

SCHOOL OF COMPUTING

UNIVERSITY OF TEESSIDE

PRAGUE COLLEGE

“NTFS MASTER FILE TABLE VIEWER SOLUTION”

[Saraa Dagvatseren]

Supervisor: Bohus Ziskal

Second reader: Filip Sedlak

Abstract

A computer security incident response team (CSIRT) solves computer security incidents within an organization. CSIRT uses information from computer to detect and respond to security incidents. Whenever a security incident such as malware delivery attempt, intrusion attempt, phishing, exploit attempt or malware infection on a computer, the CSIRT response analysts are analysing the alert and are challenged to gather evidence to support their analysis in a security investigation. A security investigation is a process where the security analyst develops and test hypotheses that answers questions about the digital events. Digital evidence is any type of digital object that holds information that support or contests a hypothesis raised by the analyst.

The current project aims to solve an existing problem within a CSIRT here referred as client by researching the security investigation methods to find the best design solution and design a tool to view digital evidence - Microsoft NTFS MFT table. The report is supported by the project requirements, design, testing, implementation, and evaluation of the client specific problem. The report and the tool are not intended for any other organization other than the client company.

TABLE OF CONTENTS

ABSTRACTII

TABLE OF CONTENTS III

FIGURES LIST V

CHAPTER 1: INTRODUCTION 6

1.1 BACKGROUND 6

1.2 THE CLIENT 6

1.3 THE PROBLEM 7

1.4 PURPOSES 7

1.5 SCOPE 7

1.6 THESIS OUTLINE 8

CHAPTER 2: REQUIREMENTS ANALYSIS 10

CHAPTER 3: METHODOLOGY 13

3.1 AGILE METHODOLOGY 15

CHAPTER 4: RESEARCH16

4.1 MFT BACKGROUND RESEARCH AND USES 16

4.1.1 MFT General Usage 17

4.2 REVIEW OF CURRENT PROCESS 17

4.2.1 INTERNAL MFT USE CASES 19

4.3 REVIEW OF EXISTING TOOLS 19

4.4 RESEARCH ON POTENTIAL TECHNOLOGIES 20

4.5 RESEARCH ON DATA STRUCTURE 21

CHAPTER 5: DESIGN 23

5.1 DESIGN DECISIONS 24

5.2	SOLUTION FUNCTIONALITIES	25
5.3	LEGAL, ETHICAL AND PROFESSIONAL ISSUES	25
CHAPTER 6:	IMPLEMENTATION	26
6.1	PARSING DATA	27
6.2	FILE DIRECTORY TREE VIEW	27
6.3	TABLE VIEW	28
6.4	ISSUES ENCOUNTERED DURING DEVELOPMENT	28
CHAPTER 7:	TESTING	30
CHAPTER 8:	EVALUATION	33
8.1	PERSONAL EVALUATION	33
8.2	FURTHER DEVELOPMENTS AND RECOMMENDATIONS	34
CONCLUSION		35
REFERENCES		36
APPENDIX A – GATHERING REQUIREMENTS		37
APPENDIX B – USER TESTING FEEDBACKS		41
APPENDIX C – PROJECT SPECIFICATION		43

Figures List

Figure 1: Waterfall Methodology (Dennis, Roth and Wixom, 2012)	13
Figure 2: Agile Iterative Development	15
Figure 3: MFT entry example	16
Figure 4: Parsed MFT csv file	19
Figure 5: Wireframe design proposal 1	23
Figure 6: Wireframe design proposal 2	24
Figure 7: Client approved final design wireframe	25

Chapter 1: Introduction

According to EC-Council (2010, p.13) disk file systems are used to store and recover on a storage device, such as hard disk, that is directly or indirectly connected to a computer. The disk file systems allow computers to locate the data into a hierarchical structure through files directories. Different operating systems manage these processes differently, this project will focus only on Microsoft New Technology File System (NTFS).

The NTFS volume holds a unique metadata file called Master File Table (MFT), MFT store a record of every file and folder on the volume. MFT is composed of entries that holds information about the file such as size, timestamps, and content. When the size of the NTFS volume is increased, the MFT is also growing. However, when a file is deleted from the disk drive, the MFT holds the entry that mentions that the file is deleted. Further research about MFT metafile has been done and is discussed in the Research chapter.

1.1 BACKGROUND

The author of this project work as a cyber-threat response analyst within the client computer security incident response team (CSIRT) in Czech Republic. The author of the project wanted to choose a project that will contribute to the teamwork by solving an existing problem. To identify a suitable project, the team manager here referring as the client representative and few other team members were interviewed, the developer suggested that developing a custom tool to view Master File Table (MFT) will be highly beneficial to the team thus this project was chosen.

1.2 THE CLIENT

The client of this project is Network Information Security Department, this team works as cyber security response team by identifying external and internal threats through monitoring tools and alerts. The future user of tool is current going to be the team which is composed of 10 analysts including the author of this report and the team lead. The analysts have different seniority, among them 5 are level 2 analysts (L2) and 5 are level 3 (L3) analysts. The client

representative stated that previously there was an attempt of developing an internal script to view MFT data however the project was not put on hold due to upcoming system upgrades.

1.3 THE PROBLEM

When investigating security incident, the security analysts have to gather evidence to support their analysis by collecting data from various sources. The client is challenged with internal policies restrictions where online or external non-approved tools are not allowed to be used. Depending on the security incident type, the analyst must view the \$MFT metadata to validate their investigation. To gather \$MFT metadata, the analyst acquires the data remotely from the user's machine using an internal Data Loss Prevention (DLP) system that monitors the users globally. The client DLP system is designed to monitor and stop malicious activities on endpoint devices.

The acquired metadata is already half-parsed from the machine format and delivered into unorganized comma-separated value (csv) file. The data delivered by the DLP is messy and needs more processing to be human-readable since the MFT data is ubiquitous and complex to interpret. The client is currently using inconvenient process and tools to view this MFT csv data. Further discussion about the current process used within the client team is available in the Research chapter.

1.4 PURPOSES

The main purpose of this project is to deliver a custom tool to view the csv format MFT metadata into a readable format that will help the security analysts in their investigation by facilitating the current process. A secondary purpose is to develop an adequate internal process to use the developed MFT viewer tool to the client. The personal objectives of the author are to contribute to the team cooperation, to improve project development skills and explore different techniques of security incident investigation from experienced security professionals.

1.5 SCOPE

The project focuses on delivering a suitable solution to the client by solving the initial given problem. The solution to be delivered in this project is to be used only by the client and is not

suitable to be used outside the client as the solution was designed specifically for the client MFT data. To validate that the delivered solution is solving the initial problem, the solution is tested with the client team. And based on the test results, improvements and adjustments are discussed in chapter 8.

First, the author conducted individual interviews with the different team members to gather the project requirements. Then, the author conducts appropriate research and based on the research and the gathered requirements, the solution is designed. Once the design was approved by the client, the solution is developed and tested. Lastly, the solution is evaluated by both the client and the author. The project excludes discussions about advanced forensics techniques as the client is security incident response team and not forensics team. The constraints of the project are the implication of client internal data and process and the time restriction.

1.6 THESIS OUTLINE

Chapter 1: Introduction provides an overview of the project, its background, purposes, and scope.

Chapter 2: Requirement analysis summarizes the project functional and non-functional requirements.

Chapter 3: Methodology is about the method used in this project and the justification about the choice of the waterfall methodology.

Chapter 4: Research discusses all the research done for this project starting with background research about MFT, the current process of viewing MFT within the client site, the different existing tools, complex data representation and future potential technologies.

Chapter 5: Design is about the proposed design for the tool and justification about the choice of the design.

Chapter 6: Implementation describes how the solution was programmed and what were the issues encountered during the development.

Chapter 7: Testing summarizes all the testing done with the client and its results.

Chapter 8: Evaluation assess the overall project outcome by reviewing what was initially promised when the project started and what was delivered. It also reviews the quality of the overall deliverable and discuss the future improvement recommendations.

Chapter 9: Conclusion discusses the summary of what was done in the project as well a discussion whether the goal of the project was met.

Chapter 2: Requirements Analysis

This chapter discusses the requirements gathered from the client. Since the project is solving an existing problem, the requirements were collected from the client based on the problem. Both functional and non-functional have been collected from the client – team member by organized one-to-one interviews. There is participation of 6 personnel out of 10 analysts where 5 analysts are L2, and 1 analyst is L3. These requirements we gathered during multiple discussions when the analysts were present in the office or when they were available. There hasn't been a meeting organised with all the members of the team due to the team agenda as some of the analysts work remotely. Though, the first meeting to approve the project has been organised by the manager of the team, the following questions have been discussed:

1. Q: When you first mentioned the project, you said previously someone from the team tried to develop an internal tool to view MFT data, can you please explain more about it.

A: Yes, it was part of a bigger project to improve the evidence gathering process. It was planned to be implemented within the Data Loss Prevention (DLP) system that we use to acquire the MFT. However, the company is planning to change DLP system, so it was on hold as of now and no one is currently working on the project. So, you can work on the development of MFT viewer tool.

2. Q: What are your requirements for the solution?

A: As I am not technician, I cannot stay the format of the tool, but I think a simple tool that is compatible with our different environments and easy to use will be good.

3. Q: I heard couple of times from the team about the bad quality of MFT data acquired from the DLP system, can you please provide more information on this?

A: Yes, this happens as you know there is no perfect system sometimes, they are discrepancy. This is related to the DLP MFT parser script. We had occasions where some data were missing from the MFT or that host machine was not available for data acquisition. We are aware of this issue, and we previously raised service request to responsible team to fix this issue.

4. Q: As the project is involving internal MFT data, how to deal with internal sensitive data?

A: You can use internal acquired MFT used as evidence in previous security incidents on your company laptop, but you are not allowed to use them outside the company assets. We can agree on the number and details when you are developing.

5. Q: What design are you expecting for the solution? A desktop application, web application, etc.?

A: I don't have anything planned you can discuss with the rest of the team. I was thinking of something very basic and light. I don't see a complex web application with server for our team because it will require the implication of other teams and the resources.

Then, a set of questions were prepared for interviewing each individual member of the team. Below is the list of questions asked during these interviews, a full transcript of the full individual interviews report is available in appendix A. Part of these interview results were also used in the research phase for the research on the internal process.

- In what case do you need to analyse MFT artefact?
- What process do you use to view the MFT artefact?
- What is the most important for you analysing the MFT metadata?
- What kind of solution is easier for you? Getting suggestions about the format of the solution.
- What kind of functionalities do you need in the solution?
- What kind of functionalities would you like to see in the solution?
- Based on these interviews, the project requirements were produced by the developer which was a combination of the merged requirements collected from individual team members. The below table represents the final project requirements which was approved by the client.

Project Requirements	
Functional requirements	User can upload an existing MFT csv file through an upload button.
	User can see all the MFT entries in an organized table where the MFT attributes are headers and each MFT entry is a row.
	User can search for specific filename to find the MFT entry.
Non-functional requirements	A light tool offering accessibility supported by various operating systems.
	A tool that is going to be handed to the client without licensing.
	A tool that does not send data to external server.

Chapter 3: Methodology

This chapter contains a through description of the methodology used for developing the project. The first challenge of the project was to choose the appropriate development methodology, this task is not easy as there is no methodology that is the best. For this project, the author decided to use the traditional waterfall development. The traditional linear-sequential life cycle model also known as waterfall development methodology. Hughey, D. (2009) states that this method was originally defined by Winston W. Royce in 1970.

The waterfall development model suits the project as the project steps are sequential which allow a traditional approach for such tool. Each phase of development is going to be developed one after other offering a great structure to the project.

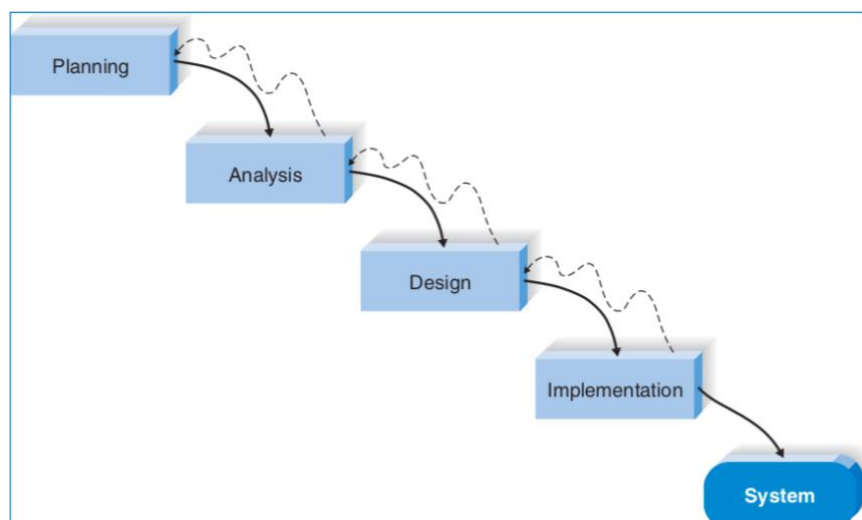


Figure 1: Waterfall Methodology

The waterfall method “makes assumption that all the requirements can be gathered up front during the requirements phase”. (Hughey, 2009). The justification of choosing the waterfall approach are as follows:

- The clarity of the requirements is clear at the beginning of the project.
- Design errors are seen prior the implementation which save time.

- The approach is very structured which helps the developer going with the progress of the project.
- Since the objective of the project is clear, the developer can work independently expect for review or approval of visual design.
- Schedule visibility, since the project had a schedule, the waterfall methodology offers great visibility on whether the project is going as planned.

The key disadvantage of waterfall methodology is that the design should be completely specified before the design begins. (Dennis, Roth and Wixom, 2012). Another disadvantage is that the methodology is not very flexible and does not allow any changes once a phase is completed.

The methodology has two variants, the parallel development, and the V-Model. For this project, the author decided to use the traditional waterfall development without choosing any variant. A waterfall method was essential for the project development, as all the requirements are predefined by the client. Much effort and time were spending during the early phases of the project to define exactly the requirements and the design because the implementation phase strongly depends on them.

Since the developer work with the client, the first meeting was organized to gather the general requirements for the viewer from the client representative. The objective of the requirement gathering was to determine what the tool should do. Once the requirements are gathered from the client, the design phase started to define technical requirements to select which technology to use, which programming language to select, and the tool is going to look like. This phase also includes the research done that will support the design decisions.

The implementation phase is when the tool is programmed and where all the previously predefined decisions and requirements. The testing phase begins when the tool is fully developed, and it is used to verify that the project is meeting the client expectations. During this phase, the testers discover issues and reports them which are later resolved by the developer. And lastly, in the system phase also known as maintenance phase is where the application is deployed.

3.1 AGILE METHODOLOGY

An agile methodology could have been chosen however the agile software development methodologies are more focused on adaptive planning where every phase is done iteratively. “The output of each iteration is a version that respond to the evolving user requirements”. (Gibson, 2011). Agile software development consists of many different methods helping project teams collaborate in efficient way with emphasis on adaptive planning, client collaboration and unique development. The above figure illustrates, the iterative approach where every iteration is a complete project. Even though, agile methods adapt easily to changes the developer decided to choose the waterfall methodology as the requirements are clear since the beginning. Also, as the developer is unexperienced with project development a traditional structure approach was chosen.

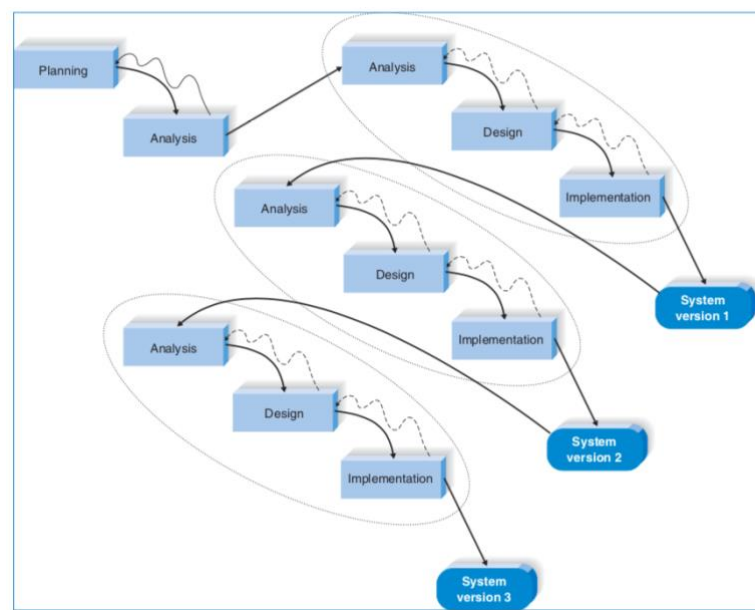


Figure 2: Agile Iterative Development

Chapter 4: Research

This chapter summarizes the completed research done for this project and their results. The goal of this chapter is to demonstrate that appropriate research has been done. Prior starting the project, the developer had identified the research areas separately. First, the developer did background research about the current process used within the team, in-depth research about MFT and its use cases, some complex data representation techniques, the existing tools, and potential technologies that can be used.

4.1 MFT BACKGROUND RESEARCH AND USES

Further research on the NTFS MFT was done by the developer to better understand what kind of information MFT holds and what analytical questions MFT artefact answer. By doing such research about MFT, it allowed the developer to have a better understand of the complex data and to identify what are the information needed from the MFT. The entry is 1024 bytes in size which 42 bytes are used for headers and 982 bytes left to store the file metadata. (Shaaban and Sapronov, 2016, p.101). MFT have different attributes that stores the metadata of file. The attribute can be either resident where all the data is within the MFT or non-resident where the MFT data some attributes are stored in the data attribute due to size limitations.

```
00C7FF3400 46 49 4C 45 30 00 03 00 43 F3 18 9B 0B 00 00 00 FILE0...C6.1...
00C7FF3410 59 00 02 00 38 00 03 00 78 02 00 00 00 04 00 00 Y...e...x...
00C7FF3420 00 00 00 00 00 00 00 00 09 00 00 00 29 00 00 00 .....
00C7FF3430 8C 06 00 00 00 00 00 00 10 00 00 00 60 00 00 00 .....
00C7FF3440 00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00 .....H...
00C7FF3450 22 3D CA 9D CA BB C4 01 14 E3 A4 11 35 24 C5 01 "=EIE>A..34 5&A..
00C7FF3460 14 E3 A4 11 35 24 C5 01 7D 28 9A 09 94 44 C5 01 .84 5&A..)(.IDA..
00C7FF3470 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00C7FF3480 00 00 00 00 29 03 00 00 00 00 00 00 00 00 00 .....
00C7FF3490 F0 37 13 29 00 00 00 00 30 00 00 00 70 00 00 00 87.)...0...p...
00C7FF34A0 00 00 00 00 00 00 04 00 52 00 00 00 18 00 01 00 .....R...
00C7FF34B0 0A E4 00 00 00 00 03 00 22 3D CA 9D CA BB C4 01 .a....."=EIE>A..
00C7FF34C0 22 3D CA 9D CA BB C4 01 22 3D CA 9D CA BB C4 01 "=EIE>A..=EIE>A..
00C7FF34D0 22 3D CA 9D CA BB C4 01 00 00 00 00 00 00 00 00 "=EIE>A.....
00C7FF34E0 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 .....
00C7FF34F0 08 02 43 00 4F 00 45 00 4E 00 33 00 35 00 7E 00 ..C.O.E.N.3.5.~.
00C7FF3500 33 00 30 00 35 00 00 00 30 00 00 00 70 00 00 00 3.0.5...0...p...
00C7FF3510 00 00 00 00 00 00 03 00 56 00 00 00 18 00 01 00 .....V...
00C7FF3520 0A E4 00 00 00 00 03 00 22 3D CA 9D CA BB C4 01 .a....."=EIE>A..
00C7FF3530 22 3D CA 9D CA BB C4 01 22 3D CA 9D CA BB C4 01 "=EIE>A..=EIE>A..
00C7FF3540 22 3D CA 9D CA BB C4 01 00 00 00 00 00 00 00 00 "=EIE>A.....
00C7FF3550 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00 .....
00C7FF3560 0A 01 63 00 6F 00 65 00 6E 00 33 00 35 00 30 00 ..c.o.e.n.3.5.0.
00C7FF3570 5F 00 30 00 35 00 00 00 40 00 00 00 28 00 00 00 _0.5...@...(.
00C7FF3580 00 00 00 00 00 00 08 00 10 00 00 00 18 00 00 00 .....
00C7FF3590 0B 8D 0A D9 1E 90 D9 11 B9 08 00 0D 56 08 E4 DB .1.0.10.1...V.a0
00C7FF35A0 90 00 00 00 58 00 00 00 00 04 18 00 00 00 07 00 .....X...
00C7FF35B0 38 00 00 00 20 00 00 00 24 00 49 00 33 00 30 00 8...$.I.3.0.
00C7FF35C0 30 00 00 00 01 00 00 00 00 10 00 00 01 00 00 00 .....
00C7FF35D0 10 00 00 00 28 00 00 00 28 00 00 00 01 00 00 00 .....
00C7FF35E0 00 00 00 00 00 00 00 00 18 00 00 00 03 00 00 00 .....
00C7FF35F0 00 00 00 00 00 00 00 00 A0 00 00 00 50 00 8C 06 .....P.I..
```

Figure 3: MFT entry example

Above figure is an example of an MFT entry on the left and its decoding on the right, each entry is exactly 1 KB in size and is addressed in 48-bit system. The MFT even has an entry for itself. (Carrier, 2005). Carrier (2005, p.200) compares the MFT entry to a large box that stores all your possessions, the outside of the box holds basic information such as the name or the address and the inside of the box is initially empty, but it can be used to store all kind of information as long as it is smaller than the box itself. This means that the MFT has no structure, and it contains several attributes that contain specific information. The first entry to any MFT is a standard entry string "FILE" as highlighted in red on the figure 3. If an entry is erroneous, the first string will be "BAAD". If a file attribute does not fit into a single entry, it is using multiple entries. Further discussion about the client produced MFT data structure is discussed in the next sub-chapter 4.5 - data structure.

4.1.1.1 MFT General Usage

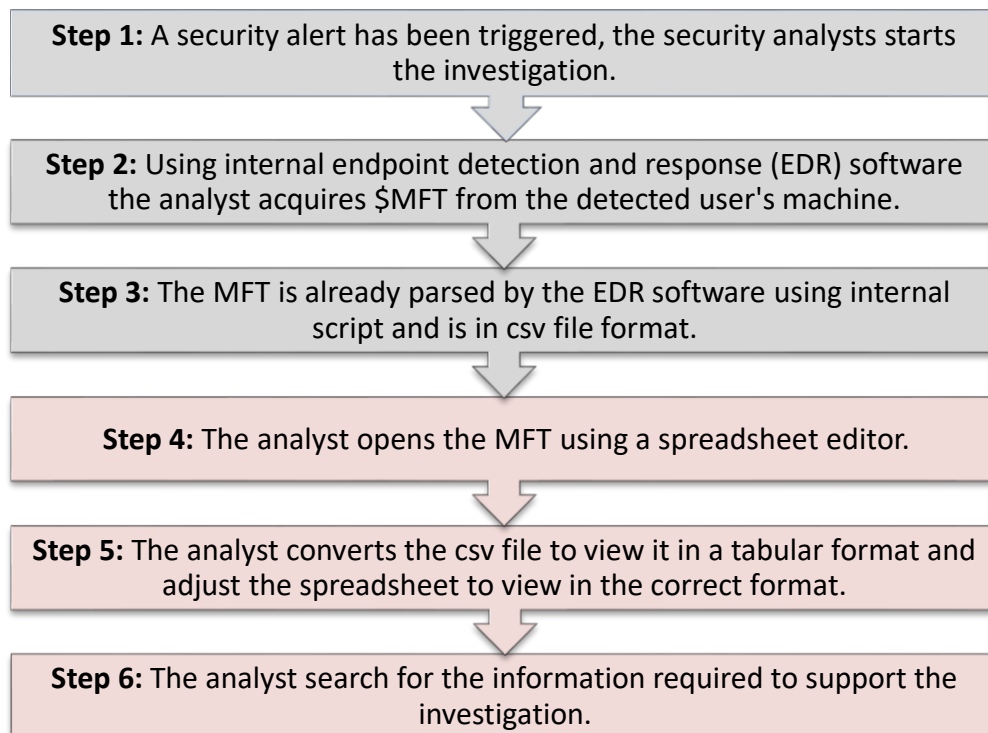
Below is a list of the analytical questions that can an MFT metafile answer for a cyber-security incident investigation:

1. What are the timestamps modified or accessed or created of a specific file or directory? NTFS has 8 timestamps divided into two groups: Standard Information Attribute (\$SI) and the Filename Information attribute (\$FN).
2. What other files or directories reside at a particular path?
3. Is the file or directory still present on the system?
4. Was the file or directory on the system even though it is not present any longer?
5. What is the file path of a particular file?
6. The client use cases of MFT metadata has been collected during the requirement gathering interviews and discussed in the next sub-chapter – Review of current process.

4.2 REVIEW OF CURRENT PROCESS

The client current process to acquire and view the MFT from a workstation within the company has been discussed in this section. The client use case of MFT data have been also prepared based on the previously conducted one-to-one interviews. The current

process used within the client team of acquiring and using the MFT has been documented below.



While collecting the current process, few flaws have been identified. Often, when investigating security incidents, the analysts are required to use internal forensic lab virtual machines and record the process. And most of the time, these machines don't have a spreadsheet software such as Excel or Libre-office pre-installed. Another disadvantage of such process is the number of adjustments and commands done in the spreadsheet editor to view the csv file into the desired correct format such as converting the text into column by delimiting the text by tab, wrapping the text to fit the column size, and applying filters to filter into an ascending order. By identifying the flaws of the current process, it has enabled the author to acquire a deeper understanding which will help to solve the problem and deliver a better process. Generally, the size of an internally parsed MFT csv file is around 100 MB. The above example figure illustrates the structure of how the MFT from the internal DLP system looks like.

```

RecNo Deleted Directory ADS Filename siCreateTime (UTC) siAccessTime (UTC) siModTime (UTC) siMFTModTime (UTC) Ac
0 0 0 0 $MFT 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 373293056 373293056
1 0 0 0 $MFTMirr 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 4096 4096 \SMF
2 0 0 0 $LogFile 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 67108864 67108864
3 0 0 0 $Volume 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 0 0 \Volume 2018
4 0 0 0 $AttrDef 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2560 4096 \SAT
5 0 1 0 . 2016-02-23 02:56:19 2018-06-04 07:34:16 2018-05-23 11:35:35 2018-05-23 11:35:35 3072 4096 \ 2018-04-24
5 0 1 1 .:DG1_DS_DIR_HDR:2016-02-23 02:56:19 2018-06-04 07:34:16 2018-05-23 11:35:35 2018-05-23 11:35:35 3072 4096
5 0 1 1 .:DG1_DS_DIR_HDR:DG1_DS_VOL_HDR:2016-02-23 02:56:19 2018-06-04 07:34:16 2018-05-23 11:35:35 2018-05-23 11:
6 0 0 0 $Bitmap 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 15089472 15089472
7 0 0 0 $Boot 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 8192 8192 \SBoot
8 0 0 0 $BadClus 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 0 0 \SBadClus 20
8 0 0 1 $BadClus:$Bad 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 494451814400
9 0 0 0 $Secure 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 \SSecure 2018
10 0 0 0 $UpCase 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 131072 131072
10 0 0 1 $UpCase:$Info 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 2018-04-24 03:54:34 32 32 \S
11 0 0 1 $Extend 2018-04-24 03:54:34 2018-06-04 07:21:04 2018-04-24 03:54:34 2018-04-24 03:54:34 \SExtend 20
24 0 0 1 $Deleted 2018-04-24 03:54:35 2018-06-03 14:28:45 2018-04-24 03:54:35 2018-04-24 03:54:35 \SExtend\

```

Figure 4: Example of internal MFT csv structure

4.2.1.1 INTERNAL MFT USE CASES

The client uses MFT data mostly for finding missing data and completing their analysis. MFT data is not often required to be consulted however for few security incident types it is needed. Below is the list of uses cases:

- To find the complete file path or location of the file.
- To identify if the file is still present on the workstation.
- To see what other files resides in a particular folder.
- To see what other files / directories were created at the time of detection of an incident.

4.3 REVIEW OF EXISTING TOOLS

While researching, it was observed that there exists currently several MFT parsing tools that are used to extract the original \$MFT from the operating system however, such tool to view partially parsed MFT data does not exist. The below table compares and review the MFT related existing tools.

Name	Functionality	Limitations
AnalyzeMFT.exe	Windows command line \$MFT parser tool.	It is a parser only to get the first \$MFT. Delivers organized data into csv file format.
Redwold MFT GUI	Windows GUI based MFT parser which allows to search by time analysis.	\$MFT parser that get the initial \$MFT and present the result into a very basic user interface. No longer in development.

Be CSV	Online csv viewer tool where the user can upload any csv files and it display the data into tabular format.	Allow to view the MFT csv data into a table but as data is internal data, the client cannot use such external tool.
Online CSV Editor and Viewer	Online csv viewer and converter where the user can upload a csv to view into a table.	Allow to view the MFT csv data in to a table but as data is internal data, the client cannot use such external tool.
Timeline Explorer	Tool to view mactime and Plaso generated CSV timelines without the need to use Excel.	Tool not designed for internally processed MFT.

None of the similar tools available are doing what the client would like to do. They are all used as parser to get the first \$MFT file from the system.

4.4 RESEARCH ON POTENTIAL TECHNOLOGIES

This part is describing the research done on the web development technologies prior the design and implementation phase. The purpose is to find potential software development methodologies, existing libraries or programming languages as the developer does not have solid knowledge on web development. This part is complimentary to the design phase.

Popular web development technologies such as HTML5, CS3, JavaScript and JavaScript libraries were researched. HMTL5 is used to structure the content of the web application, CSS3 is used to design, and JavaScript is used to parse the csv metadata to display them in the desired format. As the developer did not have much experience in JavaScript, few JavaScript libraries have been explored to find what the options were and estimate the amount of programming effort that was going to be done. Below is the list of popular technologies researched.

- VBA scripting is often used within the client team for scripting macros in excel and it is mostly used for recurrent calculation for reporting.
- jQuery is an open-source feature-rich JavaScript library.
- Papa Parse library– in-browser open-source csv parser for JavaScript. Papa Parse is different from csv parsing libraries because it does data processing locally on the client side without sending data over internet. It supports all modern browsers.
- Handsontable library is a JavaScript data grid spreadsheet component. There is a non-commercial free version available.
- FancyTree is a JavaScript tree view open-source plugin that support most modern internet browsers.
- JsTree is jQuery open-source plugin that supplies file tree structure.
- W2ui is an open-source JavaScript User Interface library that provides widgets like layout, grid, or sidebar.

4.5 RESEARCH ON DATA STRUCTURE

This research part was conducted with the objective of understanding how the internally generated csv is structured. To manipulate the data appropriately a deep understanding of how the initial data is organized is crucial. Out of the 10 different MFT analysed, it was observed that the average number of rows was 350,000 and the average size is around 130 MB. The MFT data contains alphanumerical data. The first line of the csv contains the attributes of the MFT with the following order:

	Attributes name	Description	Example data
1	RecNo	File record number.	0
2	Deleted	Deleted flag says if the file is still present on the machine.	0
3	Directory	Directory flag says if the file is a directory.	0
4	ADS	Alternate data stream flag.	0
5	Filename	File name	\$MFT

6	siCreateTime (UTC)	Standard Information create time.	24.04.2018 3:54:34
7	siAccessTime (UTC)	Standard Information access time.	24.04.2018 3:54:34
8	siModTime (UTC)	Standard Information modified time.	24.04.2018 3:54:34
9	siMFTModTime (UTC)	Standard Information MFT modified time.	24.04.2018 3:54:34
10	ActualSize	Logical size on the disk.	373293056
11	AllocSize	Physical size on the disk.	373293056
11	Ext	Extension of the file.	Hmtl
12	Fullpath	File path of the file separated by \	\\$MFT
13	fnCreateTime (UTC)	File Name create time.	24.04.2018 3:54:34
14	fnAccessTime (UTC)	File Name access time.	24.04.2018 3:54:34
15	fnModTime (UTC)	File Name modified time.	24.04.2018 3:54:34
16	fnMFTModTime (UTC)	File Name MFT modified time	24.04.2018 3:54:34
17	ReadOnly	Read-only flag indicates if the file is read-only.	0
18	Hidden	Hidden flag says if the file is hidden.	1
19	System	System flag.	1
20	Hostname	Hostname name.	localhost

When analysing the csv file, it was observed that some data might have been missing and when interviewing the client about the issue. The client is aware of this issue and when the data is incompletely parsed from the user's machine the security analysts find alternative solution to find the missing information using other available logs or tools. For example when a file path attribute data is missing, the analyst can run an inclusive search with the rest of the information through the DLP logs.

Chapter 5: Design

The following chapter provides the details about the design of the solution based on the previous chapter. The objective of this phase is to design an appropriate solution. Initially, the developer had proposed to the client two different solutions illustrated by the below wireframe figures. These two distinctive designs were proposed so that the client can chose a design that fits best his expectation. The reason behind the decision to propose two design was that when the developer was gathering the requirements and researching about the current process, it was observed that the team members had a different way of viewing the MFT data. The first proposed design solution is an offline standalone web application.

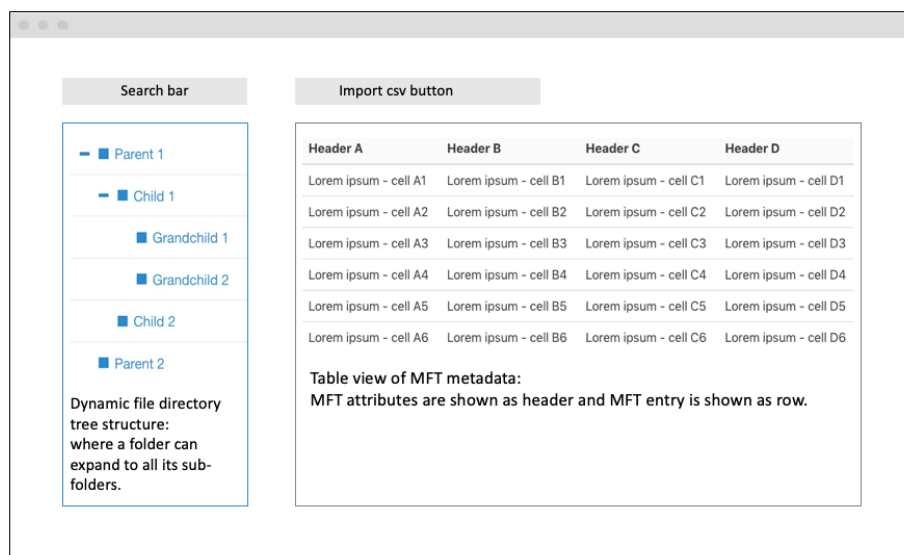


Figure 4: Wireframe design proposal 1

The web application will be used offline, it was inspired by Cyberchef tool, a tool that the client team is using offline for decoding malicious script and is “fully portable and can be downloaded locally as a simple HTML self-contained page that can run in any browsers.” (Bruneau, 2017). The team also uses similar self-contained web application which are internally developed for various tasks such sandboxing. When using the web application, the user is able to perform the following functionalities:

1. Upload the MFT data with a button upload feature.

2. The data will be then displayed into tabular form on the right.
3. The file directory structure will show the file structure and the user will be able to display the files.

The second proposed design is an excel macro plugin that will convert the csv file into a table using VBA script. The commands done by the user is automated in the macro, so the user has just to run the macro script.

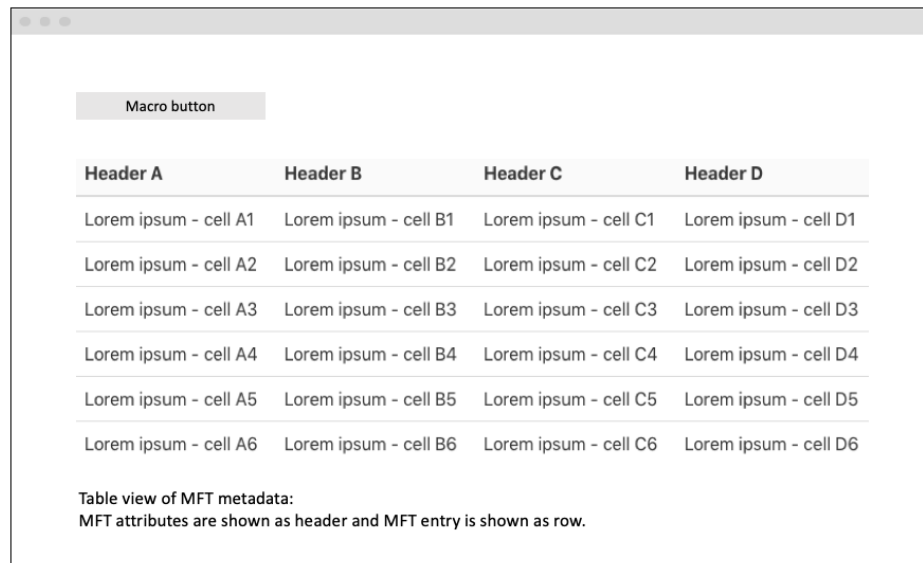


Figure 5: Wireframe design proposal 2

The macro add-in an improvement suggestion of the existing process rather than a new tool. Using this design, the user is able to:

1. Upload the MFT csv file.
2. View all the MFT entries into the table.
3. Search for functionalities
4. Ability to use all the rest of excel functionalities.

5.1 DESIGN DECISIONS

It was challenging to design a tool that satisfy every team member expectation and needs. The client had to decide which design to choose, the manager of team chose the first proposed solution design because it is meeting the project requirements and it is meeting more meeting needs. The main requirement of the client was to produce a simple portable solution which

the web application is meeting by allowing the client to not be depended on a third-party software. The MFT viewer will a browser-based application that can be run from any internet browsers offline. The web application will be using HTML5, CSS3, JavaScript, jQuery, and open-source JavaScript libraries. As the web application is intended to work on web browsers, the application is going to be developed on two different web browsers, Google Chrome, and Mozilla Firefox latest to ensure that the application is made. The client does not need a web server as the tool is going to be used internally, the client will open the HTML file in the browser to run the application.

5.2 SOLUTION FUNCTIONALITIES

The client approved the functionalities of the solution, it will have the below functionalities and each of them are addressing a functional requirement.

- Upload the MFT file into the web application.
- Present the MFT data into a table, the headers will have the MFT attributes, and each row is going to be an MFT entry.
- Search functionality to search specific filename.
- View the file structure by folders and each folder is expandable to view its sub-folders if holding any.
- When selecting a folder, the MFT entries are updated in the table displaying only the entries within the folder.
- A final meeting with the client representative was organised prior the coding phase to get all the design approved.

5.3 LEGAL, ETHICAL AND PROFESSIONAL ISSUES

The tool is intended for internal usage only thus the licensing will be fully given to the client. Concerning, the usage of programming libraries, the developer focuses on using open-source free libraries. Any internal sensitive data used during the development and testing phase will not be compromised and will be handled under the client's internal data protection policies as well as the European Data Protection Regulation. It was agreed with the client that a specimen internal MFT is going to use for development phase. The client will use 10 internal MFT files for testing purpose only. While designing and developing this project, Teesside University research ethics are followed.

Chapter 6: Implementation

This chapter describes the implementation phase of the project, this is where the solution is programmed. As a waterfall methodology was being used, the implementation started once the design phase had finished and the design was selected. Since the tool is going to be a standalone offline web application there is no server-side technologies used. The structure of the web application is using the standard web page architecture design. HTML5 and CSS3 mark-up languages are used to for the structure and the layout. JavaScript language is used to code the functions that parse all the MFT data.

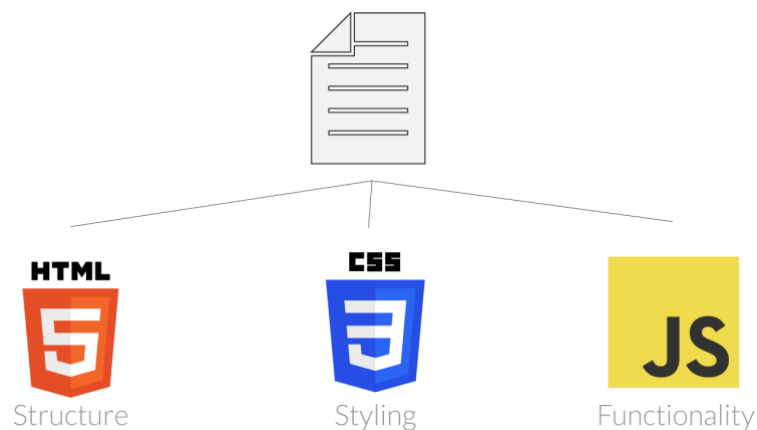


Figure 7: Architecture design of the solution

During this phase, the focus was on delivering the basic features one by one then going to the next features. On top of the technologies mentioned above, these additional technologies and tools have been used, the decisions to choose these tools have been made in the research phase:

- w2ui JavaScript grid library to display the table view with local MFT data.
- Fancytree JavaScript plugin to display the file tree directory view structure.
- jQuery library is used by the external libraries.
- jQuery-csv is a jQuery plugin used to parse multi-line csv string into 2D array.
- Atom IDE version 1.35.1

- Google Chrome browser version 72.0.3626 (64-bit) used for testing purpose.

All the external libraries have been downloaded and is going to be delivered to be stored locally. The implementation was mostly focused on delivering the parsing results than in the user interface design. The development process has been divided into smaller tasks and was implemented chronologically. Most of the effort in the implementation was in the parsing phase.

- Designing the layout user interface with HTML by adding the file upload button.
- Parsing the csv data to JSON format.
- Parsing of the file path field from the csv file to file directory tree structure on the left side.
- Displaying the parsed data into the table format using w2uid grid on the right side.
- Then populating the file tree data to the table view so that when a folder is selected the table on the right is updated showing only the entries that the file has.
- Overall testing before validating and fixing bugs.

6.1 PARSING DATA

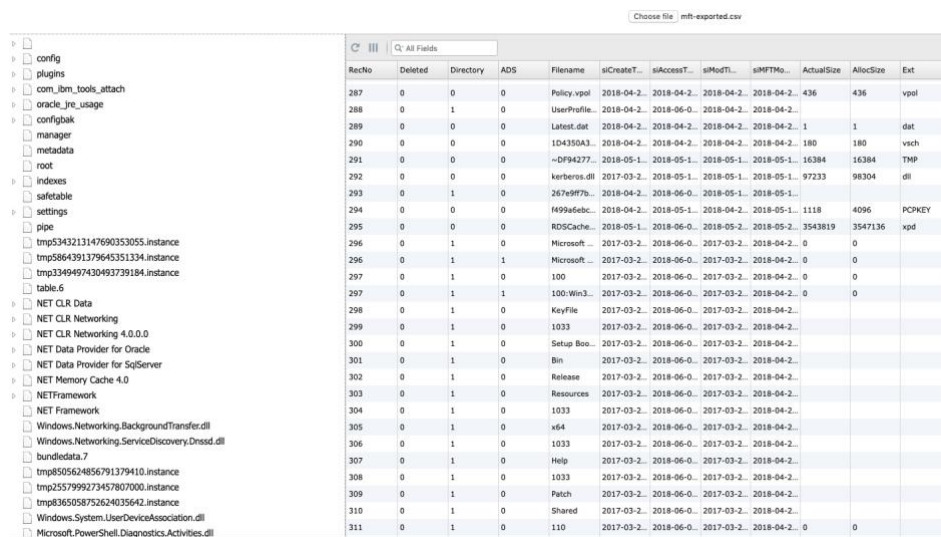
Two parser functions were implemented, the first parsing was used using recursive function and the purpose was to loop through the whole csv data to detect the full path to create a JSON file out of it with only id and name fields. By having such JSON file it allows to be used in the parser file tree view to populate data. The second parser is also using recursive function, it loops through each item in the file path and get the sub-folder id so that I can be used in the file tree directory folder view.

6.2 FILE DIRECTORY TREE VIEW

The file tree view was implemented using FancyTree functions. As mentioned below, using the data from the JSON file the directory tree structure was created to display directories and its sub-directories. The user is able to expand a directory to see the sub-directories.

6.3 TABLE VIEW

The table view was implemented using w2uid jQuery library, the MFT attributes are the table headers, and the entries are different rows. There is a functionality to hide or show columns so that the analyst can only view what they need as the table is already long and not fitting the screen. Using the functionality from w2ui library, the search functionality for specific string was implemented within the table. When implementing the search functionality, the developer tested searching alphanumerical strings which worked fine. And lastly, the table data is updated, if the user select a specific folder, the table will show only the entries that the directory has.



RecNo	Deleted	Directory	ADS	Filename	sCreateT...	sAccessT...	sModTL...	sMFTMo...	ActualSize	AllocSize	Ext
287	0	0	0	Policy.vpol	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	436	436	vpol
288	0	1	0	UserProfile...	2018-04-2...	2018-06-0...	2018-04-2...	2018-04-2...			
289	0	0	0	Latest.dat	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	1	1	dat
290	0	0	0	104350A3...	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	180	180	vsch
291	0	0	0	~DF94277...	2018-05-1...	2018-05-1...	2018-05-1...	2018-05-1...	16384	16384	TMP
292	0	0	0	kerberos.dll	2017-03-2...	2018-05-1...	2018-05-1...	2018-05-1...	97233	98304	dll
293	0	1	0	267e9f7b...	2018-04-2...	2018-06-0...	2018-05-1...	2018-05-1...			
294	0	0	0	f499a4ebc...	2018-04-2...	2018-05-1...	2018-04-2...	2018-05-1...	1118	4096	PCPKEY
295	0	0	0	RDSCache...	2018-05-1...	2018-06-0...	2018-05-2...	2018-05-2...	3543819	3547136	xpd
296	0	1	0	Microsoft...	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...	0	0	
296	0	1	1	Microsoft...	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...	0	0	
297	0	1	0	100	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...	0	0	
297	0	1	1	100:Win3...	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...	0	0	
298	0	1	0	KeyFile	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
299	0	1	0	1033	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
300	0	1	0	Setup Boo...	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
301	0	1	0	Bin	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
302	0	1	0	Release	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
303	0	1	0	Resources	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
304	0	1	0	1033	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
305	0	1	0	vs4	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
306	0	1	0	1033	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
307	0	1	0	Help	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
308	0	1	0	1033	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
309	0	1	0	Patch	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
310	0	1	0	Shared	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...			
311	0	1	0	110	2017-03-2...	2018-06-0...	2017-03-2...	2018-04-2...	0	0	

Figure 8: Screenshot of the final version rendered with MFT data without testing

6.4 ISSUES ENCOUNTERED DURING DEVELOPMENT

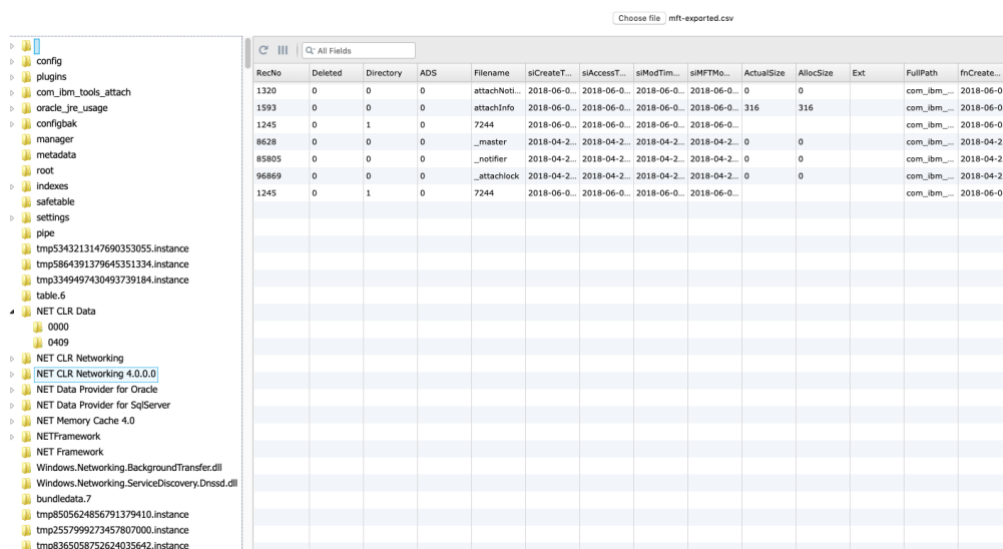
When implementing the solution few issues occurred, one of them was that sometimes the initial MFT file contained non-standard lines where a file attribute had multiple entries instead of one single entries. So, when debugging, the developer also checked the original MFT csv file and found out that some arrays were 2D instead of just strings to resolve this, the developer created a loop that goes through all the elements and just concatenate them, so they are like any other lines.

Another issue that the developer encountered was the amount of time spend checking if the data displayed in both table and file tree views were accurate and not missing anything. This

was done manually which took lots of time. Finally, the last minor issues were related to the view which were too small, so it was solved by improving user interface design.

Lastly, it occurred 4 times when testing that the browser was crashing just after file was uploaded. By debugging, this was due to the large number of data to be parsed to fix that a loop that loops by 50,000 entries was created so the file does not go through the entire table but rather by block of 50,000 entries. By doing so, the issue was fixed.

The developer had tested another time before starting the testing phase by doing so the developer had improved the user interface by adding icons to the file directory view, adding separate scrolling functionality for both views and resizing the view. The final outcome is displayed below:



The screenshot shows the 'Solution Final Rendering' of the NTFS Master File Table Viewer. On the left is a file directory tree with folders like 'config', 'plugins', 'com_ibm_tools_attach', 'oracle_jre_usage', 'configbak', 'manager', 'metadata', 'root', 'indexes', 'safetable', 'settings', 'pipe', and several instance folders. On the right is a table titled 'Choose file | mft-exported.csv' with columns: RecNo, Deleted, Directory, ADS, Filename, siCreateT..., siAccessT..., siModTim..., siMFTMo..., ActualSize, AllocSize, Ext, FullPath, and fnCreate... The table contains several rows of file data, including 'attachNoti...', 'attachInfo', '7244', '_master', '_notifier', '_attachlock', and '7244'.

RecNo	Deleted	Directory	ADS	Filename	siCreateT...	siAccessT...	siModTim...	siMFTMo...	ActualSize	AllocSize	Ext	FullPath	fnCreate...
1320	0	0	0	attachNoti...	2018-06-0...	2018-06-0...	2018-06-0...	2018-06-0...	0	0		com_ibm_...	2018-06-0...
1593	0	0	0	attachInfo	2018-06-0...	2018-06-0...	2018-06-0...	2018-06-0...	316	316		com_ibm_...	2018-06-0...
1245	0	1	0	7244	2018-06-0...	2018-06-0...	2018-06-0...	2018-06-0...				com_ibm_...	2018-06-0...
8628	0	0	0	_master	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	0	0		com_ibm_...	2018-04-2...
85805	0	0	0	_notifier	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	0	0		com_ibm_...	2018-04-2...
96869	0	0	0	_attachlock	2018-04-2...	2018-04-2...	2018-04-2...	2018-04-2...	0	0		com_ibm_...	2018-04-2...
1245	0	1	0	7244	2018-06-0...	2018-06-0...	2018-06-0...	2018-06-0...				com_ibm_...	2018-06-0...

Figure 6: Solution Final Rendering

Chapter 7: Testing

This chapter summarizes the testing activities performed to assess whether the application works as intended and to collect the client feedback. The purpose of the testing is to find bugs if it exists any and eliminate them. As the project is using waterfall methodology, this testing phase was performed once the solution was developed. However, the developer had done testing on the implemented functions throughout the implementation phase, to validate that that the new functionalities were working.

For testing purposes, the client had provided 10 different internal MFT files previously analysed in security incidents. Prior testing the application with the client, the developer had tested intensively the functionalities by herself. The functional requirements of the application were tested by both the developer and the client representative with all the provided 10 MFT files. All the testing were done on 2 different browsers in Google Chrome version 72.0.3626 and Firefox Quantum version 66.0.1. The table shows the percentage of passing score of 10 different MFT files on the 2 different browsers.

Functionalities tested	Expected Results	Passing %f or all 10 files
Uploading the csv MFT file.	MFT file is uploaded successfully.	100%
Data parsing into table.	MFT data is parsed into a table.	100%
Data parsing into file directory tree.	File path field is parsed correctly.	100%
View file directories.	Possible to view the file directory structure.	100%
Expand a folder to view sub-folder.	The user is able to expand a specific folder to view the sub-folders.	100%
View all the MFT records.	Viewing all the MFT records into the table.	100%
Search for a random string within the table.	The user is able to search for specific filename within the table and see a result if it exists any.	100%

Hide/See column for headers.	The user is able to hide/see certain columns.	100%
-------------------------------------	---	------

Once the functionalities were tested, the testing was done using predefined testing scenarios with the actual future users. The test scenarios were created in order to ensure full coverage and a first demonstration was done by the developer prior asking the user to perform the actions independently. The below table demonstrates the results done by 7 members of the CSIRT done on Google Chrome browsers. Each participant was given one different MFT file for testing.

Test scenario by performed actions	Expected results	Passing % *
Upload MFT file from your PC using the upload button.	User is able to select a file from her/his PC using the upload button.	100%
Once MFT is shown on the right in the table check that the MFT attributes are columns and entries are rows.	User is able to see the MFT entries in the table on the right side and the data is complete.	100%
Scroll down and to the right through the table to see further MFT entries.	User is able to see all the MFT entries in rows and able to scroll top-bottom and left-right.	100%
Scroll down the file directory navigation on the left.	User is able to scroll through the file directory structure independently from the table and the file path is parsed correctly.	100%
Select a folder from the file directory navigation.	The user is able to see all the entries within the selected folder in the table on the right side.	100%
Expand a folder from the file directory and select its sub-folder.	The user is able to see the sub-folder and if a sub folder is selected the entries are shown in the table on the right side.	100%

Refresh the data in the table by using the reload button.	All the MFT entries are shown in the table.	100%
Search in the table for string "Documents".	The user is able to see the results in the table for specific entries containing the string "Documents".	100%

**7 participants for 7 different MFT files. Data accuracy testing passed for all 7 files.*

The actual future users performed the user testing, tested the solution with different MFT files to view the data and the developer correlated the data to verify its accuracy. Individual feedbacks were collected from all tested users is available in appendix B including the individual feedback from users. Further description about the feedback and suggestions of improvements is discussion in the next Evaluation chapter. Due to time restriction, the testing could not yet determine, the efficiency of such MFT viewer tool for viewing MFT artefacts. Testing brought also concerns that haven't been raised in previous phases such as the speed of the parser tool and the user experience. Therefore, the client has taken decisions to continue upgrading the graphical user interface internally with the developer.

Chapter 8: Evaluation

The following chapter evaluates the overall development process and the outcome of the initial project. Firstly, the waterfall methodology used for this project was satisfactory however it was not possible to go from one phase to another without going back to the previous phase. Generally, the delivered product met all the requirements and worked as expected from the client and developer perspective. No bugs were observed during the testing.

Initially, a schedule was set with target completion dates and estimated required time. The development of the project has met the target dates however it hasn't met the required estimated time. This is due to poorly set estimation, some tasks required more time than expected. In reality, the research phase took more time than expected and was a big part of the development process for this project. Few mistakes on the assumptions were made when planning the project timeline such as on the testing phase where it actually doubles the estimated time. This is mainly to the testing with different team members. Since the target dates were quite flexible, all the phases finished on the target time expect for the report which finish 12 days later than expected.

8.1 PERSONAL EVALUATION

When developing the project, it was thought that the project was going to very complex to implement. It turned out to be less complex that first estimated. Since the developer did not have enough experience in project development, the project was developed based on a lot of background research. As the developer works within the client team, it was interesting to work on a personal side project for the team. The project offered opportunity to sit and discuss interesting investigation techniques and advice relating to more senior colleagues.

The developer did not focus enough on the user experience and the design of the solution which resulted to a less user-friendly graphical interface. There is definitely room for improvement concerning the visual design. The author also kept a project diary where the meetings, problems encountered, personal issues and project references were recorded. By

keeping, such diary it helped the developer to structure the project progress and allowed to have self-discipline.

Because the project develops had limited knowledge about web development, better architecture could have been chosen. However, the developer did not want to choose a tool that she was not an expert and did not have enough time for acquiring new skills. Also, the author's understanding of security technology operational industry standards has improved by working with the team with an independent project. I gained experience in collaborating with multiple analysts as well as my communication skills have improved. This project required lots of self-discipline and was definitely challenging to conduct as a full-time employee and student at the same time.

8.2 FURTHER DEVELOPMENTS AND RECOMMENDATIONS

As suggested by the client during testing, the overall user interface would need to be improved and few new functionalities could be added. One of them, is a feature that would allow to filter headers on the table part so that users can filter the columns by headers. Another functionality is to add an export functionality, where the user would be able export the filtered search results as evidence.

Lastly, during testing, it was observed that the speed of MFT file processing when uploading a file was relatively slow. A major improvement could be to improve the speed of the processing and a custom API could be written in the future to improve this.

Conclusion

The project produced a report which described the project implementation of the MFT Viewer solution and the MFT Viewer web application. The delivered solution allows the client team members to view MFT metadata into a human readable format and use these data as evidence to their analysis. The client requirements for the project have been met. The overall feedbacks from the client side are positive even though there is room for improvements, as discussed in Chapter 8, the MFT Viewer works as projected. The testing performed by the client show that the web application is able to present the MFT data into tabular form and the client is able to use it for security incident analysis. The developer has enjoyed working on such challenging task and believes that the solution is useful for the client. The developer believes that she had made appropriate contribution to her teamwork.

References

Bruneau, G. (2017). *CyberChef a Must Have Tool in your Tool bag*. [online] SANS Internet Storm Center. Available at: <https://isc.sans.edu/forums/diary/CyberChef+a+Must+Have+Tool+in+your+Tool+bag/22458/> [Accessed 5 Feb. 2019].

Dennis, A., Roth, R. and Wixom, B. (2012). *System Analysis and Design, Fifth Edition*. 5th ed. John Wiley & Sons.

EC-Council (2010). *Computer forensics*. 1st ed. Clifton Park, NY: Course Technology Cengage Learning, pp.12-13.

Gibson, P. (2011). *Agile Methods of Software Development*.

Hughey, D. (2009). *The Traditional Waterfall Approach*. [online] Umsl.edu. Available at: <http://www.umsl.edu/~hugheyd/is6840/waterfall.html> [Accessed 15 Oct. 2018].

Shaaban, A. and Saprionov, K. (2016). *Practical Windows forensics*. Birmingham, UK: Packt Publishing, p.101.

Carrier, B. (2005). *File system forensic analysis*. 1st ed. Upper Saddle River, NJ: Addison-Wesley Professional.

Appendix A – Gathering requirements

Interview conducted with client representative meeting notes.

User 1 – Analyst L2 interviewed on 4 October 2018:

1. Q: In what case do you need to analyse MFT artefact?

A: I mostly use MFT data to gather evidence where I need to know if the file is still present on the system and sometimes check the folder to see if there are other malicious files within the same folder.

2. Q: What current process do you use to view the MFT artefact?

A: I acquire the MFT file, and I open the file with excel and separate the tabs and see the information in a table. If I need a keyword, I just search with command + f they keyword.

3. Q: What is the most important for you analysing the MFT metadata?

A: I would say been able to search for keyword.

4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.

A: I am using cyberchef offline for decoding malicious commands and I really like how it works. If you can do something similar that would be great – it is very portable.

5. Q: What kind of functionalities do you need in the solution?

A: Searching functionality.

6. Q: What kind of functionalities would you like to see in the solution?

A: Maybe something where I can see information into a table because I am used to see the MFT into a table.

User 2 – Analyst L2 interviewed on the 5 October 2018:

1. Q: In what case do you need to analyse MFT artefact?

A: I use for looking if the file is deleted or not. I also look at the \$Filename to see if the file was copied, file was accessed or file was modified. Sometimes I use to see the file path of a malicious file because I need to acquire the file.

2. Q: What current process do you use to view the MFT artefact?

A: I get the MFT from DLP system then I see the data in excel like with any csv that we get from the different tools.

3. Q: What is the most important for you analysing the MFT metadata?

A: Been able to find what I am looking for

4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.

A: I don't mind solution; I was thinking of having an excel macro that would separate the tabs in the csv file. Or maybe something like a website.

5. Q: What kind of functionalities do you need in the solution?

A: I need the search functionalities.

6. Q: What kind of functionalities would you like to see in the solution?

A: Maybe something like I can do in excel expanding the columns or rows.

User 3 – Analyst L2 interviewed on the 17 October 2018:

1. Q: In what case do you need to analyse MFT artefact?

A: I use MFT to see if the file is still on the system.

2. Q: What current process do you use to view the MFT artefact?

A: I open the MFT file with notepad++ then look for the filename.

3. Q: What is the most important for you analysing the MFT metadata?

A: I like a fast solution because excel is slow to open large csv files.

4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.

A: You can do an internal website to see it or something like what Romulus did with IOC lookups.

5. Q: What kind of functionalities do you need in the solution?

A: Searching for filename and timestamps.

6. Q: What kind of functionalities would you like to see in the solution?

A: Something where it is sorted.

User 4 – Analyst L2 interviewed on the 23 October 2018:

1. Q: In what case do you need to analyse MFT artefact?
A: I need to see MFT information for finding the file path of a file or to see the timestamps.
2. Q: What current process do you use to view the MFT artefact?
A: I use excel to view MFT with a table and then I use the filters to search for a value.
3. Q: What is the most important for you analysing the MFT metadata?
A: Seeing table format because this is how I use.
4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.
A: Something like excel maybe I actually like the current process because I use excel with all the csv files.
5. Q: What kind of functionalities do you need in the solution?
A: Search functionality and table view are important.
6. Q: What kind of functionalities would you like to see in the solution?
A: I don't have any because excel provides everything that I need.

User 5 – Analyst L3 interviewed on the 23 October 2018:

1. Q: In what case do you need to analyse MFT artefact?
A: I use it for forensic investigation when a deeper analysis is required. Usually to see (Modified/Accessed/Created) of a specific file / directory. When I need to see what other files / directories were created around the (+ or -) time of an alert. And also, to see what another files/folder resides in a particular path.
2. Q: What current process do you use to view the MFT artefact?
A: I use Timeline Explorer to view the MFT even though they are not suited for our internal csv file format otherwise I used any text editor to open the MFT csv file.
3. Q: What is the most important for you analysing the MFT metadata?
A: I like when I can find what I am looking for easily.
4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.

A: Anything that is easy to use is good for me. Sometimes I spend lots of time using Timeline Explorer to view the data.

5. Q: What kind of functionalities do you need in the solution?

A: I need the search functionality to find the information I am looking for.

6. Q: What kind of functionalities would you like to see in the solution?

A: I would recommend a table structure and hide column functionality.

User 6 – Analyst L2 interviewed on the 30 October 2018 by phone:

1. Q: In what case do you need to analyse MFT artefact?

A: Mostly to find if the file is still present on the machine and the file path.

2. Q: What current process do you use to view the MFT artefact?

A: I use notepad++ to open csv files.

3. Q: What is the most important for you analysing the MFT metadata?

A: I would say searching functionality.

4. Q: What kind of solution is easier for you? Getting suggestions about the format of the solution.

A: Any type of table view solution maybe something like Cyberchef is also good

5. Q: What kind of functionalities do you need in the solution?

A: I need search functionality.

6. Q: What kind of functionalities would you like to see in the solution?

A: A possible feature to export the result or to filter by attributes.

Appendix B – User testing feedbacks

1. Evaluation/Feedback based on testing

User 1

The application is really good I wasn't expecting the file exploring feature, it is nice. I would recommend you improve the user experience for example add a loading wheel when a file is uploading. Also, you can error messages if it is not working or something like that in the future. I will use it in the future.

2. Evaluation/Feedback based on testing

User 2

I like that it is web-based tool, so it is portable. I think the loading is slow and search functionality can be implemented in the file browser. The tool is easy to use and intuitive.

3. Evaluation/Feedback based on testing

User 3

The tool is very easy to use, and I like that it is the portable web application. I would recommend improving the search functionality or add filtering on the table column. I like the file browser option.

4. Evaluation/Feedback based on testing

User 4

The tool is interesting, it looks good, and I like it. I don't have bad comments for now I need to use it. Just an improvement I would suggest improving the speed of processing.

5. Evaluation/Feedback based on testing

User 5

Since I usually use MFT for memory forensics – deeper investigation. I need more functionalities in the searching and filtering so I can find what exactly what I am looking. I advise you to look for more searching libraries where you can have advanced searching functionalities. But above all of that good job.

6. Evaluation/Feedback based on testing

User 6

This looks nice, I don't have comments because I need to use the tool for real investigation to let you know the feedback and the efficiency.

7. Evaluation/Feedback based on testing

User 7

I like this type of solution for web-based application. It is very basic and nice looking. Maybe improve the speed of the file loading.

Appendix C – Project Specification

Summary

The project is focused on complex data representation and manipulation which will consist of developing an appropriate graphical user interface tool to interpret a partially parsed Windows New Technology File System (NTFS) master file table (MFT) data for forensic analysis within a Security Threat Response team. This data representation tool is custom-made for this Threat Response team based on their investigation techniques, tools, and internal incident handling process. The main objective of this project is to deliver a solution that will help my colleagues in their daily workload by proposing a solution that will represent these MFT data into a simpler representation that fit their needs.

Rationale

The MFT is a file in which information about every file and directory of the operating system logical volume is stored. All information about a file, including its size, time and date stamps, permissions, and data content, is stored either in MFT entries, or in space outside the MFT that is described by MFT entries (Windows, 2018). Already over a decade ago, Carrier (2005, p. 199) confirmed that “every file and directory has at least one entry in the table, and the entries by themselves are very simple”.

Various anti-forensic techniques are used by adversary, one of them is timestamping, which can be detected by analysing the MFT data. Other malware and forensic artifact can be found by analysing MFT data. SANS (2016) that “it is hard to permanently destroy all references to files and artifacts on a host”. A data visualisation is any type of visual representation of a particular data with the purpose of providing a better understanding of that material. The data visualisation has become an important part of information security forensics as the world becomes more connected and every machine connected to an organization network produces a large amount log data. The analysis of these data is often difficult and very time consuming. There are a very few number conferences done about this topic one of them is the paper from Fowle and Schofield (2011) whom maintain

that “data visualisation is becoming increasingly important for understanding information, such as investigative data and analysis. The visual system of a human has the capability to interpret and comprehend pictures and charts much faster than reading a text of the same material”.

This project was first initiated by the Threat Response team whom I work with. As security professionals, we are often challenged with complex data analysis for forensic investigation when handling security incidents. We work with numerous types of tools daily to investigate security threats or incidents. Many of these tools are inflexible and cannot provide all the necessary evidence to complete an investigation. Regularly, we are collecting evidence one-by-one from multiple sources depending on the type of security incident to investigate. Thus, by providing such made-to-measure web application tool, the project objective will be met. My personal objectives are to have a deep understanding of forensic analysis which is an area of high interest to me and to contribute to our teamwork by developing such tool.

Background research

Prior choosing this project, appropriate background research has been conducted in order to evaluate this project’s feasibility as well as reviewing existing similar tools. As a result of these studies, multiple MFT parser tools were identified such as AnalyzeMFT by Kovar (2012), which is a tool that is “a python script designed to fully parse the MFT file from an NTFS filesystem and present the results as accurately as possible in multiple formats” (Kovar, 2012). However, while testing the application, the outcome was an unorganised data in comma-separated values file format. This first parsed data still needs to be interpreted by using another application such as a text editor.

Areas of investigation

This project has several areas of interest. The main areas of focus are forensic analysis techniques used within the team to identify which information is needed to complete a forensic investigation, how to efficiently represent complex data, software development techniques to develop the web application and how to design user interface.

Methodology and ethics

In order to complete the project, few main tasks are going to be done. One of them is collect forensic analysis process and forensic habits within my team which be collected

during a 1:1 discussion. Further specific task described below with their respective timeline. A suitable development methodology for such web application has been initially identified as waterfall development. Waterfall development model suits the project as the project steps are sequential which allow a traditional approach for such tool. Each phase of development is going to be developed one after other offering a great structure to the project. This methodology has the advantage of identifying the requirements before the programming begins and limiting changes to requirements as the project's proceeds (Dennis, Roth and Wixom, 2012).

While designing and developing this project, Teesside University research ethics are to be followed. Any harm will be avoided to any of my colleagues involved in this project. Any sensitive data used during the development and testing phase will not be compromised and will be handled according to ethic principles.

Project planning

Prior starting this project, the project requirements are going to be collected from every member of the team individually and merged into one list of requirements. Below is the project timeline including the approximate required time, starting date and end date:

Phase	Task	Required time (approx.)	Target Date
Planning	Background research about the topic and the existing tools.	20 hours	31/10/2018
	Research and chose about development technologies and methodologies to be used.	10 hours	31/10/2018
	Gather technical requirements from colleagues and required resources.	20 hours	31/10/2018
Design	Select the tools and technology to be used.	8 hours	15/11/2018
	Design the mock-up tool with the functionalities based on the requirements gathered.	10-15 hours	15/11/2018
	Get approval of the tool design from client.	8 hours	18/11/2018

Development	Develop the tool (coding part) and the user interface	100 hours	01/02/2019
	Develop the user interface.	20 hours	01/02/2019
Testing	Run testing session.	15 hours	15/02/2019
	Fixing from the testing output.	10 hours	20/02/2019
Reporting	Deliver the project finding.	30 hours	15/03/2019

Project Deliverables

The final deliverables of this project are a first prototype web application that is intended to represent any given MFT data from an already parsed comma-separated values format into a tabular form that can be used by my colleagues within Threat Response team when investigating security incidents and a detailed project report.