Forensic Analysis of Volume Shadow Service ($RecycleBin) of Win10

by

Nosheen Manzoor

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment of the requirements for the degree of MS in Information Security

June 2019

CERTIFICATE

This is to certify that **NS Nosheen Manzoor** Student of **MSIS-14** Course Reg.No **00000119618** has completed her MS Thesis title **“Forensic Analysis of Volume Shadow Service ($RecycleBin) of Win10 “**under my supervision. I have reviewed her final thesis copy and I am satisfied with her work.

Thesis Supervisor

(**Assistant Professor Mian Muhammad Waseem Iqbal**)

Dated: ______June 2019

Declaration

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

Dedication

“In the name of Allah, the most Beneficent, the most Merciful"

I dedicate this thesis to my Mother, Father, Husband and Teachers who supported me in each step of the way.

Acknowledgments

I am thankful to Allah the Almighty, for giving me the courage and strength to complete my thesis. I would also like to pay special thanks to my Supervisor, Assistant Professor Waseem Iqbal, for his continuous support and guidance throughout this thesis and believing in me. Also, I would thank my committee members; Lecturer Narmeen Shafqat, and Asst Prof Waleed Bin Shahid for their support and knowledge regarding this topic.

I am also thankful to my Family especially to my Father, Mother and my Husband for sparing me from family commitments, motivating me and praying for my success. Without their support, it would have not been possible for me to complete this research work. Last but not the least, my special gratitude to all my wonderful teachers of Department of Information Security at Military College of Signals, for imparting knowledge and enabling me to complete this task.

Abstract

In Todays' digital world everything is shifting into the smart technology and people have started relying on this digital world. Size of the storage media has also been increased day by day. When everything is shifted to the digital world therefore crime has also been shifted to the Digital crime. In digital crime finding out the digital evidence from the storage media is becoming complex and time consuming.

It is better and interesting for the investigator to start carving for the evidence from the most crucial areas like windows Volume Shadow Service.

Volume Shadow Copy is considered as gold mine for the forensic investigator as it generates differential backups. Previous versions of the files, Recycle bin and state of the $logfile get saved in Volume Shadow Copy which holds clumps of crucial data for the investigators.

Volume Shadow Copy lets the investigator to understand the state of the system on a particular date. Whatever is deleted from the system even its deleted permanently with a wipe utility may have its presence in VSC (Volume Shadow Copy).Being not accessible to the user in normal environment and being "Read only" in nature preserves the evidences to a great extent.VSC in series gives the idea of routine and activities performed by the accused in a sequential manner. An experiment with two case scenarios e.g Case scenario1, stolen financial information and Case scenario 2, modified health information of a patient has been conducted to prove the importance of the Volume Shadow Copy. Methodology has been proposed to extract the data from the Volume Shadow Copy of Windows 10 to find the evidence from Volume Shadow Copy's store which gives access to the previous version of not only the user files but also from the system files.

Table of Contents

LIST OF FIGURES

Figure Number *Page*

LIST OF TABLES

Chapter1

Introduction

1.1 Overview

Volume Shadow Service is a Microsoft's built-in service of windows. It was first introduced in Windows server 2003 named as "Shadow Copies for Shared Folder". When this service was added in windows vista then its name was changed to "Volume Shadow Copies" in 2007.Volume Shadow copy creates differential backup copies of crucial data. Volume Shadow Service provides the facility to use previous version in order to find your data which may corrupt by any virus or accidental data loss.[1][2]

Volume shadow copy is a Microsoft window's inbuilt service that permits the user to take backups of the system either its manual or automatic even when the system is in use. The block size of the hard drive of the system is 16KB. These blocks of the computer system are constantly monitored and as soon as any modification or alteration took place in a block window start implementing this modification in the block of volume shadow service after storing it on the storage location. The window creates a backup of the block in this fashion. System's settings and all important data of drive C records in Volume Shadow Copy Service. It enables the system to encounter unpredictably data deletion and from events which destabilize the system, like a virus attack or the inaccurate installation of a software package or any other hardware device. It's been remarked as a gold mine of forensic proof because of the amount of data it records.[2]

This information is very valuable from the forensic point of view due to the following reasons.

a. It lets the investigator understand the state of the system on a particular date.

b. Whatever is deleted from the system even its deleted permanently with a wipe utility may have its presence in VSC (Volume Shadow Copy)

c. Being not accessible to the user in the normal environment and being “Read-only” in nature preserves the evidence to a great extent.

d. VSC in series gives the idea of routine and activities performed by the accused in a sequential manner.

There are so many built-in features available in Windows 10 that can be utilized as a good source of forensic artifacts and Volume shadow service is from one of them. Shadow copies are created in two different ways, a differential backup or complete backup. Complete backup generates a complete copy of data available on disk. In the differentialbackup, only those changes are backed up which are made in a specified block on the original volume. To keep track of the logical construction of the volume shadow copy a buffer is used. [3]

Automatic and manual creations are two different ways to create volume shadow copies. In automatic creation volume shadow copy is generated or activated by the operating system as soon as new software is installed and when an update of the system is installed. Volume Shadow service then creates an image of only those files which are changed since the last backup. Users create manual volume shadow copy when they make changes in their files and wants to trigger volume shadow copy manually. Creation of Volume Shadow service is a three-step process. Freeze: Hard disk of the computer marked as read-only Snap: Image of the system/Hard drive created.Unfreeze: Hard disk of the system get released and volume shadow service runs in the background.[4]

There are three main components of Volume Shadow Service Writer: Volume Shadow writer is responsible for informing the backup device that how to back up the information, applications and their data. Requestor: A Volume Shadow Service requestor is responsible for starting the VSC processes. Mostly VSS requestors are backup applications. Provider: VSS provider acts as a middle layer between backup processes, operating system,and hardware.[2]

1.2 Motivation and Problem Statement

Volume Shadow Copies provides extra information with extra data that normally not available on the system.It enables the forensic investigator to know about what was happening in the system before he/she may start the investigation. It provides the point in time copies of the user as well as system data. Shadow copy is the vigilant tool to recover the previously permanently deleted files by the user.

For every newly released operating system, Forensic investigator must re-consider that new version of the operating system to determine any minor/major changes which may affect their investigation. Forensic Analysis of Windows 7 has been done but information available in the literature is not sufficient to analyze the new versions of the operating systems like windows 10.

1.3 Objective

The main objectives of the thesis are:-

a. Forensic Analysis of $Recyclebin from volume shadow service of Microsoft windows10
b. Comparison between $Recyclebin from volume shadow service of Microsoft window7 and win10

1.4 Thesis Contribution

To best of my knowledge, limited research has been done on forensic analysis of Windows 10 and especially how to extract the files and folders from the difference files. Most of the means which I have explored are blogs, presentations and articles have only given very basic knowledge about forensic analysis of Windows 10. Moreover, the internal structure of the shadow has not been discussed.

The main contribution of this research work are as follows

a) We have proposed a mechanism to identify the Volume Shadow Copy stores of Windows 10 from the system image.

b) We have discussed the internal structure of the volume Shadow Copy of Windows 10.

c) We have proposed the way to recover the files from the Volume Shadow Copy of Windows 10.

1.5 Thesis Organization

The thesis is structured as follows:

- Chapter 2 contains the literature reviewed in the thesis. The general introduction of the Volume shadow copy, working of Volume Shadow Copy, Creation process of Volume Shadow Copy, Access methods of Volume Shadow Copy used for Windows 7,Vista
- Chapter 3 contains the test beds, Experimental test case scenarios. proposed methodology for accessing the Volume Shadow Copy
- Chapter 4 Analysis of $Log file and $ Recycle.Bin, their importance and artifacts.
- Chapter 5 contains the Results and analysis reports, Winhex screenshots of experiments
- Chapter 6 contains the comparison of different freely available tools for the analysis of the Volume Shadow Copy.
- Chapter7 contains the discussion, conclusion of the thesis and Future work.

Chapter2

LiteratureReview- UnderstandingVolume Shadow Copy

In order to forensically examine volume shadow copy of windows 10 in a better way, it is important to explore research papers and articles. Unluckily limited research has been found on forensic analysis of Windows 10 and especially on forensic analysis of $Recycle bin of Volume shadow copy of Windows 10. The information available in the shape of research done by the research community. Most of the means which I have explored are blogs, presentations and articles have only given very basic knowledge about forensic analysis of volume shadow copy of windows 10.

2.1 Understanding Volume Shadow Copy

In order to understand how Volume Shadow Copy works it is important to understand its layout, structure, and configuration. This chapter discusses all necessary concepts Volume Shadow Copy, related to this research thesis.

2.1.1Volume Shadow Copy Volume Shadow Copy Service (VSS) is a Component Object Model (COM) interfaces in Microsoft Windows built-in service to perform volume backups. Excellent coordination is required between backup application, user application which is going to be backed up and hardware and software management. Volume shadow copy which was first introduced in 2003 provides the coordination between these applications.

Volume Shadow Copies are transparently maintained by the Windows. Volume Shadow Copy operates at a lower layer than NTFS as shown in fig1 [9].

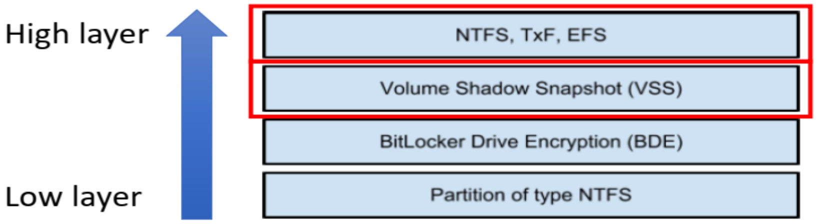

High layer
Low layer
NTFS, TxF, EFS
Volume Shadow Snapshot (VSS)
BitLocker Drive Encryption (BDE)
Partition of type NTFS

Fig:1 Layer of NTFS Volume

2.1.2 Working of Volume Shadow Copy: Volume Shadow Copy consists of thefollowing basic components.

VSS service: It coordinates with all other components of the VSS to create shadow copies smoothly

VSS Requestor: It is backup software that requests the operating system to perform a backup.Window server backup utility basically performs this duty.

VSS Writers: This software assures that the consistent backup copies of the windows are generated. They provide data integrity during backups process.

VSS Provider: VSS Providers can be VSS hardware and software and they actually generate shadow copies and then work on their maintenance as well.[2]

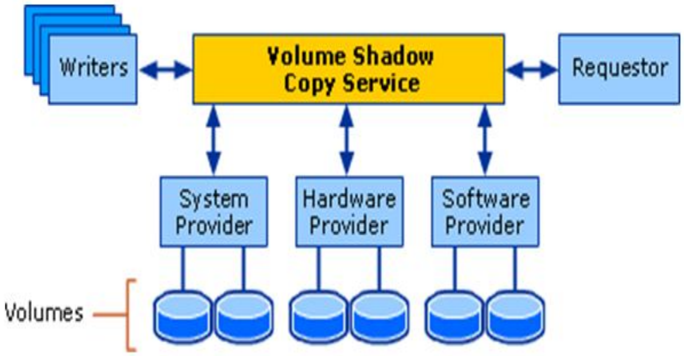


Fig 2: Architectural diagram of Volume Shadow Copy

2.1.3Creation Process of a Volume Shadow Copy

In the creationprocess, all the components of VSS service work together with a high level of coordination. Shadow copy creation process is shown in fig. 3.

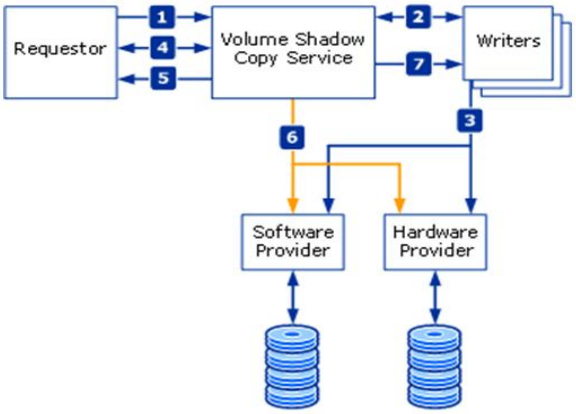


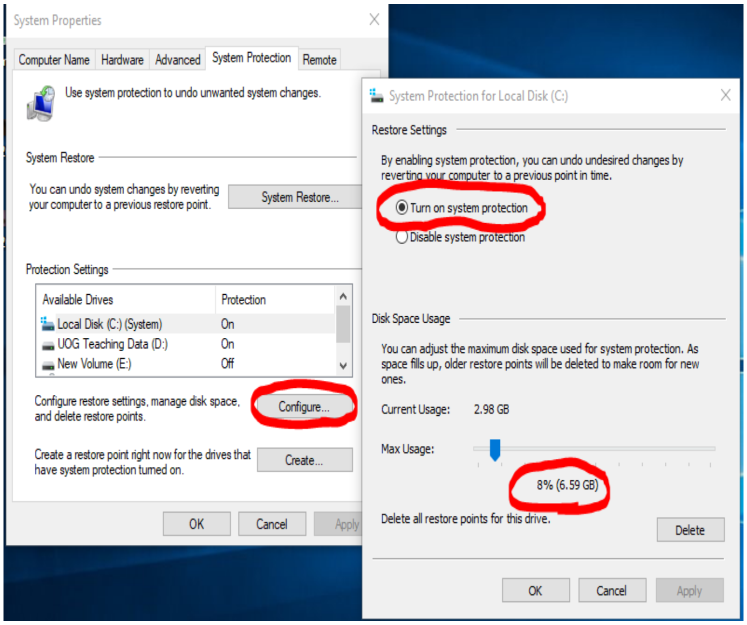
Fig3: Creation Process of Volume Shadow Copy

- Volume shadow copy service specify the writer after getting metadata for preparation of shadow copy created on the request of VSS Requestor
- XML description has been created by every writer involved in the creation of volume shadow copy
- Each writer creates an XML description of the components and data stores that need to be backed up and provides it to the Volume Shadow Copy Service. The writer also defines a restore method, which is used for all components. The Volume Shadow Copy Service
- provides the writer's description of the requester, which selects the components that will be backed up.
- All the writers then notified by the Volume Shadow service to be ready to create a shadow copy.
- All the applications are temporarily frozen for less than 60 seconds to write-up the data into the shadows.

- Within 10 seconds shadow has been created then all the I/O operation related to different applications gets released for their normal working.

2.2 Enabling Volume Shadow Copy

In order to enable the Volume Shadow Copy, we have to turn on the system Protection feature from the control panel.



Use system protection to undo unwanted system changes.

System Restore

You can undo system changes by reverting your computer to a previous restore point.

Protection Settings

Available Drives	Protection
Local Disk (C:) (System)	On
UOG Teaching Data (D:)	On
New Volume (E:)	Off

Configure restore settings, manage disk space, and delete restore points.

Create a restore point right now for the drives that have system protection turned on.

Restore Settings

By enabling system protection, you can undo undesired changes by reverting your computer to a previous point in time.

● Turn on system protection

○ Disable system protection

Disk Space Usage

You can adjust the maximum disk space used for system protection. As space fills up, older restore points will be deleted to make room for new ones.

Current Usage: 2.98 GB

Max Usage: 8% (6.59 GB)

Delete all restore points for this drive.

Fig4: Configuration of Volume Shadow Copy

Older shadow copies are deleted by the window when window run out of space. Volume Shadow Copy of a specific volume is stored in the volume itself so if the volume gets corrupted the Volume Shadow Copy of that volume will also be corrupted. Volume Shadow Copy is a block level incremental backup. Block size to be increment is 16 KB. Data of Volume Shadow Copy cannot be changed until or unless it is deleted. Volume Shadow Copy generates incremental backups and provides the facility to the user to restore the system's previous state when required. [5]

GUID{3808876b-c176-4e48-b7ae-04046e6cc752} is used by the Volume Shadow Copy to distinguish the attributes of the shadow copies which includes header files and store files.

2.3 Method of Storing Data in Volume Shadow Copy

As soon as volume shadow copy is created its corresponding catalog and store is allocated to the volume shadow copy. Data stored in 16KB blocks. As an example, 60KB file will be stored in 4 data blocks as I data block is of 16 KB shown in fig5a.[6]

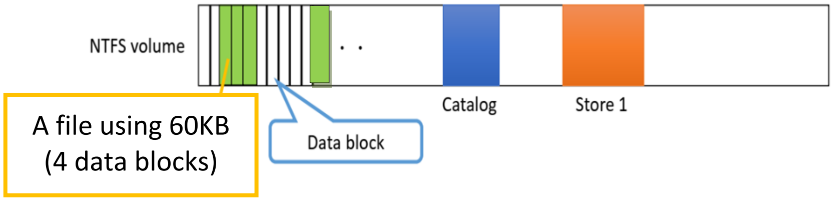


Fig: 5a Data storage method in volume shadow Copy

As soon as the data in the specific block is changed it’s been copied to the store1 as shown in the fig 5b.[6]

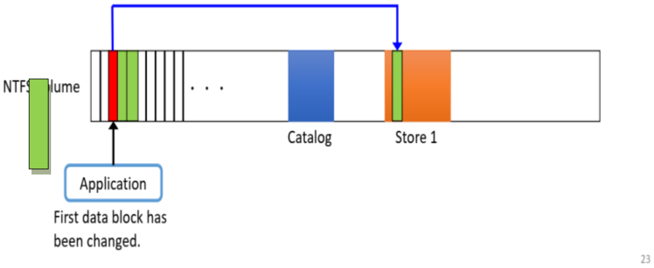


Fig: 5b Data storage method in volume shadow Copy:

When we create a second Volume Shadow Copy second store is allocated to the volume shadow copy and second entry has been made into the catalog entries. Now only those Blocks have been backed up which have been changed since last Shadow Copy [6].

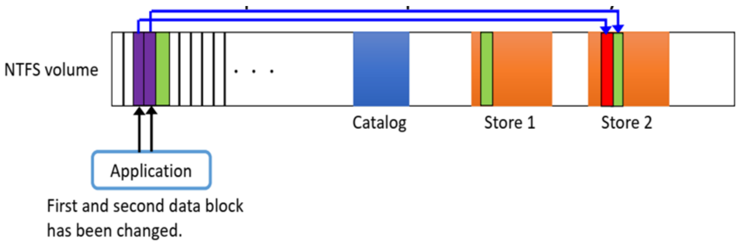


Fig5c:Data storage method in volume shadow Copy

2.4 Location and Structure of Volume Shadow Copy

Following is a layout through which windows Operating system access the volume snapshot by accessing the volume snapshot header.

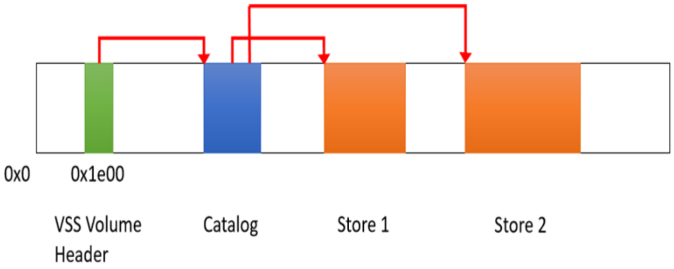


Fig 6: Volume Shadow Copy working layout

NTFS volume header contains the Volume Shadow Copy header as its part. Volume header data always starts at offset 7860(0x1e00) in windows vista, 7, 8 and in Windows 10 as well. Its size is 512 bytes equals to one sector [6].

2.4.1Structure of VSS Volume header

Volume Shadow Service is basically is located at offset 7680(1x1E00) of an NTFS volume. Its size is 512byte =1sector. Volume snapshot header consists of the Volume Shadow Copy identifier and offset of the first catalog blocks shown in table1 [7].

Table 1: Catalog Block Header[7]

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Volume Shadow Copy Identifier															
Version		Record Type(0x01)						Current offset relative to the start of the volume							
Unknown (Next offset relative to the start of the volume) 0x1e00								Unknown empty values							
Catalog offset relative to the start of the volume (value=0 if there is no catalog)								Maximum size (set to 0 if unbounded)							
Volume identifier (contains GUID)															
Shadow copy storage volume identifier (GUID)															
Unknown		Unknown empty values													

2.4.2Catalog Block Header

The catalog holds the information about each and every snapshot. Every catalog has One or more than one catalog blocks. Each catalog block consists of catalog block header and catalog block entry as shown in table 2. If Volume Shadow Copy is enabled but has no snapshot, then no catalog will exist [7][8].

Table2: Catalog Block Header[7][8]

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
VOLUME SHADOW COPY Identifier (contains GUID)															
Version 0x01				**Record Type(0x02)**				Relative catalog block list offset (The offset is relative to the start of the catalog block)							
Current catalog block list offset relative to the start of the volume								Next Catalog block offset							

Table 3: Catalog bock header types

Version	Type
0x01	Windows Vista, and 7
0x02	Windows 10

2.4.3Catalog Entries

Catalog entry started directly after the catalog block header in this case offset of first catalog entry is (0x71f0000).Catalog entry type0x03 is found directly after the catalog entry type of 0x02[8]. If the catalog entry type 0x03 is present, then it shows that stores are present on the volume. If a system hasfour-volume shadow copies it means four 0x02 and 0x03 entry type(2 entries for each Volume Shadow Copy=8 entries)

2.4.3.1Structure of Entry type 0x02

Structure of entry type 0x02 is described in table 4 [7][8].

Table4: Structure of Entry type 0x02[7][8]

<table>
<tr><td colspan="17"><b>Entry type 0x02</b></td></tr>
<tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td></tr>
<tr><td></td><td colspan="8">Catalog entry type</td><td colspan="8">Volume Size</td></tr>
<tr><td></td><td colspan="16">Store identifier (contains GUID used in store filename)</td></tr>
<tr><td></td><td colspan="8">Unknown sequence number</td><td colspan="8">Flag values (440 in windows 10) 40 in win 7,8</td></tr>
<tr><td></td><td colspan="8">Shadow copy creation time</td><td colspan="8">Unknown Empty values</td></tr>
</table>

2.4.3.2 Structure of entry type of 0x03

Structure of entry type of 0x03 described in table 5.

Table5: Entry type 0x03[10]

Entry Type 0x03	
Store identifier (contains GUID used in store filename)	
Store header offset relative to the stat of the volume	Store block range list offset relative to the start of the volume
Current bitmap offset relative to the start of the volume	NTFS metadata file reference
Allocated size	Store previous bitmap offset

2.4.4Store (Actual volume snapshot)

The actual data blocks of Volume Shadow Copy are stored in stores. Following data structure keeps track of the volume snapshot locations.

2.4.4.1Store Blocklist

It contains store block header of type 3 and is the size of 128 bytes followed by 32 bytes of index shown in table 6

Table 6: Store blocklist [7][8]

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Volume Shadow Copy Identifier contains GUID															
Version 0x01				Record Type (0x03) Store Header				Relative block offset							
Current block offset								Next block offset (0 incase of the last block)							
Size of store information								Unknown empty values							

2.4.4.2Catalog Block List

It starts directly after the store. It consists of original data block offset and relative store data block offset and store block descriptors.

2.4.4.3Store block descriptor

Flag 0x00	**Normal descriptor**
Flag 0x88	Complete block of the snapshot will be mapped to the original block
Flag 0x01	**Overlay descriptor** In an overlay a descriptor,, bitmap allocation table contains the data about to fill the block.
Flag 0x02	**Forward descriptor** The relative offset will be mapped to the original offset to the next block
Flag 0x04	**Invalid index record** The block will be ignored if the flag set to 0x04

It consists of store block header of type3 and stores block descriptor of the size of 32 bytes. Store block descriptor consists of original block offset as well as relative store data block offset. Original data block offset should be replaced by the shadow copy data block. The mapping between snapshot data and original is based on flag fields. Values of the flags can be interpreted as follows [7][8].

2.5 Accessing Volume Shadow Copy

Volume Shadow Copy creates differential backups as it only stores the changed part of the file, not a complete file. In order to access files from the Volume Shadow

Copy, all the stores and 16 KB data blocks will be read and combined with the live volume to recreate the complete file [6] [7][9].

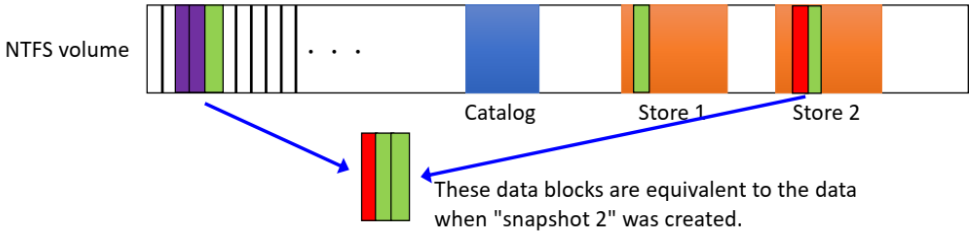

NTFS volume
. . .
Catalog
Store 1
Store 2
These data blocks are equivalent to the data when "snapshot 2" was created.

Fig:7 Reconstructing Volume Shadow Copy: These data blocks are equivalent to the data when second shadow was created

Reading of the data has been started from the most recent store to the oldest one. As in Example described in fig. data block from the current volume is combined with the data block of store reproduces the file which is equal to the file when second shadow copy has been created [7][6][8].

As we have created only two shadow copies, we have only two stores. The file reconstructed from the previous step further will be combined with the data block of stored in the first step reproduced the file equivalent to the file when shadow copy 1 was created.as shown in fig 7a

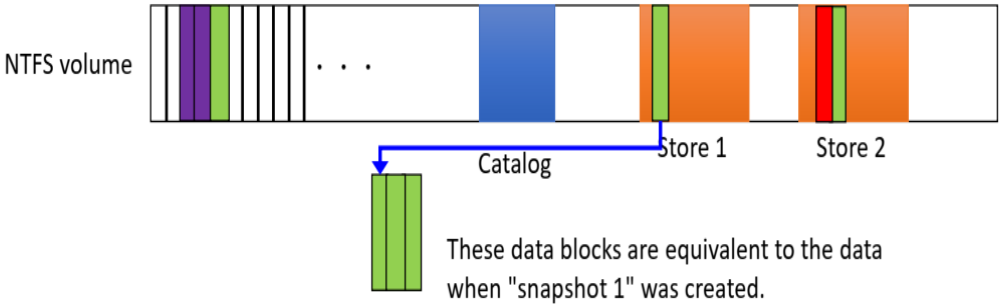

NTFS volume
. . .
Catalog
Store 1
Store 2
These data blocks are equivalent to the data when "snapshot 1" was created.

Fig7a: Reconstructing Volume Shadow Copy: These blocks represent the data when 1st shadow copy was created

Chapter3

Test Beds

This Chapter provides the details of testbed and experiments conducted for this research. Testbeds, Test Cases, image files and proposed forensic methodology has been discussed in this chapter

3.1 Detail of Test image

A test image has been created for experimental purposes. Windows 10 Operating system was used. Details are as under:

Victim Machine

The experiment was conducted on Windows 10, Intel i5 processor, 4GB Ram, and 520 GB hard disk.

Forensic machine

The investigation was conducted on Windows 10, Intel i7 processor, 8GB Ram and 2TB hard disk.

3.2 Proposed Forensic Methodology

The main purpose of this research is to find artifacts from volume shadow copy, especially from $Recycle.Bin and $logfile to prove the importance of the Volume Shadow Copy for the forensic investigators. For a complete analysis of the Volume Shadow Copy, two different case scenarios have been generated and analyzed with different volume shadow copies. Different files were created modified and deleted in order to generate evidence. After analyzing the volume shadow copy, results will be compared with existing work.

The experiment consists of two case scenarios and the experiment consists of the following steps: 1) Creation2) Extraction3) Analysis of the evidence and then comparison with the previously done work.

1st Case Scenario: In the first case scenario financial information of a client of a company has been stolen by their own employee. Later on, during investigation, it has been found that suspects have copied those stolen files on their personal computer. The forensic investigator has to dig out and find proof of stolen information. For this purpose, we will create one file from scratch and two files have been downloaded from the internet.

A file named as Account_info.txt has been created from scratch with information of the account holder.2nd File check_bank.jpg has been downloaded from the internet and placed it on the desktop of the suspect's computer. The 3rd file is Bank_Statement.jpg initially has been placed on desktop and shadow copy has been created, then these files were moved from desktop to the documents in Account_info folder. At last step Folder named as Account_info has been deleted from the system and another shadow copy has been created. All these four shadow copies will be manually analyzed to prove the importance of the shadow copies for the forensic investigators.

2nd Case Scenario: Health information of a patient has been stolen, now the investigator has to find out the evidence from suspect's computer. For this situation, three files Blood_Count.docx, Brain_Scan.jpg and Case history.pdf will be copied on desktop of the suspects computer. Blood_Count.docx has been modified and changes have been saved to the original file. Brain_scan has also been modified and saved as Brain_Scan.png. after making changes and using the files these files will be deleted from the suspects computer.

A forensic image of the suspect's computer has been taken with the help of FTK imager named as **Financial_001**. Four Shadow Copies has been created for the

Case1:

Volume Shadow Copy I: Contains the shadow copy of the whole system after copying the stolen information into the suspect's computer. Initially, files were saved on desktop.

Volume Shadow Copy II: The second shadow was triggered when the files were moved into the new folder account-info created on the desktop.

Volume Shadow Copy III: the third shadow was created when the folder Account_info was moved to the Document folder

Volume Shadow Copy IV: Fourth shadow copy has been creating when the folder Account_info has been deleted. In order to traces, the changes shadow copy has been composed using the steps discussed in section 3.3

3.3 Experimental Testbeds

A forensic image of the system has been taken with the help of FTK Imager. After successfully taken the image it was analyzed by WinHex18 academia version and for verification purposes, automated analysis with the help of different freely available forensic investigation tools has been used which gives the support to the volume shadow copy. To conduct the experiment following tools and technologies have been used.

3.3.1 FTK Imager

FTK (Forensic Tool Kit) imager is a tool to create a forensic image of the disk as a whole or in parts that may be reconstructed at the end by Access Data. It also creates MD5 Hash values of the image for verification purposes. [18]

3.3.2 WinHex

In this research WinHex, 15.2 Academia version has been used. Winhex is a universal hex editor which is helpful in computer forensic investigation, low-level data extraction. It gives the facility to carve the data. Extract and analyze all kinds of files. It gives the facility of refining volume shadows which automatically mount the Volume Shadow Copy to help the investigator. Refining volume shadow feature is not included in academia version, so we investigate our case without using this feature. In this investigation academia version has been used and it does not include refine volume shadow feature [19].

3.4 Proposed methodology of reading data from Volume Shadow Copy of Windows 10

Complete workflow of the research has been shown in fig 8.

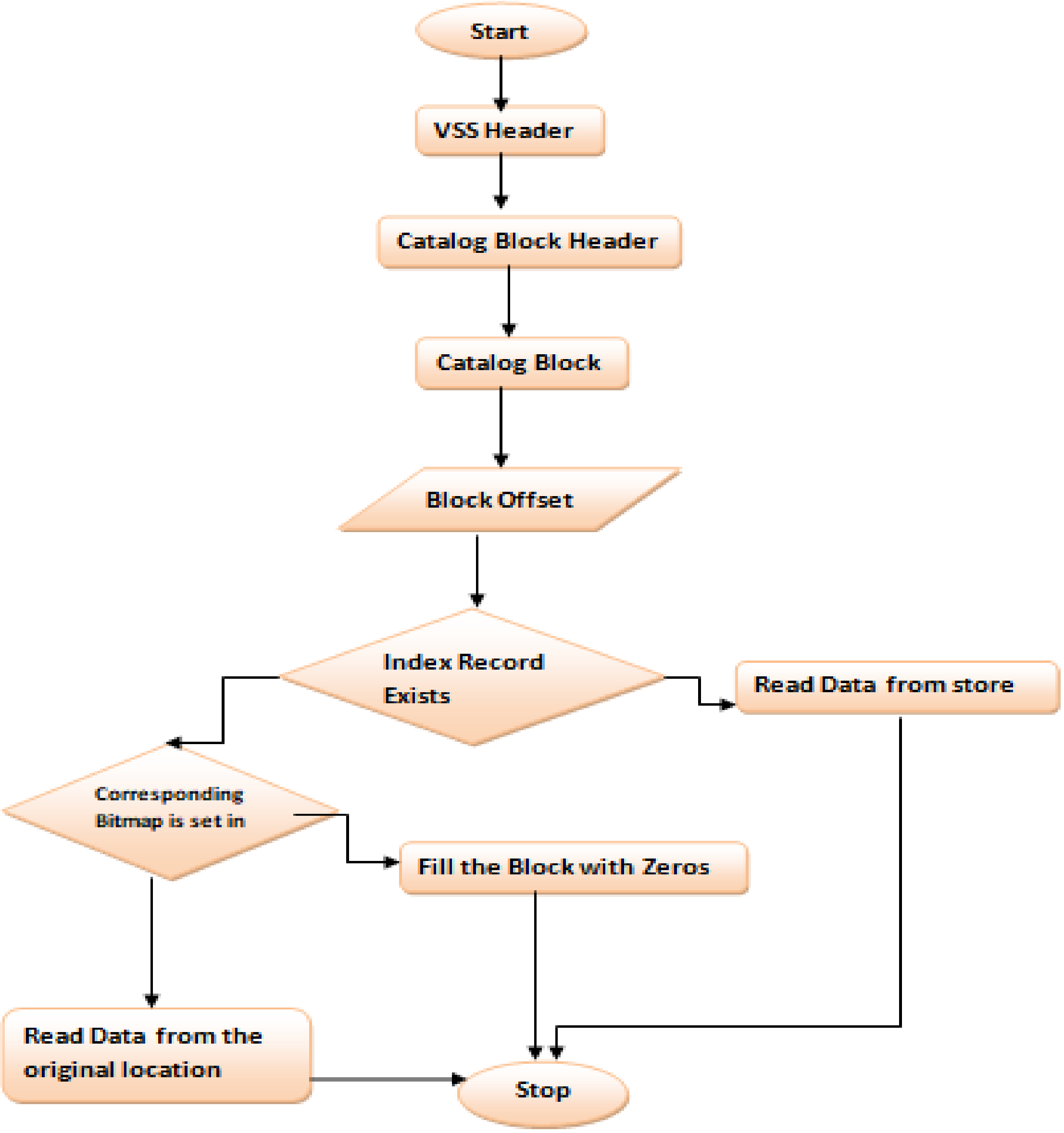


Fig8: Flowchart of reading data from Volume Shadow Copy

Chapter 4

$LogFile and $Recycle.Bin Analysis

There is a possibility that no artifacts of a deleted file exist in $MFT as it can be overwritten by new files. As all Volume Shadow Copies starts with $logfile which provides crucial information about the evidence. This research paper [10] also focuses on $logfile and discuss it in detail. $logFile has been discussed in detail which includes all types of records, the structure of the records and all information which is logged in them.

From the first 16 entries, $logfile is situated in a 2ndposition [11]. $logFile is a value-based log, recording changes of the NTFS file system. Default page size for Records is 4096 bytes or 0x1000 (which can be increased or decreased). Each record has a unique LSN ($logfile Sequence Number) which increments each time the file has been used. Log File Sequence Number is used to correlate file record of $logfile and $MFT. Each $MFT file contains LSN from 0x08-0x0F [11].

Two transactions are used by NTFS to complete filing tasks: In first transaction, files have been updated and in second transaction attributes of the $MFT has been updated. If the system fails after the first transaction, then the $Log file is used to recover the system. Every transaction is recorded in a $Log file. Every activity, e.g. renaming of a file, deleting a file have many transactions linked to complete the transaction. Therefore, a chain of operational records has been generated in $logfile to complete that activity. Because of this reason $logfile is important for forensics. Operation records of $logfile sustain data before transaction (for restoration/rollback/Undo) and data after transaction (Redo). Incase of Renaming a file following type of information is maintained [12].

4.1 Types of Log Records.

There are two types of records are recorded in two primary zones; Restart Area Records and Logging Area Records [13].

4.1.1 **Restart Area Records**. In this area, two records have been stored and both starting with “RSTR” each of length 0x1000.The 2nd record is the copy of the first record. Current LSN record in this record holds the information of last operation record. Structure of Restart area record is shown in table [13].

4.1.2 **Logging Area Records**. Real operational records are stored in the Logging are [13]. Normal page and buffer page area are two main divisions of the Logging area record. Logging area is divided into buffer Page Area and Normal Page. Buffer Page Area (0x200 to 0x4000) consists of first two pages. The second page is the copy of the first page. Last operation record is stored in Buffer Page area. Older record is pushed into the normal page area when Buffer Page area has been full. Normal Page area extends from 0x4000 to end of $logfile shown in the table7.

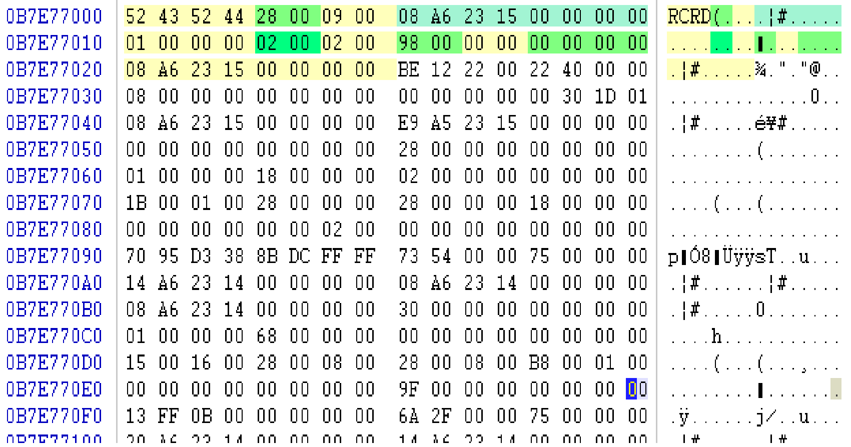
Structure of Page. Every page in the Logging area consists of a header followed by more than one operational records. Page header includes data of that page and its structure is shown in table 8 [13] with an example shown in fig 9.

Table 7: Start of the Log record

0x00	“RSTR” After checkdisk,its “CHKD”		UpdateSeq.ArrayOffset	UpdateSeq.ArraySize
	Check DiskLSN (Multi Sectorheader,allzero less “RSTR”changeto “CHKD”,lastLSNfoundbycheck disk)			
0x10	System PageSize (fatalerror if System PageSize≠Logpage)		Log Page Size	
	RestartArea Offset (from“RSTR”)	Minor Version (-1= beta,0=Transition, 1=updateseq. sp.)	Major Version (-1= beta,0=Transition, 1=updateseq. sp.)	
0x20	UpdateSeq.Array			
0x30	CurrentLSN (currentlogicalendof the logfileto facilitaterestart)			
	LogClient (maxclient sp.for thislog file)	ClientFreeList	ClientIn-useList	Flags
0x40	Seq.Number bits		RestartAreaLength	ClientArrayOffset (fromthe startof thisstructure)
	FileSize of $Logfile			
0x50	LastLSNdatalength(excludingRestartpageheader)		RecordPage Header Length	Log PageDataOffset
	RestartOpenLogCount (logfileopencount, todeterminethe changein thedisk e.g. remounting)		Padding	
0x06	Log clientArray (ClientData)			
	(Cont.)Log clientArray (ClientData)			
0x07	OldestLSN (Required tobein the logfilebythisclient)			
	ClientRestartLSN (LSNoflatestclient restartareawritten to thedisk,generallyCurrentLSN))			
0x08	Previous client(0xFF meansnoclient)	Nextclient (0xFFmeans no client)	Seq.Number(incremented on recordre-use)	Align Word (alignment field)
	Alignworld(alignt heentire record)		ClientNamelength (always 8)	
0x90	Client name (NTFSwithremainingbytes setto zero)			

Table8: Page Header of operational Record

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E
0x00	"RCRD" (signature)				Update Sequence Offset		Update Sequence Count		Last LSN orFile Offset (Last overallLSN on the page, including overlapping LSNs to next page)						
0x10	Flags				Page Count (No. of pagesused for transaction run)		Page Position (Current page no.)		Next Record Offset (Next LSN on the page)		Word Align		DWordAlign		
0x20	Last EndLSN (Last completedLSN on the page)														
0x30	Update SequenceArray (Arraycontaining theupdate seq. numberforreplacement First,two bytes of thevalue isthe UpdateSeqValue, usedevery512 bytes Followed by updatesequencearrays fora number of two-bytegroups definedinupdatedseq.countless 1)														



```
0B7E77000  52 43 52 44 28 00 09 00  08 A6 23 15 00 00 00 00  RCRD(....¦#.....
0B7E77010  01 00 00 00 02 00 02 00  98 00 00 00 00 00 00 00  ........▌.......
0B7E77020  08 A6 23 15 00 00 00 00  BE 12 22 00 22 40 00 00  .¦#.....¾.".\"@..
0B7E77030  08 00 00 00 00 00 00 00  00 00 00 00 00 30 1D 01  .............0..
0B7E77040  08 A6 23 15 00 00 00 00  E9 A5 23 15 00 00 00 00  .¦#.....é¥#.....
0B7E77050  00 00 00 00 00 00 00 00  28 00 00 00 00 00 00 00  ........(.......
0B7E77060  01 00 00 00 18 00 00 00  02 00 00 00 00 00 00 00  ................
0B7E77070  1B 00 01 00 28 00 00 00  28 00 00 00 18 00 00 00  ....(...(.......
0B7E77080  00 00 00 00 00 00 02 00  00 00 00 00 00 00 00 00  ................
0B7E77090  70 95 D3 38 8B DC FF FF  73 54 00 00 75 00 00 00  p▌Ó8▌ÜÿÿsT..u...
0B7E770A0  14 A6 23 14 00 00 00 00  08 A6 23 14 00 00 00 00  .¦#......¦#.....
0B7E770B0  08 A6 23 14 00 00 00 00  30 00 00 00 00 00 00 00  .¦#.....0.......
0B7E770C0  01 00 00 00 68 00 00 00  00 00 00 00 00 00 00 00  ....h...........
0B7E770D0  15 00 16 00 28 00 08 00  28 00 08 00 B8 00 01 00  ....(...(...¸...
0B7E770E0  00 00 00 00 00 00 00 00  9F 00 00 00 00 00 00 00  ........▌.......
0B7E770F0  13 FF 0B 00 00 00 00 00  6A 2F 00 00 75 00 00 00  .ÿ......j/..u...
```

Figure 9:Example of Page Header of $LogFile

Two main records have recordedis found in transaction Operation General record and Check Point Record, which is divided into commit and update record [13].In cases of system, failure recovery is made using Check point records and Restart area contain their LSNs. So, this can be taken as stable position of the system before the start of any transaction. Transactional updates have been traced and placed in Update Record. Last transactional record has been saved in Commit record. These are later

Converted to Check Point Record . Check Point Records and General Records have the same structure and carries necessary information to performed o run do (rollback) operation [25].

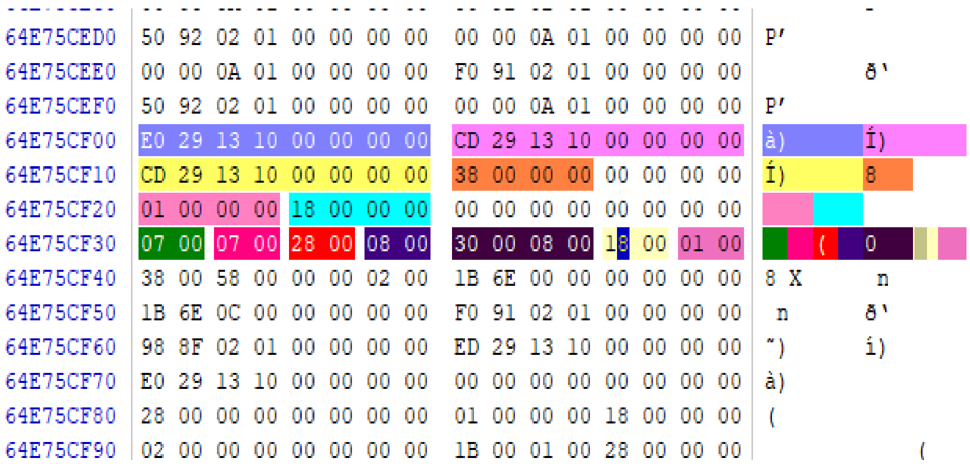
Table9: Operational Record structure of $logfile

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th></tr>
<tr><td>0x00</td><td colspan="8">Current LSN</td></tr>
<tr><td></td><td colspan="8">ClientPreviousLSN(forbacklinkingofthisrecord)</td></tr>
<tr><td>0x10</td><td colspan="8">ClientUndoLSN
(incaseoferrorrecovery,usuallythesameaspreviousLSN)</td></tr>
<tr><td></td><td colspan="4">ClientDataLength
(from“RedoOP”,thesizeofRecord)</td><td colspan="4">ClientID
(ownerofthisrecord)</td></tr>
<tr><td>0x20</td><td colspan="4">RecordType
(0x02forCheck PointRecord,0x01 forthegeneralRecord)</td><td colspan="4">TransactionID
(usedexternallybythe client (transactionmanager)togrouplogfileentries)</td></tr>
<tr><td></td><td colspan="2">PageoverflowFlags forthisrecord
(0x01-recordoverflows thepage,
0x00–nooverflow)</td><td colspan="6">Padding</td></tr>
<tr><td>0x30</td><td colspan="2">Redo OP
(code-checkcodesheet)</td><td colspan="2">UndoOP
(code-checkcodesheet)</td><td colspan="2">Redo Offset
(start ofRedodatafrom“Redo</td><td colspan="2">RedoLength</td></tr>
<tr><td></td><td colspan="2">UndoOffset
(startofundodatafrom“ RedoOP”
=RedoOffset+RedoLength)</td><td colspan="2">UndoLength</td><td colspan="2">TargetAttributeoffset
(fromRedoOP,thelocation oftarget VCN)</td><td colspan="2">LCNstoFollow
(0x01-Thereisthenextrecord)
,
0x00-</td></tr>
<tr><td>0x40</td><td colspan="2">AttributeOffset
(offsetinsideMFTrecordforappliedRedo/Undodata,if the changeaffectsanMFTrecord,otherwise0x00)</td><td colspan="2">OffsetwithinAttribute
(theoffsetofpointapplied Redo/Undodatawithinthe attributein MFTrecordorwithin the cluster)</td><td colspan="2">MFTCluster Index
(Incaseofoperation for MFTrecord,thelocationof recordappliedRedo/Undo datawithinthe cluster
0000-first,0002-Second,0004-third,0006-</td><td colspan="2">padding</td></tr>
<tr><td></td><td colspan="4">TargetVCN
(VCNof“$MFT”fileappliedRedo/Undodata,tobeappliedinconjunctionwith“MFTClusterIndex”
MultiplyVCNwithBytes/clustertoreach correspondingbytenumberandthencountthenumber ofrecordsasmentionedinMFTClusterIndextoreach thecorrespondingMFTrecord)</td><td colspan="4">padding</td></tr>
<tr><td>0x50</td><td colspan="4">TargetLCN
(LCNofthediskappliedRedo/Undodata,tobeappliedinconjunctionwith“MFTClusterIndex”
MultiplyLCNwithBytes/clustertoreach correspondingbytenumberandthencountthenumber ofrecordsinthatclusterasmentionedinMFTCluster Indextoreachthecorresponding MFTentry
IncaseofbitsofBitMap,firstfourbytesareMFTRecord Numberofthefile)</td><td colspan="4">padding</td></tr>
<tr><td>Tillthe endof data</td><td colspan="8">Data</td></tr>
</table>

Operations cheat sheet for Redo/Undo Codes. OperationCodeSheetforcodesmentionedin0x30-0x33in table 9 is listed in Table10 [13].

Table 10 -Redo/Undo Operation Code Sheet of $LogFile

Code	Operation	Code	Operation
0x00	Noop	0x0E	Add Index Entry Allocation
0x01	CompensationLog Record	0x0F	Delete Index Entry Allocation
0x02	Initialize File RecordSegment	0x12	Set Index EntryVCNAllocation
0x03	DeallocateFile RecordSegment	0x13	Update File Name Root
0x04	WriteEndof File RecordSegment	0x14	Update File NameAllocation
0x05	Create Attribute	0x15	Set Bits in Nonresident BitMap
0x06	Delete Attribute	0x16	Clear Bits in Nonresident BitMap
0x07	Update Resident Value	0x19	PrepareTransaction
0x08	Update NonresidentValue	0x1A	Commit Transaction
0x09	Update MappingPairs	0x1B	Forget Transaction
0x0A	Delete Dirty Clusters	0x1C	Open NonresidentAttribute
0x0B	Set NewAttributeSizes	0x1F	DirtyPage Table Dump
0x0C	Add Index Entry Root	0x20	Transaction Table Dump
0x0D	Delete Index EntryRoot	0x21	Update RecordDataRoot



```
64E75CED0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P’
64E75CEE0  00 00 0A 01 00 00 00 00  F0 91 02 01 00 00 00 00          ð'
64E75CEF0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P’
64E75CF00  E0 29 13 10 00 00 00 00  CD 29 13 10 00 00 00 00  à)      Í)
64E75CF10  CD 29 13 10 00 00 00 00  38 00 00 00 00 00 00 00  Í)      8
64E75CF20  01 00 00 00 18 00 00 00  00 00 00 00 00 00 00 00
64E75CF30  07 00 07 00 28 00 08 00  30 00 08 00 18 00 01 00        ( 0
64E75CF40  38 00 58 00 00 00 02 00  1B 6E 00 00 00 00 00 00  8 X      n
64E75CF50  1B 6E 0C 00 00 00 00 00  F0 91 02 01 00 00 00 00   n      ð'
64E75CF60  98 8F 02 01 00 00 00 00  ED 29 13 10 00 00 00 00  ˜)      í)
64E75CF70  E0 29 13 10 00 00 00 00  00 00 00 00 00 00 00 00  à)
64E75CF80  28 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  (
64E75CF90  02 00 00 00 00 00 00 00  1B 00 01 00 28 00 00 00              (
```

Figure 10: Example of Operation Record of $LogFile

4.2 $Recycle.Bin

$Recycle.Bin holds crucial data or evidence for the forensic investigator. Every user has his hown private bin. $Recycle.Bin folder placed inside the SID folder followed by a string which is unique for each user. Microsoft Developer Network (2009) has explained that the unique SID of each user is an alpha-numeric used to uniquely identifya user. When any user deletes a file, it will be stored in his SID folder. If there are three users are using the system will have three SID folders. One user without any permission cannot see the Bin of another user [15].

As Volume Shadow Copy generates a differential backups deleted file can be accessed by the Volume Shadow Copy if the Shadow is created before emptying the $Recycle.Bin. All activities will be maintained into the $Logfile which has also been copied into the Shadow Copy. Inorder to carryout this research it is important to understand the $Recycle. Bin structure completely.[15]

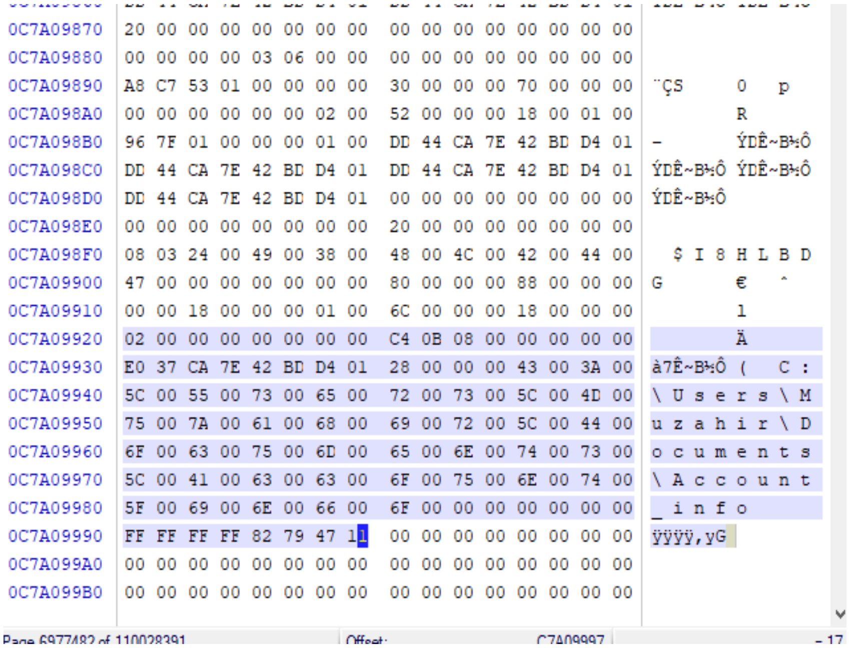
Whenever a file is deleted it produces two files:$I and $R files followed by the same six alphanumeric string. $I<abc>.<ext> contains the metadata about the deleted file and $R<abc>.<ext> holds the actual deleted file. Whenever a file is deleted its "deleted and "created" timestamps get associated with the file $Recycle.Bin is a great source of evidence for the forensic investigator.[3][15][16]

Structure of $I file is shown in Table:

Table 11:$I structure

0	8	Header(0x02)
8	16	Deleted file size
16	8	Deleted timestamp
24	4	File Name Length
28	Variable length	Filename and path

Example of $I structure of deleted folder named Account_info is shown in Fig:11



```
0C7A09870  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0C7A09880  00 00 00 00 03 06 00 00  00 00 00 00 00 00 00 00
0C7A09890  A8 C7 53 01 00 00 00 00  30 00 00 00 70 00 00 00  ¨ÇS     0   p
0C7A098A0  00 00 00 00 00 00 02 00  52 00 00 00 18 00 01 00          R
0C7A098B0  96 7F 01 00 00 00 01 00  DD 44 CA 7E 42 BD D4 01  –       ÝDÊ~B½Ô
0C7A098C0  DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
0C7A098D0  DD 44 CA 7E 42 BD D4 01  00 00 00 00 00 00 00 00  ÝDÊ~B½Ô
0C7A098E0  00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
0C7A098F0  08 03 24 00 49 00 38 00  48 00 4C 00 42 00 44 00    $ I 8 H L B D
0C7A09900  47 00 00 00 00 00 00 00  80 00 00 00 88 00 00 00  G       €   ˆ
0C7A09910  00 00 18 00 00 00 01 00  6C 00 00 00 18 00 00 00          l
0C7A09920  02 00 00 00 00 00 00 00  C4 0B 08 00 00 00 00 00          Ä
0C7A09930  E0 37 CA 7E 42 BD D4 01  28 00 00 00 43 00 3A 00  à7Ê~B½Ô (   C :
0C7A09940  5C 00 55 00 73 00 65 00  72 00 73 00 5C 00 4D 00  \ U s e r s \ M
0C7A09950  75 00 7A 00 61 00 68 00  69 00 72 00 5C 00 44 00  u z a h i r \ D
0C7A09960  6F 00 63 00 75 00 6D 00  65 00 6E 00 74 00 73 00  o c u m e n t s
0C7A09970  5C 00 41 00 63 00 63 00  6F 00 75 00 6E 00 74 00  \ A c c o u n t
0C7A09980  5F 00 69 00 6E 00 66 00  6F 00 00 00 00 00 00 00  _ i n f o
0C7A09990  FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG
0C7A099A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0C7A099B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00

Page 6977482 of 110028391        Offset:        C7A09997        = 17
```

Fig 11: Example of $I file structure of Account-info folder

At location ***0x0C7A09920*** is header and its value is 2 in Windows 10. From ***0x0C7A09928-0x0C7A9935*** is ***080BC4 = 514 KB*** size of the deleted folder. From ***0C7A9930-0C7A9937*** is time stamp for deleted folder. After converting it date time stampE037CA7E42BDD401=***2/5/2019 12:05pm*** *using convertor***.** At the Offset **0x18**timestamp start of the file path described in table 12.

Table 12: Artifacts of $I file of deleted Folder Account_Info

Offset	**Remarks**	**Findings/Artifacts**
0x0C7A09920	Header	02
0x0C7A09928-35	Size	514 KB
0x0C7A09930-37	Time stamp	2/5/2019 12:05pm
0x0C7A09938-97	Path of the deleted file	C:\User\Muzahir\Documents\Account_Info

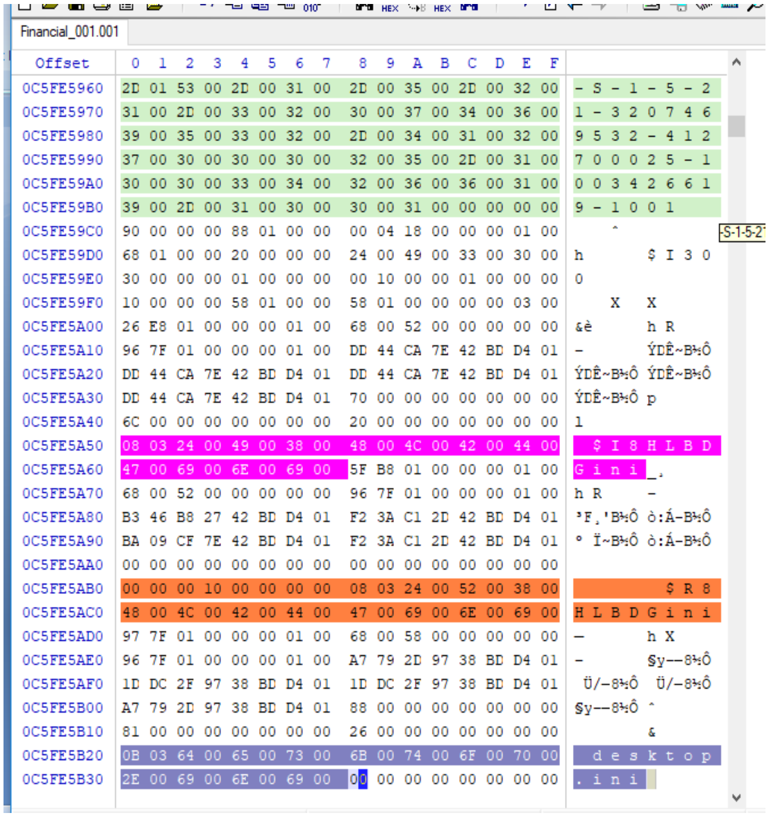
After the deletion, the two files have been generated $I and $R [15]. Both files named as $I8HLBDG for metadata of the file and $R8HBDG for the actual deleted file.

All SID folders also have index attribute files which holds the metadata of all the files contained in those folders. After analyzing the shadow

copy at offset C5FE5800 I have located file ***S-1-5-21-3207469532-412700025-1003426619-1001~$I30 (90)*** that is SID of $Recycle.Bin of the user. This file contains all the deleted files and folders' metadata.[17].$I30 is an index attribute which implements the B-Tree formation which keeps a record of the deleted or overwritten files. As in MFT all the files are not deleted/removed completely in index record all the tree nodes also not deleted only marked as deleted using their $Bitmap entry. It is not guaranteed that all the files which are presented in index record are present in volume but with the help of index record file we can find in long-lasting deleted or overwritten files with their metadata which includes [17]

- ✓ Name of the file
- ✓ Parent folders
- ✓ Creation time
- ✓ Deletion time
- ✓ MFT change time
- ✓ Access time

An example of an index attribute file for SID ***S-1-5-21-3207469532-412700025-1003426619-1001~$I30***been shown in fig 11.



Financial_001.001

```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0C5FE5960 2D 01 53 00 2D 00 31 00  2D 00 35 00 2D 00 32 00  - S - 1 - 5 - 2
0C5FE5970 31 00 2D 00 33 00 32 00  30 00 37 00 34 00 36 00  1 - 3 2 0 7 4 6
0C5FE5980 39 00 35 00 33 00 32 00  2D 00 34 00 31 00 32 00  9 5 3 2 - 4 1 2
0C5FE5990 37 00 30 00 30 00 30 00  32 00 35 00 2D 00 31 00  7 0 0 0 2 5 - 1
0C5FE59A0 30 00 30 00 33 00 34 00  32 00 36 00 36 00 31 00  0 0 3 4 2 6 6 1
0C5FE59B0 39 00 2D 00 31 00 30 00  30 00 31 00 00 00 00 00  9 - 1 0 0 1
0C5FE59C0 90 00 00 00 88 01 00 00  00 04 18 00 00 00 01 00      ˆ
0C5FE59D0 68 01 00 00 20 00 00 00  24 00 49 00 33 00 30 00  h       $ I 3 0
0C5FE59E0 30 00 00 00 01 00 00 00  00 10 00 00 01 00 00 00  0
0C5FE59F0 10 00 00 00 58 01 00 00  58 01 00 00 00 00 03 00      X   X
0C5FE5A00 26 E8 01 00 00 00 01 00  68 00 52 00 00 00 00 00  &è      h R
0C5FE5A10 96 7F 01 00 00 00 01 00  DD 44 CA 7E 42 BD D4 01  –       ÝDÊ~B½Ô
0C5FE5A20 DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
0C5FE5A30 DD 44 CA 7E 42 BD D4 01  70 00 00 00 00 00 00 00  ÝDÊ~B½Ô p
0C5FE5A40 6C 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00  l
0C5FE5A50 08 03 24 00 49 00 38 00  48 00 4C 00 42 00 44 00    $ I 8 H L B D
0C5FE5A60 47 00 69 00 6E 00 69 00  5F B8 01 00 00 00 01 00  G i n i _¸
0C5FE5A70 68 00 52 00 00 00 00 00  96 7F 01 00 00 00 01 00  h R     –
0C5FE5A80 B3 46 B8 27 42 BD D4 01  F2 3A C1 2D 42 BD D4 01  ³F¸'B½Ô ò:Á-B½Ô
0C5FE5A90 BA 09 CF 7E 42 BD D4 01  F2 3A C1 2D 42 BD D4 01  º Ï~B½Ô ò:Á-B½Ô
0C5FE5AA0 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0C5FE5AB0 00 00 00 10 00 00 00 00  08 03 24 00 52 00 38 00              $ R 8
0C5FE5AC0 48 00 4C 00 42 00 44 00  47 00 69 00 6E 00 69 00  H L B D G i n i
0C5FE5AD0 97 7F 01 00 00 00 01 00  68 00 58 00 00 00 00 00  —       h X
0C5FE5AE0 96 7F 01 00 00 00 01 00  A7 79 2D 97 38 BD D4 01  –       §y-—8½Ô
0C5FE5AF0 1D DC 2F 97 38 BD D4 01  1D DC 2F 97 38 BD D4 01   Ü/—8½Ô  Ü/—8½Ô
0C5FE5B00 A7 79 2D 97 38 BD D4 01  88 00 00 00 00 00 00 00  §y-—8½Ô ˆ
0C5FE5B10 81 00 00 00 00 00 00 00  26 00 00 00 00 00 00 00          &
0C5FE5B20 0B 03 64 00 65 00 73 00  6B 00 74 00 6F 00 70 00    d e s k t o p
0C5FE5B30 2E 00 69 00 6E 00 69 00  00 00 00 00 00 00 00 00  . i n i
```

Fig12: $I30Index Attribute of the SID

🟧 Name of the file SID of the user

🟪 $I file

🟫 $R file

🟦 Desktop.ini is included in all SID folders

This chapter highlights the importance and structure of the $logfile and $Recycle.bin for the Forensic Investigator as they contain valuable information to dig out the evidence to prove in court of law.

Chapter5

Results and Analysis

In this chapter all the artifacts, findings of the experiment and results have been discussed. Volume Shadow Copy is incremental backups of 16 KB and created automatically after specific time intervals or manually whenever the user wants. By investigating Volume Shadow Copy one can view the previous version of the files. By following all the steps already discussed in Chapter II and given in the proposed forensic methodology we find the artifacts.

5.1 Composing Volume Shadow Copy from Forensic Image of Case Scenario1:

As discussed in Chapter III composing the Volume Shadow Copies is a multistep process step which includes store header, catalogs, descriptors etc. Following steps have been followed to construct a Volume Shadow Copy from forensic image for Case scenario 1(Stolen financial information) forensic image of the suspect's computer has been taken with the help of FTK imager named as **Financial_001**. Four Shadow Copies has been created for the Case1:

Volume Shadow Copy I: Contains the shadow copy of the whole system after copying the stolen information into the suspect's computer. Initially, files were saved on desktop.

Volume Shadow Copy II: The second shadow was triggered when the files were moved into the new folder account-info created on the desktop.

Volume Shadow Copy III: The third shadow was created when the folder Account_info was moved to the Document folder.

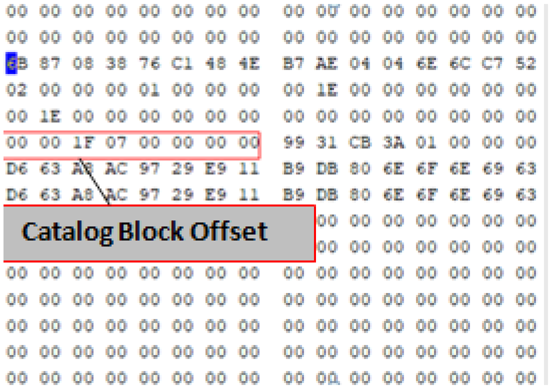
Volume Shadow Copy IV: Fourth shadow copy has been creating when the folder Account_info has been deleted

In order to traces, the changes shadow copy has been composed using the following steps.

5.1.1Presence of Volume Shadow Copy: In order to check whether the volume shadow copy is present in the system or not, catalog block offset from volume shadow header from location ***0x1E00***has been checked. After converting it into little Indian it was ***0x071F0000.*** It is the location of the first catalog block offset of the first shadow copy.

Table: Artifacts found from Volume Shadow Copy header and catalog block header

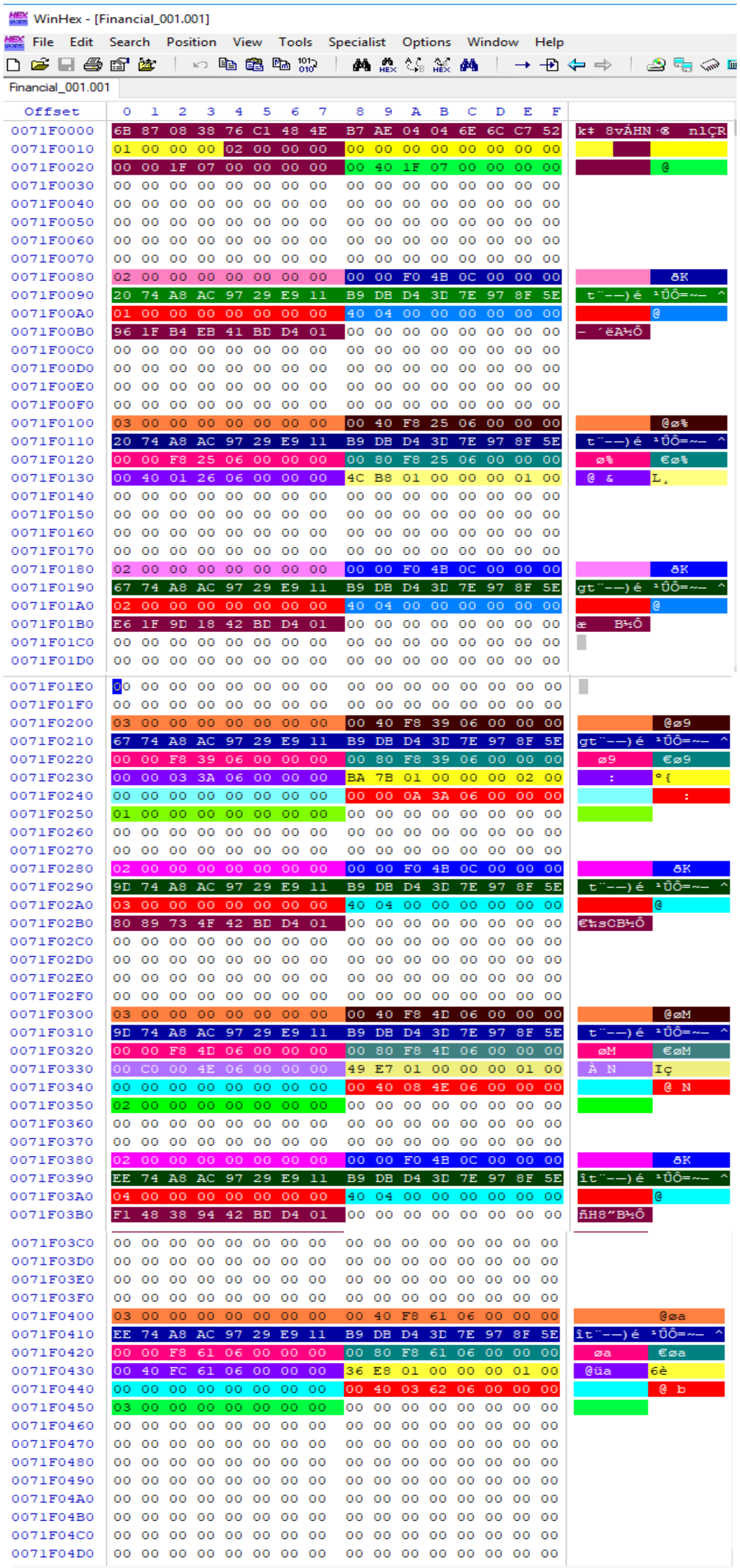
Offset	**Findings/Artifacts**	**Remarks**
0x1E00	VSS header	VSS Header always start from this location
0x71f0000	Catalog Block header	Contains the information of all shadows present in the system
0x071f0080	Entry type 0x02	Contains the metadata about the shadow copy Size ,Creation time, Shadow identifier
0x071f0100	Entry type 0x03	Contains the store offsets , store range offset ,store bitmap offset



```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52
02 00 00 00 01 00 00 00  00 1E 00 00 00 00 00 00
00 1E 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 1F 07 00 00 00 00  99 31 CB 3A 01 00 00 00
D6 63 A8 AC 97 29 E9 11  B9 DB 80 6E 6F 6E 69 63
D6 63 A8 AC 97 29 E9 11  B9 DB 80 6E 6F 6E 69 63
                            00 00 00 00 00 00 00
                            00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig13: Volume Shadow Copy Headers

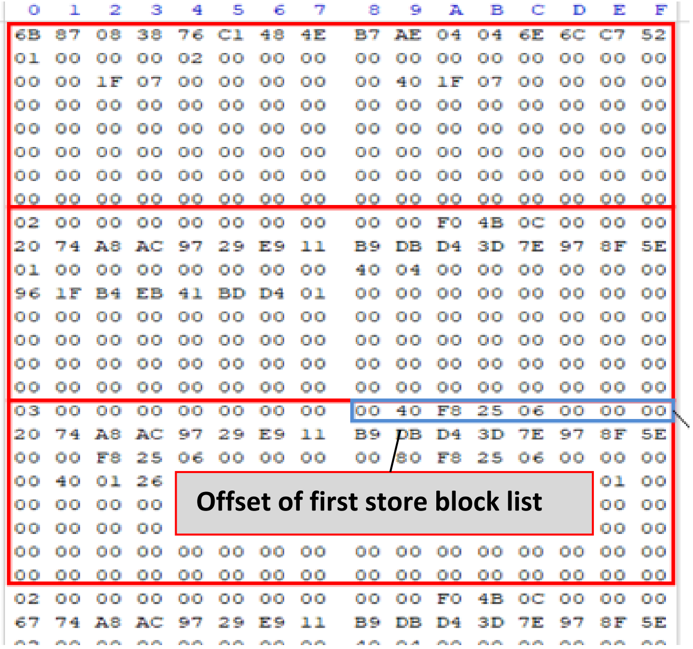
By accessing the Catalog Block from the above-mentioned Catalog Block header has been decoded as shown in fig 14. There are total 8 catalog entries against 4 shadow copies. Entry type 0x02 contains the metadata of the Volume Shadow Copy and entry type 0x03 contains data about the actual volume snapshots shown in fig 14. These two entry types are decoded to according to the Table 4 and Table5 described in chapter II.



```
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0071F0000  6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52   k‡ 8vÁHN·®  nlÇR
0071F0010  01 00 00 00 02 00 00 00  00 00 00 00 00 00 00 00
0071F0020  00 00 1F 07 00 00 00 00  00 40 1F 07 00 00 00 00           @
0071F0030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0080  02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00             ðK
0071F0090  20 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E    t¨¬—)é ¹ÛÔ=~— ^
0071F00A0  01 00 00 00 00 00 00 00  40 04 00 00 00 00 00 00           @
0071F00B0  96 1F B4 EB 41 BD D4 01  00 00 00 00 00 00 00 00   – ´ëA½Ô
0071F00C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F00D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F00E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F00F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0100  03 00 00 00 00 00 00 00  00 40 F8 25 06 00 00 00            @ø%
0071F0110  20 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E    t¨¬—)é ¹ÛÔ=~— ^
0071F0120  00 00 F8 25 06 00 00 00  00 80 F8 25 06 00 00 00     ø%      €ø%
0071F0130  00 40 01 26 06 00 00 00  4C B8 01 00 00 00 01 00    @ &     L¸
0071F0140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0180  02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00             ðK
0071F0190  67 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E   gt¨¬—)é ¹ÛÔ=~— ^
0071F01A0  02 00 00 00 00 00 00 00  40 04 00 00 00 00 00 00           @
0071F01B0  E6 1F 9D 18 42 BD D4 01  00 00 00 00 00 00 00 00   æ    B½Ô
0071F01C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F01D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F01E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F01F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0200  03 00 00 00 00 00 00 00  00 40 F8 39 06 00 00 00            @ø9
0071F0210  67 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E   gt¨¬—)é ¹ÛÔ=~— ^
0071F0220  00 00 F8 39 06 00 00 00  00 80 F8 39 06 00 00 00     ø9      €ø9
0071F0230  00 00 03 3A 06 00 00 00  BA 7B 01 00 00 00 02 00      :     º{
0071F0240  00 00 00 00 00 00 00 00  00 00 0A 3A 06 00 00 00              :
0071F0250  01 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0260  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0270  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0280  02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00             ðK
0071F0290  9D 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E    t¨¬—)é ¹ÛÔ=~— ^
0071F02A0  03 00 00 00 00 00 00 00  40 04 00 00 00 00 00 00           @
0071F02B0  80 89 73 4F 42 BD D4 01  00 00 00 00 00 00 00 00   €‰sOB½Ô
0071F02C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F02D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F02E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F02F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0300  03 00 00 00 00 00 00 00  00 40 F8 4D 06 00 00 00            @øM
0071F0310  9D 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E    t¨¬—)é ¹ÛÔ=~— ^
0071F0320  00 00 F8 4D 06 00 00 00  00 80 F8 4D 06 00 00 00     øM      €øM
0071F0330  00 C0 00 4E 06 00 00 00  49 E7 01 00 00 00 01 00    À N     Iç
0071F0340  00 00 00 00 00 00 00 00  00 40 08 4E 06 00 00 00            @ N
0071F0350  02 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0370  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0380  02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00             ðK
0071F0390  EE 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E   ît¨¬—)é ¹ÛÔ=~— ^
0071F03A0  04 00 00 00 00 00 00 00  40 04 00 00 00 00 00 00           @
0071F03B0  F1 48 38 94 42 BD D4 01  00 00 00 00 00 00 00 00   ñH8”B½Ô
0071F03C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F03D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F03E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F03F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0400  03 00 00 00 00 00 00 00  00 40 F8 61 06 00 00 00            @øa
0071F0410  EE 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E   ît¨¬—)é ¹ÛÔ=~— ^
0071F0420  00 00 F8 61 06 00 00 00  00 80 F8 61 06 00 00 00     øa      €øa
0071F0430  00 40 FC 61 06 00 00 00  36 E8 01 00 00 00 01 00    @üa     6è
0071F0440  00 00 00 00 00 00 00 00  00 40 03 62 06 00 00 00            @ b
0071F0450  03 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0460  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0470  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0480  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F0490  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F04A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F04B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F04C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0071F04D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig14: Catalog Block entries type 0x02 and 0x03

The offset of the first store block has been found and decoded from catalog block header as shown in fig 15.



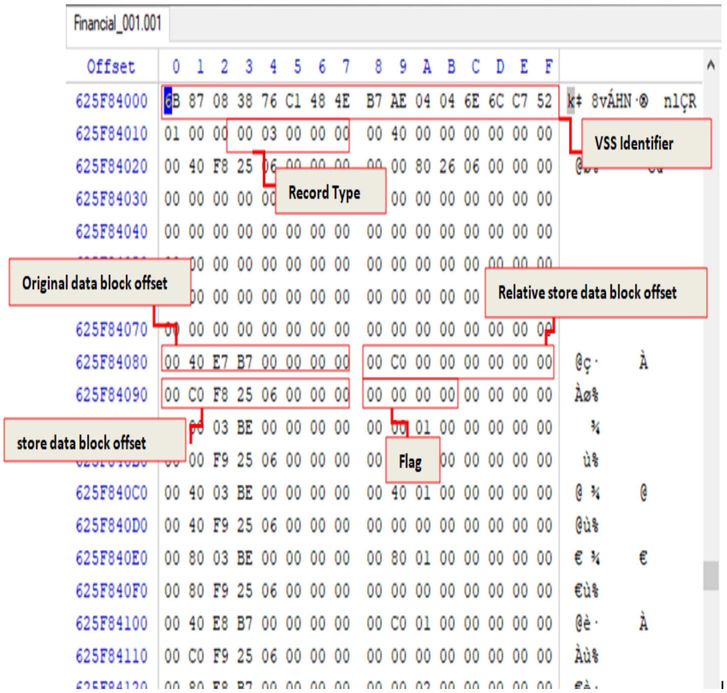
```
 0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52
01 00 00 00 02 00 00 00  00 00 00 00 00 00 00 00
00 00 1F 07 00 00 00 00  00 40 1F 07 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00
20 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E
01 00 00 00 00 00 00 00  40 04 00 00 00 00 00 00
96 1F B4 EB 41 BD D4 01  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
03 00 00 00 00 00 00 00  00 40 F8 25 06 00 00 00
20 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E
00 00 F8 25 06 00 00 00  00 80 F8 25 06 00 00 00
00 40 01 26                                 01 00
00 00 00 00                                 00 00
00 00 00 00                                 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
02 00 00 00 00 00 00 00  00 00 F0 4B 0C 00 00 00
67 74 A8 AC 97 29 E9 11  B9 DB D4 3D 7E 97 8F 5E
```

Offset of first store block list

Fig 15: Catalog block header and catalog entry types

5.1.2Locating Catalog Store Block List

From entry type 0x03 we locate the offset of first store block list. After converting it into little Endean we reached first store block list located at **0x625F84000** shown in fig16.



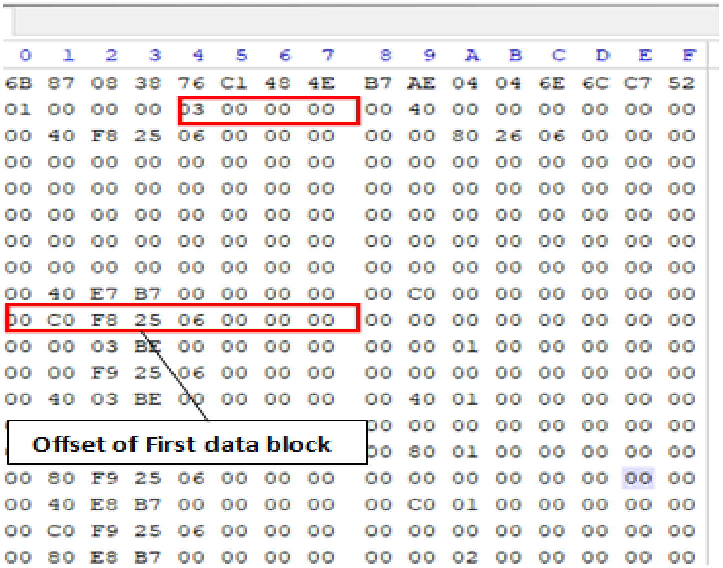
```
Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
625F84000  6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52  k‡ 8vÁHN·® nlÇR
625F84010  01 00 00 00 03 00 00 00  00 40 00 00 00 00 00 00
625F84020  00 40 F8 25 06 00 00 00  00 00 80 26 06 00 00 00
625F84030  00 00 00 00 00              00 00 00 00 00 00 00
625F84040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00  00 00 00 00 00
625F84070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F84080  00 40 E7 B7 00 00 00 00  00 C0 00 00 00 00 00 00  @ç·     À
625F84090  00 C0 F8 25 06 00 00 00  00 00 00 00 00 00 00 00  Àø%
              00 03 BE 00 00 00 00  00 00 01 00 00 00 00 00    ¾
              00 F9 25 06 00 00 00  00       00 00 00 00 00   ù%
625F840C0  00 40 03 BE 00 00 00 00  00 40 01 00 00 00 00 00  @ ¾     @
625F840D0  00 40 F9 25 06 00 00 00  00 00 00 00 00 00 00 00  @ù%
625F840E0  00 80 03 BE 00 00 00 00  00 80 01 00 00 00 00 00  € ¾     €
625F840F0  00 80 F9 25 06 00 00 00  00 00 00 00 00 00 00 00  €ù%
625F84100  00 40 E8 B7 00 00 00 00  00 C0 01 00 00 00 00 00  @è·     À
625F84110  00 C0 F9 25 06 00 00 00  00 00 00 00 00 00 00 00  Àù%
```

Fig 16: Shadow Store block list of first Shadow

Table :Artifacts found from store block list

Offset	**Findings/artifacts**	**Remarks**
0x625f84000	VSS Identifier	Volume shadow copy name
0x625f84014	Record type	Store block list
0x625f84080	Original data block offset	Data block of active volume
0x625f84090	Store data block offset	Offset of first volume shadow copy
0x625f84098	Flag value	Descriptor value 0=Normal descriptor

From store block list header type 0x03 offset of first store block is found ***0x0625f8000*** as shown in fig17.

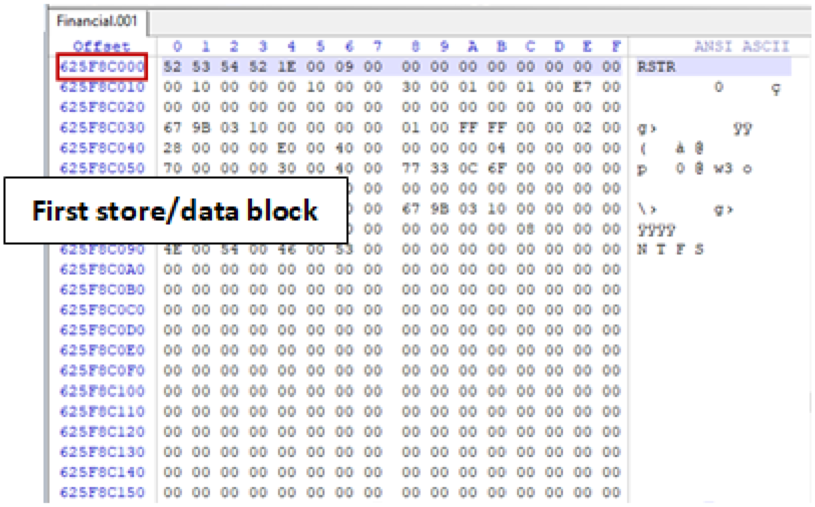


```
 0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52
01 00 00 00 03 00 00 00  00 40 00 00 00 00 00 00
00 40 F8 25 06 00 00 00  00 00 80 26 06 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 40 E7 B7 00 00 00 00  00 C0 00 00 00 00 00 00
00 C0 F8 25 06 00 00 00  00 00 00 00 00 00 00 00
00 00 03 BE 00 00 00 00  00 00 01 00 00 00 00 00
00 00 F9 25 06 00 00 00  00 00 00 00 00 00 00 00
00 40 03 BE 00 00 00 00  00 40 01 00 00 00 00 00
                         00 00 00 00 00 00 00 00
                         00 80 01 00 00 00 00 00
00 80 F9 25 06 00 00 00  00 00 00 00 00 00 00 00
00 40 E8 B7 00 00 00 00  00 C0 01 00 00 00 00 00
00 C0 F9 25 06 00 00 00  00 00 00 00 00 00 00 00
00 80 E8 B7 00 00 00 00  00 00 02 00 00 00 00 00
```

Offset of First data block

Fig 17: Shadow Store block list header

Shadow block list header is of type 0x03 and contains the store data block offset and flag descriptor value which is 0 indicates normal block descriptor.



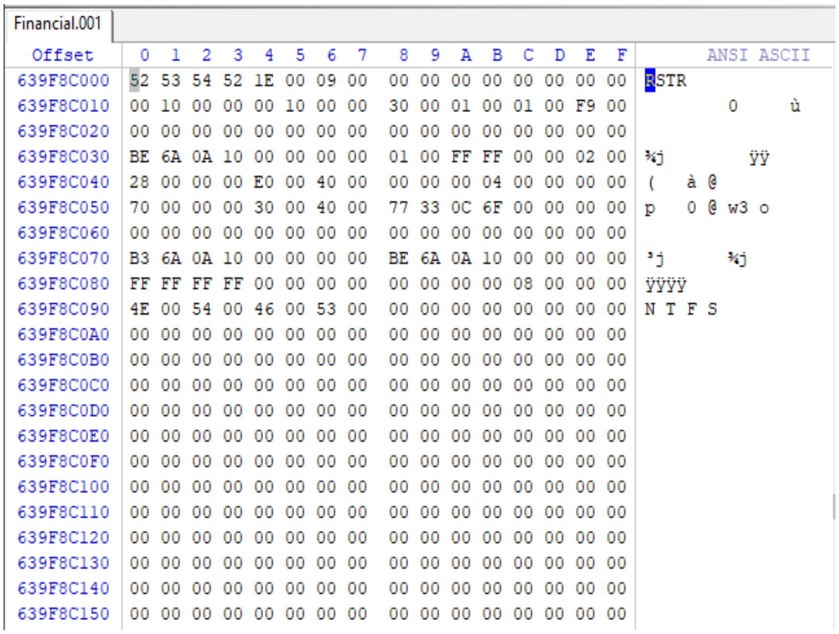
```
Financial.001
  Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
625F8C000   52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00   RSTR
625F8C010   00 10 00 00 00 10 00 00  30 00 01 00 01 00 E7 00         0     ç
625F8C020   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C030   67 9B 03 10 00 00 00 00  01 00 FF FF 00 00 02 00   g›        ÿÿ
625F8C040   28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00   (    à @
625F8C050   70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00   p    0 @ w3 o
                                 00  00 00 00 00 00 00 00 00
                                 00  67 9B 03 10 00 00 00 00   \›        g›
                                 00  00 00 00 00 08 00 00 00   ÿÿÿÿ
625F8C090   4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00   N T F S
625F8C0A0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0B0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0C0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0D0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0E0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C100   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C110   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C120   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C130   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C140   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C150   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

First store/data block

Fig 18: Snapshot of First data block

First data store indicating $logfile captured in 1st Volume Shadow Copy. It only saves the changes to the original $logfile. Hence the traces can also be traced from $logfile as well.

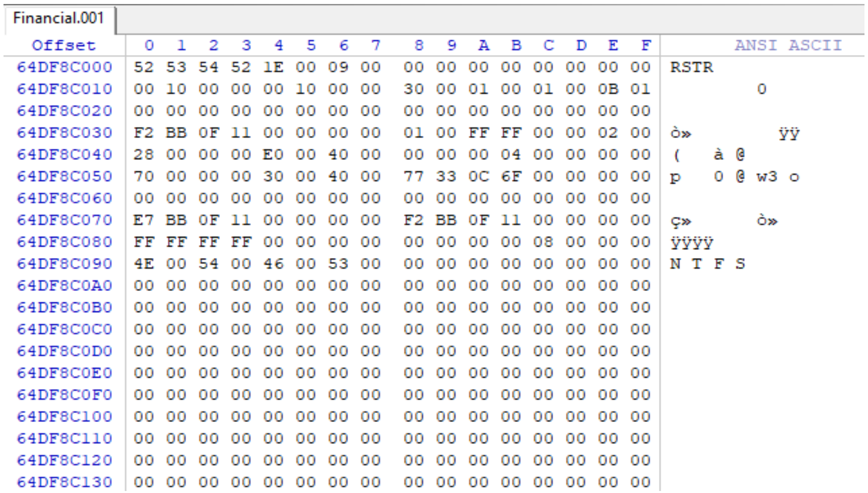
Second data store found at the location of ***0x639f84000***as shown in fig19.



```
Financial.001
  Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F      ANSI ASCII
639F8C000   52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00  RSTR
639F8C010   00 10 00 00 00 10 00 00  30 00 01 00 01 00 F9 00          0     ù
639F8C020   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C030   BE 6A 0A 10 00 00 00 00  01 00 FF FF 00 00 02 00  ¾j          ÿÿ
639F8C040   28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00  (   à @
639F8C050   70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00  p   0 @ w3 o
639F8C060   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C070   B3 6A 0A 10 00 00 00 00  BE 6A 0A 10 00 00 00 00  ³j      ¾j
639F8C080   FF FF FF FF 00 00 00 00  00 00 00 00 08 00 00 00  ÿÿÿÿ
639F8C090   4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00  N T F S
639F8C0A0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C0B0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C0C0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C0D0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C0E0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C0F0   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C100   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C110   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C120   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C130   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C140   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
639F8C150   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

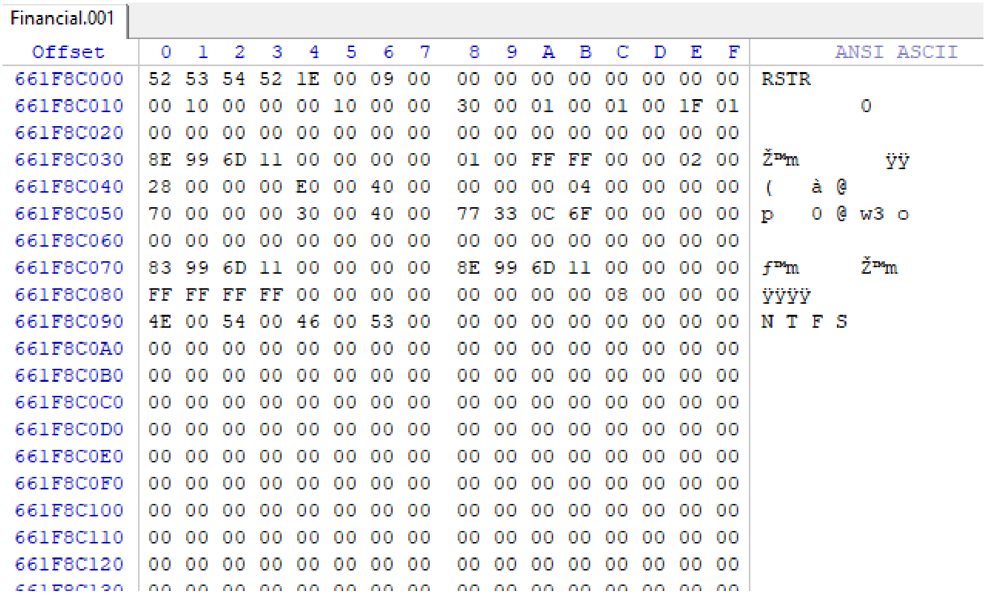
Fig: 19 Snapshot of Second Store: starts with $logfile

In the same way, third and fourth data stores have been traced from catalog block header entry type 0x03 and it is observed that theses all blocks start with $logfile shown in fig 20 and 21.



```
Financial.001
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
64DF8C000  52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00  RSTR
64DF8C010  00 10 00 00 00 10 00 00  30 00 01 00 01 00 0B 01          0
64DF8C020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C030  F2 BB 0F 11 00 00 00 00  01 00 FF FF 00 00 02 00  ò»          ÿÿ
64DF8C040  28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00  (   à @
64DF8C050  70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00  p   0 @ w3 o
64DF8C060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C070  E7 BB 0F 11 00 00 00 00  F2 BB 0F 11 00 00 00 00  ç»      ò»
64DF8C080  FF FF FF FF 00 00 00 00  00 00 00 00 08 00 00 00  ÿÿÿÿ
64DF8C090  4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00  N T F S
64DF8C0A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C0B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C0C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C0D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C0E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C0F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
64DF8C130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig:20 Snapshot of Third Store



```
Financial.001
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
661F8C000  52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00  RSTR
661F8C010  00 10 00 00 00 10 00 00  30 00 01 00 01 00 1F 01          0
661F8C020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C030  8E 99 6D 11 00 00 00 00  01 00 FF FF 00 00 02 00  Ž™m          ÿÿ
661F8C040  28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00  (   à @
661F8C050  70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00  p   0 @ w3 o
661F8C060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C070  83 99 6D 11 00 00 00 00  8E 99 6D 11 00 00 00 00  ƒ™m     Ž™m
661F8C080  FF FF FF FF 00 00 00 00  00 00 00 00 08 00 00 00  ÿÿÿÿ
661F8C090  4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00  N T F S
661F8C0A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C0B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C0C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C0D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C0E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C0F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
661F8C130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig:21 Snapshot of Fourth Store

The Data stores are manipulated in a way that the changes traced from the latest store first and then to the next latest and the oldest snapshot will be manipulated at the end. Every 16 KB block has to be manipulated to verify the modification of the original contents. Flag in index record was set as 0x00 in all four shadow copies which indicate normal block.

In case of normal block whole block is mapped to the original block to get the data. In order to generate a Volume Shadow Copy of the forensic investigator has to check all 16 KB blocks. If 16 KB block is not modified, then read data from the original position. If the index record of the block does not exist, then read the corresponding bit from the bitmap.[8]

5.2 Reading snapshot from store data:

BD=Block Descriptor *CV=Current Volume*

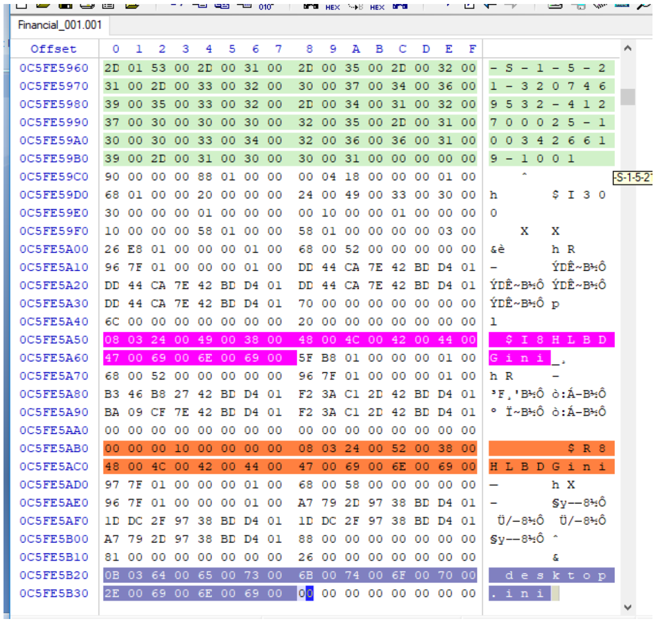
FD=Forward block descriptor *RBD=Reverse Block Descriptor*

```
If BD=1 then
    If BD=Over_Lay then
    Use overlay block descriptors
    If FD=1 and Next_store=1 then
    Read data from store   *Using relative store offset
    Else
    Read data from the original block
  Else
    IF next_store =1 then
    Read data from next store
    Else
    IF RBD =1 then
    Read data from CV
    Else If Active_store=most recent
    Read data from original Volume
```

5.3 Analysis of $Recycle.Bin

Whenever a file is deleted its two copies have been generated into the $Recycle.Bin. $R file and $I file with the same random integer values. If a file named hello.txt is deleted its two files $Igh4563 and $RIgh4563 files will be generated. $R contains the actual data of the file that has been deleted and $I file contains the metadata of the deleted file. The very first problem that a forensic investigator has to face is a large amount of data to be processed as 2 files are generated in case of deletion of 1 file [3].

After analyzing the shadow copy at offset ***0xC5FE5800*** I have located file ***S-1-5-21-3207469532-412700025-1003426619-1001~$I30 (90)*** that is SID of $Recycle.Bin ofthe user. This file contains the all the deleted files and folders' metadata.$I30 is an index attribute which implements the B-Tree formation which keeps a record of the deleted or overwritten files.AS in MFT all the files are not deleted/removed completely in index record all the tree nodes also not deleted only marked as deleted using their $Bitmap entry. Attribute Index has shown in fig 22.



```
Financial_001.001

  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
0C5FE5960  2D 01 53 00 2D 00 31 00  2D 00 35 00 2D 00 32 00  - S - 1 - 5 - 2
0C5FE5970  31 00 2D 00 33 00 32 00  30 00 37 00 34 00 36 00  1 - 3 2 0 7 4 6
0C5FE5980  39 00 35 00 33 00 32 00  2D 00 34 00 31 00 32 00  9 5 3 2 - 4 1 2
0C5FE5990  37 00 30 00 30 00 30 00  32 00 35 00 2D 00 31 00  7 0 0 0 2 5 - 1
0C5FE59A0  30 00 30 00 33 00 34 00  32 00 36 00 36 00 31 00  0 0 3 4 2 6 6 1
0C5FE59B0  39 00 2D 00 31 00 30 00  30 00 31 00 00 00 00 00  9 - 1 0 0 1
0C5FE59C0  90 00 00 00 88 01 00 00  00 04 18 00 00 00 01 00      ˆ
0C5FE59D0  68 01 00 00 20 00 00 00  24 00 49 00 33 00 30 00  h       $ I 3 0
0C5FE59E0  30 00 00 00 01 00 00 00  00 10 00 00 01 00 00 00  0
0C5FE59F0  10 00 00 00 58 01 00 00  58 01 00 00 00 00 03 00      X   X
0C5FE5A00  26 E8 01 00 00 00 01 00  68 00 52 00 00 00 00 00  &è      h R
0C5FE5A10  96 7F 01 00 00 00 01 00  DD 44 CA 7E 42 BD D4 01  –       ÝDÊ~B½Ô
0C5FE5A20  DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
0C5FE5A30  DD 44 CA 7E 42 BD D4 01  70 00 00 00 00 00 00 00  ÝDÊ~B½Ô p
0C5FE5A40  6C 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00  l
0C5FE5A50  08 03 24 00 49 00 38 00  48 00 4C 00 42 00 44 00    $ I 8 H L B D
0C5FE5A60  47 00 69 00 6E 00 69 00  5F B8 01 00 00 00 01 00  G i n i _¸
0C5FE5A70  68 00 52 00 00 00 00 00  96 7F 01 00 00 00 01 00  h R     –
0C5FE5A80  B3 46 B8 27 42 BD D4 01  F2 3A C1 2D 42 BD D4 01  ³F¸'B½Ô ò:Á-B½Ô
0C5FE5A90  BA 09 CF 7E 42 BD D4 01  F2 3A C1 2D 42 BD D4 01  º Ï~B½Ô ò:Á-B½Ô
0C5FE5AA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0C5FE5AB0  00 00 00 10 00 00 00 00  08 03 24 00 52 00 38 00            $ R 8
0C5FE5AC0  48 00 4C 00 42 00 44 00  47 00 69 00 6E 00 69 00  H L B D G i n i
0C5FE5AD0  97 7F 01 00 00 00 01 00  68 00 58 00 00 00 00 00  —       h X
0C5FE5AE0  96 7F 01 00 00 00 01 00  A7 79 2D 97 38 BD D4 01  –       §y-—8½Ô
0C5FE5AF0  1D DC 2F 97 38 BD D4 01  1D DC 2F 97 38 BD D4 01   Ü/—8½Ô  Ü/—8½Ô
0C5FE5B00  A7 79 2D 97 38 BD D4 01  88 00 00 00 00 00 00 00  §y-—8½Ô ˆ
0C5FE5B10  81 00 00 00 00 00 00 00  26 00 00 00 00 00 00 00          &
0C5FE5B20  0B 03 64 00 65 00 73 00  6B 00 74 00 6F 00 70 00    d e s k t o p
0C5FE5B30  2E 00 69 00 6E 00 69 00  00 00 00 00 00 00 00 00  . i n i
```

S-1-5-21

Fig:22 $I 30 index attributes of the file

🟧	Name of the file SID of the user
🟪	$I file
🟫	$R file
🟦	Desktop.ini is included in all SID folders

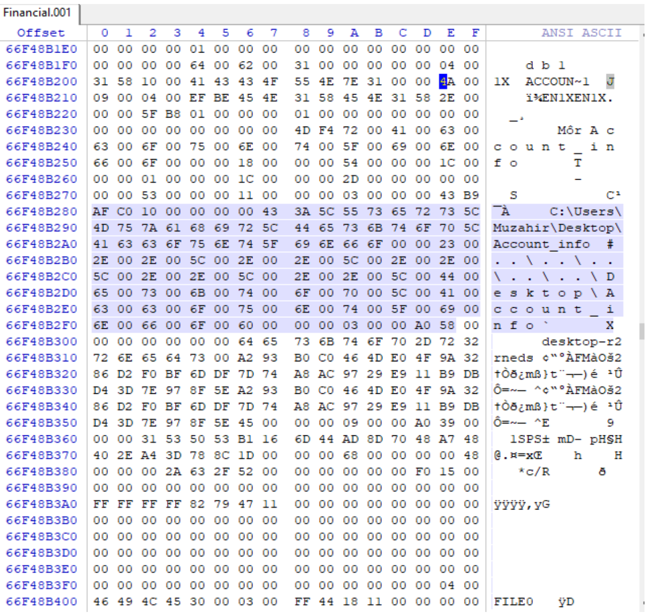
Structure of the $I file is shown in table.12

Table.12 Structure of $I file

0	8	Header(0x02)
8	16	Deleted file size
16	8	Deleted timestamp
24	4	File Name Length
28	Variable length	Filename and path

n NTFS file system each file has an MFT table entry. The file which was deleted from the system actually it is not deleted only its entry in MFT table marked as deleted actual contents remain there until or unless its overwritten [21]. But if the contents of the files are overwritten then it cannot be recovered. Whenever any file deleted accidentally or by choice its contents are saved before deletion into the Volume Shadow Copy. So, the file can be recovered using Volume Shadow Copy. First Volume Shadow Copy is the complete copy. From Second to onward only changes have been copied. In our experiment, we have changed the location of the stolen files and then at the end, they were deleted so in shadows only the differences have been saved.

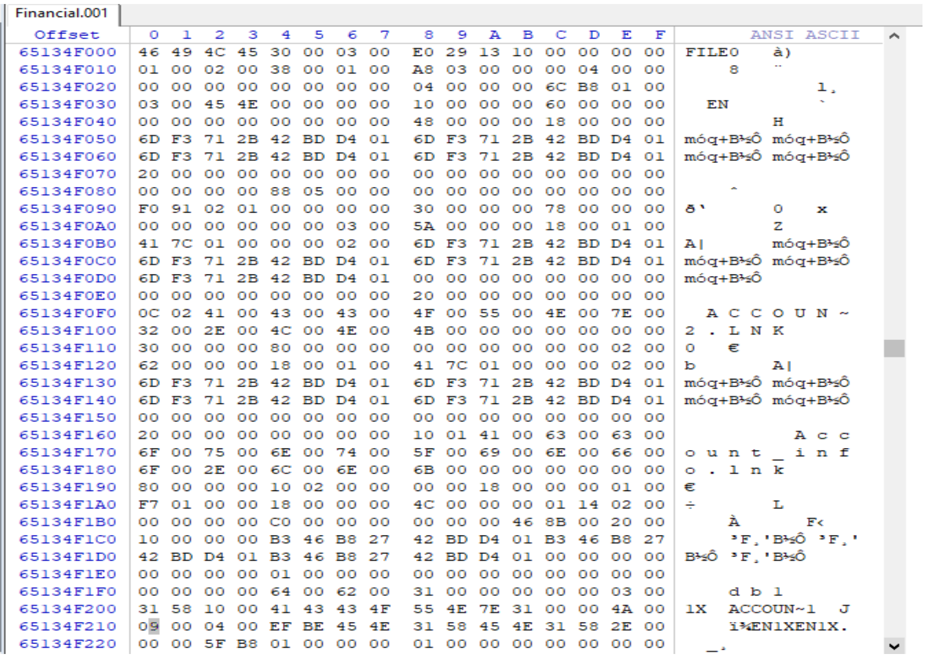
The second shadow shows that the Account_info folder is placed at the desktop shown in fig 23.



```
Financial.001
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F    ANSI ASCII
66F48B1E0  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00
66F48B1F0  00 00 00 00 64 00 62 00  31 00 00 00 00 00 04 00      d b 1
66F48B200  31 58 10 00 41 43 43 4F  55 4E 7E 31 00 00 4A 00  1X  ACCOUN~1  J
66F48B210  09 00 04 00 EF BE 45 4E  31 58 45 4E 31 58 2E 00      ï¾EN1XEN1X.
66F48B220  00 00 5F B8 01 00 00 00  01 00 00 00 00 00 00 00    _¸
66F48B230  00 00 00 00 00 00 00 00  4D F4 72 00 41 00 63 00          Môr A c
66F48B240  63 00 6F 00 75 00 6E 00  74 00 5F 00 69 00 6E 00  c o u n t _ i n
66F48B250  66 00 6F 00 00 00 18 00  00 00 54 00 00 00 1C 00  f o         T
66F48B260  00 00 01 00 00 00 1C 00  00 00 2D 00 00 00 00 00              -
66F48B270  00 00 53 00 00 00 11 00  00 00 03 00 00 00 43 B9    S            C¹
66F48B280  AF C0 10 00 00 00 00 43  3A 5C 55 73 65 72 73 5C  ¯À      C:\Users\
66F48B290  4D 75 7A 61 68 69 72 5C  44 65 73 6B 74 6F 70 5C  Muzahir\Desktop\
66F48B2A0  41 63 63 6F 75 6E 74 5F  69 6E 66 6F 00 00 23 00  Account_info  #
66F48B2B0  2E 00 2E 00 5C 00 2E 00  2E 00 5C 00 2E 00 2E 00  . . \ . . \ . .
66F48B2C0  5C 00 2E 00 2E 00 5C 00  2E 00 2E 00 5C 00 44 00  \ . . \ . . \ D
66F48B2D0  65 00 73 00 6B 00 74 00  6F 00 70 00 5C 00 41 00  e s k t o p \ A
66F48B2E0  63 00 63 00 6F 00 75 00  6E 00 74 00 5F 00 69 00  c c o u n t _ i
66F48B2F0  6E 00 66 00 6F 00 60 00  00 00 03 00 00 A0 58 00  n f o `        X
66F48B300  00 00 00 00 00 00 64 65  73 6B 74 6F 70 2D 72 32        desktop-r2
66F48B310  72 6E 65 64 73 00 A2 93  B0 C0 46 4D E0 4F 9A 32  rneds ¢"°ÀFMàOš2
66F48B320  86 D2 F0 BF 6D DF 7D 74  A8 AC 97 29 E9 11 B9 DB  †Òð¿mß}t¨¬—)é ¹Û
66F48B330  D4 3D 7E 97 8F 5E A2 93  B0 C0 46 4D E0 4F 9A 32  Ô=~— ^¢"°ÀFMàOš2
66F48B340  86 D2 F0 BF 6D DF 7D 74  A8 AC 97 29 E9 11 B9 DB  †Òð¿mß}t¨¬—)é ¹Û
66F48B350  D4 3D 7E 97 8F 5E 45 00  00 00 09 00 00 A0 39 00  Ô=~— ^E        9
66F48B360  00 00 31 53 50 53 B1 16  6D 44 AD 8D 70 48 A7 48    1SPS± mD- pH§H
66F48B370  40 2E A4 3D 78 8C 1D 00  00 00 68 00 00 00 00 48  @.¤=xŒ    h    H
66F48B380  00 00 00 2A 63 2F 52 00  00 00 00 00 00 F0 15 00     *c/R      ð
66F48B390  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
66F48B3A0  FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG
66F48B3B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
66F48B3C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
66F48B3D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
66F48B3E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
66F48B3F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 04 00
66F48B400  46 49 4C 45 30 00 03 00  FF 44 18 11 00 00 00 00  FILE0   ÿD
```

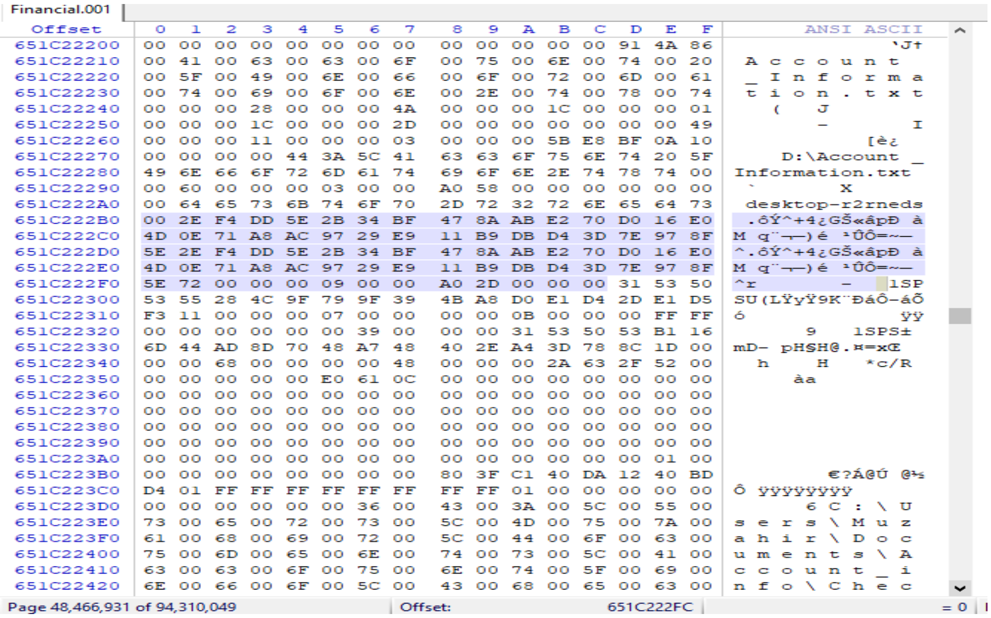
Fig:23Account_info in 2ndin Shadow copy

After the generation of second shadow copy folder was moved on the document folder and shown in fig 24 and 25.



```
Financial.001
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F     ANSI ASCII
65134F000  46 49 4C 45 30 00 03 00  E0 29 13 10 00 00 00 00  FILE0    à)
65134F010  01 00 02 00 38 00 01 00  A8 03 00 00 00 04 00 00      8    ¨
65134F020  00 00 00 00 00 00 00 00  04 00 00 00 6C B8 01 00              l¸
65134F030  03 00 45 4E 00 00 00 00  10 00 00 00 60 00 00 00    EN          `
65134F040  00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00          H
65134F050  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
65134F060  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
65134F070  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
65134F080  00 00 00 00 88 05 00 00  00 00 00 00 00 00 00 00      ˆ
65134F090  F0 91 02 01 00 00 00 00  30 00 00 00 78 00 00 00  ð'      0   x
65134F0A0  00 00 00 00 00 00 03 00  5A 00 00 00 18 00 01 00          Z
65134F0B0  41 7C 01 00 00 00 02 00  6D F3 71 2B 42 BD D4 01  A|      móq+B½Ô
65134F0C0  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
65134F0D0  6D F3 71 2B 42 BD D4 01  00 00 00 00 00 00 00 00  móq+B½Ô
65134F0E0  00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
65134F0F0  0C 02 41 00 43 00 43 00  4F 00 55 00 4E 00 7E 00    A C C O U N ~
65134F100  32 00 2E 00 4C 00 4E 00  4B 00 00 00 00 00 00 00  2 . L N K
65134F110  30 00 00 00 80 00 00 00  00 00 00 00 00 00 02 00  0   €
65134F120  62 00 00 00 18 00 01 00  41 7C 01 00 00 00 02 00  b       A|
65134F130  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
65134F140  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
65134F150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
65134F160  20 00 00 00 00 00 00 00  10 01 41 00 63 00 63 00            A c c
65134F170  6F 00 75 00 6E 00 74 00  5F 00 69 00 6E 00 66 00  o u n t _ i n f
65134F180  6F 00 2E 00 6C 00 6E 00  6B 00 00 00 00 00 00 00  o . l n k
65134F190  80 00 00 00 10 02 00 00  00 00 18 00 00 00 01 00  €
65134F1A0  F7 01 00 00 18 00 00 00  4C 00 00 00 01 14 02 00  ÷       L
65134F1B0  00 00 00 00 C0 00 00 00  00 00 00 46 8B 00 20 00      À       F‹
65134F1C0  10 00 00 00 B3 46 B8 27  42 BD D4 01 B3 46 B8 27      ³F¸'B½Ô ³F¸'
65134F1D0  42 BD D4 01 B3 46 B8 27  42 BD D4 01 00 00 00 00  B½Ô ³F¸'B½Ô
65134F1E0  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00
65134F1F0  00 00 00 00 64 00 62 00  31 00 00 00 00 00 03 00      d b 1
65134F200  31 58 10 00 41 43 43 4F  55 4E 7E 31 00 00 4A 00  1X  ACCOUN~1  J
65134F210  09 00 04 00 EF BE 45 4E  31 58 45 4E 31 58 2E 00      ï¾EN1XEN1X.
65134F220  00 00 5F B8 01 00 00 00  01 00 00 00 00 00 00 00    _¸
```

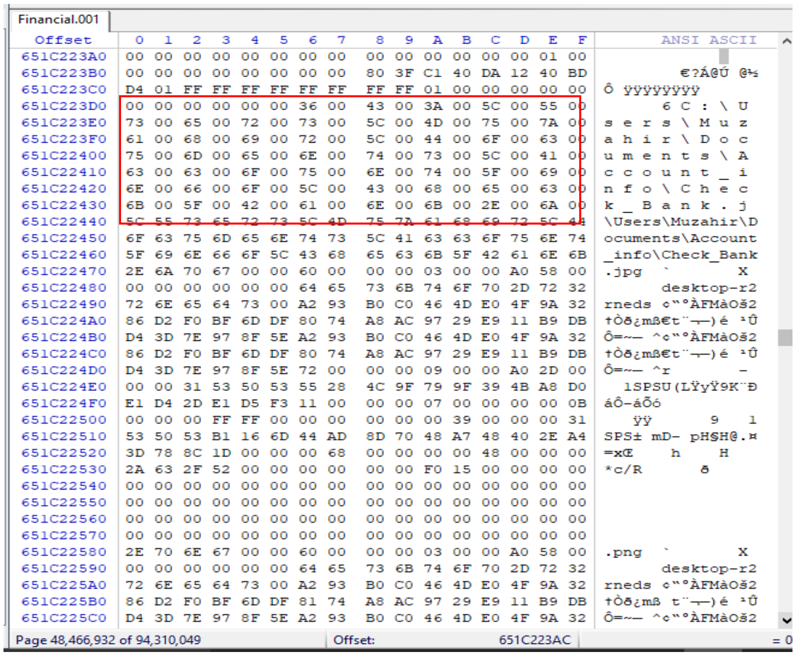
Fig 24: Account_info file in a 4th shadow copy



```
Financial.001
 Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F    ANSI ASCII
651C22200  00 00 00 00 00 00 00 00  00 00 00 00 00 91 4A 86              'J†
651C22210  00 41 00 63 00 63 00 6F  00 75 00 6E 00 74 00 20    A c c o u n t
651C22220  00 5F 00 49 00 6E 00 66  00 6F 00 72 00 6D 00 61    _ I n f o r m a
651C22230  00 74 00 69 00 6F 00 6E  00 2E 00 74 00 78 00 74   t i o n . t x t
651C22240  00 00 00 28 00 00 00 4A  00 00 00 1C 00 00 00 01      (   J
651C22250  00 00 00 1C 00 00 00 2D  00 00 00 00 00 00 00 49          -       I
651C22260  00 00 00 11 00 00 00 03  00 00 00 5B E8 BF 0A 10              [è¿
651C22270  00 00 00 00 44 3A 5C 41  63 63 6F 75 6E 74 20 5F      D:\Account _
651C22280  49 6E 66 6F 72 6D 61 74  69 6F 6E 2E 74 78 74 00   Information.txt
651C22290  00 60 00 00 00 03 00 00  A0 58 00 00 00 00 00 00    `       X
651C222A0  00 64 65 73 6B 74 6F 70  2D 72 32 72 6E 65 64 73    desktop-r2rneds
651C222B0  00 2E F4 DD 5E 2B 34 BF  47 8A AB E2 70 D0 16 E0    .ôÝ^+4¿GŠ«âpÐ à
651C222C0  4D 0E 71 A8 AC 97 29 E9  11 B9 DB D4 3D 7E 97 8F   M q¨¬—)é ¹ÛÔ=~—
651C222D0  5E 2E F4 DD 5E 2B 34 BF  47 8A AB E2 70 D0 16 E0   ^.ôÝ^+4¿GŠ«âpÐ à
651C222E0  4D 0E 71 A8 AC 97 29 E9  11 B9 DB D4 3D 7E 97 8F   M q¨¬—)é ¹ÛÔ=~—
651C222F0  5E 72 00 00 00 09 00 00  A0 2D 00 00 00 31 53 50   ^r        -   1SP
651C22300  53 55 28 4C 9F 79 9F 39  4B A8 D0 E1 D4 2D E1 D5   SU(LŸyŸ9K¨ÐáÔ-áÕ
651C22310  F3 11 00 00 00 07 00 00  00 00 0B 00 00 00 FF FF   ó              ÿÿ
651C22320  00 00 00 00 00 00 39 00  00 00 31 53 50 53 B1 16         9   1SPS±
651C22330  6D 44 AD 8D 70 48 A7 48  40 2E A4 3D 78 8C 1D 00   mD- pH§H@.¤=xŒ
651C22340  00 00 68 00 00 00 00 48  00 00 00 2A 63 2F 52 00     h     H   *c/R
651C22350  00 00 00 00 00 E0 61 0C  00 00 00 00 00 00 00 00        àa
651C22360  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22370  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22380  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22390  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C223A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 01 00
651C223B0  00 00 00 00 00 00 00 00  80 3F C1 40 DA 12 40 BD           €?Á@Ú @½
651C223C0  D4 01 FF FF FF FF FF FF  FF FF 01 00 00 00 00 00   Ô ÿÿÿÿÿÿÿÿ
651C223D0  00 00 00 00 00 00 36 00  43 00 3A 00 5C 00 55 00         6 C : \ U
651C223E0  73 00 65 00 72 00 73 00  5C 00 4D 00 75 00 7A 00   s e r s \ M u z
651C223F0  61 00 68 00 69 00 72 00  5C 00 44 00 6F 00 63 00   a h i r \ D o c
651C22400  75 00 6D 00 65 00 6E 00  74 00 73 00 5C 00 41 00   u m e n t s \ A
651C22410  63 00 63 00 6F 00 75 00  6E 00 74 00 5F 00 69 00   c c o u n t _ i
651C22420  6E 00 66 00 6F 00 5C 00  43 00 68 00 65 00 63 00   n f o \ C h e c
Page 48,466,931 of 94,310,049     Offset:     651C222FC     = 0
```

Fig 25: Account_information.txt file in 3rd shadow copy is found D director

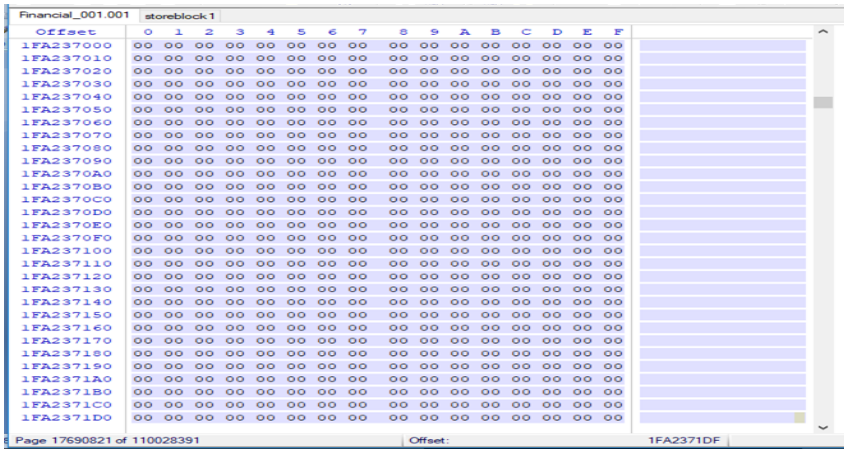
In fourth Shadow Copy file, Check-bank.jpg file has been moved from desktop to the Document folder, so in 4 the Volume Shadow Copy only the change which is the path of the file saved. The change path of the file is shown in 26.



```
Financial.001
 Offset     0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F    ANSI ASCII
651C223A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 01 00
651C223B0  00 00 00 00 00 00 00 00  80 3F C1 40 DA 12 40 BD           €?Á@Ú @½
651C223C0  D4 01 FF FF FF FF FF FF  FF FF 01 00 00 00 00 00   Ô ÿÿÿÿÿÿÿÿ
651C223D0  00 00 00 00 00 00 36 00  43 00 3A 00 5C 00 55 00         6 C : \ U
651C223E0  73 00 65 00 72 00 73 00  5C 00 4D 00 75 00 7A 00   s e r s \ M u z
651C223F0  61 00 68 00 69 00 72 00  5C 00 44 00 6F 00 63 00   a h i r \ D o c
651C22400  75 00 6D 00 65 00 6E 00  74 00 73 00 5C 00 41 00   u m e n t s \ A
651C22410  63 00 63 00 6F 00 75 00  6E 00 74 00 5F 00 69 00   c c o u n t _ i
651C22420  6E 00 66 00 6F 00 5C 00  43 00 68 00 65 00 63 00   n f o \ C h e c
651C22430  6B 00 5F 00 42 00 61 00  6E 00 6B 00 2E 00 6A 00   k _ B a n k . j
651C22440  5C 55 73 65 72 73 5C 4D  75 7A 61 68 69 72 5C 44   \Users\Muzahir\D
651C22450  6F 63 75 6D 65 6E 74 73  5C 41 63 63 6F 75 6E 74   ocuments\Account
651C22460  5F 69 6E 66 6F 5C 43 68  65 63 6B 5F 42 61 6E 6B   _info\Check_Bank
651C22470  2E 6A 70 67 00 00 60 00  00 00 03 00 00 A0 58 00   .jpg  `        X
651C22480  00 00 00 00 00 00 64 65  73 6B 74 6F 70 2D 72 32         desktop-r2
651C22490  72 6E 65 64 73 00 A2 93  B0 C0 46 4D E0 4F 9A 32   rneds ¢"°ÀFMàOš2
651C224A0  86 D2 F0 BF 6D DF 80 74  A8 AC 97 29 E9 11 B9 DB   †Òð¿mß€t¨¬—)é ¹Û
651C224B0  D4 3D 7E 97 8F 5E A2 93  B0 C0 46 4D E0 4F 9A 32   Ô=~— ^¢"°ÀFMàOš2
651C224C0  86 D2 F0 BF 6D DF 80 74  A8 AC 97 29 E9 11 B9 DB   †Òð¿mß€t¨¬—)é ¹Û
651C224D0  D4 3D 7E 97 8F 5E 72 00  00 00 09 00 00 A0 2D 00   Ô=~— ^r         -
651C224E0  00 00 31 53 50 53 55 28  4C 9F 79 9F 39 4B A8 D0     1SPSU(LŸyŸ9K¨Ð
651C224F0  E1 D4 2D E1 D5 F3 11 00  00 00 07 00 00 00 00 0B   áÔ-áÕó
651C22500  00 00 00 FF FF 00 00 00  00 00 00 39 00 00 00 31      ÿÿ      9   1
651C22510  53 50 53 B1 16 6D 44 AD  8D 70 48 A7 48 40 2E A4   SPS± mD- pH§H@.¤
651C22520  3D 78 8C 1D 00 00 00 68  00 00 00 00 48 00 00 00   =xŒ    h    H
651C22530  2A 63 2F 52 00 00 00 00  00 00 F0 15 00 00 00 00   *c/R      ð
651C22540  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22550  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22560  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22570  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
651C22580  2E 70 6E 67 00 00 60 00  00 00 03 00 00 A0 58 00   .png  `        X
651C22590  00 00 00 00 00 00 64 65  73 6B 74 6F 70 2D 72 32         desktop-r2
651C225A0  72 6E 65 64 73 00 A2 93  B0 C0 46 4D E0 4F 9A 32   rneds ¢"°ÀFMàOš2
651C225B0  86 D2 F0 BF 6D DF 81 74  A8 AC 97 29 E9 11 B9 DB   †Òð¿mß t¨¬—)é ¹Û
651C225C0  D4 3D 7E 97 8F 5E A2 93  B0 C0 46 4D E0 4F 9A 32   Ô=~— ^¢"°ÀFMàOš2
Page 48,466,932 of 94,310,049     Offset:     651C223AC     = 0
```

Fig: 26 Check_bank in 4 the shadow copy

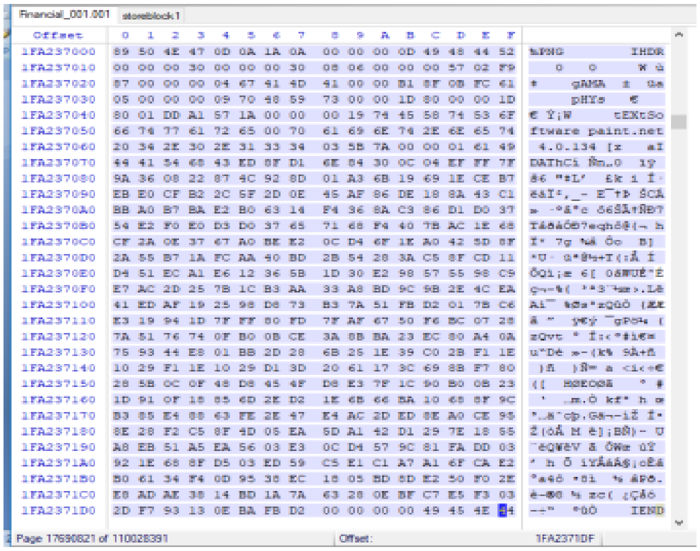
Bank_statement file was deleted and its deletion through recycle.bin is saved in 4th shadow copy as shown in fig 27:



```
Financial_001.001   storeblock 1
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
1FA237000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237010  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2370F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237170  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237180  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA237190  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2371A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2371B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2371C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
1FA2371D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
Page 17690821 of 110028391          Offset:          1FA2371DF
```

Fig.27: Deleted file Bank_ statement.jpg

The file Bak_statement is recovered back by mapping the shadow copy to the original data block and successfully recover shown in fig 29.

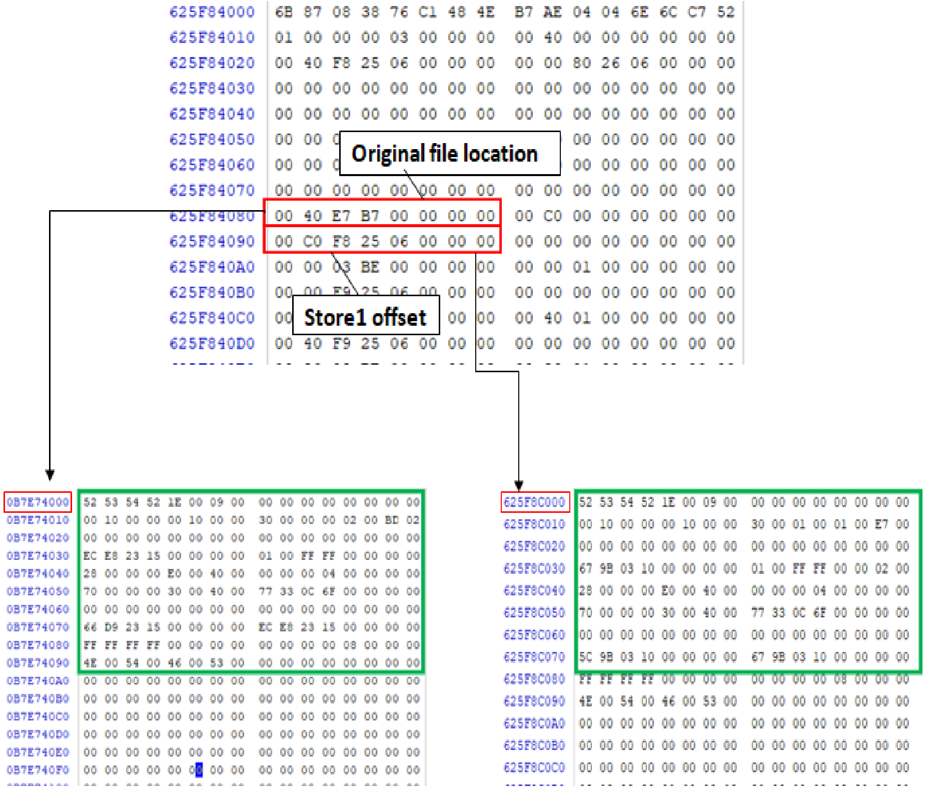


```
Financial_001.001   storeblock 1
  Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F
1FA237000  89 50 4E 47 0D 0A 1A 0A  00 00 00 0D 49 48 44 52   ‰PNG        IHDR
1FA237010  00 00 00 30 00 00 00 30  08 06 00 00 00 57 02 F9      0   0     W ù
1FA237020  87 00 00 00 04 67 41 4D  41 00 00 B1 8F 0B FC 61   ‡    gAMA  ±  üa
1FA237030  05 00 00 00 09 70 48 59  73 00 00 1D 80 00 00 1D        pHYs   €
1FA237040  80 01 DD A1 57 1A 00 00  00 19 74 45 58 74 53 6F   € Ý¡W     tEXtSo
1FA237050  66 74 77 61 72 65 00 70  61 69 6E 74 2E 6E 65 74   ftware paint.net
1FA237060  20 34 2E 30 2E 31 33 34  03 5B 7A 00 00 01 61 49    4.0.134 [z   aI
1FA237070  44 41 54 68 43 ED 8F D1  6E 84 30 0C 04 EF FF 7F   DAThCí Ñn„0  ïÿ
1FA237080  9A 36 08 22 87 4C 92 8D  01 A3 6B 19 69 1E CE B7   š6 "‡L’  £k i Î·
1FA237090  EB E0 CF B2 2C 5F 2D 0E  45 AF 86 DE 18 8A 43 C1   ëàÏ²,_- E¯†Þ ŠCÁ
1FA2370A0  BB A0 B7 BA E2 B0 63 14  F4 36 8A C3 86 D1 D0 37   » ·ºâ°c ô6ŠÃ†ÑÐ7
1FA2370B0  54 E2 F0 E0 D3 D0 37 65  71 68 F4 40 7B AC 1E 68   TâðàÓÐ7eqhô@{¬ h
1FA2370C0  CF 2A 0E 37 67 A0 BE E2  0C D4 6F 1E A0 42 5D 8F   Ï* 7g ¾â Ôo  B]
1FA2370D0  2A 55 B7 1A FC AA 40 BD  2B 54 28 3A C5 8F CD 11   *U· üª@½+T(:Å Í
1FA2370E0  D4 51 EC A1 E6 12 36 5B  1D 30 E2 98 57 55 98 C9   ÔQì¡æ 6[ 0â˜WU˜É
1FA2370F0  E7 AC 2D 25 7B 1C B3 AA  33 A8 BD 9C 9B 2E 4C EA   ç¬-%{ ³ª3¨½œ›.Lê
1FA237100  41 ED AF 19 25 98 D8 73  B3 7A 51 FB D2 01 7B C6   Aí¯ %˜Øs³zQûÒ {Æ
1FA237110  E3 19 94 1D 7F FF 80 FD  7F AF 67 50 F6 BC 07 28   ã ”  ÿ€ý ¯gPö¼ (
1FA237120  7A 51 76 74 0F B0 0B CE  3A 8B BA 23 EC 80 A4 0A   zQvt ° Î:‹º#ì€¤
1FA237130  75 93 44 E8 01 BB 2D 28  6B 25 1E 39 C0 2B F1 1E   u“Dè »-(k% 9À+ñ
1FA237140  10 29 F1 1E 10 29 D1 3D  20 61 17 3C 69 8B F7 80    )ñ  )Ñ= a <i‹÷€
1FA237150  28 5B 0C 0F 48 D8 45 4F  D8 E3 7F 1C 90 B0 0B 23   ([  HØEOØã   ° #
1FA237160  1D 91 0F 18 85 6D 2E D2  1E 6B 66 BA 10 68 8F 9C    ‘  …m.Ò kfº h œ
1FA237170  B3 85 E4 88 63 FE 2E 47  E4 AC 2D ED 8E A0 CE 95   ³…äˆcþ.Gä¬-íŽ Î•
1FA237180  8E 28 F2 C5 8F 4D 05 EA  5D A1 42 D1 29 7E 18 55   Ž(òÅ M ê]¡BÑ)~ U
1FA237190  A8 EB 51 A5 EA 56 03 E3  0C D4 57 9C 81 FA DD 03   ¨ëQ¥êV ã ÔWœ úÝ
1FA2371A0  92 1E 68 8F D5 03 ED 59  C5 E1 C1 A7 A1 6F CA E2   ’ h Õ íYÅáÁ§¡oÊâ
1FA2371B0  B0 61 34 F4 0D 95 38 EC  18 05 BD 8D E2 50 F0 2E   °a4ô •8ì  ½ âPð.
1FA2371C0  E8 AD AE 38 14 BD 1A 7A  63 28 0E BF C7 E5 F3 03   è®8 ½ zc( ¿Çåó
1FA2371D0  2D F7 93 13 0E BA FB D2  00 00 00 00 49 45 4E 44   -÷“ ºûÒ    IEND
Page 17690821 of 110028391          Offset:          1FA2371DF
```

Fig.29: Bank _statement.jpg file is recovered

It might possible that all the artifacts related to the deleted file is no longer present into the MFT but after doing the analysis of $log file examiner can dig out the important artifacts.

Every shadow copy or store starts with the $logfile. As the first store is placed at offset ***0x625f84000*** **.** From index record store’s relative offset and original file offset pointing to the $log file from original volume and shadow copy found which is shown in fig 29.



```
625F84000  6B 87 08 38 76 C1 48 4E  B7 AE 04 04 6E 6C C7 52
625F84010  01 00 00 00 03 00 00 00  00 40 00 00 00 00 00 00
625F84020  00 40 F8 25 06 00 00 00  00 00 80 26 06 00 00 00
625F84030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F84040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F84050  00 00 0                        00 00 00 00 00 00
625F84060  00 00 0                        00 00 00 00 00 00
625F84070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F84080  00 40 E7 B7 00 00 00 00  00 C0 00 00 00 00 00 00
625F84090  00 C0 F8 25 06 00 00 00  00 00 00 00 00 00 00 00
625F840A0  00 00 03 BE 00 00 00 00  00 00 01 00 00 00 00 00
625F840B0  00 00 F9 25 06 00 00 00  00 00 00 00 00 00 00 00
625F840C0  00                 00 00  00 40 01 00 00 00 00 00
625F840D0  00 40 F9 25 06 00 00 00  00 00 00 00 00 00 00 00
```

```
0B7E74000  52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00
0B7E74010  00 10 00 00 00 10 00 00  30 00 00 00 02 00 BD 02
0B7E74020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E74030  EC E8 23 15 00 00 00 00  01 00 FF FF 00 00 00 00
0B7E74040  28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00
0B7E74050  70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00
0B7E74060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E74070  66 D9 23 15 00 00 00 00  EC E8 23 15 00 00 00 00
0B7E74080  FF FF FF FF 00 00 00 00  00 00 00 00 08 00 00 00
0B7E74090  4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00
0B7E740A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E740B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E740C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E740D0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E740E0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0B7E740F0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

```
625F8C000  52 53 54 52 1E 00 09 00  00 00 00 00 00 00 00 00
625F8C010  00 10 00 00 00 10 00 00  30 00 01 00 01 00 E7 00
625F8C020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C030  67 9B 03 10 00 00 00 00  01 00 FF FF 00 00 02 00
625F8C040  28 00 00 00 E0 00 40 00  00 00 00 04 00 00 00 00
625F8C050  70 00 00 00 30 00 40 00  77 33 0C 6F 00 00 00 00
625F8C060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C070  5C 9B 03 10 00 00 00 00  67 9B 03 10 00 00 00 00
625F8C080  FF FF FF FF 00 00 00 00  00 00 00 00 08 00 00 00
625F8C090  4E 00 54 00 46 00 53 00  00 00 00 00 00 00 00 00
625F8C0A0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0B0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
625F8C0C0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

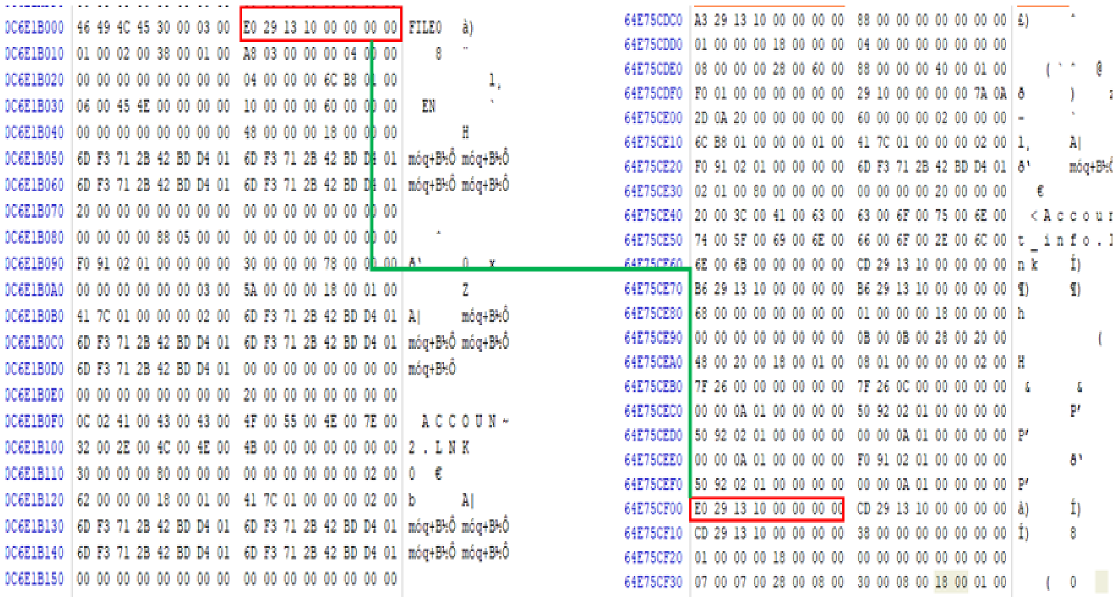
Fig 29: Original volume offset, and First Volume Shadow copy Offset

Table:16 Artifacts from the first catalog block list

Offset		Remarks	Findings/Artifacts
625F84080-87	8 Bytes	Original file location	E7B74000 (Original $Logfile Location)
625F84090-97	8Bytes	Location of the Store1(1st Shadow copy)	625F8C000 (Offset of the 1st shadow Copy)

The original file location is a location of the $log file of the Active volume and store1's offset represents the start of the Volume snapshot which contains $log file values which save the state of the $logfile for that specific point in time when shadow copy was created. It's important from the forensic point of view that investigator can investigate from that $logfile which saves the state of the system.

Every file has LSN and can be identified with its LSN in a $log file. Folder Account_info has LSN ***0E72AC23*** in its $MFT entry. In $Log file LSN of folder Account_info which was deleted and contains the all three files 1) Bank_Staement.jpg, Bankcheck.jpg,and Account_information.txt have been found at the location ***0x64E75CDE0***.Corelation between $MFT and $log file for the said folder has been shown in fig 30.



```
0C6E1B000  46 49 4C 45 30 00 03 00  E0 29 13 10 00 00 00 00  FILE0   à)
0C6E1B010  01 00 02 00 38 00 01 00  A8 03 00 00 00 04 00 00      8   ¨
0C6E1B020  00 00 00 00 00 00 00 00  04 00 00 00 6C B8 01 00          l¸
0C6E1B030  06 00 45 4E 00 00 00 00  10 00 00 00 60 00 00 00    EN     `
0C6E1B040  00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00          H
0C6E1B050  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
0C6E1B060  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
0C6E1B070  20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
0C6E1B080  00 00 00 00 88 05 00 00  00 00 00 00 00 00 00 00      ˆ
0C6E1B090  F0 91 02 01 00 00 00 00  30 00 00 00 78 00 00 00  ð'      0   x
0C6E1B0A0  00 00 00 00 00 00 03 00  5A 00 00 00 18 00 01 00          Z
0C6E1B0B0  41 7C 01 00 00 00 02 00  6D F3 71 2B 42 BD D4 01  A|      móq+B½Ô
0C6E1B0C0  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
0C6E1B0D0  6D F3 71 2B 42 BD D4 01  00 00 00 00 00 00 00 00  móq+B½Ô
0C6E1B0E0  00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
0C6E1B0F0  0C 02 41 00 43 00 43 00  4F 00 55 00 4E 00 7E 00    A C C O U N ~
0C6E1B100  32 00 2E 00 4C 00 4E 00  4B 00 00 00 00 00 00 00  2 . L N K
0C6E1B110  30 00 00 00 80 00 00 00  00 00 00 00 00 00 02 00  0   €
0C6E1B120  62 00 00 00 18 00 01 00  41 7C 01 00 00 00 02 00  b       A|
0C6E1B130  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
0C6E1B140  6D F3 71 2B 42 BD D4 01  6D F3 71 2B 42 BD D4 01  móq+B½Ô móq+B½Ô
0C6E1B150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

```
64E75CDC0  A3 29 13 10 00 00 00 00  88 00 00 00 00 00 00 00  £)      ˆ
64E75CDD0  01 00 00 00 18 00 00 00  04 00 00 00 00 00 00 00
64E75CDE0  08 00 00 00 28 00 60 00  88 00 00 00 40 00 01 00      ( `  ˆ   @
64E75CDF0  F0 01 00 00 00 00 00 00  29 10 00 00 00 00 7A 0A  ð       )     z
64E75CE00  2D 0A 20 00 00 00 00 00  60 00 00 00 02 00 00 00  -       `
64E75CE10  6C B8 01 00 00 00 01 00  41 7C 01 00 00 00 02 00  l¸      A|
64E75CE20  F0 91 02 01 00 00 00 00  6D F3 71 2B 42 BD D4 01  ð'      móq+B½Ô
64E75CE30  02 01 00 80 00 00 00 00  00 00 00 00 20 00 00 00     €
64E75CE40  20 00 3C 00 41 00 63 00  63 00 6F 00 75 00 6E 00    < A c c o u n
64E75CE50  74 00 5F 00 69 00 6E 00  66 00 6F 00 2E 00 6C 00  t _ i n f o . l
64E75CE60  6E 00 6B 00 00 00 00 00  CD 29 13 10 00 00 00 00  n k     Í)
64E75CE70  B6 29 13 10 00 00 00 00  B6 29 13 10 00 00 00 00  ¶)      ¶)
64E75CE80  68 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  h
64E75CE90  00 00 00 00 00 00 00 00  0B 00 0B 00 28 00 20 00              (
64E75CEA0  48 00 20 00 18 00 01 00  08 01 00 00 00 00 02 00  H
64E75CEB0  7F 26 00 00 00 00 00 00  7F 26 0C 00 00 00 00 00   &       &
64E75CEC0  00 00 0A 01 00 00 00 00  50 92 02 01 00 00 00 00          P'
64E75CED0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P'
64E75CEE0  00 00 0A 01 00 00 00 00  F0 91 02 01 00 00 00 00          ð'
64E75CEF0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P'
64E75CF00  E0 29 13 10 00 00 00 00  CD 29 13 10 00 00 00 00  à)      Í)
64E75CF10  CD 29 13 10 00 00 00 00  38 00 00 00 00 00 00 00  Í)      8
64E75CF20  01 00 00 00 18 00 00 00  00 00 00 00 00 00 00 00
64E75CF30  07 00 07 00 28 00 08 00  30 00 08 00 18 00 01 00      (   0
```

Fig 30: LSN of a file in $MFT and $Logfile of Volume Shadow Copy

5.4 File Deletion case.

Two types of file deletion have been supported by NTFS: using Recycle Bin and bypassing the Recycle bin.

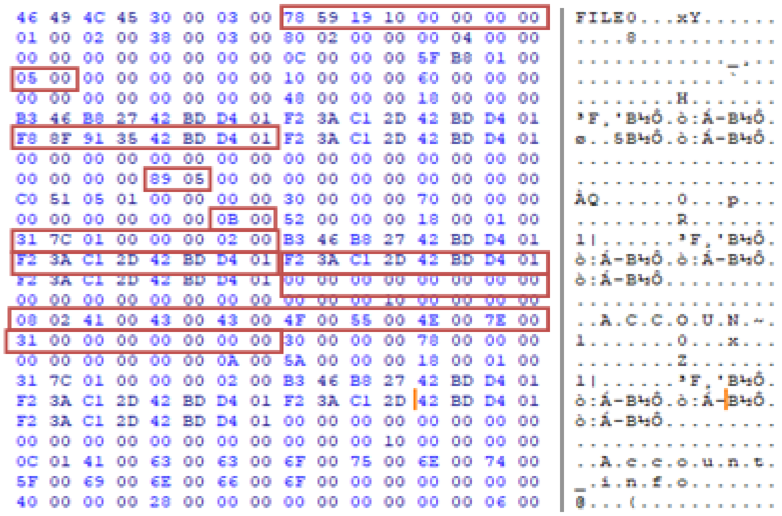
Deleting files using Recycle Bin. As soon as the file is deleted it is moved into the recycle bin after renaming. Changes also made in $MFT file records and $logfile transactional records.

Changes in MFT Record of File is moved to Recycle.Bin after Deletion

- Log file Sequence Number is updated at offset0x08-0x0F.
- Next attribute ID at offset0x28 is incremented by 1, as previous “next attribute ID” has been assigned to attribute 0x30 as it has been changed.
- Updatesequencenumberatoffset0x30-0x31,0x1FE-0x1FFand0x3FE-0x3FFare updated.
- MFT update time at offset 0x60 is updated in attribute 0x10 $Standard_Information.
- Security ID is updated at offset 0x84 is updated in attribute 0x10 $Standard_Information.

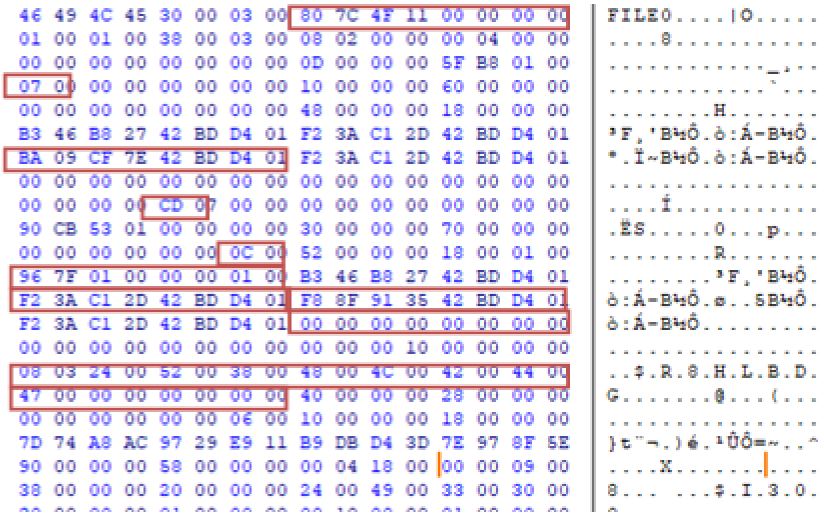
- As the name of the file is changed, so attribute length is changed at offset 0x9C-0x9 -0x30 $File_Name.
- Attribute ID at 0xA6-0xA7of attribute 0x30 $File_Name is incremented to “Next attribute ID” mentioned in the original record, as this attribute has been updated and will be accordingly mentioned in $logFile to differentiate the updated attribute 0x30 from the previous record.
- The file is indexed under the current user folder in recycle bin instead of the folder under which it was previously held and accordingly “File reference to the parent directory” at offset 0xB0-0xB7 is updated.
- File modified time, and MFT update time are updated at offset 0xC0-0xC7 and 0xC8-0xC Fin 0x30$File_Name, respectively.
- Allocated and real size (of payload of attribute 0x80 $Data) is updated at offset 0xD8-0xDFand 0xE0-0xE7 in 0x30 $File_Name, respectively.
- File name length, file name namespace and file name are updated starting from attribute0xF0in0x30 $File_Name.

The comparative screenshot is shown in Figure 32. Most of the above- mentioned offsets are specific to following screenshot for ease of understanding.



```
46 49 4C 45 30 00 03 00 78 59 19 10 00 00 00 00   FILE0...xY......
01 00 02 00 38 00 03 00 80 02 00 00 00 04 00 00   ....8...........
00 00 00 00 00 00 00 00 0C 00 00 00 5F B8 01 00   ............_¸..
05 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00   ............`...
00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00   ........H.......
B3 46 B8 27 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ³F¸'B½Ô.ò:Á-B½Ô.
F8 8F 91 35 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ø..5B½Ô.ò:Á-B½Ô.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 89 05 00 00 00 00 00 00 00 00 00 00   ................
C0 51 05 01 00 00 00 00 30 00 00 00 70 00 00 00   ÀQ......0...p...
00 00 00 00 00 00 0B 00 52 00 00 00 18 00 01 00   ........R.......
31 7C 01 00 00 00 02 00 B3 46 B8 27 42 BD D4 01   1|......³F¸'B½Ô.
F2 3A C1 2D 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ò:Á-B½Ô.ò:Á-B½Ô.
F2 3A C1 2D 42 BD D4 01 00 00 00 00 00 00 00 00   ò:Á-B½Ô.........
00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00   ................
08 02 41 00 43 00 43 00 4F 00 55 00 4E 00 7E 00   ..A.C.C.O.U.N.~.
31 00 00 00 00 00 00 00 30 00 00 00 78 00 00 00   1.......0...x...
00 00 00 00 00 00 0A 00 5A 00 00 00 18 00 01 00   ........Z.......
31 7C 01 00 00 00 02 00 B3 46 B8 27 42 BD D4 01   1|......³F¸'B½Ô.
F2 3A C1 2D 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ò:Á-B½Ô.ò:Á-B½Ô.
F2 3A C1 2D 42 BD D4 01 00 00 00 00 00 00 00 00   ò:Á-B½Ô.........
00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00   ................
0C 01 41 00 63 00 63 00 6F 00 75 00 6E 00 74 00   ..A.c.c.o.u.n.t.
5F 00 69 00 6E 00 66 00 6F 00 00 00 00 00 00 00   _.i.n.f.o.......
40 00 00 00 28 00 00 00 00 00 00 00 00 00 06 00   @...(...........
```

Folder before deletion



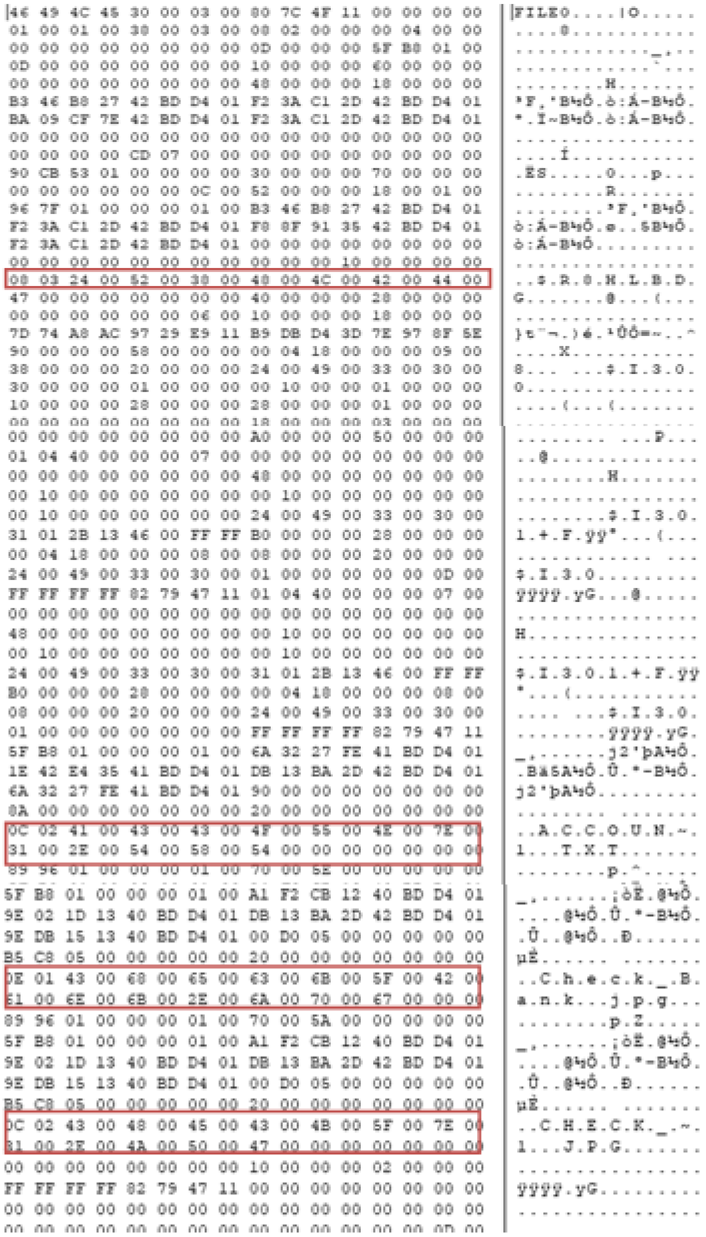
```
46 49 4C 45 30 00 03 00 80 7C 4F 11 00 00 00 00   FILE0....|O.....
01 00 01 00 38 00 03 00 08 02 00 00 00 04 00 00   ....8...........
00 00 00 00 00 00 00 00 0D 00 00 00 5F B8 01 00   ............_¸..
07 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00   ............`...
00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00   ........H.......
B3 46 B8 27 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ³F¸'B½Ô.ò:Á-B½Ô.
BA 09 CF 7E 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   º.Ï~B½Ô.ò:Á-B½Ô.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 CD 07 00 00 00 00 00 00 00 00 00 00   ....Í...........
90 CB 53 01 00 00 00 00 30 00 00 00 70 00 00 00   .ËS.....0...p...
00 00 00 00 00 00 0C 00 52 00 00 00 18 00 01 00   ........R.......
96 7F 01 00 00 00 01 00 B3 46 B8 27 42 BD D4 01   ........³F¸'B½Ô.
F2 3A C1 2D 42 BD D4 01 F8 8F 91 35 42 BD D4 01   ò:Á-B½Ô.ø..5B½Ô.
F2 3A C1 2D 42 BD D4 01 00 00 00 00 00 00 00 00   ò:Á-B½Ô.........
00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00   ................
08 03 24 00 52 00 38 00 48 00 4C 00 42 00 44 00   ..$.R.8.H.L.B.D.
47 00 00 00 00 00 00 00 40 00 00 00 28 00 00 00   G.......@...(...
00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00   ................
7D 74 A8 AC 97 29 E9 11 B9 DB D4 3D 7E 97 8F 5E   }t¨¬.)é.¹ÛÔ=~..^
90 00 00 00 58 00 00 00 00 04 18 00 00 00 09 00   ....X...........
38 00 00 00 20 00 00 00 24 00 49 00 33 00 30 00   8... ...$.I.3.0.
```

Folder after deletion

Figure32: Comparative Screen Shot of MFT for File Deletion

Changes in MFT Record of New Parent Folder (Current User Folder in RecycleBin)

When the **file is moved to RecycleBin** . MFT record of parent folder under which the file after deletion has been indexed (moved) (shown at offset 0xB0-0xB7 in Figure32) is shown in Figure 33.

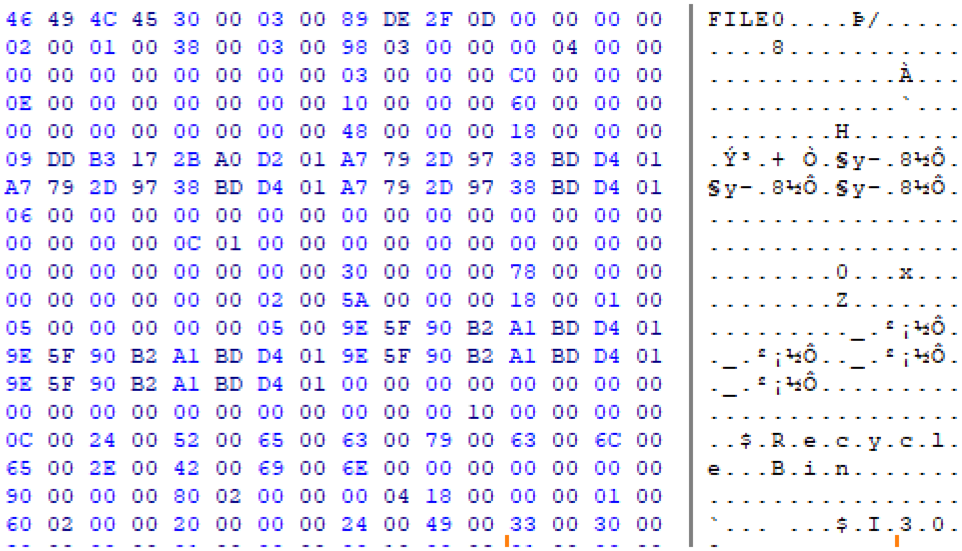


```
46 49 4C 45 30 00 03 00 80 7C 4F 11 00 00 00 00  |FILE0....|O.....
01 00 01 00 38 00 03 00 08 02 00 00 00 04 00 00   ....8...........
00 00 00 00 00 00 00 00 0D 00 00 00 5F B8 01 00   ............_¸..
0D 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00   ............`...
00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00   ........H.......
B3 46 B8 27 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   ³F¸'B½Ô.ò:Á-B½Ô.
BA 09 CF 7E 42 BD D4 01 F2 3A C1 2D 42 BD D4 01   º.Ï~B½Ô.ò:Á-B½Ô.
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 CD 07 00 00 00 00 00 00 00 00 00 00   ....Í...........
90 CB 53 01 00 00 00 00 30 00 00 00 70 00 00 00   .ËS.....0...p...
00 00 00 00 00 00 0C 00 52 00 00 00 18 00 01 00   ........R.......
96 7F 01 00 00 00 01 00 B3 46 B8 27 42 BD D4 01   ........³F¸'B½Ô.
F2 3A C1 2D 42 BD D4 01 F8 8F 91 35 42 BD D4 01   ò:Á-B½Ô.ø..5B½Ô.
F2 3A C1 2D 42 BD D4 01 00 00 00 00 00 00 00 00   ò:Á-B½Ô.........
00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00   ................
08 03 24 00 52 00 38 00 48 00 4C 00 42 00 44 00   ..$.R.8.H.L.B.D.
47 00 00 00 00 00 00 00 40 00 00 00 28 00 00 00   G.......@...(...
00 00 00 00 00 00 06 00 10 00 00 00 18 00 00 00   ................
7D 74 A8 AC 97 29 E9 11 B9 DB D4 3D 7E 97 8F 5E   }t¨¬.)é.¹ÛÔ=~..^
90 00 00 00 58 00 00 00 00 04 18 00 00 00 09 00   ....X...........
38 00 00 00 20 00 00 00 24 00 49 00 33 00 30 00   8... ...$.I.3.0.
30 00 00 00 01 00 00 00 00 10 00 00 01 00 00 00   0...............
10 00 00 00 28 00 00 00 28 00 00 00 01 00 00 00   ....(...(.......
00 00 00 00 00 00 00 00 18 00 00 00 03 00 00 00   ................
00 00 00 00 00 00 00 00 A0 00 00 00 50 00 00 00   ........ ...P...
01 04 40 00 00 00 07 00 00 00 00 00 00 00 00 00   ..@.............
00 00 00 00 00 00 00 00 48 00 00 00 00 00 00 00   ........H.......
00 10 00 00 00 00 00 00 00 10 00 00 00 00 00 00   ................
00 10 00 00 00 00 00 00 24 00 49 00 33 00 30 00   ........$.I.3.0.
31 01 2B 13 46 00 FF FF B0 00 00 00 28 00 00 00   1.+.F.ÿÿ°...(...
00 04 18 00 00 00 08 00 08 00 00 00 20 00 00 00   ............ ...
24 00 49 00 33 00 30 00 01 00 00 00 00 00 0D 00   $.I.3.0.........
FF FF FF FF 82 79 47 11 01 04 40 00 00 00 07 00   ÿÿÿÿ.yG...@.....
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
48 00 00 00 00 00 00 00 00 10 00 00 00 00 00 00   H...............
00 10 00 00 00 00 00 00 00 10 00 00 00 00 00 00   ................
24 00 49 00 33 00 30 00 31 01 2B 13 46 00 FF FF   $.I.3.0.1.+.F.ÿÿ
B0 00 00 00 28 00 00 00 00 04 18 00 00 00 08 00   °...(...........
08 00 00 00 20 00 00 00 24 00 49 00 33 00 30 00   .... ...$.I.3.0.
01 00 00 00 00 00 00 00 FF FF FF FF 82 79 47 11   ........ÿÿÿÿ.yG.
5F B8 01 00 00 00 01 00 6A 32 27 FE 41 BD D4 01   _¸......j2'þA½Ô.
1E 42 E4 35 41 BD D4 01 DB 13 BA 2D 42 BD D4 01   .Bä5A½Ô.Û.º-B½Ô.
6A 32 27 FE 41 BD D4 01 90 00 00 00 00 00 00 00   j2'þA½Ô.........
8A 00 00 00 00 00 00 00 20 00 00 00 00 00 00 00   ........ .......
0C 02 41 00 43 00 43 00 4F 00 55 00 4E 00 7E 00   ..A.C.C.O.U.N.~.
31 00 2E 00 54 00 58 00 54 00 00 00 00 00 00 00   1...T.X.T.......
89 96 01 00 00 00 01 00 70 00 5E 00 00 00 00 00   ........p.^.....
5F B8 01 00 00 00 01 00 A1 F2 CB 12 40 BD D4 01   _¸......¡òË.@½Ô.
9E 02 1D 13 40 BD D4 01 DB 13 BA 2D 42 BD D4 01   ....@½Ô.Û.º-B½Ô.
9E DB 15 13 40 BD D4 01 00 D0 05 00 00 00 00 00   .Û..@½Ô..Ð......
B5 C8 05 00 00 00 00 00 20 00 00 00 00 00 00 00   µÈ...... .......
0E 01 43 00 68 00 65 00 63 00 6B 00 5F 00 42 00   ..C.h.e.c.k._.B.
61 00 6E 00 6B 00 2E 00 6A 00 70 00 67 00 00 00   a.n.k...j.p.g...
89 96 01 00 00 00 01 00 70 00 5A 00 00 00 00 00   ........p.Z.....
5F B8 01 00 00 00 01 00 A1 F2 CB 12 40 BD D4 01   _¸......¡òË.@½Ô.
9E 02 1D 13 40 BD D4 01 DB 13 BA 2D 42 BD D4 01   ....@½Ô.Û.º-B½Ô.
9E DB 15 13 40 BD D4 01 00 D0 05 00 00 00 00 00   .Û..@½Ô..Ð......
B5 C8 05 00 00 00 00 00 20 00 00 00 00 00 00 00   µÈ...... .......
0C 02 43 00 48 00 45 00 43 00 4B 00 5F 00 7E 00   ..C.H.E.C.K._.~.
31 00 2E 00 4A 00 50 00 47 00 00 00 00 00 00 00   1...J.P.G.......
00 00 00 00 00 00 00 00 10 00 00 00 02 00 00 00   ................
FF FF FF FF 82 79 47 11 00 00 00 00 00 00 00 00   ÿÿÿÿ.yG.........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 0D 00
```

Figure 33-ExampleofCurrentUserFolder of Recycle MFTRecord

This MFT record is, in fact, a folder of current user in recyclebin folder referred at

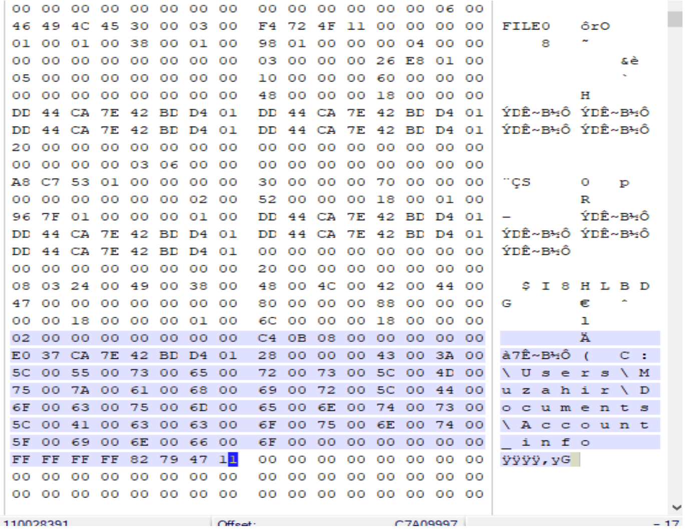
Offset 0xB0-0xB7, shown inFigure34.



```
46 49 4C 45 30 00 03 00 89 DE 2F 0D 00 00 00 00  FILE0....Þ/.....
02 00 01 00 38 00 03 00 98 03 00 00 00 04 00 00  ....8...........
00 00 00 00 00 00 00 00 03 00 00 00 C0 00 00 00  ............À...
0E 00 00 00 00 00 00 00 10 00 00 00 60 00 00 00  ............`...
00 00 00 00 00 00 00 00 48 00 00 00 18 00 00 00  ........H.......
09 DD B3 17 2B A0 D2 01 A7 79 2D 97 38 BD D4 01  .Ý³.+ Ò.§y-.8½Ô.
A7 79 2D 97 38 BD D4 01 A7 79 2D 97 38 BD D4 01  §y-.8½Ô.§y-.8½Ô.
06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 0C 01 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 30 00 00 00 78 00 00 00  ........0...x...
00 00 00 00 00 00 02 00 5A 00 00 00 18 00 01 00  ........Z.......
05 00 00 00 00 00 05 00 9E 5F 90 B2 A1 BD D4 01  ........._.²¡½Ô.
9E 5F 90 B2 A1 BD D4 01 9E 5F 90 B2 A1 BD D4 01  ._.²¡½Ô.._.²¡½Ô.
9E 5F 90 B2 A1 BD D4 01 00 00 00 00 00 00 00 00  ._.²¡½Ô.........
00 00 00 00 00 00 00 00 00 00 00 10 00 00 00 00  ................
0C 00 24 00 52 00 65 00 63 00 79 00 63 00 6C 00  ..$.R.e.c.y.c.l.
65 00 2E 00 42 00 69 00 6E 00 00 00 00 00 00 00  e...B.i.n.......
90 00 00 00 80 02 00 00 00 04 18 00 00 00 01 00  ................
60 02 00 00 20 00 00 00 24 00 49 00 33 00 30 00  `... ...$.I.3.0.
```

Figure 34: Example of Recycle Bin MFT Record

When any file or folder has been deleted two pair of files are generated as discussed in chapter 2.$I and $R with random characters the same for both files.$I contains the metadata and $R is an actual deleted file[22].The interesting thing is when the folder is deleted its pair of files have been generated with new names but the contents of the folder remain the same with their original names.



```
00 00 00 00 00 00 00 00  00 00 00 00 00 00 06 00
46 49 4C 45 30 00 03 00  F4 72 4F 11 00 00 00 00  FILE0   ôrO
01 00 01 00 38 00 01 00  98 01 00 00 00 04 00 00      8    ˜
00 00 00 00 00 00 00 00  03 00 00 00 26 E8 01 00              &è
05 00 00 00 00 00 00 00  10 00 00 00 60 00 00 00              `
00 00 00 00 00 00 00 00  48 00 00 00 18 00 00 00          H
DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
20 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 03 06 00 00  00 00 00 00 00 00 00 00
A8 C7 53 01 00 00 00 00  30 00 00 00 70 00 00 00  ¨ÇS     0   p
00 00 00 00 00 00 02 00  52 00 00 00 18 00 01 00          R
96 7F 01 00 00 00 01 00  DD 44 CA 7E 42 BD D4 01  –       ÝDÊ~B½Ô
DD 44 CA 7E 42 BD D4 01  DD 44 CA 7E 42 BD D4 01  ÝDÊ~B½Ô ÝDÊ~B½Ô
DD 44 CA 7E 42 BD D4 01  00 00 00 00 00 00 00 00  ÝDÊ~B½Ô
00 00 00 00 00 00 00 00  20 00 00 00 00 00 00 00
08 03 24 00 49 00 38 00  48 00 4C 00 42 00 44 00    $ I 8 H L B D
47 00 00 00 00 00 00 00  80 00 00 00 88 00 00 00  G       €   ˆ
00 00 18 00 00 00 01 00  6C 00 00 00 18 00 00 00          l
02 00 00 00 00 00 00 00  C4 0B 08 00 00 00 00 00          Ä
E0 37 CA 7E 42 BD D4 01  28 00 00 00 43 00 3A 00  à7Ê~B½Ô (   C :
5C 00 55 00 73 00 65 00  72 00 73 00 5C 00 4D 00  \ U s e r s \ M
75 00 7A 00 61 00 68 00  69 00 72 00 5C 00 44 00  u z a h i r \ D
6F 00 63 00 75 00 6D 00  65 00 6E 00 74 00 73 00  o c u m e n t s
5C 00 41 00 63 00 63 00  6F 00 75 00 6E 00 74 00  \ A c c o u n t
5F 00 69 00 6E 00 66 00  6F 00 00 00 00 00 00 00  _ i n f o
FF FF FF FF 82 79 47 11  00 00 00 00 00 00 00 00  ÿÿÿÿ‚yG
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
110028391            Offset:            C7A09997            = 17
```

Figure 35 -$I8HLBDG MFTRecord Structure

Transactions Logged in $LogFile when File is pushed to RecycleBin. Deletion process initiates an operational records' long chain which is recorded in $LogFile. This chain starts from logging the transaction of allocating MFT record in $MFT for newly created $Ixxx file and terminates at updating index entry of root record 05 of $I. This chain contains numerous small subchain each representing a sequence of transactions. This chain contains numerous small sub chains each representing a sequence of transactions. Important transactions logged in $logfile(in sequential order) are discussed below.

SubChain1–MFTRecordGenerationof$Ixxx

- A new file record($Ixxx) generated for Current User Folder of Recycle Bin to point to MFT location of deleted File
- New File($Ixxx) indexed in Current User Folder of RecycleBin
- Initialization of MFT Record of $Ixxx

SubChain2–UpdateRecycleBin

- Update MFT of RecycleBin
- UpdateIndexEntryof Current User in RecycleBin

SubChain3. Update 0x80 of MFT record of $Ixxx.

Sub Chain 4. Update Index Entry of $Ixxx in Current User Folder of Recycle Bin.

SubChain5–Renaming Deleted File and Updating Associated Records

- Removing Deleted file entry(Long Filename) from Parent Folder
- Removing Original File Name from MFT record of Deleted File
- Removing Deleted file entry(Short File name) from Parent Folder
- Renaming Deleted File to $Rxxx in MFT Record of Deleted File
- Clearing Index Entries from Recycle Bin of Current User Folder

A new Cluster Allocated for Non-Resident INDX of Current User Folder of RecycleBin in $BitMap

- Creating Attr. 0xA0 in MFT in RecycleBin of Current User Folder
- Creating Attr. 0xB0 in MFT in RecycleBin of Current User Folder
- Updating Attr. 0xB0 inMFT in RecycleBin of Current User Folder
- Updating Attr. 0xA0 in MFT in RecycleBin of Current User Folder
- Add index entries less Renamed Deleted File $Rxxxin MFT in RecycleBin of Current User Folder.
- Creating Attr. 0x90 in MFT in RecycleBin of Current User Folder
- Add index entry of renamed file $Rxxxto INDX of Attr.0xA0

SubChain6–UpdatingRecycleBin

- Update MFT Record in RecycleBin of Current User Folder
- Updating MFT Record in RecycleBin of Current User Folder

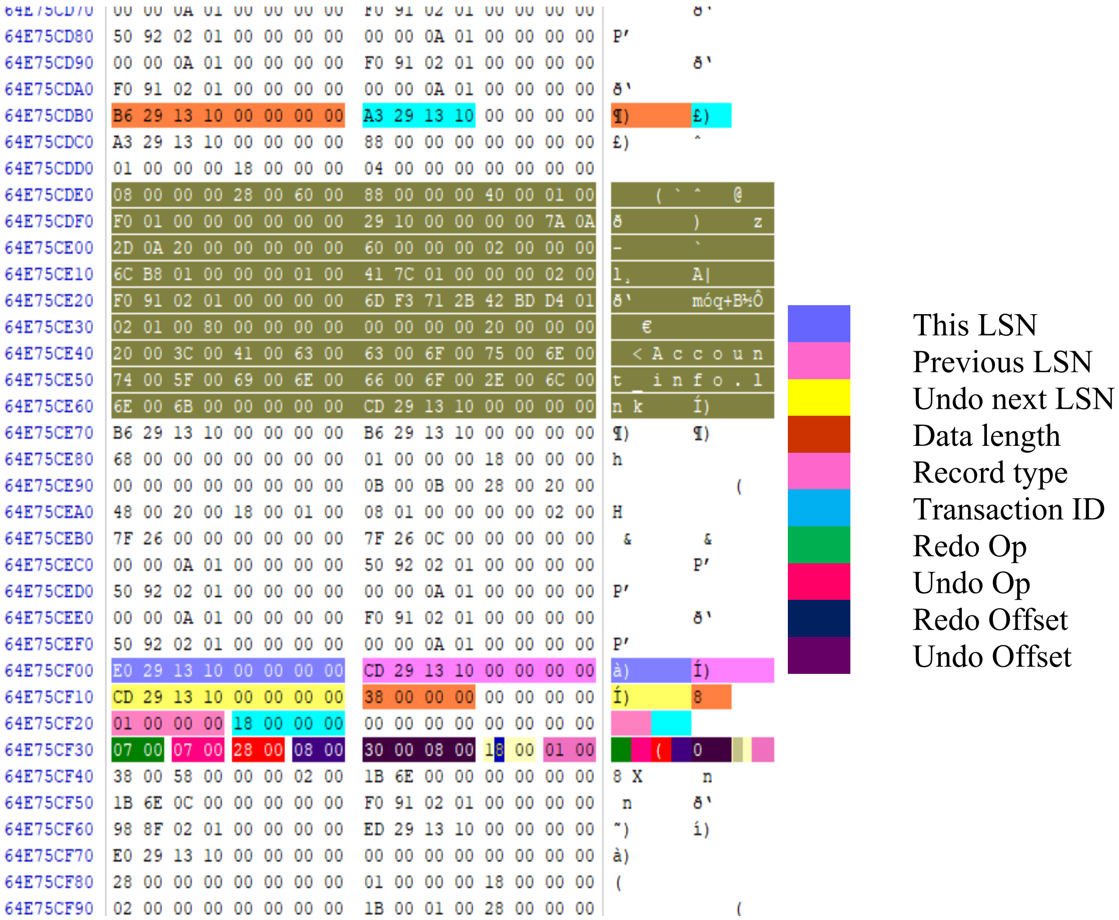
SubChain7.

- Updat MFT Record of Renamed Deleted File $Rxxx (Attr.0x10 only Security ID, Quota charged an Update Sequence Number).

Sub Chain 9 – Updating MFT Records in Chain of Parent Folders

- Update MFT Record of Parent Folder
- Update MFT Record of Parent of the Parent Folder

Example of operational record has been shown in fig 36.

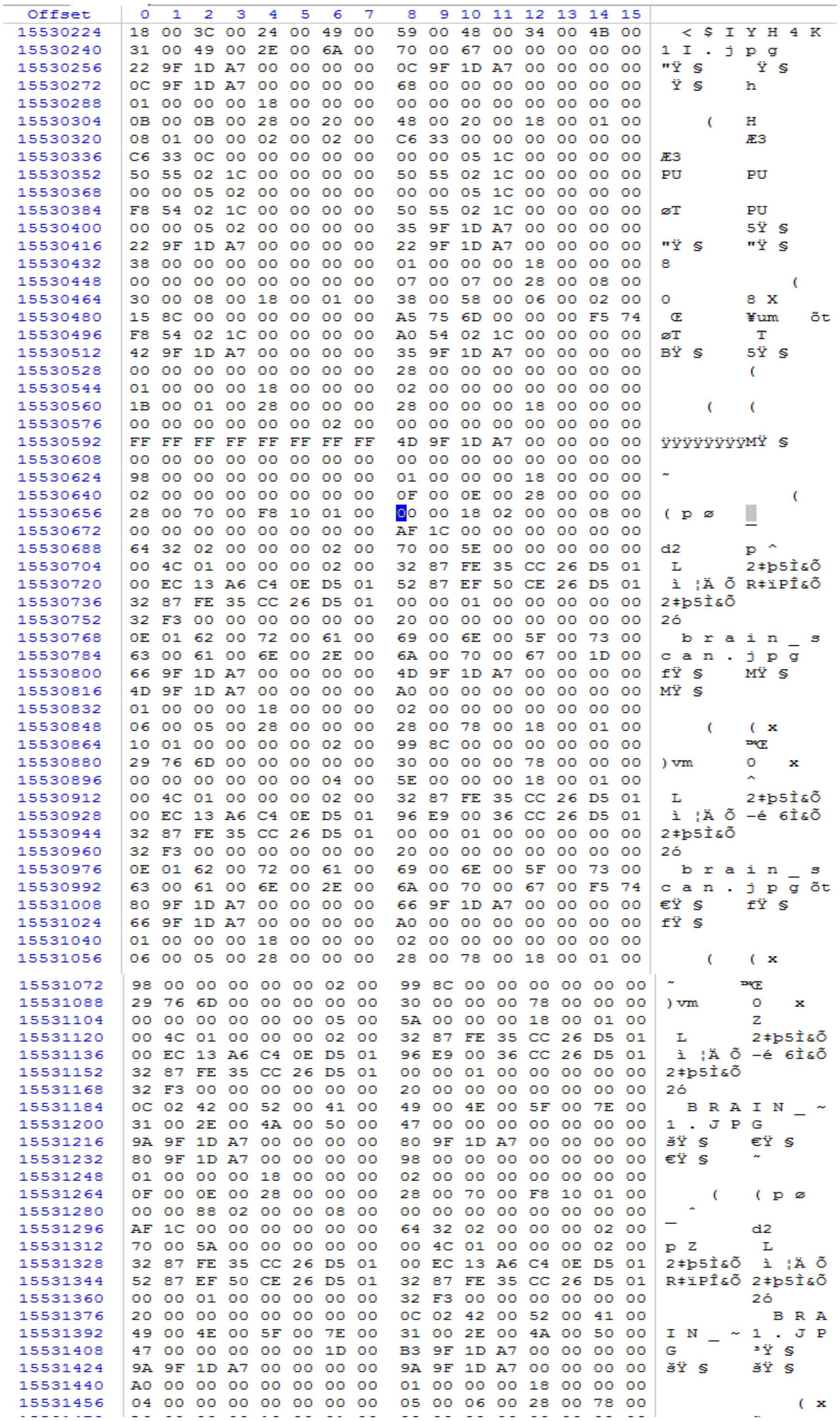


```
64E75CD70  00 00 0A 01 00 00 00 00  F0 91 02 01 00 00 00 00          ð'
64E75CD80  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P'
64E75CD90  00 00 0A 01 00 00 00 00  F0 91 02 01 00 00 00 00          ð'
64E75CDA0  F0 91 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  ð'
64E75CDB0  B6 29 13 10 00 00 00 00  A3 29 13 10 00 00 00 00  ¶)      £)
64E75CDC0  A3 29 13 10 00 00 00 00  88 00 00 00 00 00 00 00  £)      ˆ
64E75CDD0  01 00 00 00 18 00 00 00  04 00 00 00 00 00 00 00
64E75CDE0  08 00 00 00 28 00 60 00  88 00 00 00 40 00 01 00      ( `  ˆ   @
64E75CDF0  F0 01 00 00 00 00 00 00  29 10 00 00 00 00 7A 0A  ð       )     z
64E75CE00  2D 0A 20 00 00 00 00 00  60 00 00 00 02 00 00 00  -       `
64E75CE10  6C B8 01 00 00 00 01 00  41 7C 01 00 00 00 02 00  l¸      A|
64E75CE20  F0 91 02 01 00 00 00 00  6D F3 71 2B 42 BD D4 01  ð'      móq+B½Ô
64E75CE30  02 01 00 80 00 00 00 00  00 00 00 00 20 00 00 00     €
64E75CE40  20 00 3C 00 41 00 63 00  63 00 6F 00 75 00 6E 00   < A c c o u n
64E75CE50  74 00 5F 00 69 00 6E 00  66 00 6F 00 2E 00 6C 00  t _ i n f o . l
64E75CE60  6E 00 6B 00 00 00 00 00  CD 29 13 10 00 00 00 00  n k     Í)
64E75CE70  B6 29 13 10 00 00 00 00  B6 29 13 10 00 00 00 00  ¶)      ¶)
64E75CE80  68 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  h
64E75CE90  00 00 00 00 00 00 00 00  0B 00 0B 00 28 00 20 00              (
64E75CEA0  48 00 20 00 18 00 01 00  08 01 00 00 00 00 02 00  H
64E75CEB0  7F 26 00 00 00 00 00 00  7F 26 0C 00 00 00 00 00   &       &
64E75CEC0  00 00 0A 01 00 00 00 00  50 92 02 01 00 00 00 00          P'
64E75CED0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P'
64E75CEE0  00 00 0A 01 00 00 00 00  F0 91 02 01 00 00 00 00          ð'
64E75CEF0  50 92 02 01 00 00 00 00  00 00 0A 01 00 00 00 00  P'
64E75CF00  E0 29 13 10 00 00 00 00  CD 29 13 10 00 00 00 00  à)      Í)
64E75CF10  CD 29 13 10 00 00 00 00  38 00 00 00 00 00 00 00  Í)      8
64E75CF20  01 00 00 00 18 00 00 00  00 00 00 00 00 00 00 00
64E75CF30  07 00 07 00 28 00 08 00  30 00 08 00 18 00 01 00      (   0
64E75CF40  38 00 58 00 00 00 02 00  1B 6E 00 00 00 00 00 00  8 X      n
64E75CF50  1B 6E 0C 00 00 00 00 00  F0 91 02 01 00 00 00 00   n      ð'
64E75CF60  98 8F 02 01 00 00 00 00  ED 29 13 10 00 00 00 00  ˜)      í)
64E75CF70  E0 29 13 10 00 00 00 00  00 00 00 00 00 00 00 00  à)
64E75CF80  28 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  (
64E75CF90  02 00 00 00 00 00 00 00  1B 00 01 00 28 00 00 00              (
```

Fig 36: Operational Record of folder Account_info

At location, ***0x64E5CF20*** contains the Record type which is ***0x01***showing that it's a general record. Transaction ID has 0x18 page overflow flag and its value at location 0x 64E5CF28 is 0x00 which indicate that page is not overflowing. In the same way, remaining operational record has been.

$logfile data facts of Blood_count.docx represent the traces of the all the transactions done with the file as shown in fig 37



```
Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15
15530224  18 00 3C 00 24 00 49 00  59 00 48 00 34 00 4B 00    < $ I Y H 4 K
15530240  31 00 49 00 2E 00 6A 00  70 00 67 00 00 00 00 00  1 I . j p g
15530256  22 9F 1D A7 00 00 00 00  0C 9F 1D A7 00 00 00 00  "Ÿ §     Ÿ §
15530272  0C 9F 1D A7 00 00 00 00  68 00 00 00 00 00 00 00   Ÿ §    h
15530288  01 00 00 00 18 00 00 00  00 00 00 00 00 00 00 00
15530304  0B 00 0B 00 28 00 20 00  48 00 20 00 18 00 01 00      (   H
15530320  08 01 00 00 02 00 02 00  C6 33 00 00 00 00 00 00          Æ3
15530336  C6 33 0C 00 00 00 00 00  00 00 05 1C 00 00 00 00  Æ3
15530352  50 55 02 1C 00 00 00 00  50 55 02 1C 00 00 00 00  PU      PU
15530368  00 00 05 02 00 00 00 00  00 00 05 1C 00 00 00 00
15530384  F8 54 02 1C 00 00 00 00  50 55 02 1C 00 00 00 00  øT      PU
15530400  00 00 05 02 00 00 00 00  35 9F 1D A7 00 00 00 00          5Ÿ §
15530416  22 9F 1D A7 00 00 00 00  22 9F 1D A7 00 00 00 00  "Ÿ §    "Ÿ §
15530432  38 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  8
15530448  00 00 00 00 00 00 00 00  07 00 07 00 28 00 08 00               (
15530464  30 00 08 00 18 00 01 00  38 00 58 00 06 00 02 00  0       8 X
15530480  15 8C 00 00 00 00 00 00  A5 75 6D 00 00 00 F5 74   Œ      ¥um   õt
15530496  F8 54 02 1C 00 00 00 00  A0 54 02 1C 00 00 00 00  øT       T
15530512  42 9F 1D A7 00 00 00 00  35 9F 1D A7 00 00 00 00  BŸ §    5Ÿ §
15530528  00 00 00 00 00 00 00 00  28 00 00 00 00 00 00 00          (
15530544  01 00 00 00 18 00 00 00  02 00 00 00 00 00 00 00
15530560  1B 00 01 00 28 00 00 00  28 00 00 00 18 00 00 00      (   (
15530576  00 00 00 00 00 00 02 00  00 00 00 00 00 00 00 00
15530592  FF FF FF FF FF FF FF FF  4D 9F 1D A7 00 00 00 00  ÿÿÿÿÿÿÿÿMŸ §
15530608  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
15530624  98 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00  ˜
15530640  02 00 00 00 00 00 00 00  0F 00 0E 00 28 00 00 00               (
15530656  28 00 70 00 F8 10 01 00  00 00 18 02 00 00 08 00  ( p ø       _
15530672  00 00 00 00 00 00 00 00  AF 1C 00 00 00 00 00 00
15530688  64 32 02 00 00 00 02 00  70 00 5E 00 00 00 00 00  d2      p ^
15530704  00 4C 01 00 00 00 02 00  32 87 FE 35 CC 26 D5 01   L      2‡þ5Ì&Õ
15530720  00 EC 13 A6 C4 0E D5 01  52 87 EF 50 CE 26 D5 01   ì ¦Ä Õ R‡ïPÎ&Õ
15530736  32 87 FE 35 CC 26 D5 01  00 00 01 00 00 00 00 00  2‡þ5Ì&Õ
15530752  32 F3 00 00 00 00 00 00  20 00 00 00 00 00 00 00  2ó
15530768  0E 01 62 00 72 00 61 00  69 00 6E 00 5F 00 73 00    b r a i n _ s
15530784  63 00 61 00 6E 00 2E 00  6A 00 70 00 67 00 1D 00  c a n . j p g
15530800  66 9F 1D A7 00 00 00 00  4D 9F 1D A7 00 00 00 00  fŸ §    MŸ §
15530816  4D 9F 1D A7 00 00 00 00  A0 00 00 00 00 00 00 00  MŸ §
15530832  01 00 00 00 18 00 00 00  02 00 00 00 00 00 00 00
15530848  06 00 05 00 28 00 00 00  28 00 78 00 18 00 01 00      (   ( x
15530864  10 01 00 00 00 00 02 00  99 8C 00 00 00 00 00 00          ™Œ
15530880  29 76 6D 00 00 00 00 00  30 00 00 00 78 00 00 00  )vm     0   x
15530896  00 00 00 00 00 00 04 00  5E 00 00 00 18 00 01 00          ^
15530912  00 4C 01 00 00 00 02 00  32 87 FE 35 CC 26 D5 01   L      2‡þ5Ì&Õ
15530928  00 EC 13 A6 C4 0E D5 01  96 E9 00 36 CC 26 D5 01   ì ¦Ä Õ –é 6Ì&Õ
15530944  32 87 FE 35 CC 26 D5 01  00 00 01 00 00 00 00 00  2‡þ5Ì&Õ
15530960  32 F3 00 00 00 00 00 00  20 00 00 00 00 00 00 00  2ó
15530976  0E 01 62 00 72 00 61 00  69 00 6E 00 5F 00 73 00    b r a i n _ s
15530992  63 00 61 00 6E 00 2E 00  6A 00 70 00 67 00 F5 74  c a n . j p g õt
15531008  80 9F 1D A7 00 00 00 00  66 9F 1D A7 00 00 00 00  €Ÿ §    fŸ §
15531024  66 9F 1D A7 00 00 00 00  A0 00 00 00 00 00 00 00  fŸ §
15531040  01 00 00 00 18 00 00 00  02 00 00 00 00 00 00 00
15531056  06 00 05 00 28 00 00 00  28 00 78 00 18 00 01 00      (   ( x

15531072  98 00 00 00 00 00 02 00  99 8C 00 00 00 00 00 00  ˜       ™Œ
15531088  29 76 6D 00 00 00 00 00  30 00 00 00 78 00 00 00  )vm     0   x
15531104  00 00 00 00 00 00 05 00  5A 00 00 00 18 00 01 00          Z
15531120  00 4C 01 00 00 00 02 00  32 87 FE 35 CC 26 D5 01   L      2‡þ5Ì&Õ
15531136  00 EC 13 A6 C4 0E D5 01  96 E9 00 36 CC 26 D5 01   ì ¦Ä Õ –é 6Ì&Õ
15531152  32 87 FE 35 CC 26 D5 01  00 00 01 00 00 00 00 00  2‡þ5Ì&Õ
15531168  32 F3 00 00 00 00 00 00  20 00 00 00 00 00 00 00  2ó
15531184  0C 02 42 00 52 00 41 00  49 00 4E 00 5F 00 7E 00    B R A I N _ ~
15531200  31 00 2E 00 4A 00 50 00  47 00 00 00 00 00 00 00  1 . J P G
15531216  9A 9F 1D A7 00 00 00 00  80 9F 1D A7 00 00 00 00  šŸ §    €Ÿ §
15531232  80 9F 1D A7 00 00 00 00  98 00 00 00 00 00 00 00  €Ÿ §    ˜
15531248  01 00 00 00 18 00 00 00  02 00 00 00 00 00 00 00
15531264  0F 00 0E 00 28 00 00 00  28 00 70 00 F8 10 01 00      (   ( p ø
15531280  00 00 88 02 00 00 08 00  00 00 00 00 00 00 00 00    ˆ
15531296  AF 1C 00 00 00 00 00 00  64 32 02 00 00 00 02 00  ¯       d2
15531312  70 00 5A 00 00 00 00 00  00 4C 01 00 00 00 02 00  p Z      L
15531328  32 87 FE 35 CC 26 D5 01  00 EC 13 A6 C4 0E D5 01  2‡þ5Ì&Õ  ì ¦Ä Õ
15531344  52 87 EF 50 CE 26 D5 01  32 87 FE 35 CC 26 D5 01  R‡ïPÎ&Õ 2‡þ5Ì&Õ
15531360  00 00 01 00 00 00 00 00  32 F3 00 00 00 00 00 00          2ó
15531376  20 00 00 00 00 00 00 00  0C 02 42 00 52 00 41 00            B R A
15531392  49 00 4E 00 5F 00 7E 00  31 00 2E 00 4A 00 50 00  I N _ ~ 1 . J P
15531408  47 00 00 00 00 00 1D 00  B3 9F 1D A7 00 00 00 00  G       ³Ÿ §
15531424  9A 9F 1D A7 00 00 00 00  9A 9F 1D A7 00 00 00 00  šŸ §    šŸ §
15531440  A0 00 00 00 00 00 00 00  01 00 00 00 18 00 00 00
15531456  04 00 00 00 00 00 00 00  05 00 06 00 28 00 78 00               ( x
```

Fig:36 traces of transaction in Brain_scan.jpg

Workflow of deletion process has been illustrated in fig 38

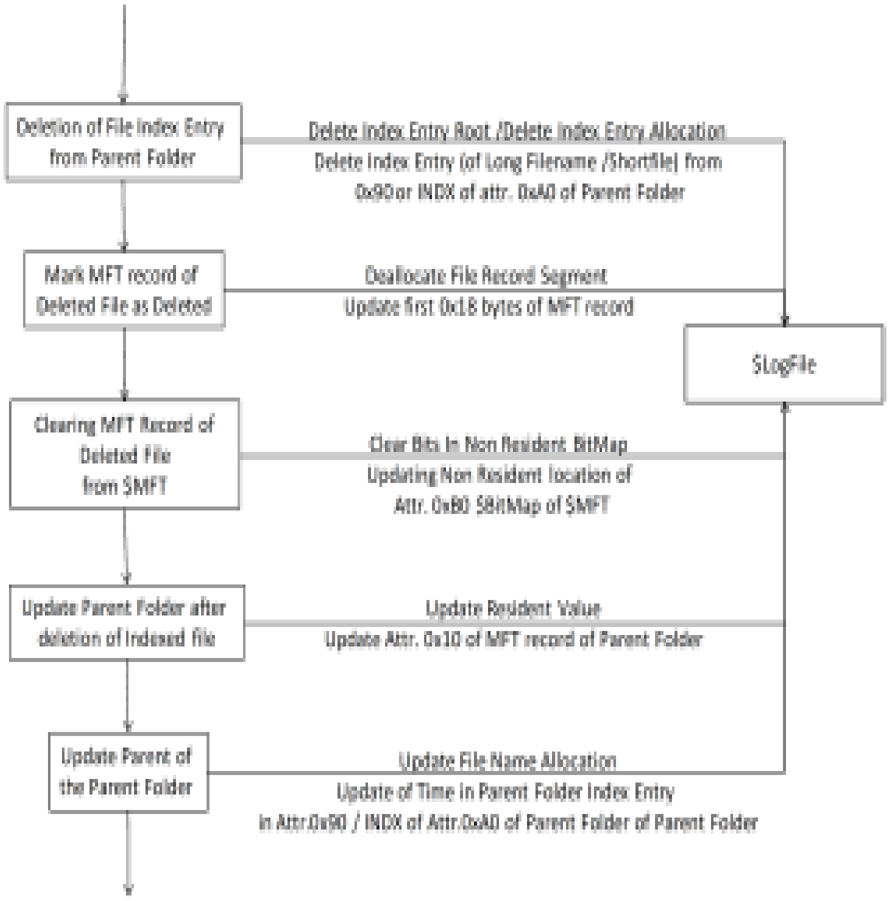

Deletion of File Index Entry from Parent Folder
Delete Index Entry Root /Delete Index Entry Allocation
Delete Index Entry (of Long Filename /Shortfile) from 0x90or INDX of attr. 0xA0 of Parent Folder
Mark MFT record of Deleted File as Deleted
Deallocate File Record Segment
Update first 0x18 bytes of MFT record
$LogFile
Clearing MFT Record of Deleted File from $MFT
Clear Bits In Non Resident BitMap
Updating Non Resident location of Attr. 0xB0 $BitMap of $MFT
Update Parent Folder after deletion of Indexed file
Update Resident Value
Update Attr. 0x10 of MFT record of Parent Folder
Update Parent of the Parent Folder
Update File Name Allocation
Update of Time in Parent Folder Index Entry
in Attr.0x90 / INDX of Attr.0xA0 of Parent Folder of Parent Folder

Fig 38: Deletion Process of a file in the $logfile

5.5 Summary-Forensic Artifacts

File creation and deletion process as discussed in chapter4 left many artifacts and they are backedup by Volume Shadow Copy.

Table 14: Forensic Artifacts of deleted files using Recycle.Bin

	$MFT		**$Logfile**	
Artifacts	**File**	**Parent**	**File**	**Parent**
File Name	Record of MFT remain unchanged Only the record header does not change Attr.0x30 (name started with $R followed by six random integers) Metadata of deleted MFT record is stored in $Ixxx file which containsthe name and location of the file.	Deleted folder and files are completely removed from theparent's folder and moved to the Recycle.bin folder of current user renaming to $Rxxx.Index entry will be renamed (name+time+size) of $Rxxx in Recycle bin of Current User Folder If original / renamed file was indexed at the end of 0x90 or INDX of Attr. 0xA0, deletion will leave this index entry (name+time+size) as file slack	$logfile records remain unchanged until overwritten by a new one when $logfile is out of space but during the deletionprocess, new records have been generated Delete Index Entry Root / Index Entry Allocation: Index Record of $Ixxx in Attr. 0x90/INDX of Attr. 0xA0 of Current User Folder of Recycle Bin	$logfile records remain unchanged until overwritten by a new one when $logfile is out of space but during the deletionprocess, new records have been generated Delete Index Entry Root / Index Entry Allocation: Index Record of $Rxxx in Attr. 0x90 / INDX of Attr.0xA0 of Current User Folder of Recycle Bin
File Time	Attr. 0x10 (Time Info related to entire MFT Record). This time will also give information on file deletion to Recycle Bin Attr. 0x30 (Time Info Related to File Name). This time will also give information of file deletion to Recycle Bin	Renamed Index Entry (name+time+size) of $Rxxx in Current User Folder of Recycle Bin If original / renamed file was indexed at the end of 0x90 or INDX of Attr. 0xA0, deletion will leave this index entry (name+time+size) as file slack	All $logfile records created during File Creation, remain unchanged. They are overwritten with new records after entire $Logfile space has been consumed. However, the deletion process generates its own records Delete Index Entry Root / Index Entry Allocation: Index Record of $Ixxx in Attr. 0x90/INDX of Attr. 0xA0 of Current User Folder of Recycle Bin	All $logfile records created during File Creation, remain unchanged. They are overwritten with new records after the entire$Logfile space has been consumed. However, the deletion process generates itsown recordsUpdate File Name Allocation: Time and Size info of Index entry of Current User Folder (longfilename and short film) in INDX of Attr. 0xA0 of Recycle Bin after clearing recycle bin reveals the time of recycle bin empty operation. Update Resident Value: Attr. 0x10 of parent folder after the deleted file has been de-indexedReveals the time of file deletion to recycle bin

During the experiment following important artifacts at following offsets found.

Table 15: Artifacts found in the experiment

Offset	Remarks	Findings/Artifacts
0x1E00	Start of the Shadow Service	Volume Shadow Service Header always starts from this offset
0x71f0000	Offset of the first catalog block offset	First catalog block
625f84000		The offset of first store block extracted from entry type of 0x03
625f8000	Relative offset of first store block	625f8C000
	Time Stamp Creation time of 1st Volume Shadow	*01D4BD41EBB41F96(5-2-2019 11:31)*
639f84000	Relative offset of second store block	639f84000+0xC000=639f84C00
	Creation time of 2nd Volume Shadow	*01D4BD42189D1FE6 (5-2-2019 11:43)*
64DF8400	Relative offset of third store block	64DF84000+0xC000 =64DF84000
	Creation time of 3rd Volume Shadow	*01D4BD424f738980(5-2-2019 11:57)*
661f84000	Relative offset of fourth store block	661f84000+0xc000=661f84c000
	Creation time of 4th Volume Shadow	Creation time of 4th Volume Shadow *01D4BD42943848F1(5-2-2019 12:16)*
C5FE5800	*Indexed attribute of deleted folder*	*S-1-5-21-3207469532-412700025-1003426619-1001~$I30 (90)*
C7A09920	*Offset of $I file of deleted folder Account_Info*	*$I8HLBDG file*
C5EAFDB8	*Account_information.txt File was permanently deleted from the system but present in the Volume Shadow Copy)*	*Account_information.txt file contents*
1FA237000	*Bank statement.png file which was deleted but recovered using Volume Shadow Copy)*	*Bank statement.png file*
0x64E75CDE0	*LSN of Account_info folder in $logfile*	*LSN of Account_info folder in $logfile*

After analyzing the Volume Shadow copy Recycle.Bin little difference has been found between $I file of $Recycle.Bin of Windows 7,8,10

Table 16: Comparison between $I file of Window 7,8 and 10

	Header (0x01)		Deleted File Size		Deleted Time Stamp		File Name Length		File Name Path	
	S.B*	Size	S.B*	Size	S.B*	Size	S.B*	Size	S.B*	Size
Windows 7,8	0	8	8	16	16	8	24			
Windows 10	0	8	8	16	16	8	24	4	2 8	Variable length

*Chapter*6

Review of tools used for the analysisof Volume Shadow Copy

After manual analysis of the Volume Shadow Copy for verification purpose, automated analysis of the Volume Shadow Copy has also been done. For automated analysis different freely available tools which support the Volume Shadow Copy have been used.

Many Digital Forensic Tools does not support Volume Shadow Copy.

Shadow Explorer

Shadow Explorer is a digital forensic tool specially made for analysis of Volume Shadow Copy. Allow the users of windows vista/7/8 and 10 to access the point in time shadow copies of the system.It does not support raw images. It only accesses the Volume Shadow Copies from the live volume. user can export any folder/file. [*23*] . The interface of the shadow explorer has been shown in Fig 37

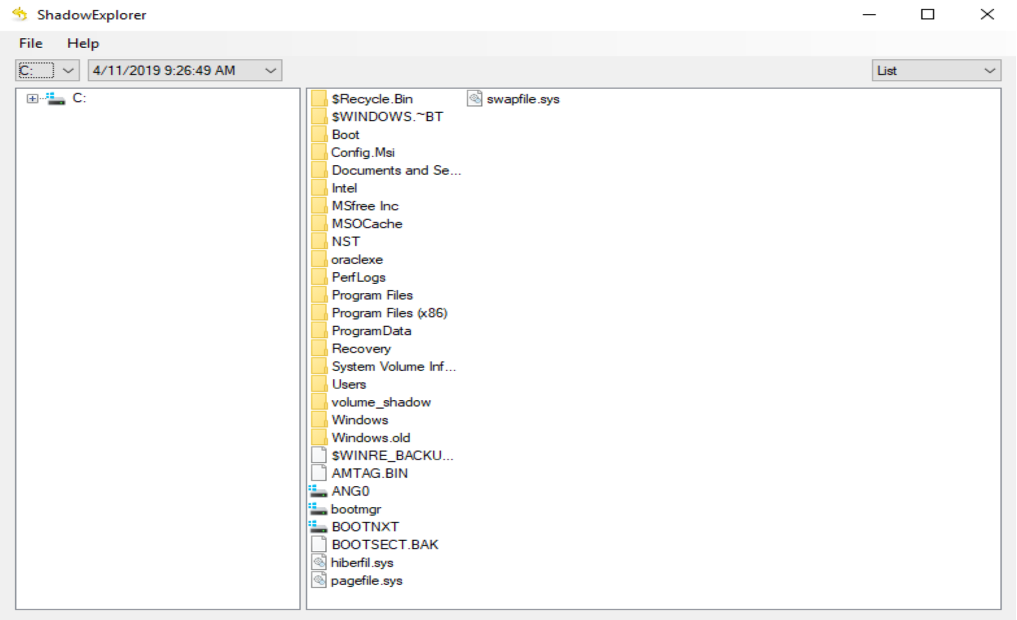
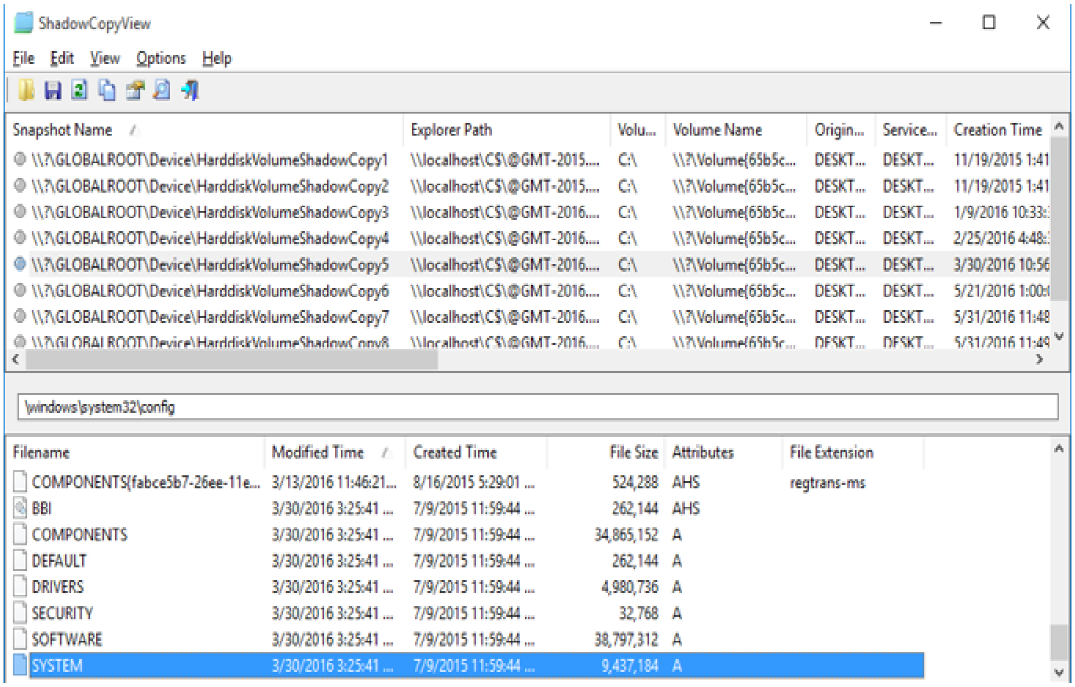


Fig: 37 Screen Shot of Shadow Explorer

Shadow Copy View

Shadow copy View is a simple tool that let the user explore the volume shadow copies of his system. It also does not support images of the system. It gives the edge to the investigator to see the directories before manual analysis of the volume Shadow copies [*26*]. The simple view of the Volume Shadow View is shown in Fig:33



Snapshot Name	Explorer Path	Volu...	Volume Name	Origin...	Service...	Creation Time
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1	\\localhost\C$\@GMT-2015....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	11/19/2015 1:41
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy2	\\localhost\C$\@GMT-2015....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	11/19/2015 1:41
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy3	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	1/9/2016 10:33:
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy4	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	2/25/2016 4:48:
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy5	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	3/30/2016 10:56
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy6	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	5/21/2016 1:00:
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy7	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	5/31/2016 11:48
\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy8	\\localhost\C$\@GMT-2016....	C:\	\\?\Volume{65b5c...	DESKT...	DESKT...	5/31/2016 11:49

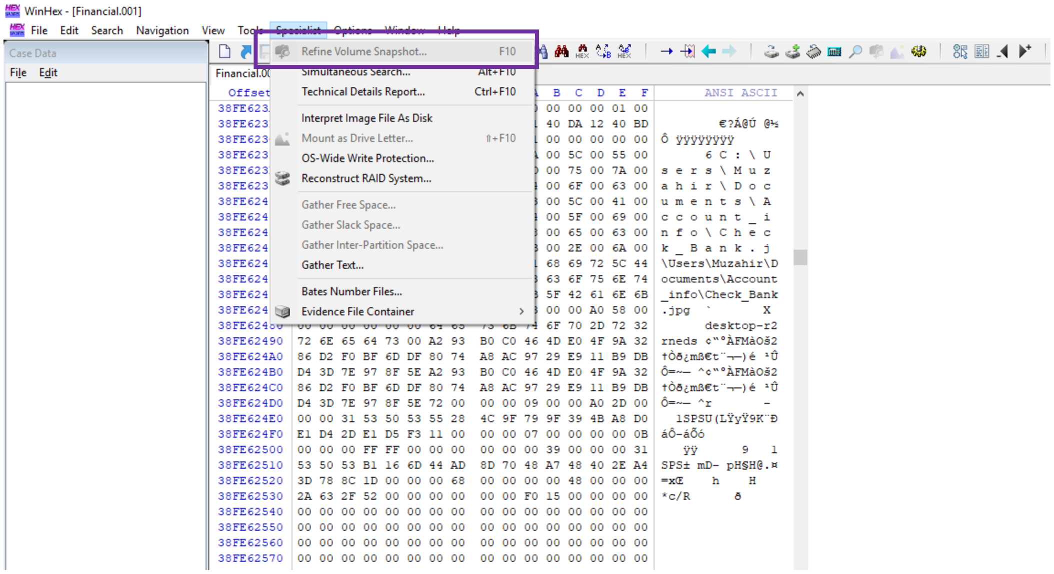
\windows\system32\config

Filename	Modified Time	Created Time	File Size	Attributes	File Extension
COMPONENTS{fabce5b7-26ee-11e...	3/13/2016 11:46:21...	8/16/2015 5:29:01 ...	524,288	AHS	regtrans-ms
BBI	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	262,144	AHS	
COMPONENTS	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	34,865,152	A	
DEFAULT	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	262,144	A	
DRIVERS	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	4,980,736	A	
SECURITY	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	32,768	A	
SOFTWARE	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	38,797,312	A	
SYSTEM	3/30/2016 3:25:41 ...	7/9/2015 11:59:44 ...	9,437,184	A	

Fig 38: Screenshot of ShadowCopyView

Winhex

Winhex is a powerfuldigital forensic toolfor the investigators to investigate and find the evidence. Its academia version is freely available. In Winhex Refine Snapshot feature is included but only in licensed version under the specialist option shown in fig 39. It gives the option of a manual analysis of the Volume Shadow Copy briefly explained in the above chapters.



WinHex - [Financial.001]
File Edit Search Navigation View Tools Specialist Options Window Help
Case Data
File Edit
Financial.00
Refine Volume Snapshot... F10
Simultaneous Search... Alt+F10
Technical Details Report... Ctrl+F10
Interpret Image File As Disk
Mount as Drive Letter... ⇑+F10
OS-Wide Write Protection...
Reconstruct RAID System...
Gather Free Space...
Gather Slack Space...
Gather Inter-Partition Space...
Gather Text...
Bates Number Files...
Evidence File Container

```
Offset                                 B  C  D  E  F       ANSI ASCII
38FE623                             00 00 00 01 00
38FE623                          40 DA 12 40 BD           €?Á@Ú @½
38FE623                          00 00 00 00 00    Ô ÿÿÿÿÿÿÿÿ
38FE623                          00 5C 00 55 00          6 C : \ U
38FE623                          00 75 00 7A 00    s e r s \ M u z
38FE623                          00 6F 00 63 00    a h i r \ D o c
38FE624                          00 5C 00 41 00    u m e n t s \ A
38FE624                          00 5F 00 69 00    c c o u n t _ i
38FE624                          00 65 00 63 00    n f o \ C h e c
38FE624                          00 2E 00 6A 00    k _ B a n k . j
38FE624                          68 69 72 5C 44    \Users\Muzahir\D
38FE624                          63 6F 75 6E 74    ocuments\Account
38FE624                          5F 42 61 6E 6B    _info\Check_Bank
38FE624                          00 00 A0 58 00    .jpg  `        X
38FE624                          6F 70 2D 72 32          desktop-r2
38FE62490  72 6E 65 64 73 00 A2 93  B0 C0 46 4D E0 4F 9A 32    rneds ¢“°ÀFMàOš2
38FE624A0  86 D2 F0 BF 6D DF 80 74  A8 AC 97 29 E9 11 B9 DB    †Òð¿mß€t¨¬—)é ¹Û
38FE624B0  D4 3D 7E 97 8F 5E A2 93  B0 C0 46 4D E0 4F 9A 32    Ô=~— ^¢“°ÀFMàOš2
38FE624C0  86 D2 F0 BF 6D DF 80 74  A8 AC 97 29 E9 11 B9 DB    †Òð¿mß€t¨¬—)é ¹Û
38FE624D0  D4 3D 7E 97 8F 5E 72 00  00 00 09 00 00 A0 2D 00    Ô=~— ^r        -
38FE624E0  00 00 31 53 50 53 55 28  4C 9F 79 9F 39 4B A8 D0      1SPSU(LŸyŸ9K¨Ð
38FE624F0  E1 D4 2D E1 D5 F3 11 00  00 00 07 00 00 00 00 0B    áÔ-áÕó
38FE62500  00 00 00 FF FF 00 00 00  00 00 00 39 00 00 00 31       ÿÿ       9   1
38FE62510  53 50 53 B1 16 6D 44 AD  8D 70 48 A7 48 40 2E A4    SPS± mD- pH§H@.¤
38FE62520  3D 78 8C 1D 00 00 00 68  00 00 00 00 48 00 00 00    =xŒ    h    H
38FE62530  2A 63 2F 52 00 00 00 00  00 00 F0 15 00 00 00 00    *c/R      ð
38FE62540  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
38FE62550  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
38FE62560  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
38FE62570  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

Fig 39: Screenshot of winhex academia version

Comparative analysis of these Forensic tools has been shown in table 13:

Table13: Comparative analysis of Volume Shadow Copy

Name	Freely available	Deals with Image files	System files	Shadow Copy Analysis facility
Shadow Explorer	Yes	No	Yes	No (Only gives the view of directory
ShadowCopyView	Yes	No	Yes	No (Only gives the view of directory)
Forensic Explorer	No (Only 30 days Trial with limited features)	Yes	Yes	Yes (Only in Licensed version)
WinHex	No (Only Academia Version is available)	Yes	Yes	Yes (Only in Licensed version)

Chapter 7

Discussion and Future work

This research thesis has focused on the Volume Shadow Copy its importance and way of analyzing the Volume Shadow Copy of windows 10. Volume Shadow Copy is a goldmine for forensic investigators as it contains an older version of the files with $logfile backed up with each Volume Shadow Copy and $Recycle.Bin both of these files are enriched with forensic artifacts.

It is essential for the computer forensic investigators to know about the importance and complications of the Windows Volume Shadow Copy to extract the evidence. Volume shadow copy contains the time in point snapshots of the system. Instead of saving the whole file at once it just saves the difference from the previously stored file. so, locating just only the difference file is a little bit tricky and complicated. It does not save the whole file every time the shadow is created. If any file is deleted and before deletion Volume Shadow Copy has been created it will be saved even if it is deleted bypassing the Recycle Bin. Volume Shadow Copy also contains the $logfile till that point of time provides the clumps of important data that can be investigated to find the evidence. Manual Analysis of the Volume shadow copy from a system image has been done which is accurate but lengthy and time-consuming process. Volume Shadow copy is a treasure for the forensic investigator, so more precise and accurate tools should be developed which can analyze the volume shadow copy accurately and precisely as most of the tools provide only the view of the volume shadow copies of live systems and do not provide the facility to analyze volume shadow copy from system images.

To conduct the experiment two case scenarios have been generated and tested. To successfully conduct the experiment different files have been created and modified and moved around different directories to create the traces. After taking the images they have been analyzed through Winhex Academia version.

Different shadow copies have been created for modified, changed the path and deleted files/folders. Different traces have been found in different shadow copies. The $logfile and $Recycle.Bin has also been analyzed. The changed state of the $logfile for that point of time has also been copied and analyzed. $Recycle.Bin is important as before emptying the recycle.bin if it is copied by the Volume Shadow Service, it’s all contents will also be saved and can be a great point of interest for the forensic investigators.

In this thesis, only $logfile and $Recycle.bin has been analyzed but Volume Shadow Copy has a lot more than these two files. It not only has the previous versions of the user files but also has the previous versions of the system files. Complete analysis of Volume Shadow Copy with all feature e.g $logfile in each shadow can be taken as a continuation of this research work.

References

1. Jewan Bang, Sangjin Lee: Silhouette: Volume Shadow Copy Analyzer. In: James J. Park, Victor C.M. Leung, Cho-Li. Wang, Taeshik Shon. Future Information Technology, Application, and Services, Vol. 1,pp. 721-730. Springer, Heidelberg (2012).
2. https://docs.microsoft.com/en-us/windows-server/storage/file-server/volume-shadow-copy-service)
3. Timothy R. Leschke,"$Recycle.Bin Forensics for Windows 7 and Windows Vista Shadow Volumes"
4. James Crabtree, Gary Evans: Reliably recovering evidential data from Volume Shadow Copies in Windows Vista and Windows 7.
5. http://www.forensicswiki.org/wiki/Windows_Shadow_Volumes
6. Minoru Kobayashi, HiroshiSuzuki,"Reconstruct the world from deleted shadows"4-9-2018
7. Joachim Metz: Volume Shadow Snapshot (Volume Shadow Copy) Analysis the Windows NT Volume Shadow Copy format
8. Sareeja S c, c Balan 2016 International Conference on Emerging Technological Trends [ICETT]
9. Joachim Metz: Windowless shadow copies Snapshot (Volume Shadow Copy) Analysis the Windows NT Volume Shadow Copy format
10. David Cowen. (2013, January) [Online].http://hackingexposedcomputerforensicsblog.blogspot.com/2013/01/ntfstriforce-deeper-look-inside.html
11. Sameer H. MahantB.B.Meshram: " NTFS Deleted Files Recovery: Forensics View" RACIST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 2, No.3, June 2012

12. Junghoon Oh. (2014, July) NTFS-log-tracker. [Online]. http://forensicinsight.org/wp-content/uploads/2013/06/F-INSIGHT-NTFSLog-TrackerEnglish.pdf?ckattempt=1

13. NTFS.com. [Online]. http://ntfs.com/transaction.htm

14. Peter Baer Galvin and Abraham Silberschatz, *Operating System Concepts*., 1998.

15. $Recycle.Bin Forensics forWindows 7 and Windows VistaTimothy R. LeschkeForensic Computer EngineerU.S. Department of Defense Cyber Crime Institute911 Elkridge Landing Road, Suite 450Linthicum, MD 21090Timothy.Leschke.ctr@dc3.mi

16. https://www.magnetforensics.com/blog/artifact-profile-recycle-bin/

17. https://digital-forensics.sans.org/blog/2011/09/20/ntfs-i30-index-attributes-evidence-of-deleted-and-overwritten-files

18. https://accessdata.com/product-download/

19. http://www.winhex.com/winhex/

20. Derek Newton. (2010, June) Recycle Bin Forensics in Windows 7 and Vista. [Online].http://dereknewton.com/2010/06/recycle-bin-forensics-in-windows-7-and-vista/

21. Jason Medeiros: Exploring NTFS- Forensic Extraction of Data. Grayscale Research 2008

22. Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein, "BTrees," in *Introduction to Algorithms*, Second, Ed.: MIT Press and McGraw-Hill, 2001, ch. 18, pp. 434–454

23. [http://www.forensicexplorer.com/]

24. NTFS Documentation [Online] http://iplsqldeveloper.sourceforge.net/ntfs/concepts/file_record.html

25. Nirsofer "www.nirsofer.net "Para 1 2016-2017. [online available]:http:\\nirsoft.net/shadow_copy_view.html[accessed Apr, 30 2019].

26. B. Yoo, J. Park, J. Bang and S. Lee, "A Study on a Carving Method for Deleted NTFS Compressed Files," in Human-Centric Computing (HumanCom), 2010 3rd International Conference, Cebu, Philippines, 2010

27. E. Casey, "Chapter 5: Windows Forensic Analysis," in Handbook of Digital Forensics and Investigation, United States of America, Elsevier Inc., 2010, pp. 209-300

28. Z. Kai, C. En and G. Qinquan, "Analysis and Implementation of NTFS File System Based on Computer Forensics," in The Second International Workshop on Education Technology and Computer Science, Wuhan, Hubei, China, 2010.