuses on the identification, examination, collection, preservation, and analysis of data from any device that can store electronic/digital information, such as computers and mobile phones. The science is applied in both criminal and civil investigations in a court of law, and in the private sector when investigating internal issues or intrusions.

Discord - a voice chat application popular with gamers and streamers, that allows for Voice over Internet Protocol (VOIP) and messaging between users.

Dropbox - a file hosting service operated by Dropbox, Inc. It offers cloud storage, file synchronization, and client software that allows users to upload data into storage that is kept on Dropbox's servers.

EnCase - a suite of digital forensics tools created by Guidance Software. The software comes in several forms designed for forensic, cyber security, and ediscovery use.

Registry Viewer - application that is used to view and search through Windows Registry Files.

Rename Windows Registry (Windows Registry or Registry) -in Windows operating systems, the registry contains information about the hardware, network connections, user preferences, installed software, and other critical information.

Slack - an business focused group Messaging Web Application, allowing companies to create channel templates for communication between employees.

SQLite - an embedded database for local/client storage in application software (Most Widely Deployed SQL Database).

SQLite Browser - a client side application allowing for the viewing of SQLite database files.

Virtual Machine (VMs) - a software-based computer that executes and runs programs like a physical machine.

.VMDK - a VMWare file format that emulates a hard drive

Web Application* - an application in which all or some parts of the software are downloaded from the Web each time it is run. It may refer to browser-based apps that run within the user's Web browser, or to "rich client" desktop apps that do not use a browser or to mobile apps that access the Web for additional information.

Methodology and Methods

Prior to the start of testing, we established three virtual machines across three different Operating Systems (OS). The OS choices we had for testing were Windows 7, Windows 10, and Mac OS Sierra. Each clean install was updated and then powered off. The reason for this was to create copies of each VM for the different applications that the group was testing. Each application had the same exact baseline, and from a testing standpoint consistency is key. The app data gen process was similar between Slack, Discord, and Dropbox, but each yielded different results. Slack was the first app the team decided to analyze. We sent messages to generate data between the virtual machines. After sending several messages to each other through Slack, we suspended our VM's and created the clones. After we created the clone, we were able to use the .vmdk file from Slack and process and acquire it in EnCase. Dropbox followed a similar process but the actual data gen was slightly different. Dropbox has a drag and drop feature for file uploads, so most of us were able to upload images or text documents with ease. After we generated enough to deem sufficient, we moved on to the clone and analyzed it through EnCase, which finished up our data gen for Dropbox. Discord was the last app for data gen. Every last feature of Discord was tested (file upload, role change, user banning, link sharing, normal messages, emojis, custom emojis, and so on) to ensure we could find as much as possible. We originally thought Discord would hide the most, but thanks to this thorough data gen we found more user generated artifacts than predicted.

VMware - VMware was our primary tool for the duration of the Application Analysis project. VMware was used to virtualize computers. In order to start our data gen on our applications, we first had to install our selected operating systems on VMware. VMware would then allow us to create a clone of the virtual machine. We would take the clone .vmdk file and then export it, so we could further investigate what was left on the machine that we did our data gen on.

EnCase - The forensic tool we used for this project was EnCase. EnCase is a piece of software that contains multiple tools that allowed our team to look further into virtual machine files. After creating the cloned virtual machine files, the team would take each file and process and acquire it inside of EnCase. This tool allowed us to view cache files directly, getting info a user usually could not access. Links, pictures, and even user input that was recorded in cache were all found within the caches of the web apps that we looked at.

Appendix A shows the steps taken with each of the Applications.

Equipment Used

Device/Software	Version		Comments
iMac	Late 2013		Used to host the Mac VM
VMware Fusion	8.4.3		Running on iMac
VMware Workstation	12.5.2		Running both Windows test machines
EnCase Forensic Training	8.01.00.77		Acquire and process the data generated
Mac OS Sierra	10.12.3		Running on Both iMac and Mac OS Sierra VM
Windows 7	X15-65805 x64		Professional
Windows 10	6851151 x64		Technical Preview
Slack	Mac: 2.4.1	Windows: 2.5.1	
Dropbox	Mac: 20.4.19	Windows: 16.4.3	
Discord	Mac: 0.0.247	Windows: 0.0.297	
010	v7.0.2		Hex Editor
SQLite Browser	v3		Allows for easier viewing of SQLite Databases

Analysis

The data used was acquired and processed using EnCase v8.01. Further file analysis was done using SQLite Browser and 010, a hex editor. The group focused on the differences in investigating Mac and Windows as well as data handling post deletion. An Excel spreadsheet was used to record the process of our data generation. The group parsed through the images and investigated what files contained evidence of user activity as well as what files the user uploaded. The files that contained data were highlighted in the notes and the information they contained was recorded in Google Drive. To search image files and browse SQLite Databases both 010 and SQLite Browser were used to analyze the information that is logged and stored by the Web Apps. The same processes were done on the Images post app deletion. The acquisition and processing was done by EnCase and further data analysis done with 010 and SQLite Browser.

Results

Slack

After doing our analysis, we found a Slack folder located at: *Macintosh HD\Users\<username>\Library\Application Support\Slack* for Mac, or *C\Users\<username>\AppData\Roaming\Slack*. This folder contained all of the user related data (username,

profile pictures, etc.) that Slack stored on the system.

1	Cache	94	Folder
2	dictionaries	108	Folder
3	GPUCache	100	Folder
4	Local Storage	110	Folder
5	logs	92	Folder
6	storage	98	Folder
7	temp	92	Folder
8	Cookies	19,456	Unknown
9	Cookies-journal	0	Unknown

Figure 1 Storage locations for Slack

The *Cache* folder contained images that were found in Slack. There were many gifs that were used as well as images people uploaded. In the *GPUCache* folder, we also found profile pictures of the users in the Slack channel after Encase parsed the file. We were also able to find users who had previously been in our Slack channel, but no longer are. These were located in the files *Data_0*, *Data_1*, *Data_2*, and *Data_3*. This can be useful to an investigator because they can see users who are in the same Slack team as the user as well as users who used to be in that Slack team but no longer are.

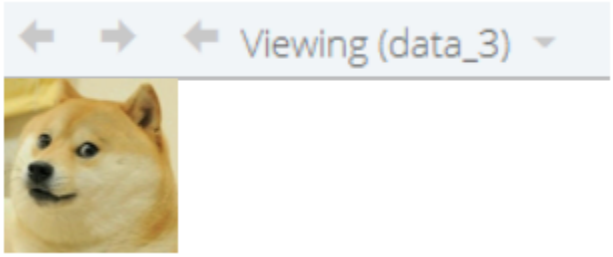


Figure 2 Image found in the data_3 files

In the Local Storage folder, we found a file that contained information about the channels the user has access to by ID. The following screenshot shows the IDs of each channel the user has access to (there are 8 channels). The ID of the Slack team is listed with the active_history immediately after it.

12	U260ZBQSW_active_history	*BLOB*	["G43U1RP9N","C2EKD7P0C","G35UJMQF9","C1HTJB53L","D44MMCA5T","G3A4YRN3T","D260GBFBJ","D260FU4K0"]

Figure 3 U260ZBQSW_active_history

The logs folder stored log files for Slack. Of particular interest, the webview-1.log file contained information

about the user, the team they are on, and other information about their Slack client. This file contains the name of the user on the Slack team, the team url, the ID of the team and user, as well as the team's logo. This file also includes timestamps of when these logs were collected.

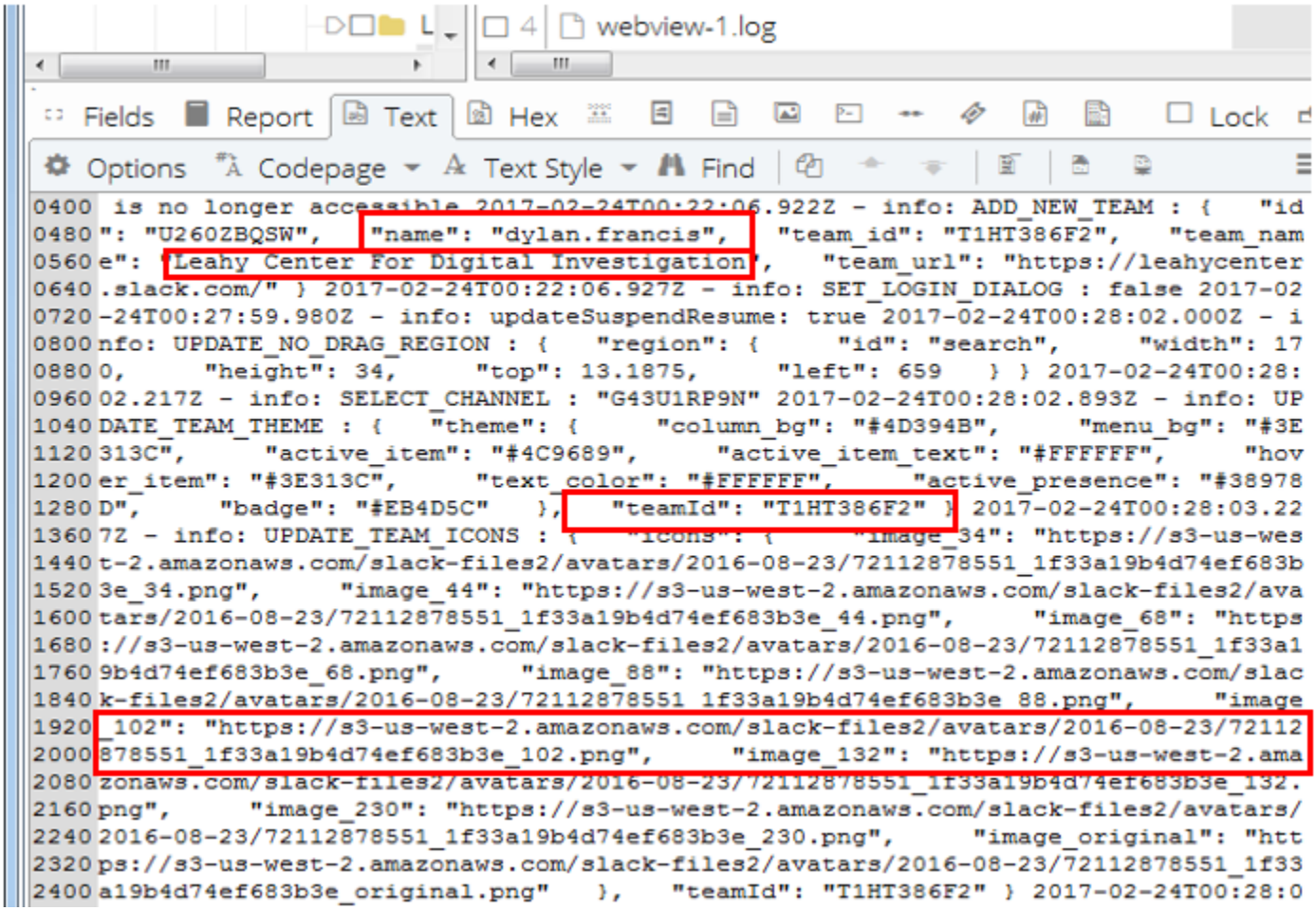


Figure 4 Text of the webview-1.log file

Dropbox - Windows 7

The most interesting artifact was the *C:\Users\User\AppData\Local\Dropbox\instance1\aggregation.dbx* file. Below is one of the examples of the data stored within the file. These are the files we uploaded for testing and each one can be found along with timestamps.

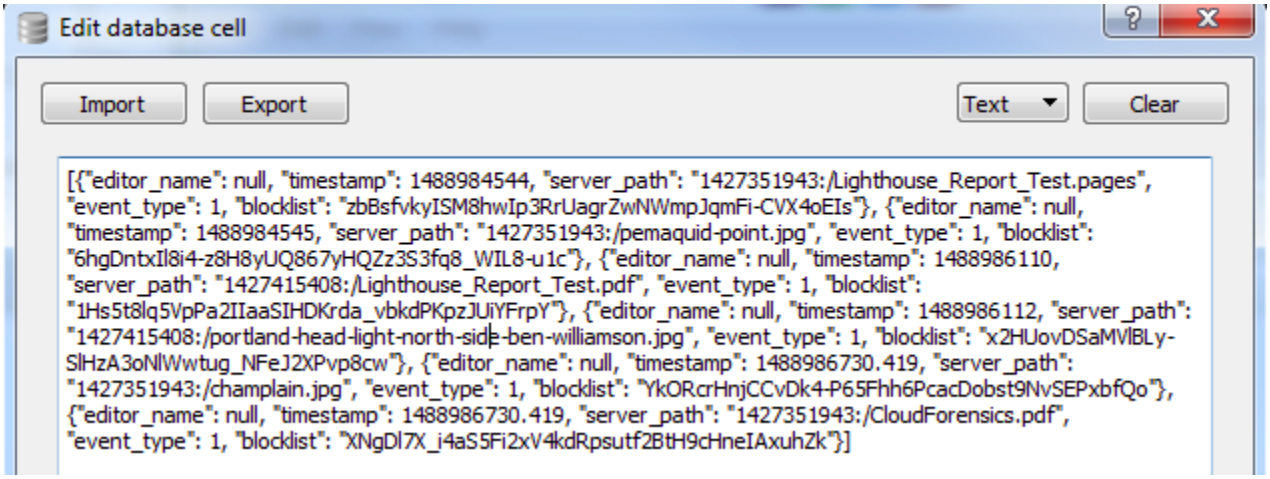


Figure 5 Contents of aggregation.dbx

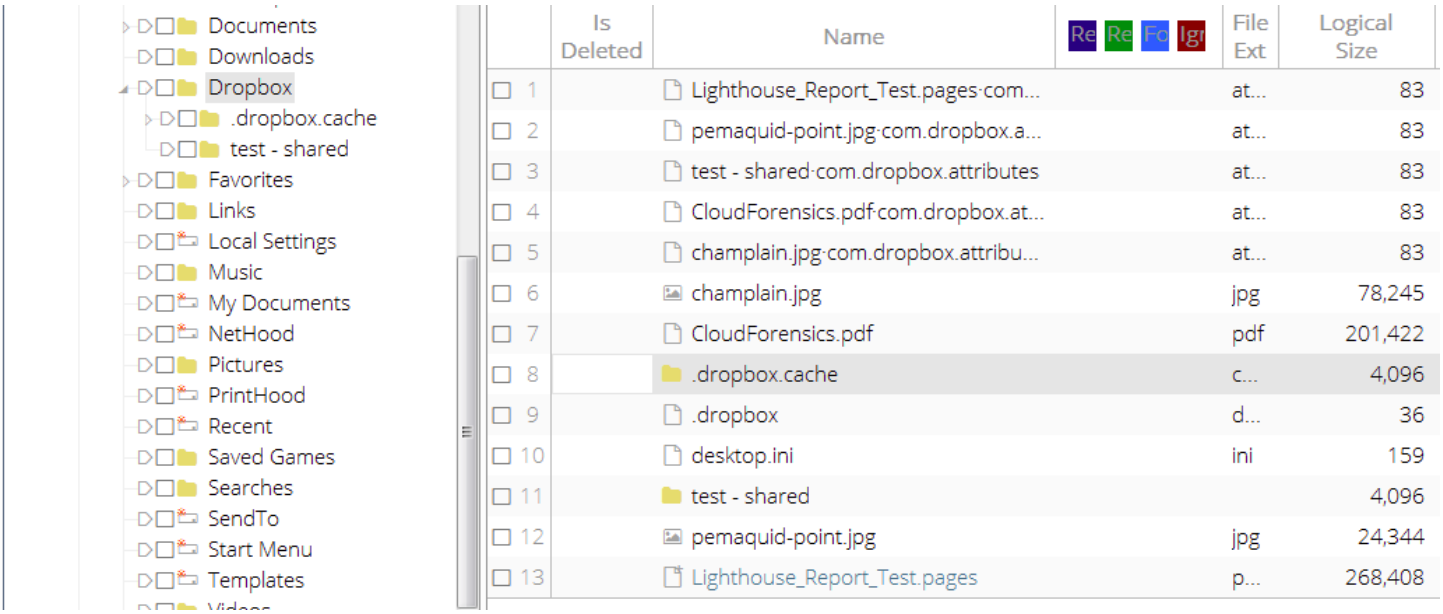


Figure 6 Dropbox User folder being visible

The AES-256 Encryption does not extend to the client machine. Below is seen the Dropbox user folder post deletion of the app. The folder was not encrypted on the client server nor was it deleted after the app was deleted. Dropbox/dropbox.cache also contained traces of deleted files that were removed from the Dropbox account before it was installed and accessed on the machine.

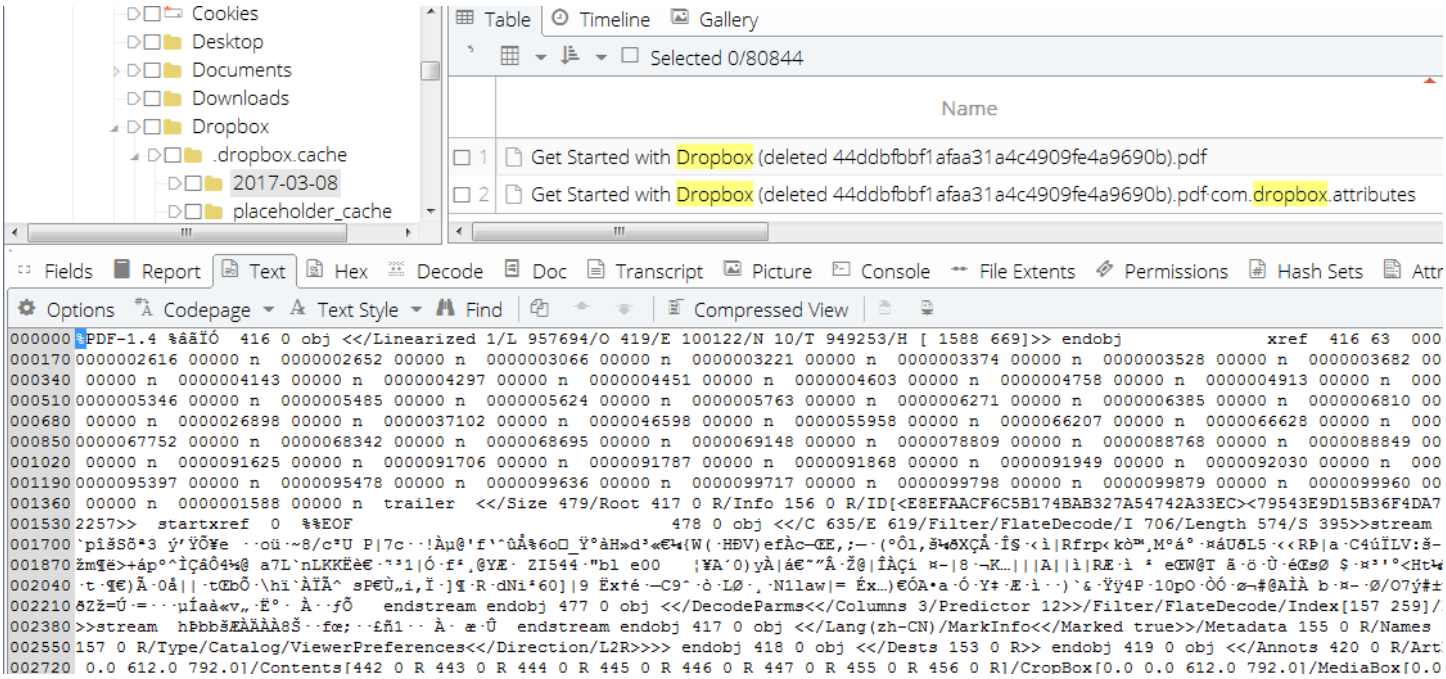


Figure 7Deleted Dropbox file being visible in the Dropbox.cache folder

Discord

The group found way more data and artifacts on Discord than we originally thought we would. Inside of Discord's cache we were able to see the chat logs and each individual message that we sent. The image below contains an example of this with the message highlighted.

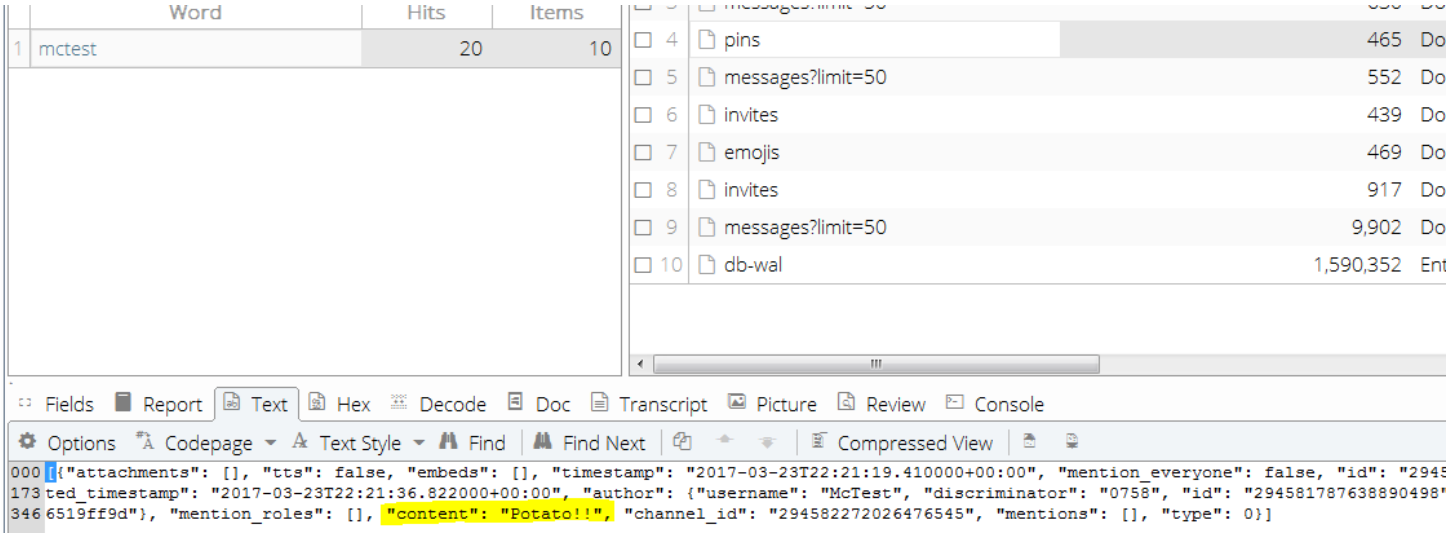


Figure 8 Messages?limit=50 with the message content highlighted

Discord for Mac

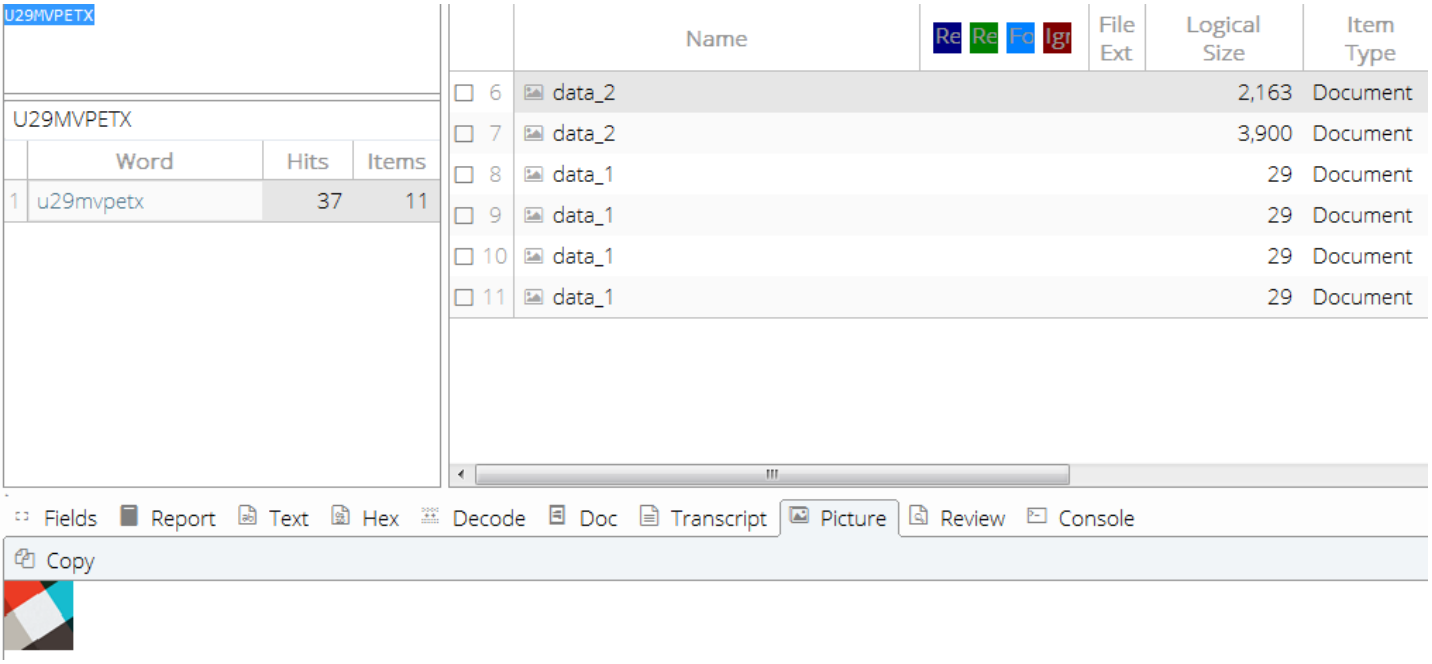


Figure 9 User U29MVPETXs profile photo found be user id

Discord was by far the most interesting as, after parsing the data, the group was able to find the message logs the Users had in *messages?limit=50.* The files contained the message with the content along with any images being sent.

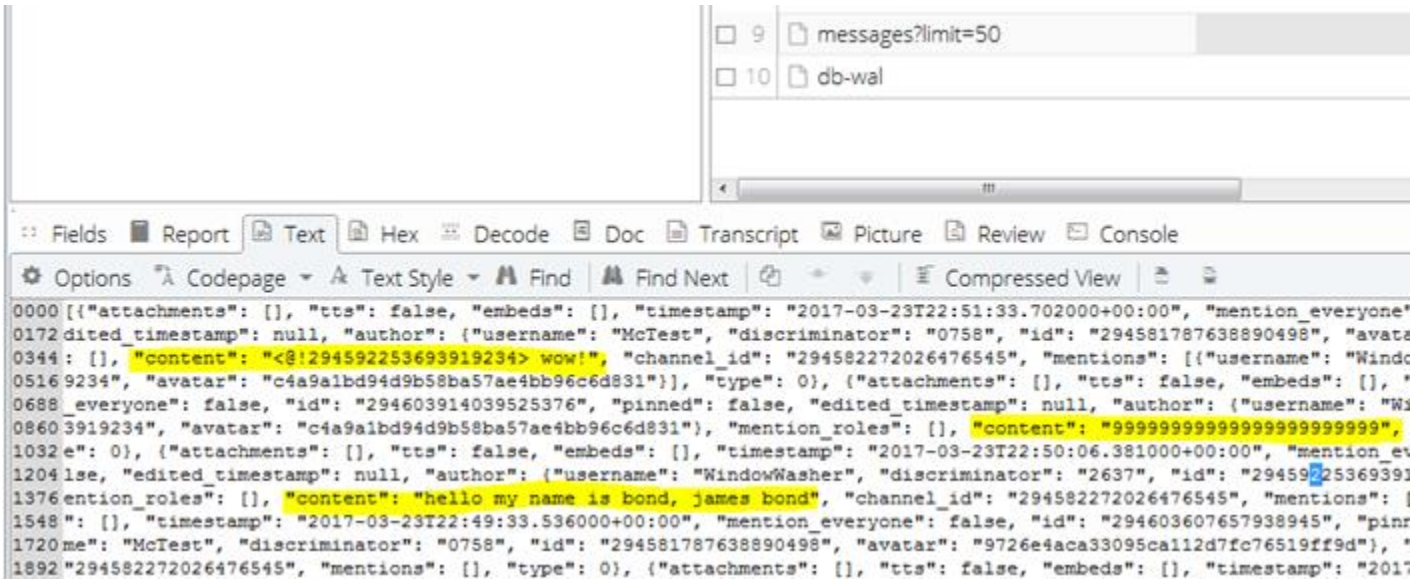


Figure 10 The messages?limit=50 with the messages being highlighted

Figure 11 The messages?limit=50 with the messages being highlighted

The folder path to the image above: *C:\Users\<username>\AppData\Roaming\discord\Cache* which shows the *messages?limit=50* file. In the cache, each OS followed the same basic outline with Discord, as each user found the message logs along with the images sent to the channels. The final artifacts found were in the local database. Which provided us with information such as the email and username of the person who logged on as seen in the figure below:

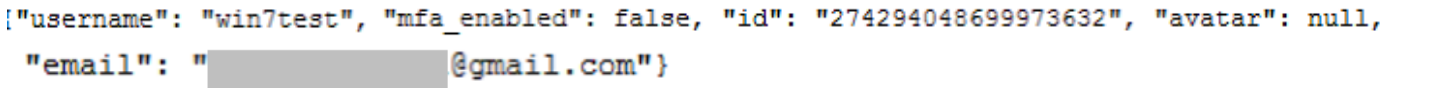


Figure 12 The user's login information found in the local database

Conclusion

In conclusion, our original hypothesis decided that Discord would store the least artifacts, and Twitter would store the most artifacts. We were wrong. In the end, we couldn't actually take on Twitter due to no current app existing for Windows 7. Discord had the most user generated found artifacts and left behind the most data. Dropbox and Slack showed us differing results, but Discord by far had the most to give. The usage of local SQL storage on cache was a trademark of all the tested applications with everything from file uploads to emoji usage being tracked. Across all of the platforms, the Web Apps followed a standardized format and data could be found in a similar manner. The only major difference was that Windows 10 did not cache profile images in Slack.

Further Work

The Web Application Analysis team covered a lot in this project, but there are so many apps that could have also been investigated. We originally wanted to investigate Twitter and Steam, but due to personal student projects and the lack of a modern app for Windows 7, we were unable to look at these applications. Further work on this project should try to cover Steam and Twitter, if they become available. Other apps are available to work on as well. As for specifics, it will be up to the team or project leader at the time. This type of project is very flexible in regards to what one wants to work on.

Appendix

Appendix A - Data Gen

Table 1 Mac OS Sierra – Base Image

Time Conducted		What was done?
02/02/2017 19:35		Logged on Installed VMware Fusion 8.5
02/06/2017 8:55		Updated VMware Fusion to 8.5.3
11:07		Mac OS Sierra on VM
02/08/2017 10:02		Reinstalled Mac OS Sierra on the VM. Virtual Machine was not on the machine upon power up
10:03		Working on installing Sierra on VMware Fusion (Default Settings: English for the main language, Install Mac OS (Upgrade or Install a new copy of Mac OS), Continue, Agree, Agree, Install on Macintosh HD, Install (Wait approximately 7 minutes for install) / (Machine just restarted and has approximately 15 minutes remaining for install) Note: Installing on a virtual machine with 2gb of ram))
10:49		Changed Date and Time Settings (Original Configuration was Pacific Standard time instead of Eastern Standard Time)
11:16		Virtual Machine Restarts/Finishes installing
11:25		Install Complete (Default Settings)
11:46		Logged into Clean Install
02/09/2017 19:24		Upon Login ran Updates
19:35		Once Updates Completed the Virtual Machine was powered down
19:36		The VM was then cloned into three full clones (One for Slack, Dropbox, and Discord)

Table 2 Slack – Mac OS

Time Conducted		What was done?
02/09/2017 19:40		Powered on VM
19:43		Installed Slack via the Apple App Store
19:54		Paused VM
02/13/2017 8:00		Resumed VM
8:02		Login Using Established Credentials
8:40		Created a board called testchannel2
8:42		Added the user @cedric to testchannel2
8:43		Sent the message "Hi" to @dylan
8:49		Sent the message "Hackers...hackers everywhere" in testchannel2
9:00		Sent written message "Hello World" from @dylan
9:02		Sent a Hacker Photo to @dylan
9:03		Sent @cedric a message "Hello"

9:04		Sent a /giphy forensics on the channel
9:10		Powered Off VM
11:00		Copied the .vmdk files onto Google Drive and will look at more files in EnCase
02/15/2017 10:06		Began acquisition of the evidence using EnCase v8.01
11:08		Processing of evidence is completed

Table 3 Slack – Windows 10

Time Conducted		What was done?
01/25/2017 8:00		Created Windows 10 VM
8:30		Move VM to Network Drive
02/22/2017 8:50		Initialized Windows 10 in VMWare
9:16		VM would not open properly. Will try through ESXI server.
10:30		I now have Slack installed on the VM, will make clone. The network is slowing this down but I just have to create a clone and then I can run the messages through.
02/23/2017 16:23		Began Slack Tests
16:30		Sent “Test for 10” on testchannel1
16:32		Removed “Test for 10” on testchannel1
16:33		Sent “test for 10” on testchannel1

Table 4 Slack – Windows 7

Time Conducted		What was done?
02/02/2017 17:30		Windows 7 Virtual Machine was Created
02/08/2017 8:30		Downloaded and installed Google Chrome on the Win7 VM
8:35		Downloaded Slack (Install failed needed to download .NET 4.5 Manually)
8:45		Installed NET 4.5 Framework from MS manually
8:47		Installed Slack Successfully created Desktop Icon
10:03		Initial Launch of Slack
10:18		Initial Logon on Slack. Kept Default settings
10:25		Sent test messages to board, @dylan, and personal, sent txt file to myself, desktop notifications off
10:27		Created board testchannel1 added all group members
10:46		Closed Slack waited for notification, pop up came Slack opened
10:55		Reverted Win7 machine to snapshot post login(snapshot2) changed the default settings to enchant privacy
11:11		Conducted test messages. Used the procedure from testrun1 private message, personal message board message, set txt to myself
11:24		Received test message in Slack on default settings

11:40		Uninstalled Slack (uninstalled Slack through control panel)
02/09/2017 16:43		Began retesting Slack with streamlined processes
17:21		Used default settings to test Slack
17:25		Deleted snapshot through windows control panel, took snapshot
02/15/2017 10:39		Reverted win7 machine to posttest snapshot, cloned it
02/16/2017 17:11		Keyword searched word list Found hits on Leahy, Slack, LCDI, sgorski, text.txt
02/17/2017 18:27		Search log files for timestamps Showed install time and when launched
18:50		Found channel ID info Does not show words just numerical tags
18:51		Keyword searched other channel names Found unrelated hits, Slack does not seem to store channel names, only through numerical tags that correspond to the channel

Table 5 Dropbox – Mac OS

Time Conducted		What was done?
03/01/2017 11:08		Started Data generation using Dropbox
03/06/2017 8:30		Dropbox was installed with all of the default configurations
8:48		Moved test files onto our VM and will upload to Dropbox one by one
8:49		champlain.jpg
8:49		CloudForensics.pdf
8:50		LCDI.png
8:50		Lighthouse_Report_Test.pages
8:50		Lighthouse_Report_Test.pdf.
8:51		Pemaquid-point.jpg
8:51		Portland-head-light-north-side-ben-williamson.jpg
8:51		Virtual Hard Disk Forensics Using EnCase-Nading-5-20-2015.pdf
8:52		Deleted LCDI.png
03/09/2017 15:58		Powered Down Virtual Machine
16:02		Copied the .vmdk files onto Google Drive and will look at more files in EnCase
16:10		Began acquisition of the evidence using EnCase v8.01
16:22		Processing of evidence is completed
16:58		Started analyzing the artifacts from Dropbox

Table 6 Dropbox – Windows 7

Time Conducted		What was done?
03/08/2017 8:35		Installed Dropbox on fresh install
10:20		Placed CloudForensics.pdf and champlain.jpg on Dropbox
10:42		Uninstalled Dropbox, unpinned from taskbar, used control panel uninstall program to uninstall Dropbox
11:44		Processed Dropbox post testing image

Table 7 Discord – Mac OS

Time Conducted		What was done?
03/23/2017 17:25		Powered on Virtual Machine
17:30		Created a Discord account McTest and a Discord server Test-Discord
17:33		Created two doge emojis :DankMemes: and :Doge:
17:39		Created a testchannel and testchanneladmin text channel, as well as TestVoice1
17:48		Wrote "Hello World" in the #general section
18:26		Added a poop emoji reaction
03/27/2017 20:30		Data Gen Ended and data extraction was started and .vmdk was moved onto a Network Drive for EnCase Analysis

Table 8 Discord – Windows 7

Time Conducted		What was done?
03/23/2017 17:50		Begin the installation process on the windows 7 VM
17:55		Create Discord account Username: win7test
18:07		Data gen added image to mac Discord channel
18:28		Logged in sent message on got the handle mcsteak on testchannel
18:30		Began Data Gen-uploaded images and .txt to the Win7 test channel as well as the Mac Test Channel
19:20		Data Gen ended
19:27		Made clone of VM uninstalled Discord, through control panel moved under Research Projects

References

Katz, M. & Montelbano, R.(2013, November 04). Cloud Forensics, Retrieved February 09, 2017, From Senator Leahy Center for Digital Investigations.

Encyclopedia. (n.d.). Retrieved February 20, 2017, from http://www.pcmag.com/encyclopedia/

Yang, T. Y., Dehghantanha, A., Choo, K. R., & Muda, Z. (2016, March 16). Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies. Retrieved February 23, 2017, from http://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0150300