2015

Cloud Signature Creator: An Application to Establish Cloud-Computing Application Artifacts

Gerald W. Gent
University of Rhode Island, ggent@cs.uri.edu

CLOUD SIGNATURE CREATOR

AN APPLICATION TO ESTABLISH CLOUD-COMPUTING APPLICATION ARTIFACTS

BY

GERALD W. GENT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER SCIENCE AND STATISTICS

UNIVERSITY OF RHODE ISLAND

2015

MASTER OF SCIENCE THESIS

OF

GERALD GENT

APPROVED:

Thesis Committee:

Major Professor Victor Fay-Wolfe

Lisa DiPippo

Haibo He

Nasser H. Zawia
DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2015

ABSTRACT

As the percentage of computer users that utilize cloud-computing services grows, more potential evidence for state and local law enforcement investigators is being stored with these cloud services and not on a local computer's hard drive. To address this problem, this project created a tool called *Cloud Signature Creator* as a solution that allows an investigator to locate potential areas of a computer's file system that contains evidence useful to their investigation. The Cloud Signature Creator solution leverages existing technologies and implements a new software application that provides the end user with a listing of files and locations that might indicate a cloud service was utilized on a suspect computer.

ACKNOWLEDGMENTS

The work in this thesis would not have been completed without the support and assistance from so many. First I would like to thank all of the members of the Digital Forensics and Cyber Security Center. Namely, I would like to thank Jacob Fonseca for pushing me to get the work done, when I might have been able to find something else I would rather be doing.

Additionally, I would like to thank my family for their patience and support in this endeavor. I would like to thank my wife, Chelsea, for dealing with me while I was stressed out attempting to make a deadline and thank you to my parents for instilling within me a work ethic and positive attitude that made it possible to complete this thesis.

Finally, I would like to thank Dr. Victor Fay-Wolfe for the wonderful opportunities academically, professionally, and personally. Without his guidance, I would likely not be involved in the field of digital forensics. It is because of his support and commitment to his students that I am able to accomplish this milestone.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1
INTRODUCTION

Today's Internet allows its users to store, create, and share documents and other files in the cloud. Suspects of a crime can store files of potential evidentiary value in these cloud-computing applications. State and local law enforcement investigators need to have a way to uncover this evidence, which may no longer be stored on a user's physical workstation.

1.1 Statement Of The Problem

The increase in Internet users utilizing cloud-computing applications for the purpose of data storage, running a desktop environment, or processing some type of data provides some unique new challenges to state and local law enforcement when performing digital forensics investigations. Due to the rise of the use of such applications, criminals may no longer need to store the evidence of their crimes on a local device that a law enforcement officer is capable of seizing. As a result, law enforcement requires a means of determining whether certain cloud-computing applications were utilized on a computer, and to determine information about the user that might allow them to request information from the service provider.

1.2 Justification For And Significance Of The Study

According to the National Institute of Standards and Technology (Mell & Grance, 2011), cloud computing is a model for on-demand access to configurable computing resources such as storage, applications, and services. A Fortune article

(Griffith, 2014) cites Dropbox, Microsoft's OneDrive, and Google's Drive as the top consumer cloud storage providers. Dropbox claims to service 300 million users as of May 2014, Microsoft claims over 250 million, and Google provides for 240 million users as of September 2014. Each of these companies offer free data storage plans, making the option attractive to computer users. When a user stores a photo, video, or document with a cloud storage provider they can access it from any location on any device with an Internet connection. This ultimate portability provides a challenge for state and local law enforcement in performing digital forensics investigations.

Due to 4th Amendment limitations on the scope of search warrants, when a law enforcement officer requests data from a cloud service provider company, the officer must provide enough information to identify the user, the time range, and the type of information requested for which probable cause has been established. The use for an application that establishes cloud-computing application artifacts is to provide assistance to the investigating law enforcement officer as to where important information might be stored on the local computer.

1.3 Goals

The goal of this project is to create a solution that may be used to determine specific files and locations that are modified during the use of a cloud-computing service. In order to accomplish this goal, the solution must:

1. Monitor file system changes that are a result of usage of a cloud service.

2. Accept information collected from the monitoring system and hash the files, removing duplicates from the list to direct an investigator to specific storage locations for cloud services.
3. Compare the hash lists from multiple monitoring instances to show files that are present in both instances to narrow an investigator's focus for potential artifacts of cloud services.

1.4 Summary Of Accomplishments

The result of this project was the creation of the *Cloud Signature Creator* application that may be used to provide the investigator with a listing of files with potential evidentiary value. The *Cloud Signature Creator* application met the goals specified in Section 1.3 by utilizing pre-existing software and implementing new software that parses data from the monitoring software.

CHAPTER 2

REVIEW OF LITERATURE

This chapter discusses conceptual and technical materials that aided in the development of the *Cloud Signature Creator* application that corresponds to the goals of Section 1.3 by providing context and foundation to the research. The chapter will begin by discussing technologies that the application leverages to accomplish stated goals. Next, it will discuss related works that serve as the basis and inspiration behind this project. Lastly, the target audience of the *Cloud Signature Creator* application is defined.

2.1 Technologies

This section elaborates on the technical components required to build an application to reveal artifacts of a cloud-computing application's use. Specifically, this section will describe any software tools required and programming components that are utilized by the resulting application.

2.1.1 Sysinternals' Process Monitor

Sysinternals' Process Monitor is a monitoring tool for Windows that reports activity to file system, registry, and process/thread objects. (Russinovich & Cogswell, 2014) This Windows utility will be leveraged in this project to track changes made to the files stored on the computer's hard drive and report them to the investigator. The software developed on this thesis project will pare down the information that Process

Monitor reports to provide only data that is relevant to obtaining cloud-based evidence.

2.1.2 Microsoft Visual Studio

Microsoft Visual Studio (Microsoft Visual Studio) is an integrated development environment that is used to develop and test computer programs for the Microsoft Windows environment. Visual studio supports a multitude of computer programming languages, to include C, C++, and C#. C# has been selected for the primary programming language in this application due to it being a simple, modern, object-oriented programming language.

2.2 Related Works

This section provides insight into several related works that serve as both a basis and an inspiration to the *Cloud Signature Creator* application. These works include previous research projects that are a launching point for this project and commercial products used in the digital forensics industry.

2.2.1 Internet Evidence Finder

Magnet Forensics' Internet Evidence Finder, or IEF, (Magnet Forensics) is a software solution used to find, analyze, and present evidence found on computers, smart phones, and tablets related to Internet activity. One of the subsets of applications that IEF supports is cloud-computing applications. IEF supports a large number of applications, but with more cloud-computing applications being released there is a delay on when user's of the software will have access to evidence from a new cloud-computing application. The user selects the artifacts that the program should search for

from the IEF interface. The number of artifacts the user has selected to search for will alter the amount of time the user must wait for the results to be completely reported. IEF falls short for users when a cloud application updates its client application or storage method or a new cloud application is released. When this occurs, the user needs to wait for the IEF support staff to notice the problem and release an updated version of the tool. This can cause an investigation to grind to a halt, especially if there is not another way to locate the information desired by the investigator.

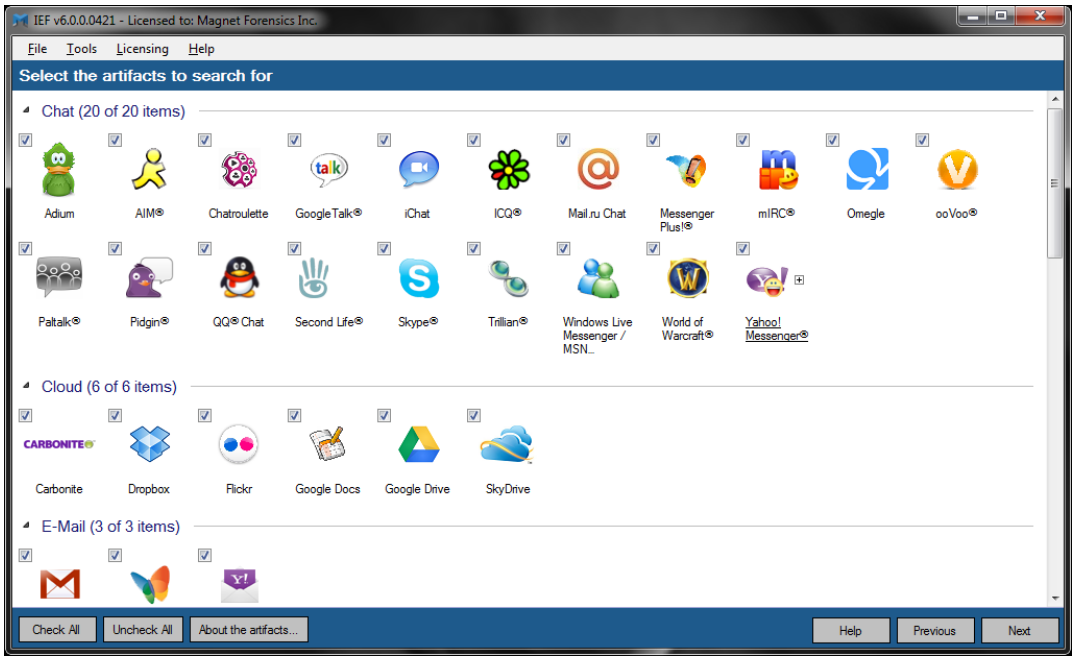

Figure 1 - Internet Evidence Finder Interface

2.2.2 EnCase Forensic

Guidance Software's EnCase Forensic software (Guidance Software) is a forensic solution that examiners frequently utilize to analyze hard drives and removable media. EnCase is an effective tool for data collection and investigations for active and deleted files, but is a full forensic tool for investigations of the entire

contents of a hard drive. EnCase does not specifically search for and parse information of cloud services that had been utilized on the device it is analyzing, however scripts can be created to attempt to parse information when one knows where that data is stored.

2.2.3 FTK

AccessData's Forensic Toolkit (FTK) (AccessData) is another digital investigation platform that examiners utilize to analyze hard drives and removable media. FTK is a powerful tool for searching and filtering data on the devices, but does not specifically target any cloud applications. The examiner would need to know where the data they are interested in is being stored by the cloud application in order to extract any information that might be able to further their investigation of the suspect's use of a particular cloud service.

2.2.4 Cloud Signature

Cloud Signature (Koppen, 2012) is a software tool created by the University of Rhode Island's Digital Forensics and Cyber Security Center (DFCSC) to provide the user with information about a specific list of supported cloud-computing applications that it detects on a computer. This tool parses the information it has detected in specific files known to be associated with supported cloud-computing applications, but the release of new applications and updates to existing supported applications causes the tool to be out-of-date too quickly.

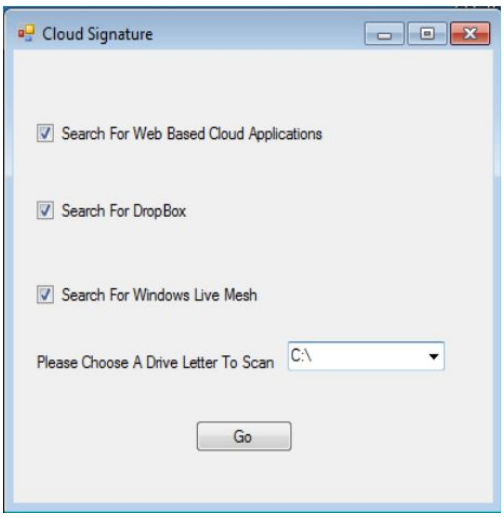

Figure 2 - Cloud Signature Interface

2.3 Target Audiences

Based on the goals of this project, two target audiences have been identified as potential users of the *Cloud Signature Creator* application: *Forensic Application Developers* and *Computer Crime Investigators*.

2.3.1 Forensic Application Developers

Forensic application developers that are interested in adding new cloud applications to their current products may have a need for this application to point out changes in the file system while using cloud services. These changes may indicate locations from which useful information might be able to be regularly extracted

2.3.2 Computer Crime Investigators

Computer crime investigators are users that have a particular case that existing tools do not support a cloud service that a suspect is known to be utilizing. This investigator must have a technical knowledge in order to use the service that their

suspect is utilizing, but the use of this application might lead the investigator to some new evidence that could result in further legal process or an arrest.

CHAPTER 3

METHODS

This chapter consists of three sections. The first section discusses the procedures used to develop the *Cloud Signature Creator* application and discusses design decisions made throughout the process. The second section describes different use cases for the end user and explains how one's workflow might be. The third section discusses testing procedures used to measure the effectiveness of the application and to determine that the project met the goals stated in Section 1.3.

3.1 Application Development

The development of the application can be divided into three main areas: the overall conceptual design of the application, the design and development of the user interface, and the design and development of the reporting function of the application. The conceptual design of the application takes Koppen's work on *Cloud Signature* (Koppen, 2012) adapts it to be able to detect general cloud-based installations on a computer to add to the tool's ability to stay relevant with quickly changing cloud applications. The user interface is both a control center for the application and instructions to the user as the application leverages other tools to work properly. The reporting function of the new tool produces two different reports that the user may be able to utilize to further their investigation.

3.1.1 Conceptual Design

The initial concept of *Cloud Signature* was to create a tool that parses through a hard drive and provides a user with the data that might be able to help them to sufficiently fulfill the needs of a service provider to respond to legal process. Keeping this tool up-to-date, not only on the cloud services initially supported, but also for support of new cloud services, requires significant effort. In fact, this became more evident with Magnet Forensics' purchase of Internet Evidence Finder and commercializing a similar product (Magnet Forensics). The very high rate at which the cloud services were updating and releasing new applications dictates the rate at which updates to *Cloud Signature* would need to be released in order to keep up with this growing industry. Keeping up with this manually, as *Cloud Signature* was originally designed, was prohibitive. The *Cloud Signature Creator* tool created in this project was designed so that it could stay relevant by assisting in the processing a hard drive of a suspect that is known to be using a cloud service that is not yet supported by the industry tools. The *Cloud Signature Creator* tool is to be utilized as a part of a forensic analysts procedure. The tool will not take the place of an entire suite of tools that one might utilize to conduct and investigation, but will perform a smaller task that will help the investigator get the job done. To do this the application monitors changes made to a system while a investigator is utilizing a cloud service on a testing machine. These changes are monitored over several uses of the service and then generated into a report that provides file locations for an investigator to manually parse through to locate some data of potential evidentiary value on their suspect's hard drive. The following diagram highlights the space in an investigative process that *Cloud*

Signature Creator would fit. Please note that this sample workflow might change during the course of an investigation as new developments arise.

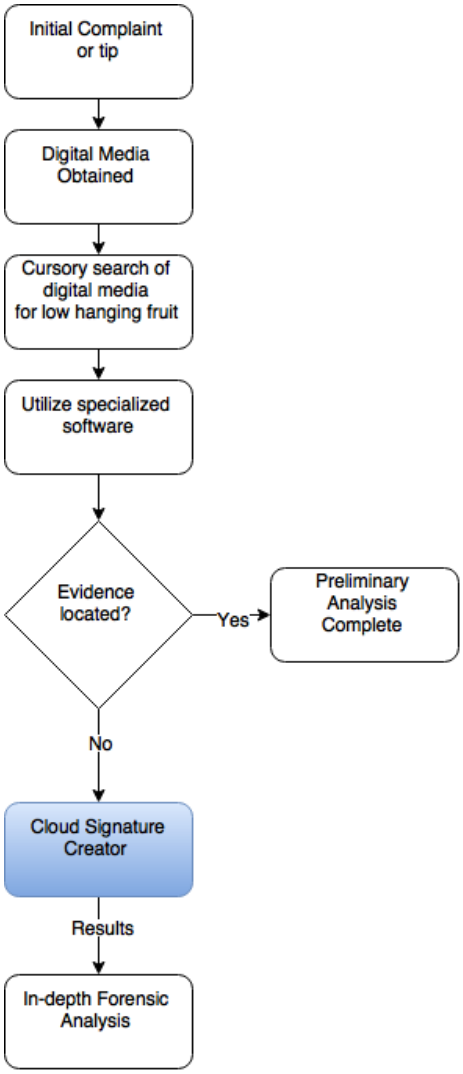

Figure 3 - Investigative Workflow

The *Cloud Signature Creator* solution monitors the file system's changes to determine which files were being created and/or modified during the use of a cloud service. To do this, Process Monitor was utilized to follow a specified process running on a Windows computer. (Luttgens, Pepe, & Mandia, 2014) Process Monitor is

already known to be capable of tracking file system changes like file creations and modifications. It can also be directed at one specific process, such as a web browser, and filter the results to include only changes made as a result of that process. This tool is a good basis to be leveraged for the purpose of file system monitoring in the *Cloud Signature Creator* application.

After determining how to address the file system changes, the project added the ability to compare several collections of the changes during the use of the cloud service in question. This collection of data results in a listing of files and hash values that were created and modified during the use of the cloud service, and the comparison takes two of these collections and results in a listing of hash values that are common across the different runs. Unfortunately, some of the files that would be most important to the investigator will not have the exact same contents across multiple data collections because of the actual file contents being modified. Therefore, the comparison phase also results in a listing of all files that were created and/or modified in both collection runs. The investigator can then utilize this information to manually look at the contents of any or all of these files for pertinent data. The investigator is looking for contents of the files that would provide probable cause for a law enforcement agency to obtain a search warrant for the entire contents of the cloud account.

3.1.2 Implementation

For this project, the selected programming language was C# with Microsoft Visual Studio as a development environment. This environment was chosen for its

ease of generating Windows graphical user interfaces and the robustness of the standard libraries associated with the C# language (C# Programming Guide). The C# language's ability to create classes of objects and its library's prebuilt forms make it very simple to create Windows forms and dialogs.

3.1.3 User Interface

The *Cloud Signature Creator user* interface has two phases. The first phase is the *data collection phase*. In order for this application to work to its fullest potential, the cloud service must be run several times while tracking the file system changes in order to remove some file anomalies that are a result of normal web browsing and computer usage. More runs of the service will result in the most concise results of files that are exactly the same across the collection runs, but any slight changes to the user's content will result in the files not being reported.

The second phase is the *comparison phase*. The comparison phase takes input of two listings of files that were collected from previous runs of the cloud service as a result of the collection phase. The collected file listings are compared and produce a report of files that contain the same content across both collections. The figure below is the opening screen of the *Cloud Signature Creator* application.

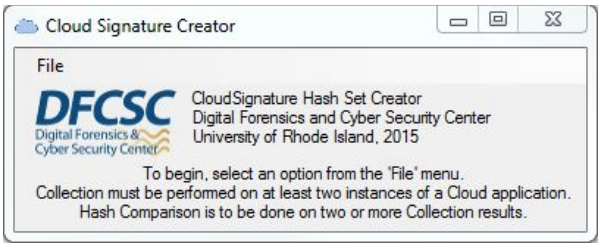

Figure 4 - Cloud Signature Creator Interface

3.1.3.1 Collection Phase

The collection phase leverages Process Monitor to track the changes in the file system while a cloud service is being used. In order to do this, the user must set up the filter in Process Monitor to show results only for the web browser or other client with which the user is accessing the cloud service. The user is directed to start Process Monitor and adjust the filter before browsing to, and using, a cloud service. The interface asks for a location to a comma-separated value (CSV) file produced by Process Monitor, a Unique Identifier (UID) for the test, and a save location for the resulting CSV.

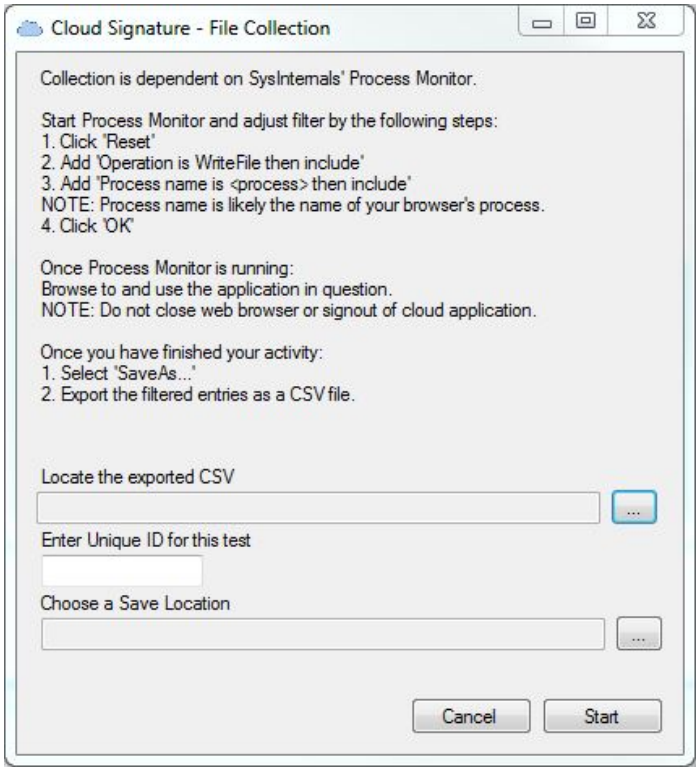

Figure 5 - Cloud Signature Creator File Collection Dialog

The *Cloud Signature Creator* application will process the CSV exported from Process Monitor to generate MD5 hash values of the listed files and output a CSV of the hash values for use in the comparison phase.

3.1.3.2 Comparison Phase

The comparison phase of the *Cloud Signature Creator* application is utilized following the collection of at least two runs of the cloud service in question. The application requests the two CSV files generated from the collection phase to compare hash values. The result of the comparison phase is a listing of hash values that are the same across both runs of the collection phase. Additionally, the application provides an output of all file names across both runs. The hash values that match are files that were created, modified, or accessed during both runs and have the exact same content. While these files may be useful to determine that a particular website or cloud service was visited, it is not particularly useful to gather information from specific files accessed in the cloud service if they are not identical. The files that do not have matching hash values, but are stored in the same locations, would be the files that might contain data useful to the examiners investigation. Therefore, the *Cloud Signature Creator* tool will provide the user with a list of common hash values and a list of all files modified during the use of the cloud service. Once the user has these things, they can conduct a more focused investigation.

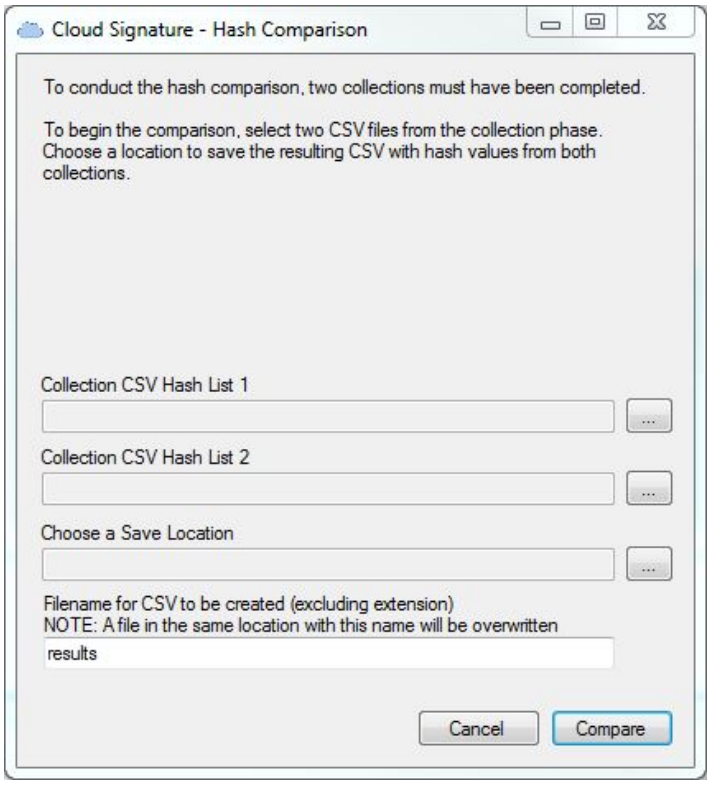

Figure 6 - Cloud Signature Creator Hash Comparison Dialog

3.2 Use Cases And Workflow

In this section, two different use cases will be examined and a workflow will be introduced. The *Cloud Signature Creator* tool is essentially the combination of two different use cases for the application that are combined to make one workflow.

3.2.1 Use Case 1 - Collection

The first use case that will be discussed will be the file collection use case. The user in this case would start the File Collection Dialog and follow directions listed on dialog. This includes the use of Process Monitor and *Cloud Signature Creator*. This activity can be utilized to show changes that occur from very specific actions that can be scripted by the user. At the end of this scenario, the user has a list of files that had

been created or modified during the use of the application, and the hash values associated with those files. The following sequence diagram shows the users interaction with the applications involved for this use case.

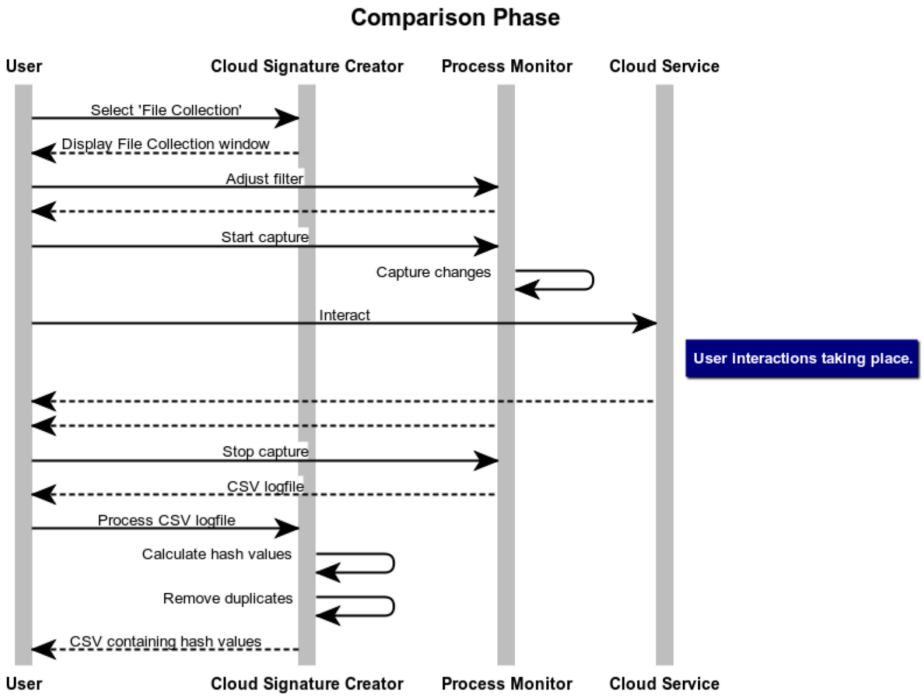

Figure 7 - Comparison Phase Sequence Diagram

3.2.2 Use Case 2 – Find Common Hash Values

The second use case is to compare hash values from two lists and receive a hash list with only values that are present in both lists. The user in this case would start with the Hash Comparison dialog in the *Cloud Signature Creator* and follow directions listed on the dialog. This method will generate a list of common MD5 hash values with any two comma-separated value (CSV) documents, so long as the first value in each row is an MD5 hash value. The following sequence diagram shows the users interaction with the applications involved for this use case.

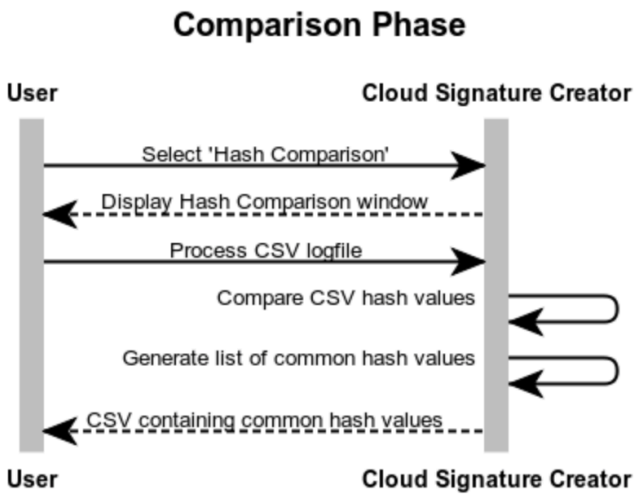

Figure 8 - Comparison Phase Sequence Diagram

3.2.3 Workflow

The minimum workflow for this solution is presented as a list below. This is the minimum workflow because it will include only two collections with Process Monitor. The more collections that are performed, the more narrowed down the list of common MD5 hash values should be. *Cloud Signature Creator* at first is a user intensive solution, however this is a necessity in some respects due to the nature of the problem being solved. This solution is user intensive because the user starts the two different applications and adjusts the filter in Process Monitor to suit their needs for a particular cloud service. Once the program is monitoring, the user then has to perform some actions while using the cloud service in an effort to simulate the usage of the cloud service as a suspect might utilize it. The minimum workflow is as follows:

1. Launch *Cloud Signature Creator*
2. Open File Collection dialog.

3. Launch Process Monitor.
4. Adjust the Process Monitor filter.
5. Start the Process Monitor capture.
6. Interact with the suspect cloud service.
7. End the Process Monitor capture.
8. Save the filtered data into a CSV file.
9. Process the CSV file with *Cloud Signature Creator* to hash and remove duplicate records.
10. Repeat steps 5-9 as a second collection run for comparison.
11. Open Hash Comparison dialog.
12. Locate two CSV files from the collection phase.
13. Process the hash lists with *Cloud Signature Creator* to create a list of common MD5 hash values.

3.3 Testing Procedures

This section describes the methodologies used to test the implementation of the *Cloud Signature Creator* solution. This section will identify the experiments that were conducted to evaluate how effectively this solution meets the goals defined in Section 1.3.

3.3.1 Testing Procedure 1

The purpose of Testing Procedure 1 is to determine that Process Monitor is capable of monitoring file system operations that will be appropriate to use for this solution. The Process Monitor application has dozens of different operations that can

be the basis for filtering. With the number of files system changes that occur during normal usage, it is important to filter the results to get the most useful information possible. The default filter in Process Monitor removes any entries that are a result of Process Monitor running. This testing process will conclude with an analysis of the Process Monitor log and filtering the data to determine that the data can be filtered to show only changes made by the user while using the cloud service. The testing procedure is as follows:

1. Start Process Monitor.
2. Select Default Filter.
3. Begin capture.
4. Perform testing activity.
5. Stop capture.
6. Conduct analysis.

Testing activity includes opening files, folders, and applications, and browsing the Internet as normal usage of the system. Once the capture has completed and the analysis begins, the Process Monitor filters will be used to determine the best filters for the final solution's implementation.

3.3.2 Testing Procedure 2

The purpose of Testing Procedure 2 is to determine if *Cloud Signature Creator* accepts input of a Process Monitor log and is able to properly calculate a Message-Digest 5 (MD5) hash value. In order to conduct this test, a sample set of data will be

created of a reasonable size to allow for hashing to be performed on each of the files in question with an external application. The testing procedure is as follows:

1. Start *Cloud Signature Creator*.
2. Open File Collection dialog.
3. Browse to sample Process Monitor log.
4. Enter Unique Identifier for this test.
5. Choose a save location.
6. Click “Start”.
7. When completed, view results.
8. Verify the hash values generated match those from the external application.

Once the results are available in the save location, the resulting files will be analyzed for accuracy. The MD5 values will be externally verified using AccessData’s FTK Imager to ensure that the hash values are being correctly calculated.

3.3.3 Testing Procedure 3

The purpose of Testing Procedure 3 is to verify that *Cloud Signature Creator* successfully and accurately removes duplicates from the collection phase be hash value. A sample set of data will be created of a reasonable size to allow for duplicate removal to be performed manually. The testing procedure is as follows:

1. Start *Cloud Signature Creator*.
2. Open File Collection dialog.
3. Browse to sample Process Monitor log.
4. Enter Unique Identifier for this test.

5. Choose a save location.
6. Click “Start”.
7. When completed, view results.
8. Verify that known duplicate entries are removed.

To be considered a success, this test will result in the removal of duplicate hash values from the resulting CSV file. Verification will be conducted manually on a sample set of data.

3.3.4 Testing Procedure 4

The purpose of Testing Procedure 4 is to determine that the Comparison interface correctly compares output from different runs of the collection phase. Two sample data sets will be created to allow manual comparison. The testing procedure is as follows:

1. Start *Cloud Signature Creator*.
2. Open Hash Comparison dialog.
3. Locate two sample data sets.
4. Choose a save location.
5. Input a filename for resulting CSV.
6. Click “Compare”.
7. When complete, view results.
8. Verify that all common hash values are reported.

A successful test will result in the resulting CSV file being a single list of hash values that are contained in both collection hash lists along with the file names associated with the hash values from both collection CSV files.

3.3.5 Testing Procedure 5

The purpose of Testing Procedure 5 is to determine if the *Cloud Signature Creator* solution is a viable solution for directing the user towards a set of specific files for further analysis in the suspected use of a particular cloud service used via the Internet. The testing procedure is as follows:

1. Start *Cloud Signature Creator*.
2. Start Process Monitor.
3. Perform test-specific operations.
4. Analyze results.

3.3.6 Testing Procedure 6

The purpose of Testing Procedure 6 is to determine if the *Cloud Signature Creator* solution is a viable solution for directing the end user towards a set of specific files for further analysis in the suspected use of a particular cloud service used via a locally installed client application. The testing procedure is as follows:

5. Start *Cloud Signature Creator*.
6. Start Process Monitor.
7. Perform test-specific operations.
8. Analyze results.

CHAPTER 4

FINDINGS

This chapter discusses the tests performed to validate the *Cloud Signature Creator* solution. Discussion will consist of results of testing procedures from Section 3.3 and dialogue on the strengths and limitations of the *Cloud Signature Creator* solution.

4.1 Testing Procedure 1 Results

This set of testing is based on Testing Procedure 1, from Section 3.3.1. After the testing procedure was completed, results of the Process Monitor capture were reviewed and analyzed. For every second of computer usage, there are hundreds of operations that Process Monitor reports. The most commonly occurring operations, of thirty-six operations, were CreateFile, WriteFile, CloseFile, and ReadFile. Based on the analysis of these results, it has been determined to filter in only entries with these operation types.

The Process Monitor user can filter based on process name, which will only show operations selected that were triggered by a specific process. For example, a Process Monitor filter for use with this solution and a cloud service being run in Internet Explorer would look like this:

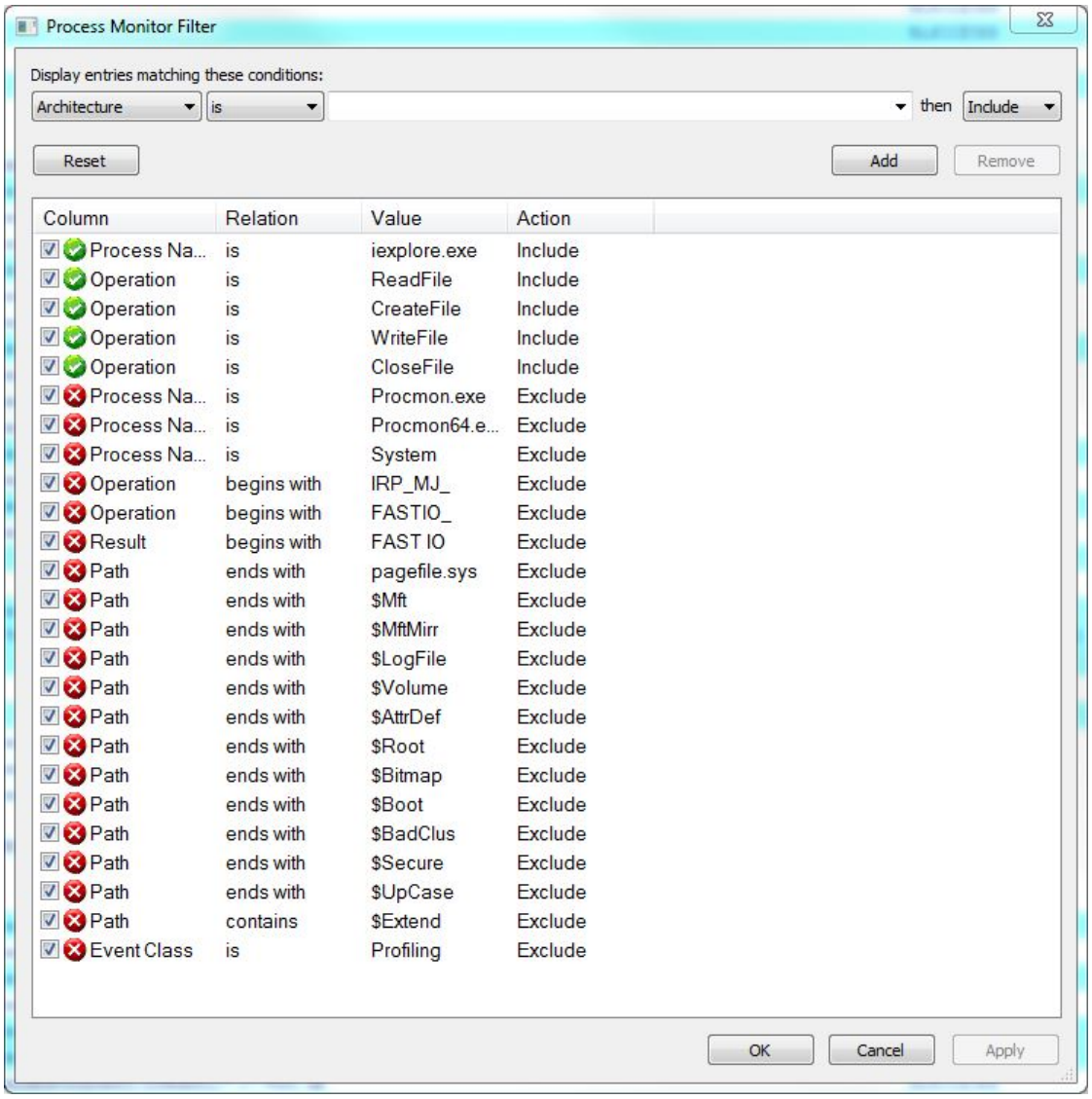

Figure 9 - Process Monitor Filter

In the Process Monitor Filter window, the green check marked rows are the filtering options that are to be included and the red check marked rows are the filtering options that are to be excluded from the final results. In this example, the above filtering resulted in a reduction of the number of events displayed to 13,196 from 272,883. This is only 4.8 percent of the total events, resulting in much less manual filtering and analysis from the end user.

Although this example filtered in a web browser, the user may have instead filtered in a client application for a particular cloud service. This would result in the events shown only being a result of actions taken by the cloud services application, rather than the web browser.

4.2 Testing Procedure 2 Results

This set of testing was intended to determine that the *Cloud Signature Creator* tool was properly calculating MD5 hash values. For this test, a sample set of twenty entries was created in the format of a Process Monitor log. These known files were located and hashed with the use of FTK Imager (FTK Imager version 3.4.0, 2015). The files were located using the path provided in the Process Monitor log and were added to an AD1 custom content image file. An AD1 custom content image is a container to store multiple files without altering their contents. Once the files were added to the image, the image was processed with FTK Imager and a hash list of the enclosed files was generated. After running the test, the hash list generated by the *Cloud Signature Creator* tool was compared with the generated hash values from FTK Imager. In order to compare these values, both sets were loaded into an Excel spreadsheet and compared using Excel's comparison functionality (Microsoft Knowledge Base). As a result of the comparison, it was determined that the *Cloud Signature Creator* tool correctly calculated the MD5 hash values for the files in question. The results were also manually verified.

4.3 Testing Procedure 3 Results

This set of testing was intended to determine that *Cloud Signature Creator* properly removes duplicate hash value entries in the file collection phase. For this test, a sample set of twenty entries, to include five duplicates, was created in the format of a Process Monitor log. The files were located using the path provided in the Process Monitor log and were added to an AD1 custom content image file. Once the files were added to the image, the image was processed with FTK Imager and a hash list of the enclosed files was generated. The hash values were loaded into an Excel spreadsheet and duplicates were removed using the Remove Duplicates function (Microsoft Office Support). After the test was conducted using the *Cloud Signature Creator* tool, the resulting list and the sample set with its duplication removed were loaded into an Excel spreadsheet and compared using Excel comparison functionality. As a result of the comparison, it was determined that the *Cloud Signature Creator* tool correctly removed duplicate hash values for the files in question. The results were also manually verified.

4.4 Testing Procedure 4 Results

This set of testing was intended to determine that *Cloud Signature Creator* properly compares the data from two CSV files and reports only the entries that report having that same hash value. For this test, a two sample data sets with twenty entries each were created. The entries contained a hash value and a file name. The files were located using the path provided in the Process Monitor log and were added to an AD1 custom content image file. Once the files were added to the image, the image was processed with FTK Imager and a hash list of the enclosed files was generated. Both data sets were loaded into an Excel spreadsheet and the columns were compared using

Excel's built-in comparison features. After conducting the test, the resulting list of hash values and file names were reported correctly. This conclusion was reached after loading the resulting hash list and the generated data set into an Excel spreadsheet and utilizing Excel's built-in comparison features. The results were also manually verified.

4.5 Testing Procedure 5 Results

This set of testing was designed to determine the viability of the *Cloud Signature Creator* solution in the investigation of cloud services by a forensic investigator. In this test a cloud service is accessed via Internet Explorer 11. Internet Explorer was used in testing because it stores its browsing data and caches files in a format that lends itself to parsing. For this test, a Drop box account was setup with a small amount of known files. One Word document, one JPEG image file, one Excel Spreadsheet, one PDF file, and one text document were uploaded to Dropbox prior to the start of the test. These known files were accessed and opened via the Dropbox web portal. The same steps for the acquisition of these files from Dropbox were used on two separate occasions for the file collection phase and then passed on to the hash comparison phase. After running the test procedure, the individual runs of the collection phase reported 484 and 453 hash values. These sets were run through the comparison phase and the resulting list of files in common between the two runs contained 434 files. Following the test runs, the two hash sets from the collection run were loaded into an Excel spreadsheet and compared using Excel's comparison features. The results confirmed those of the *Cloud Signature Creator* solution.

After an analysis of the results, I observed that some of the documents were able to be located in the file system after the testing was concluded and the web browser was shutdown. This includes the text document, the PDF file, and the JPEG file. The Word and Excel documents were not plainly observed, but evidence that a spreadsheet or document were viewed is present in the form of a JavaScript file that is setting the frame for the files. Contents were not located in the file set. The majority of the files found to be common across both collection phase runs were graphics and windows DLL files. These files could be used to confirm that the cloud service was utilized, but do not seem to contain any additional user data.

4.6 Testing Procedure 6 Results

This set of testing was designed to determine the viability of the *Cloud Signature Creator* solution in the investigation of cloud services by a forensic investigator. In this test, a cloud service is accessed via a locally installed client application. The cloud service being tested is Drop box. . For this test, a Dropbox account was setup with a small amount of known files. One Word document, one JPEG image file, one Excel Spreadsheet, one PDF file, and one text document were uploaded to Dropbox prior to the start of the test. These known files were accessed and opened via the Dropbox folder. In a locally installed client environment, Dropbox has a background process that monitors the usage of the Dropbox folder to determine if new files or changes need to be synced with its servers to provide the user with the most up-to-date data in all locations. The same steps for the acquisition of these files from Dropbox were used on two separate occasions for the file collection phase and then passed on to the hash comparison phase. After running the test procedure, the

individual runs of the collection phase reported forty-two and sixty hash values. These sets were run through the comparison phase and the resulting list of files in common between the two runs contained twenty-six files. Following the test runs, the two hash sets from the collection run were loaded into an Excel spreadsheet and compared using Excel's comparison features. The results confirmed those of the *Cloud Signature Creator* solution.

After an analysis of the results, I observed that many of the common files were system files. Additionally, several of the other files that were reported were some database files that would change had new files been modified, uploaded, or removed from the Dropbox folder. Note that the files being accessed are actually resident on the local computer, so their contents, if not encrypted, will be available in a Dropbox folder on the hard drive.

CHAPTER 5

DISCUSSION

5.1 Conclusions

The previous chapter presented the results collected from several tests that were outlined in Chapter 3. These tests were designed to address the goals laid out in Section 1.3 and the overall viability of the solution proposed to help state and local law enforcement investigators identify files and file locations that might contain information to further an investigation involving cloud-computing services.

5.1.1 Goal 1 Conclusions

The first goal of this project was to monitor file system changes that are a result of the cloud service usage. The Testing Procedure 1 Results, described in Section 4.1, discuss the findings of a test for file system monitoring and Process Monitor log filtering. The monitoring system utilized for this solution does a good job of tracking changes to files and folders in the file system. Additionally, the Windows Registry can be monitored with different operations than those that were selected as filtering options. The Windows Registry is a hierarchical database that stores data about the users and current configuration of a Windows system. With this solution, ignoring the registry was a decision that was made because of the fact that hash values were being utilized. Because the registry is always available while the system is running, and so many reads and writes are made to it, hash values of the registry are constantly changing. More useful to the *Cloud Signature Creator Solution* would be to

take note of exactly which registry keys and values are being written and read and to provide these to an investigator as a specific location of potential evidence.

5.1.2 Goal 2 Conclusions

The second goal of this project was to develop an application that receives input from the monitoring service and locates the files listed in a log file. Once the application locates a file, an MD5 hash value of the file is calculated, and duplicate hash values are removed from the application's report. While the MD5 value is calculated correctly, the hash value is only calculated if the file can be located after the collection run. Perhaps more important than the files that can be located in the current file system after the cloud service was used would be the files that are no longer able to be located. Most likely, these files have been removed or deleted by the cloud service cleaning up after itself when usage is completed. As a result, the files that are deleted are never reported by the *Cloud Signature Creation* application.

The duplicate removal is successful and does seem to be a helpful feature. If the same file is created, read, written, or closed more than one time, being reported that files hash value or location several times only contributes to the growing number of results for the end user to sift through.

5.1.3 Goal 3 Conclusions

The third goal of this project was to compare the hash lists from multiple monitoring instances to show files that are present in both instances. After the user performs two collection runs of with the *Cloud Signature Creator* solution, the resulting hash lists are compared to determine which hash values are common between

the two lists. The *Cloud Signature Creator* solution was successful at comparing these hash values, but the results of the comparison are interesting. The resulting hash list of common files are, generally, system files or graphics that might be loaded with a particular web page in the cloud service's usage.

More important to the end user are the file's with hash values that do not match across different collection runs of *Cloud Signature Creator*. These files might actually be the ones that contain the contents of files that were accessed through the cloud service or information about the user of a cloud service. All files that could be located and hashed are reported by the collection phase, so the investigator could look into the files and locations that are listed in that location. However, files that could not be located because they had been deleted are never actually reported to the investigator besides being in the Process Monitor log.

5.2 Future Work

The *Cloud Signature Creator* solution developed for this project was able to meet the goals described above. However, certain limitations exist in the solution that lends it to the possibility of additional work to improve the tool. The following sections discuss some of these areas of future work.

5.2.1 Monitoring System

Currently, the monitoring system is Process Monitor, which limits the use of this application to working only with Windows workstations. Additionally, the fact that this solution leverages an external tool for this aspect means that it relies on the support of the external tool for continued availability. If Process Monitor support is

discontinued, the *Cloud Signature Creator* solution is in jeopardy of discontinuation. Aside from the support being discontinued, if the Process Monitor log file format is adjusted, the *Cloud Signature Creator* log reading functions may not work correctly.

As future work to help sustain the usability of this solution, the monitoring system should be integrated into the *Cloud Signature Creator* application. As a secondary advantage to this work, integrating the monitoring system would allow for a more automated process that would require less user input when it comes to working with the cloud service.

5.2.2 File Hashing Algorithms

In an effort to advance the usage of this solution, hashing algorithms could be updated to include other algorithms, such as SHA-1 and SHA-256 algorithms. A more interesting adjustment to the file hashing aspect of the application would be the use of a fuzzy hashing algorithm (Hurlbut, 2009). Fuzzy hashing is a hashing method that hashes files in smaller sections, so as to be able to identify parts of files that are the same. In this case, it would be useful to know that two files are partially the same, indicating that maybe the rest of the file is session dependent and includes user data or date and time information.

5.2.3 Reporting

The reporting features of the *Cloud Signature Creator* application can be expanded to include some other data that would likely be useful to the investigator. One way to adjust the reporting of the current set of data given to the user would be to separate the graphic files and system files from the rest of the results to possibly show

the end user files that might contain actual data that could identify some type of evidence.

Additionally, Process Monitor is capable of monitoring changes to the Windows Registry. As briefly discussed in Goal 1 Conclusions above, these changes were ignored in this project. However, the registry values can be an invaluable source of information to an investigator. It would be extremely helpful to an investigator to be reported which specific registry keys were accessed or created as a result of the use of a cloud service or client application. With this data in hand, an investigator can greatly reduce time spent sifting through registry keys and values hoping to find information of use to their investigation.

Another possible change to the reporting would be to utilize the National Software Reference Library (NSRL) data set to reduce the hash values provided to the investigator by checking for known system files that are common on all computer systems and would have no information about the usage of the suspect cloud service.

Finally, the reporting of the *Cloud Signature Creator* application can include the reporting of files that were unable to be located and hashed due to them being deleted. These files might be useful to an investigator that could then attempt to carve the file's contents from unallocated areas of the hard drive. In order to do this, the investigator would also need some indication that these files might actually contain some information that is useful to their investigation, which the file's name and location may or may not provide them.

5.3 Conclusion

In conclusion, the *Cloud Signature Creator* solution was successful in meeting the goals described in Section 1.3. It has potential to be helpful in an investigation for state and local law enforcement when the suspect is known to be using a cloud service that is not supported by current industry tools. Although this project was successful at meeting its goals, there are some clear areas for improvement that will allow the solution to be more helpful to state and local law enforcement agencies.

BIBLIOGRAPHY

AccessData. (n.d.). *Forensic Toolkit (FTK)*. Retrieved November 11, 2015, from AccessData: accessdata.com/solutions/digital-forensics/forensic-toolkit-ftk

C# Programming Guide. (n.d.). Retrieved November 11, 2015, from Microsoft Developer Network: https://msdn.microsoft.com/en-us/library/67ef8sbd.aspx

FTK Imager version 3.4.0. (2015, March 16). Retrieved April 23, 2015, from AccessData: accessdata.com/product-download/digital-forensics/ftk-imager-version-3.4.0

Griffith, E. (2014, November 6). *Who's winning the consumer storage wars?* Retrieved April 23, 2015, from Fortune: http://fortune.com/2014/11/06/dropbox-google-drive-microsoft-onedrive/

Guidance Software. (n.d.). *Computer Forensic Software - Encase Forensic*. Retrieved November 11, 2015, from Guidance Software: https://www2.guidancesoftware.com/products/Pages/encase-forensic/overview.aspx

Hurlbut, D. (2009). *Fuzzy Hashing for Digital Forensic Investigators.* AccessData.

Koppen, J. (2012). *Cloud Signature: An Application to Detect Cloud-Computing Application Artifacts.* University of Rhode Island, Department of Computer Science and Statistics, Kingston.

Luttgens, J., Pepe, M., & Mandia, K. (2014). *Incident Response & Computer Forensics* (Third Edition ed.). McGraw-Hill Education.

Magnet Forensics. (n.d.). *Internet Evidence Finder*. Retrieved April 23, 2015, from Magnet Forensics: http://www.magnetforensics.com/mfsoftware/internet-evidence-finder/

Mell, P., & Grance, T. (2011). *The NIST Definition of Cloud Computing* (Vols. 800-145). NIST Special Publication .

Microsoft Knowledge Base. (n.d.). *How to compare data in two columns to find duplicates in Excel*. Retrieved November 11, 2015, from Microsoft Support: https://support.microsoft.com/en-us/kb/213367

Microsoft Office Support. (n.d.). *Filter for unique values or remove duplicate values*. Retrieved November 11, 2015, from Microsoft Office Support: https://support.office.com/en-us/article/Filter-for-unique-values-or-remove-duplicate-calues-ccf664b0-81d6-449b-bbe1-8daaec1e83c2

Microsoft Visual Studio. (n.d.). *Home Page*. Retrieved April 23, 2015, from Visual Studio - Microsoft Developer Tools: https://www.visualstudio.com

Russinovich, M., & Cogswell, B. (2014, March 7). *Process Monitor v3.1*. Retrieved April 23, 2015, from Microsoft TechNet: https://technet.microsoft.com/en-us/library/bb896645.aspx