

402 – Find suspicious Files

Team Information

Team Name: ISEGYE_IDOL

Team Member: Eungchang Lee, Sojeong Kim, Mingyu Seong, Donghyun HA

Email Address: dfc-isegyeidol@googlegroups.com

Instructions

Description While analyzing the drug suspect's PC, the police found traces of accessing Dropbox through a web browser while the USB was connected. The download history of the web browser is all deleted, so it is not known for sure, but it is suspected that some files were downloaded from Dropbox to USB. Analyze the USB image to find the files that are suspected to have been downloaded from Dropbox and find related files.

Target	Hash (MD5)
USB.dd	add65ecdb718cee281a47a297b65d7df

Questions

1. What file(s) did the suspect download from Dropbox? (50 points)
2. Find completely deleted files in USB. (20 points)
3. What is the content of the first file that was completely deleted? (Hint: Person) (50 points)
4. What is the content of the second file that was completely deleted? (Hint: Appointment) (80 points)
5. Implement and submit a tool to track the history of changes to file data. (The tool should be implemented to solve problems #1, #3, and #4) (200 points)

Teams must:

- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).

Tools used:

Name:	FTK Imager	Publisher:	Access Data
Version:	4.2.1.4		
URL:	https://accessdata.com/product-download/ftk-imager-version-4-5		

Name:	HashTab	Publisher:	Implbits Software
Version:	6.0.0		
URL:	http://implbits.com		

Name:	HxD	Publisher:	Maël Hörz
Version:	2.5.0.0		
URL:	www.mh-nexus.de		

Name:	NTFS Log Tracker	Publisher:	Junghoon Oh
Version:	1.7.1		
URL:	https://sites.google.com/site/forensicnote/ntfs-log-tracker		

Name:	DB browser for SQLite	Publisher:	Digital Ocean
Version:	3.12.2		
URL:	https://sqlitebrowser.org/		

Name:	dfir_ntfs	Publisher:	msuhanov
Version:	-		
URL:	https://github.com/msuhanov/dfir_ntfs		

Step-by-step methodology:

다운받은 USB.dd 이미지에 대한 md5 값을 통해 원본 파일과 일치함을 확인한다.

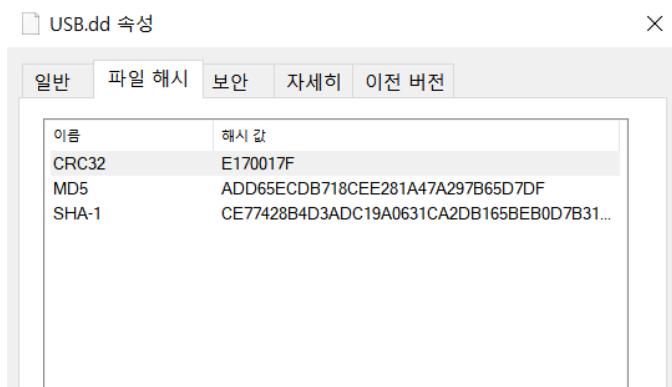


그림 1. USB.dd의 MD5값 확인

1. What file(s) did the suspect download from Dropbox? (50 points)

FTK Imager로 USB.dd 이미지를 열고 파일에 대한 다운로드 기록과 완전 삭제 기록, 그리고 삭제된 파일의 내용을 확인하기 위해 \$LogFile과 \$MFT파일을 추출한다.

Evidence Tree		File List			
		Name	Size	Type	Date Modified
USB.dd	NONAME [NTFS]	BadClus	0	Regular File	2022-03-15 오전 7:19:47
	[Orphan]	Bitmap	120	Regular File	2022-03-15 오전 7:19:47
	[root]	\$Boot	8	Regular File	2022-03-15 오전 7:19:47
		\$I30	4	NTFS Index...	2022-04-05 오전 5:44:38
		\$LogFile	8,480	Regular File	2022-03-15 오전 7:19:47
		\$MFT	256	Regular File	2022-03-15 오전 7:19:47
		\$MFTMirr	4	Regular File	2022-03-15 오전 7:19:47
		\$Secure	1	Regular File	2022-03-15 오전 7:19:47
		STVFS DATA	1	NTFS Log	2022-04-05 오전 5:44:38

그림 2. \$LogFile과 \$MFT 파일 추출

해당 이미지 파일에서 \$UsnJrnI은 주어지지 않았다.

The screenshot shows the NTFS Log Tracker interface with the following configuration:

- Target Path:** \$LogFile File Path: C:\Users\Wehfeh\Desktop\WDFCW402 - Find suspicious files\ logfile
- \$UsnJrnI File Path:** (for UsnJrnI Record Carving)
- Source Files Folder Path:** (for UsnJrnI Record Carving)
- Option:**
- \$MFT File Path:** C:\Users\Wehfeh\Desktop\WDFCW402 - Find suspicious files\ mft
- Open SQLite DB File:**
- SQLite DB File Path:**

The results table displays log entries from LSN 4449391 to 6000. The columns include LSN, EventTime(UTC+0), Event, Date, File/Directory Name, Full Path (from MFT), Create Time, Modified Time, MFT_Modified T..., Access Time, and Redo. The log includes various events such as file creation, deletion, and renaming, along with their corresponding file paths and timestamps.

그림 3. NTFS Log Tracker를 통해 \$LogFile, \$MFT parsing

NTFS Log Tracker로 \$LogFile과 \$MFT파일에 대한 파싱된 데이터를 db파일로 뽑아낸다. 그 후, DB Browser for SQLite로 저장된 db파일을 열어 분석을 진행했다.

The screenshot shows a SQLite database browser displaying a table named 'Logfile' with the following schema:

LSN	EventTime	Event	Detail	FileName
582	2022-04-05 14:26:41	File Deletion		8_3_20_b.txt
583	2022-04-05 14:26:53			Job3
584	2022-04-05 14:27:50	Directory Creation		새 폴더
585	2022-04-05 14:27:52	Renaming Directory	새 폴더 -> Job2	Job2
586	2022-04-05 14:29:38	File Creation		Area2
587	2022-04-05 14:29:38	File Deletion		Info5.docx
588	2022-04-05 14:29:38	File Creation	Data Runs(in Volume) : 919728(5)	Info5.docx
589	2022-04-05 14:29:38	File Creation	Data Runs(in Volume) : 919728(5)	미 확인 212633.crdownload
590	2022-04-05 14:29:38			\$\$Secure
591	2022-04-05 14:29:38	Renaming File	미 확인 212633.crdownload -> Info5.docx	Info5.docx
592	2022-04-05 14:29:38			Info5.docx
593	2022-04-05 14:31:32	File Creation		2_2_5_a.txt
594	2022-04-05 14:31:32	File Deletion		2_2_5_a.txt
595	2022-04-05 14:31:32	File Creation		2_2_5_a.txt
596	2022-04-05 14:31:32	Writing Content of Non-Resident File	Data Runs(in Volume) : 919726(1)	2_2_5_a.txt
597	2022-04-05 14:31:32	Writing Content of Non-Resident File	Data Runs(in Volume) : 919733(1)	2_2_5_a.txt
598	2022-04-05 14:31:32	Writing Content of Non-Resident File	Data Runs(in Volume) : 919734(1)	2_2_5_a.txt
599	2022-04-05 14:31:32	Writing Content of Non-Resident File	Data Runs(in Volume) : 919735(1)	2_2_5_a.txt
600	2022-04-05 14:31:32	Writing Content of Non-Resident File	Data Runs(in Volume) : 919736(1)	2_2_5_a.txt

그림 4. crdownload 흔적 확인

LSN(Logfile Sequence Number)가 4440584인 부분에서 파일 생성 이벤트가 발생했고, cluster volume 919728에 저장되었으며 파일 이름은 미확인 212633.crdownload이다.

실제로 저장된 것을 확인하기 위해 Hxd에서 USB.dd 이미지를 열고, cluster기 때문에 $919728 * 8$ 을 계산한 값 7357824로 섹터 이동을 진행했다.

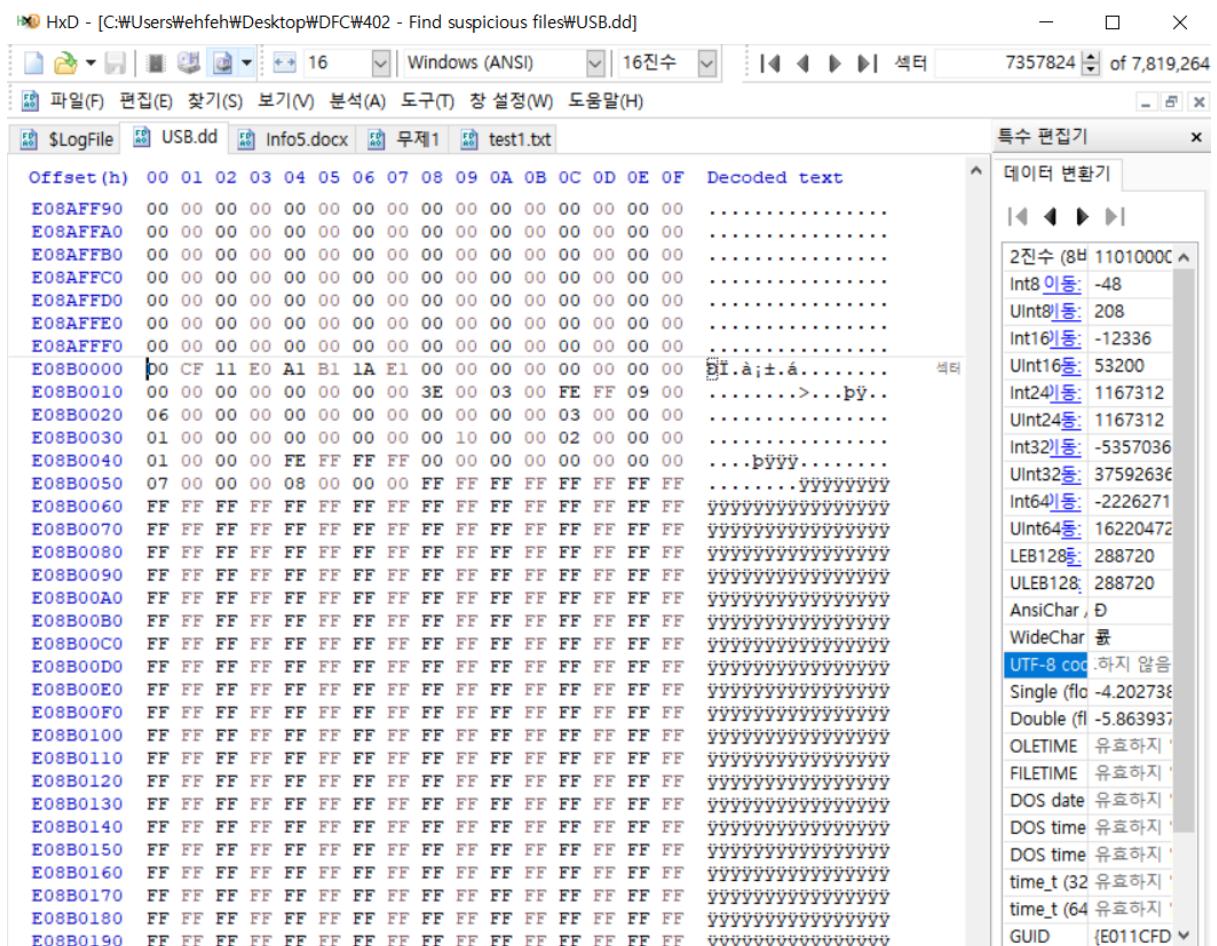


그림 5. 섹터 7357824에 존재하는 미확인 212633.crdownload 확인

해당 파일은 OLE 형식으로 되어있었고, 앞선 그림 4에서 Info5.docx로 renaming된 것을 보아 docx 파일이 암호화되어 있음을 알 수 있다. 따라서, 해당 파일은 Renaming되었다는 사실을 알 수 있고 docx파일을 암호화할 때 사용된 Key값들이 해당 파일안에 존재하기 때문에, USB.dd에서 Area2₩Job2 폴더에 있는 Info5.docx를 export해서 파일을 비교했다.

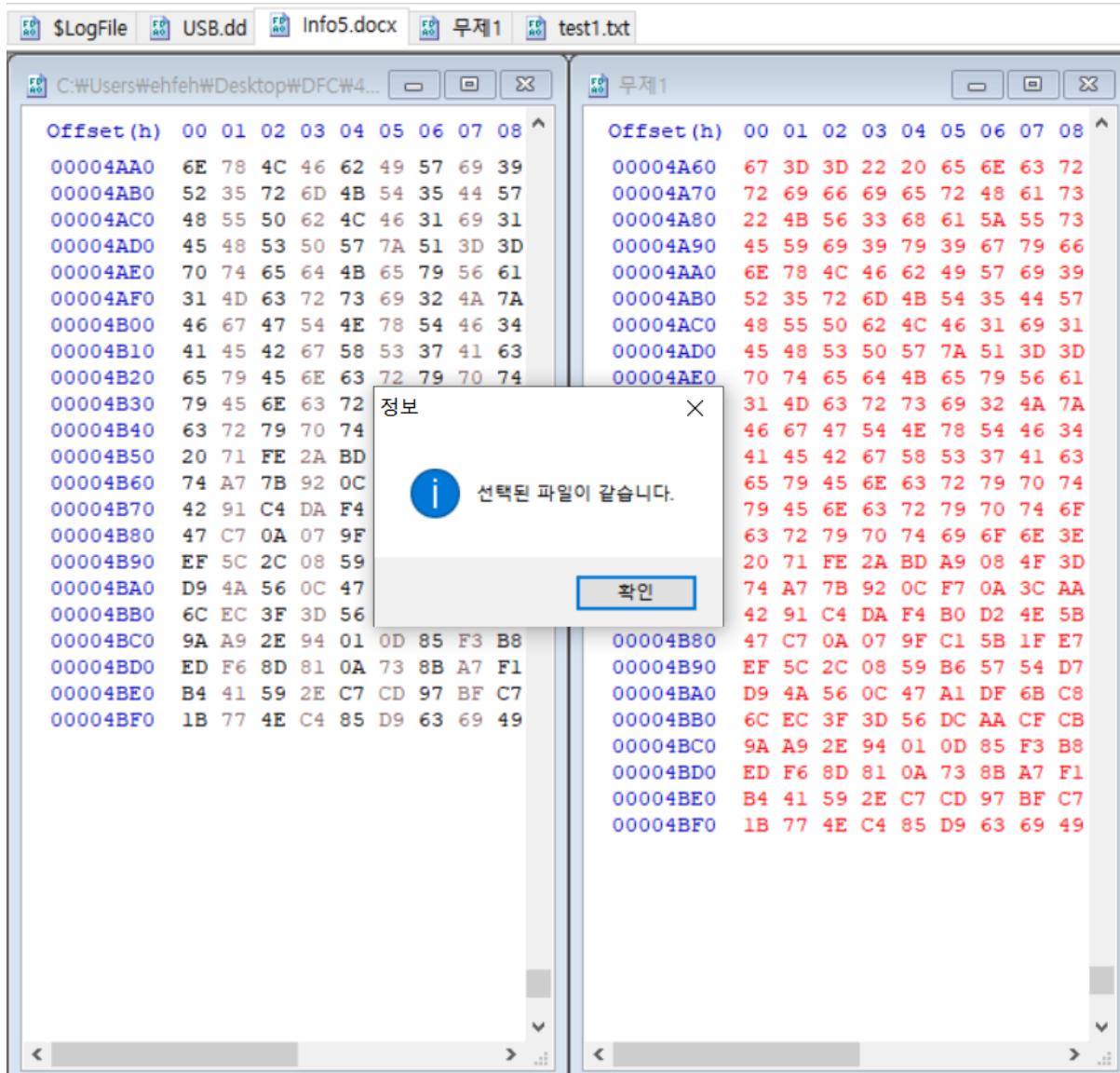


그림 6. Info5.docx와의 비교

앞서 섹터 7357824인 0xE08B000부터 파일의 끝까지 hex값을 블록선택 및 복사하여 새로운 곳에 붙여넣기 한 후, Hxd의 [분석]-[데이터 비교]-[비교]를 통해 확인한 결과 동일함을 알 수 있었다.

한편, 파일이 Dropbox에서 다운된 기록을 알기 위해서 LSN 기반으로 분석했다.

LSN	EventTime	Event	Detail	FileName
필터	필터	필터	필터	필터
582	4438216 2022-04-05 14:26:41	File Deletion		8_3_20_b.txt
583	4436253			Job3
584	4439366 2022-04-05 14:27:50	Directory ...		새 폴더
585	4439541 2022-04-05 14:27:52	Renaming ...	새 폴더 -> Job2	Job2
586	4439607			Area2
587	4440326 2022-04-05 14:29:38	File Creation		Info5.docx
588	4440409 2022-04-05 14:29:38	File Deletion		Info5.docx
589	4440584 2022-04-05 14:29:38	File Creation	Data Runs(in Volume) : 919728(5)	미확인 212633.crdownload \$Secure
590	4440954			
591	계산기	- □ ×	212633.crdownload -> Info5.docx	Info5.docx
592	☰ 프로그래머			Info5.docx
593				2_2_5.a.txt
594				2_2_5.a.txt
595				2_2_5.a.txt
596				2_2_5.a.txt
597	HEX 43 C208			2_2_5.a.txt
598	DEC 4,440,584			2_2_5.a.txt
599	OCT 20 741 010			2_2_5.a.txt
600	BIN 0100 0011 1100 0010 0000 1000			2_2_5.a.txt

그림 7. download 파일의 LSN 값으로 Hxd상 \$LogFile에서 조회

해당 파일을 다운로드 하고 Volume 919728에 기록된 시점의 LSN 4440584(0x43C208)을 Hxd에서 검색해보면, 0x1E1040에서 \$MFT와 결합된 구조로 데이터가 나타난다.

001E1040	08 C2 43 00 00 00 00 00 E1 C1 43 00 00 00 00 00 00	.ÁC.....áAC.....
001E1050	E1 C1 43 00 00 00 00 00 98 01 00 00 00 00 00 00 00	áAC.....~.....
001E1060	01 00 00 00 18 00 00 00 04 00 00 00 00 00 00 00 00(.p.~.....
001E1070	02 00 00 00 28 00 70 01 98 01 00 00 18 00 01 00#.....
001E1080	00 00 00 00 02 00 02 00 23 00 00 00 00 00 00 00 00	#.....FILE0.....
001E1090	23 00 04 00 00 00 00 00 46 49 4C 45 30 00 03 00	ÉAC.....8.....
001E10A0	C9 C1 43 00 00 00 00 00 04 00 01 00 38 00 01 00	p.....
001E10B0	70 01 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00`.....
001E10C0	03 00 00 00 8D 00 00 00 01 00 00 00 00 00 00 00 00	H.....%SS@HØ.....
001E10D0	10 00 00 00 60 00 00 00 00 00 00 00 00 00 00 00 00	%SS@HØ.....%SS@HØ.....
001E10E0	48 00 00 00 18 00 00 00 1C 89 A7 24 AE 48 D8 01	%SS@HØ.....%SS@HØ.....
001E10F0	1C 89 A7 24 AE 48 D8 01 1C 89 A7 24 AE 48 D8 01`.....
001E1100	1C 89 A7 24 AE 48 D8 01 20 00 00 00 00 00 00 00 00^.....
001E1110	00 00 00 00 00 00 00 00 00 00 00 00 00 09 01 00 00^.....
001E1120	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00^.....
001E1130	30 00 00 00 88 00 00 00 00 00 00 00 00 00 00 02 00^.....
001E1140	6C 00 00 00 18 00 01 00 8C 00 00 00 00 00 00 03 00^.....
001E1150	1C 89 A7 24 AE 48 D8 01 1C 89 A7 24 AE 48 D8 01^.....
001E1160	1C 89 A7 24 AE 48 D8 01 1C 89 A7 24 AE 48 D8 01^.....
001E1170	00 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00^.....
001E1180	20 00 00 00 00 00 00 00 15 00 F8 BB 55 D6 78 C7ø»UÖxç.....
001E1190	20 00 32 00 31 00 32 00 36 00 33 00 33 00 2E 00	.2.1.2.6.3.3....
001E11A0	63 00 72 00 64 00 6F 00 77 00 6E 00 6C 00 6F 00	c.r.d.o.w.n.l.o.....
001E11B0	61 00 64 00 00 00 00 00 80 00 00 00 48 00 00 00	a.d....€...H.....
001E11C0	01 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00 00@.....
001E11D0	04 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 00@.....
001E11E0	00 50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00@.....
001E11F0	00 00 00 00 00 00 00 31 05 B0 08 0E 00 D5 50l.º...ÕP.....
001E1200	FF FF FF FF 82 79 47 11 41 C2 43 00 00 00 00 00	VÝÝÝ, yg ÁAC.....

그림 8. LSN 4440584에 대한 data

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
This LSN								Previous LSN							
Client Undo LSN					Client Data Length				Client ID						
Record Type		Transaction ID			Flags	Alignment or Reserved									
Redo OP	Undo OP	Redo Offset	Redo Length		Undo Offset	Undo Length	Target Attribute	LCNs to follows							
Record Offset	Attr Offset	MFT Cluster Index	Alignment or Reserved		Target VCN		Alignment or Reserved								
Target LCN		Alignment or Reserved													

그림 9. 작업 레코드 구조

1

¹ <http://forensicinsight.org/wp-content/uploads/2013/06/F-INSIGHT-NTFS-Log-TrackerKorean.pdf>

그림 8에서 그림 9를 참고하면, 다음과 같이 정리를 할 수 있다.

Attribute	값
This LSN	0x43C208(4440584)
Previous LSN	0x43C1E1
Client Undo LSN	0x43C1E1
Client Data Length	0x198
Record Type	0x1
Redo OP	0x2
Target VCN	0x23

표 1. LSN 4440584에 대한 작업 레코드 정보

또한, \$MFT 구조를 참고하여 필요한 정보를 다음과 같이 정리할 수 있다.

Attribute	값
File record	0x46494C45(FILE)
\$LogFile Sequence Number	0x43C1C9
\$FILE_NAME에서의 file Name	(offset 0x1E118A ~ 0x1E11B2) 미확인212633.crdownload
Non-resident	(offset 0x1E11C0) 0x01
\$DATA에서의 RunList Data Offset	(offset 0x1E11FA ~ 0x1E11FC) 0xB0080E

여기서 RunList Data Offset인 0xB0080E 값은 10진수로 919728로, 해당 cluster는 앞서 Info5.docx와 같은지에 대한 여부를 살펴보는데 사용되었다.

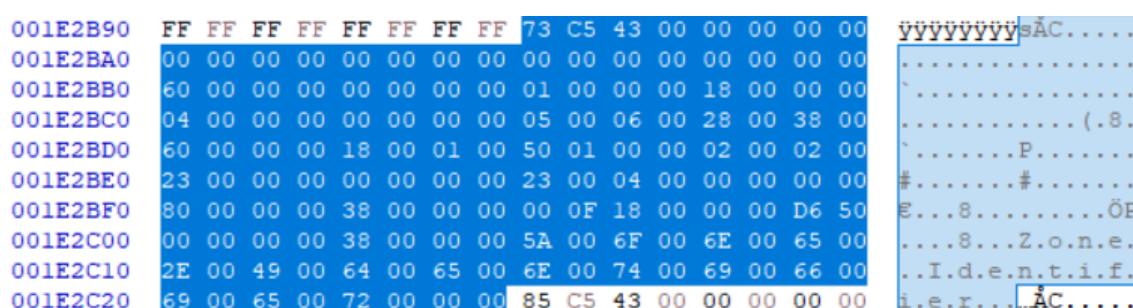


그림 10. Zone.Identifier 확인

웹에서 파일 다운로드 시 생성될 수 있는 Zone.Identifier의 LSN 값은 0x43C573(4441459)이다.

또한, 그림 11에서는 LSN 0x43C6EC(4441836)에 대한 정보를 확인할 수 있다. ZoneId값이 3이면 인터넷에서 파일을 다운로드했다고 볼 수 있으며, HostUrl를 살펴보면 Dropbox에서 다운받은 기록임을 확인할 수 있다.

591	4441275	2022-04-05 14:29:38	Renaming File	미확인 212633.crdownload -> Info5.docx	Info5.docx
592	4442863				Info5.docx

그림 12. LSN 4441275와 4442863

```
001E4480 00 00 00 00 00 00 00 00 EC C6 43 00 00 00 00 00 00 00 00  
001E4490 92 C8 43 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
001E44A0 00 00 00 00 00 00 00 00 00 00 C0 01 00 00 00 00 00 00 00  
001E44B0 01 00 00 00 18 00 00 00 00 02 00 00 00 00 00 00 00 00  
001E44C0 06 00 05 00 28 00 00 00 28 00 98 01 18 00 01 00  
001E44D0 50 01 00 00 02 00 02 00 23 00 00 00 00 00 00 00 00 00  
001E44E0 23 00 04 00 00 00 00 00 80 00 00 00 98 01 00 00  
001E44F0 00 0F 18 00 00 00 04 00 5B 01 00 00 38 00 00 00 00  
001E4500 5A 00 6F 00 6E 00 65 00 2E 00 49 00 64 00 65 00  
001E4510 6E 00 74 00 69 00 66 00 69 00 65 00 72 00 00 00  
001E4520 5B 5A 6F 6E 65 54 72 61 6E 73 66 65 72 5D 0D 0A  
001E4530 5A 6F 6E 65 49 64 3D 33 0D 0A 52 65 66 65 72 72  
001E4540 65 72 55 72 6C 3D 68 74 74 70 73 3A 2F 2F 77 77  
001E4550 77 2E 64 72 6F 70 62 6F 78 2E 63 6F 6D 2F 0D 0A  
001E4560 48 6F 73 74 55 72 6C 3D 68 74 74 70 73 3A 2F 2F  
001E4570 75 63 62 35 66 33 39 31 36 32 65 30 62 35 66 63  
001E4580 38 66 64 34 65 36 31 34 36 64 66 66 2E 64 6C 2E  
001E4590 64 72 6F 70 62 6F 78 75 73 65 72 63 6F 6E 74 65  
001E45A0 6E 74 2E 63 6F 6D 2F 63 64 2F 30 2F 67 65 74 2F  
001E45B0 42 6A 62 32 69 54 30 75 45 69 6B 39 43 6A 63 42  
001E45C0 53 36 76 7A 65 76 58 6B 39 56 57 69 57 6F 30 67  
001E45D0 6C 6A 71 6E 5F 39 75 5F 65 6B 46 44 35 64 74 70  
001E45E0 79 6A 5A 72 61 65 48 75 56 6A 4E 50 6E 54 5F 4B  
001E45F0 49 61 66 34 5F 4A 51 67 33 5A 41 47 6A 39 D8 50  
001E4600 6B 35 39 4F 35 75 32 44 36 4F 57 5A 64 67 53 41  
001E4610 42 43 69 6D 74 4B 54 50 42 32 70 48 78 4F 58 31  
001E4620 6D 58 6B 56 74 72 39 68 68 49 43 72 70 64 75 62  
001E4630 71 63 6F 67 78 32 68 76 58 58 53 49 62 68 6F 58  
001E4640 4C 52 4A 61 70 6F 62 6B 33 79 52 71 55 7A 50 39  
001E4650 30 43 41 67 7A 51 35 4B 6C 7A 51 2D 30 4B 50 75  
001E4660 6D 47 78 5A 2D 46 74 73 73 55 66 41 4C 30 59 54  
001E4670 67 6B 49 2F 66 69 6C 65 23 0D 0A 00 00 00 00 00
```

그림 11. Dropbox에서 다운로드 흔적 확인

LSN 4441275와 4442863 내에 위의 두 작업 레코드가 기록되어 있으므로, 해당 미확인 212633.crdownload는 dropbox에서 다운로드 되었다고 볼 수 있다.

2. Find completely deleted files in USB. (20 points)

USB에서 완전 삭제된 파일을 찾아야 한다.

614	4455267	2022-04-05 14:34:11	Renaming File	2_2_5_b.txt -> `7a}w377k[P	`7a}w377k[P
615	4455580	2022-04-05 14:34:11	Renaming File	`7a}w377k[P -> ZuTeo4zuR`q	ZuTeo4zuR`q
616	4455885	2022-04-05 14:34:11	Renaming File	ZuTeo4zuR`q -> iB(jRcfcrK=	iB(jRcfcrK=
617	4456200	2022-04-05 14:34:11	Renaming File	iB(jRcfcrK= -> 7pvuDpZgl_Y	7pvuDpZgl_Y
618	4456516	2022-04-05 14:34:11	Renaming File	7pvuDpZgl_Y -> S3vzt70Bi=3	S3vzt70Bi=3
619	4456821	2022-04-05 14:34:11	Renaming File	S3vzt70Bi=3 -> /}ykHG-bpa!	/}ykHG-bpa!
620	4457134	2022-04-05 14:34:11	Renaming File	/}ykHG-bpa! -> +w78CQ}--=D1	+w78CQ}--=D1
621	4457364	2022-04-05 14:34:11	File Deletion	Abnormal Timestamp (1601-01-01 09:00:00)	+w78CQ}--=D1
622	4458164	2022-04-05 14:34:11	Renaming File	2_2_5_a.txt -> 3bG[Jaf{iczA	3bG[Jaf{iczA
623	4459101	2022-04-05 14:34:11	Renaming File	3bG[Jaf{iczA -> VF+i1YD}ege	VF+i1YD}ege
624	4459721	2022-04-05 14:34:11	Renaming File	VF+i1YD}ege -> YX3XOAFk12Y	YX3XOAFk12Y
625	4460026	2022-04-05 14:34:11	Renaming File	YX3XOAFk12Y -> UpPsjG97iv!	UpPsjG97iv!
626	4460242	2022-04-05 14:34:11	File Deletion	Abnormal Timestamp (1601-01-01 09:00:00)	UpPsjG97iv!

그림 13. 파일 삭제 흔적

그림 13은 Eraser를 통해 파일을 완전 삭제했을 때 log가 남는 흔적이다. 이에 대한 이유는 관련 논문 자료[1]와 작년 디지털 포렌식 챌린지 201번 문제를 참고해볼 수 있다. Eraser 프로그램을 통해 완전 삭제 시 해당 파일명의 길이는 원본 파일명 길이와 동일하고 파일명 자체는 숫자, 문자, 특수문자 조합으로 변경된다. 또한, 마지막에 파일 삭제가 일어날 때는 비정상적인 파일 시간 정보를 가지게 되기 때문이다.

또한, 3번과 4번 문제에서 첫번째 파일과 두번째 파일의 삭제 파일 내용을 묻는 것으로 보아, 2_2_5_a.txt 와 2_2_5_b.txt 라는 파일이 완전 삭제되었다는 판단을 확실히 내렸다.

완전 삭제된 파일	2_2_5_a.txt, 2_2_5_b.txt
-----------	--------------------------

3.What is the content of the first file that was completely deleted? (Hint: Person) (50 points)

완전 삭제된 첫번째 파일의 내용을 묻는 문제이다. 2_2_5_a.txt를 먼저 살펴보았다.

593	4444553	2022-04-05 14:31:32	File Creation		2_2_5_a.txt
594	4444636	2022-04-05 14:31:32	File Deletion		2_2_5_a.txt
595	4444792	2022-04-05 14:31:32	File Creation		2_2_5_a.txt
596	4444988		Writing Content of Non-Resident File	Data Runs(in Volume) : 919726(1)	2_2_5_a.txt
597	4445891		Writing Content of Non-Resident File	Data Runs(in Volume) : 919733(1)	2_2_5_a.txt
598	4446201		Writing Content of Non-Resident File	Data Runs(in Volume) : 919734(1)	2_2_5_a.txt
599	4446519		Writing Content of Non-Resident File	Data Runs(in Volume) : 919735(1)	2_2_5_a.txt
600	4447422		Writing Content of Non-Resident File	Data Runs(in Volume) : 919726(1)	2_2_5_a.txt
601	4448143		Writing Content of Non-Resident File	Data Runs(in Volume) : 919733(1)	2_2_5_a.txt

그림 14. 2_2_5_a.txt에 대한 \$logfile metadata

해당 파일은 앞서 살펴본 다운로드 기록처럼 Cluster Number가 Detail 정보에 나와있다. 해당 파일은 Non-Resident 특성으로 내용이 기록되었다.

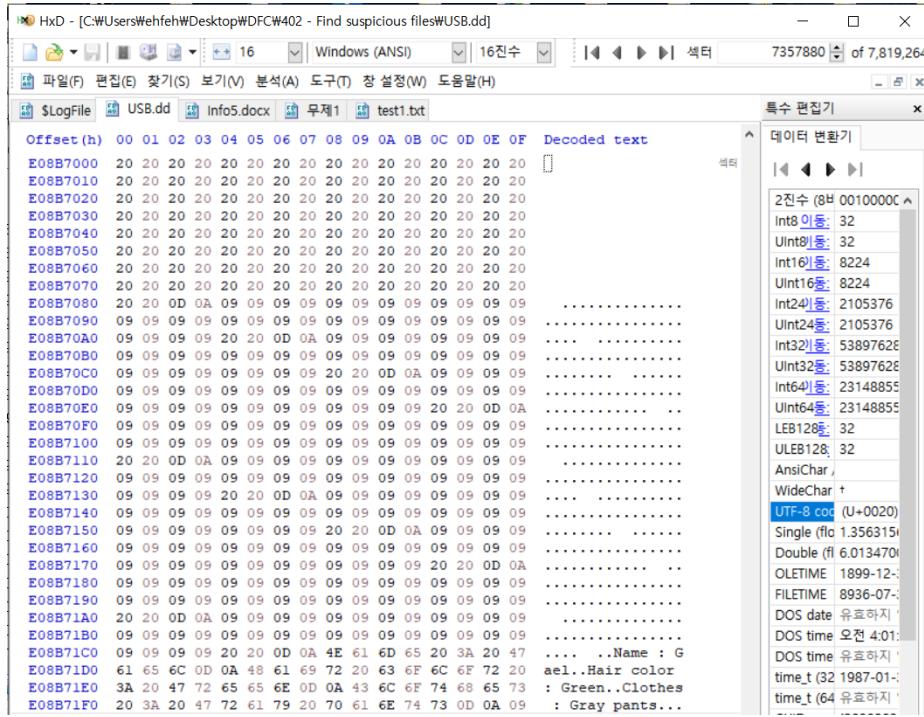


그림 15. cluster num 919735에 대한 sector 정보

해당 Data Runs 정보에서 919735 Cluster Number에 대해 8을 곱한 섹터 7357880을 살펴보았다.

```

E08B71C0 09 09 09 09 20 20 0D 0A 4E 61 6D 65 20 3A 20 47    .... .Name : G
E08B71D0 61 65 6C 0D 0A 48 61 69 72 20 63 6F 6C 6F 72 20  ael..Hair color
E08B71E0 3A 20 47 72 65 65 6E 0D 0A 43 6C 6F 74 68 65 73 : Green..Clothes
E08B71F0 20 3A 20 47 72 61 79 20 70 61 6E 74 73 0D 0A 09 : Gray pants...

```

그림 16. 2_2_5_a.txt에서 특정할 수 있는 정보

2_2_5_a.txt에서는 3번 문제의 힌트가 Person이라는 점을 감안할 때 사람에 대한 이름과 인상 착의 정보가 기록되어 있다. 또한, 2_2_5_a.txt 파일의 내용을 기록하는 시점의 작업 레코드를 통해 탐색한 정보이므로 해당 정보가 완전 삭제된 파일의 내용이라고 판단할 수 있다.

001EC9B0	00 00 00 00 00 00 00 00 37 D9 43 00 00 00 00 00 00	7ÙC.....
001EC9C0	26 D9 43 00 00 00 00 00 26 D9 43 00 00 00 00 00 00	ÙC.....ÙC.....
001EC9D0	38 00 00 00 00 00 00 00 01 00 00 00 18 00 00 00 00	8.....
001EC9E0	00 00 00 00 00 00 00 00 09 00 09 00 28 00 08 00(...
001EC9F0	30 00 08 00 18 00 01 00 08 01 40 00 00 00 00 E0 50	0.....@...àP
001ECA00	24 00 00 00 00 00 00 00 24 00 04 00 00 00 00 00 00	S.....S.....
001ECA10	31 01 B7 08 0E 00 00 00 00 00 00 00 00 00 00 00 00	1.....
001ECA20	44 D9 43 00 00 00 00 00 37 D9 43 00 00 00 00 00 00	DÙC.....7ÙC.....
001ECA30

그림 17. \$LogFile에서 해당 LSN에 대한 작업 레코드 정보

또한, \$LogFile에서도 Cluster Num 919735에 기록할 때의 LSN인 4446519(0x43D937)을 검색해보면 그림 17과 같은 정보가 나온다. Offset 0x001ECA10부터 0x001ECA14까지를 살펴보면, 31 01 B7 08 0E를 나타낸다. 해당 hex값은 Run List구조로 판단하였다.

Non-Resident 속성은 실제 데이터를 속성 내용이 위치한 곳에 Run List 구조를 사용하여 데이터가 저장되어 있는 클러스터의 오프셋 및 클러스터 개수를 기록한다.

따라서, 0xE08B7은 919735로 이 값은 Cluster offset을 가리키는 것을 확인할 수 있다.

599	4446519	Writing Content of Non-Resident File	Data Runs(in Volume) : 919735(1)	2_2_5_a.txt	Update Mapping Pairs
600	4447422	Writing Content of Non-Resident File	Data Runs(in Volume) : 919726(1)	2_2_5_a.txt	Update Mapping Pairs
601	4448143	Writing Content of Non-Resident File	Data Runs(in Volume) : 919733(1)	2_2_5_a.txt	Update Mapping Pairs

그림 18. 2_2_5_a.txt에 대한 Redo/Undo 정보 확인

또한, 그림 17과 그림 18을 보면 해당 작업 레코드에서 Redo, Undo opcode값은 0x09로 NTFS Log Tracker에서도 보여지는 UpdateMappingPairs로 같다.

첫번째 삭제된 파일의 내용은 다음과 같다.

첫번째 삭제된 파일	내용
2_2_5_a.txt	Name : Gael Hair color : Green Clothes : Gray pants

4.What is the content of the second file that was completely deleted? (Hint: Appointment) (80 points)

완전 삭제된 두번째 파일의 내용을 묻는 문제이다.

602	4449501	2022-04-05 14:32:16	File Creation		2_2_5_b.txt
603	4449584	2022-04-05 14:32:16	File Deletion		2_2_5_b.txt
604	4449732	2022-04-05 14:32:16	File Creation		2_2_5_b.txt
605	4449800		Writing Content of Resident File	Writing Size : 69	2_2_5_b.txt
606	4450760		Writing Content of Resident File	Writing Size : 67	2_2_5_b.txt
607	4451571		Writing Content of Resident File	Writing Size : 69	2_2_5_b.txt

그림 19. 2_2_5_b.txt에 대한 metadata

2_2_5_b.txt는 Resident 특성으로 내용이 기록되었음을 알 수 있다. 해당 이벤트에 대한 LSN들을 따라가보면,

그림 20. LSN 4449800에 대한 작업 레코드

	HEX	DEC	OCT	계산기	프로그래머	4,450,760
001F4E40	C8 E9 43 00 00 00 00 00 00 00 00 00 00 00 00 00(.				
001F4E50	00 00 00 00 00 00 00 00 28 00 00 00 00 00 00 00(.C.(.				
001F4E60	01 00 00 00 18 00 00 00 06 00 00 00 00 00 00 00	0.....\$.				
001F4E70	07 00 07 00 28 00 43 00 28 00 00 00 18 00 01 00	\$.....ÓéC..				
001F4E80	30 01 18 00 02 00 02 00 24 00 00 00 00 00 00 00	ÓéC.....				
001F4E90	24 00 04 00 00 00 00 00 D3 E9 43 00 00 00 00 00	(.....(.				
001F4EA0	C8 E9 43 00 00 00 00 00 00 00 00 00 00 00 00 00(.				
001F4EB0	28 00 00 00 00 00 00 00 01 00 00 00 18 00 00 00(.				
001F4EC0	02 00 00 00 00 00 00 00 1B 00 01 00 28 00 00 00(.				
001F4ED0	28 00 00 00 18 00 00 00 00 00 00 00 00 00 02 00(.				
001F4EE0	00 00 00 00 00 00 00 00 FF FF FF FF FF FF FF FFVVVVVVVV				
001F4EF0	DE E9 43 00 00 00 00 00 00 00 00 00 00 00 00 00	ÓéC.....				
001F4FO0	00 00 00 00 00 00 00 00 A8 00 00 00 00 00 00 00(.				

그림 21. LSN 4450760에 대한 작업 레코드

HEX	DEC	OCT	BIN
FF FF FF FF FF FF F3 EC 43 00 00 00 00 00 00	4,451,571	20 766 363	0100 0011 1110 1100 1111 0011
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			
28 00 00 00 00 00 00 00 00 01 00 00 18 00 00 00			
06 00 00 00 00 00 00 00 07 00 07 00 28 00 45 00			
28 00 00 00 18 00 01 00 30 01 18 00 02 00 02 00			
24 00 00 00 00 00 00 00 24 00 04 00 00 00 00 00			
FE EC 43 00 00 00 00 00 F3 EC 43 00 00 00 EA 50			
00 00 00 00 00 00 00 00 28 00 00 00 00 00 00 00			
01 00 00 00 18 00 00 00 02 00 00 00 00 00 00 00			
1B 00 01 00 28 00 00 00 28 00 00 00 18 00 00 00			
00 00 00 00 00 00 02 00 00 00 00 00 00 00 00 00			
FF FF FF FF FF FF F9 ED 43 00 00 00 00 00 00			
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00			

그림 22 ISBN 4451571에 대한 작업 레코드

세 작업 레코드에서 3번 문제처럼 LSN tracing으로 직접적인 데이터는 얻지 못했지만, 강조된 부분을 살펴보면 Resident File 데이터 Update가 이루어진 사실을 알 수 있다.

관련 자료²를 참고해보면, Redo 작업이 opcode 0x07로 Update Resident Value이고 Record Offset

² <http://forensicinsight.org/wp-content/uploads/2013/06/F-INSIGHT-NTFS-Log-TrackerKorean.pdf>

이 0xF8이상, 그리고 Attr Offset이 0x18이상이면 \$DATA 속성에 대한 업데이트 작업이라고 볼 수 있다. 하지만, Win7부터 적용이 되지 않아 Redo, Undo에 대한 데이터를 확인할 수는 없어서 LSN과 함께 다음 작업 레코드가 오는 것을 확인할 수 있다.

하지만, 앞서 살펴본 LSN 4449800과 LSN 4451571 사이에서 4번 문제 힌트인 appointment와 관련된 정보가 두 번 발견된다.

001F4A70	4E E9 43 00 00 00 00 00 00 43 E9 43 00 00 00 00 00 00	NÉC.....CéC.
001F4A80	43 E9 43 00 00 00 00 00 00 88 00 00 00 00 00 00 00 00	CéC.....^.....
001F4A90	01 00 00 00 18 00 00 00 00 02 00 00 00 00 00 00 00 00
001F4AA0	06 00 05 00 28 00 00 00 28 00 60 00 18 00 01 00 00 00(....(.`
001F4AB0	30 01 00 00 02 00 02 00 24 00 00 00 00 00 00 00 00 00	0.....\$.....
001F4AC0	24 00 04 00 00 00 00 00 80 00 00 00 60 00 00 00 00 00	\$.....€.....
001F4AD0	00 00 18 00 00 00 01 00 45 00 00 00 18 00 00 00 00 00E.....
001F4AE0	44 61 74 65 20 26 20 54 69 6D 65 20 3A 20 32 30	Date & Time : 20
001F4AF0	32 32 2F 30 36 2F 31 31 20 31 33 3A 30 30 20 7E	22/06/11 13:00 ~
001F4B00	20 31 33 3A 33 30 0D 0A 4C 6F 63 61 74 69 6F 6E	13:30..Location
001F4B10	20 3A 20 31 33 20 73 6F 75 74 68 20 31 31 72 64	: 13 south 11rd
001F4B20	20 53 74 0D 0A 00 00 00 65 E9 43 00 00 00 00 00 00	St.....eéC.....

그림 23. LSN 0x4EE943(4450638)에 대한 작업 레코드

먼저 LSN 4450638은 그림 19에서 Writing Content of Resident File 이벤트의 첫번째 LSN(4449800)과 두번째 LSN(4450760) 사이에서 발견된다.

001F63C0	FF FF FF FF FF FF FF FF	79 EC 43 00 00 00 00 00 00	yyyyyyyy	viC
001F63D0	6E EC 43 00 00 00 00 00 00	6E EC 43 00 00 00 00 00 00	niC.....niC	
001F63E0	88 00 00 00 00 00 00 00 01	00 00 00 00 18 00 00 00 00	^.....	
001F63F0	02 00 00 00 00 00 00 00 06	00 05 00 28 00 EA 50 (.êP	
001F6400	28 00 60 00 18 00 01 00 30	01 00 00 02 00 02 00 00 00	(. `..... 0	
001F6410	24 00 00 00 00 00 00 00 24	00 04 00 00 00 00 00 00 00	\$..... \$	
001F6420	80 00 00 00 60 00 00 00 00	00 00 18 00 00 00 04 00	E....`	
001F6430	43 00 00 00 18 00 00 00 44	61 74 65 20 26 20 54	C.....Date & T	
001F6440	69 6D 65 20 3A 20 32 30 32	32 32 2F 30 36 2F 31 31	ime : 2022/06/11	
001F6450	20 31 33 3A 30 30 20 7E 20	31 33 3A 33 30 0D 0A	13:00 ~ 13:30..	
001F6460	4C 6F 63 61 74 69 6F 6E 20	3A 20 31 33 20 73 6F	Location : 13 so	
001F6470	75 74 68 20 31 31 72 64 20	53 74 00 00 00 00 00 00	uth llrd St.....	

그림 24. LSN 0x43EC79(4451449)에 대한 작업 레코드

두번째 LSN 4451449는 그림 19에서 Writing Content of Resident File 이벤트의 두번째 LSN(4450760)과 세번째 LSN(4451571)사이에서 발견된다.

두 작업 레코드 모두 \$DATA + 0x8에서 값이 00이므로 resident file임을 알 수 있고, 그에 따라 앞서 살펴본 Non-resident file인 2_2_5_a.txt처럼 RunList 구조로 Cluster Offset이 오는 것이 아니라,

\$DATA + 0x18부분에서 데이터가 appointment에 관한 정보라는 것을 바로 확인할 수 있다.

FF FF FF FF FF FF FF	79 EC 43 00 00 00 00 00 00 00 00 00 00 00 00 00	yyyyyyyyyyiC..
6E EC 43 00 00 00 00 00 00 00 00 00 00 00 00 00	6E EC 43 00 00 00 00 00 00 00 00 00 00 00 00 00	niC.....niC..
88 00 00 00 00 00 00 00 00 01 00 00 00 00 18 00 00 00	02 00 00 00 00 00 00 00 00 06 00 05 00 28 00 EA 50	^.....(..êP
28 00 60 00 18 00 01 00 30 01 00 00 02 00 02 00	28 00 60 00 18 00 01 00 30 01 00 00 02 00 02 00	(..`....0.....
24 00 00 00 00 00 00 00 24 00 04 00 00 00 00 00 00 00	80 00 00 00 60 00 00 00 00 00 18 00 00 00 00 04 00	\$.....\$.....
43 00 00 00 18 00 00 00 44 61 74 65 20 26 20 54	43 00 00 00 18 00 00 00 44 61 74 65 20 26 20 54	é.....`.....
69 6D 65 20 3A 20 32 30 32 32 2F 30 36 2F 31 31	20 31 33 3A 30 30 20 7E 20 31 33 3A 33 30 0D 0A	C.....Date & T
20 31 33 3A 30 30 20 7E 20 31 33 3A 33 30 0D 0A	4C 6F 63 61 74 69 6F 6E 20 3A 20 31 33 20 73 6F	ime : 2022/06/11
4C 6F 63 61 74 69 6F 6E 20 3A 20 31 33 20 73 6F	75 74 68 20 31 31 72 64 20 53 74 00 00 00 00 00 00	13:00 ~ 13:30..
75 74 68 20 31 31 72 64 20 53 74 00 00 00 00 00 00 00	90 EC 43 00 00 00 00 00 79 EC 43 00 00 00 00 00 00 00	Location : 13 so
90 EC 43 00 00 00 00 00 79 EC 43 00 00 00 00 00 00 00 00	90 EC 43 00 00 00 00 00 90 EC 43 00 00 00 00 00 00 00 00	uth 11rd St.....
90 EC 43 00 00 00 00 00 90 EC 43 00 00 00 00 00 00 00 00 00	40 00 00 00 00 00 00 00 01 00 00 00 18 00 00 00 00 00	.iC.....yìC..
40 00 00 00 00 00 00 00 01 00 00 00 18 00 00 00 00 00 00	04 00 00 00 00 00 00 00 05 00 06 00 28 00 18 00	yìC.....(.....
04 00 00 00 00 00 00 00 05 00 06 00 28 00 18 00	40 00 00 00 18 00 01 00 30 01 00 00 02 00 02 00	%...(.È.(.....
40 00 00 00 18 00 01 00 30 01 00 00 02 00 02 00	24 00 00 00 00 00 00 00 24 00 04 00 00 00 00 00 00 00	8.....\$.....
24 00 00 00 00 00 00 00 24 00 04 00 00 00 00 00 00 00 00	80 00 00 00 18 00 00 00 00 00 18 00 00 00 00 05 00	\$.....>iC..
80 00 00 00 18 00 00 00 00 00 18 00 00 00 00 00 05 00	00 00 00 00 18 00 00 00 A9 EC 43 00 00 00 00 00 00 00	.iC.....iC..
00 00 00 00 18 00 00 00 A9 EC 43 00 00 00 00 00 00 00 00		@.....@.....
		\$.....\$.....
		é.....é.....
		©iC.....

그림 25. 이후 작업 레코드 확인

두 작업 레코드 모두 redo opcode가 0x06이고 undo가 0x05이고 이후 작업 레코드에서는 redo opcode가 0x05, undo opcode가 0x06으로 바뀌면서 \$DATA +0x18 부분에 적힌 해당 appointment 관련 정보를 읽을 수 있게된다.

이는 앞서 살펴본 관련 자료에서 \$FILE_NAME일때의 파일명 변경 작업을 참고하면, 해당 상황에서는 \$DATA에 저장되어 있던 정보가 없어지는 변경 작업이 이루어졌다고 판단했다. 그리고, 작업 전에는 데이터를 생성하려는 CreateAttribute고 작업 후에는 데이터를 삭제하려는 DeleteAttribute를 가리키는 것을 보아, 삭제 작업을 진행하려는 데이터는 동일하다.

또한, redo opcode 0x25는 ZeroEndOfFileRecord를 의미하는데, 이는 resident data에 대한 업데이트로 볼 수 있다.

따라서, 그림 23과 그림 24에서의 작업 레코드를 보고 삭제 전에는 해당 데이터가 쓰여있음을 확인할 수 있고, 그림 20, 21, 22에서는 쓰여 있던 데이터에 대한 파일 Update가 일어났음을 짐작해볼 수 있

다.

NTFS Log Tracker 1.7.1 버전에서는 파싱되지 않는 LSN들을 자세히 보고 교차 검증하기 위해, dfir_ntfs open source tool을 이용하여 \$MFT와 \$LogFile을 대상으로 파싱 결과를 분석했다.

LSN: 4449708

Transaction ID: 24

Log record, redo operation: AddIndexEntryRoot, undo operation: DeleteIndexEntryRoot

Target (file number): 140

Target path (from \$MFT, likely wrong if the file was deleted later): /Area2/Job2

Offset in target: 464

LCN(s): 262179

Redo data:

00000000 91 00 00 00 00 00 02 00-68 00 58 00 00 00 00 00h.X....

00000010 8C 00 00 00 00 00 03 00-C4 0C CF 82 AE 48 D8 01H..

00000020 C4 0C CF 82 AE 48 D8 01-C4 0C CF 82 AE 48 D8 01H.....H..

00000030 C4 0C CF 82 AE 48 D8 01-00 00 00 00 00 00 00 00H.....

00000040 00 00 00 00 00 00 00 00-20 00 00 00 00 00 00 00

00000050 0B 00 32 00 5F 00 32 00-5F 00 35 00 5F 00 62 00 ..2._2._5._b.

00000060 2E 00 74 00 78 00 74 00t.x.t.

Undo data:

-

그림 26. Resident file 생성 작업1

LSN: 4449732
 Transaction ID: 24
 Log record, redo operation: InitializeFileRecordSegment, undo operation: Noop
 Target (file number): 145
 Offset in target: 0
 LCN(s): 262180
 Redo data:
 00000000 46 49 4C 45 30 00 03 00-A0 E5 43 00 00 00 00 00 FILE0....C....
 00000010 02 00 01 00 38 00 01 00-28 01 00 00 00 04 00 008...(.....
 00000020 00 00 00 00 00 00 00 00-03 00 00 00 91 00 00 00` ...
 00000030 01 00 00 00 00 00 00 00-10 00 00 00 60 00 00 00` ...
 00000040 00 00 00 00 00 00 00 00-48 00 00 00 18 00 00 00H.....
 00000050 C4 0C CF 82 AE 48 D8 01-C4 0C CF 82 AE 48 D8 01H.....H..
 00000060 C4 0C CF 82 AE 48 D8 01-C4 0C CF 82 AE 48 D8 01H.....H..
 00000070 20 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
 00000080 00 00 00 00 09 01 00 00-00 00 00 00 00 00 00 00
 00000090 00 00 00 00 00 00 00 00-30 00 00 00 70 00 00 000..p...
 000000A0 00 00 00 00 00 00 02 00-58 00 00 00 18 00 01 00X.....
 000000B0 8C 00 00 00 00 03 00-C4 0C CF 82 AE 48 D8 01H..
 000000C0 C4 0C CF 82 AE 48 D8 01-C4 0C CF 82 AE 48 D8 01H.....H..
 000000D0 C4 0C CF 82 AE 48 D8 01-00 00 00 00 00 00 00 00H.....
 000000E0 00 00 00 00 00 00 00 00-20 00 00 00 00 00 00 00
 000000F0 0B 00 32 00 5F 00 32 00-5F 00 35 00 5F 00 62 00 ..2._2._5._b.
 00000100 2E 00 74 00 78 00 74 00-80 00 00 00 18 00 00 00 ..t.x.t.....
 00000110 00 00 18 00 00 00 01 00-00 00 00 00 18 00 00 00
 00000120 FF FF FF FF 82 79 47 11VG.

그림27. Resident file 생성 작업2

또한, 그림 26과 그림27을 통해 AddIndexEntryRoot/DeleteIndexEntryRoot에서 InitializeFileRecordSegment/Noop로에 대한 작업은 resident file 생성 작업임을 아까 언급했던 각주2에서 알 수 있다. 여기서 LCN(Logical Cluster Number)는 262180이라는 것도 확인할 수 있다.

LSN: 4450638
Transaction ID: 24
Log record, redo operation: DeleteAttribute, undo operation: CreateAttribute
Target (file number): 145
Offset in target: 304
LCN(s): 262180
Redo data:
-

Undo data:
00000000 80 00 00 00 60 00 00 00-00 00 18 00 00 00 01 00
00000010 45 00 00 00 18 00 00 00-44 61 74 65 20 26 20 54 E.....Date & T
00000020 69 6D 65 20 3A 20 32 30-32 32 2F 30 36 2F 31 31 im : 2022/06/11
00000030 20 31 33 3A 30 30 20 7E-20 31 33 3A 33 30 0D 0A 13:00 ~ 13:30..
00000040 4C 6F 63 61 74 69 6F 6E-20 3A 20 31 33 20 73 6F Location : 13 so
00000050 75 74 68 20 31 31 72 64-20 53 74 0D 0A 00 00 00 uth 11rd St....

그림28. 2_2_5_b.txt에 대한 내용1

LSN: 4451449
Transaction ID: 24
Log record, redo operation: DeleteAttribute, undo operation: CreateAttribute
Target (file number): 145
Offset in target: 304
LCN(s): 262180
Redo data:
-

Undo data:
00000000 80 00 00 00 60 00 00 00-00 00 18 00 00 00 04 00
00000010 43 00 00 00 18 00 00 00-44 61 74 65 20 26 20 54 C.....Date & T
00000020 69 6D 65 20 3A 20 32 30-32 32 2F 30 36 2F 31 31 im : 2022/06/11
00000030 20 31 33 3A 30 30 20 7E-20 31 33 3A 33 30 0D 0A 13:00 ~ 13:30..
00000040 4C 6F 63 61 74 69 6F 6E-20 3A 20 31 33 20 73 6F Location : 13 so
00000050 75 74 68 20 31 31 72 64-20 53 74 00 00 00 00 00 uth 11rd St....

그림29. 2_2_5_b.txt에 대한 내용2

그림 28과 그림 29에서 해당 내용들은 앞서 그림 27에서 살펴본 2_2_5_b.txt의 Logical Cluster Number 262180과 동일하다는 것을 알 수 있다. 삭제 작업에 대한 기록이 남아 있어 해당 기록을 통해 파일의 내용으로 판단하였다.

따라서, 해당 4번 문제의 답은 다음과 같다.

두번째 삭제된 파일	내용
2_2_5_b.txt	Date & Time : 2022/06/11 13:00 ~ 13:30 Location : 13 south 11rd St

5. Implement and submit a tool to track the history of changes to file data. (The tool should be implemented to solve problems #1, #3, and #4) (200 points)

● 실행 알고리즘

Webbrowser File Download Discovery	1. NTFS Log Tracker v1.7에서 추출된 DB 읽기 (LogFile TABLE) 2. [Detail] 컬럼에서 ".crdownload" 발견하면, 첫 LSN 값 저장 3. 다음 열부터 [Redoinfo]가 "Update Resident Value" 발견하면, 마지막 LSN 값 저장 4. USB.dd에서 추출한 \$LogFile 읽기 5. \$LogFile에서 [ZoneTransfer] 문자열 발견 6. ZoneTransfer 가 저장된 LSN 이 startLSN 과 lastLSN 의 사이에 존재한다면, 웹브라우저로 다운로드한 파일(.crdownload)의 ZONEID 임을 증명
ERASER DISCOVERY	1. NTFS Log Tracker v1.7 에서 추출된 DB 읽기 (LogFile TABLE) 2. [Event] 컬럼에서 "File Deletion" 발견 3. [MFT_ModifiedTime] 컬럼 값이 [CreateTime]보다 빠를 경우 4. 동일한 [filename]의 Logfile 중에서 [Event]가 "Renaming File" 발견하면, 완전 삭제(Eraser) 흔적 증명
FILE RECOVERY	1. USB.dd 를 읽어오기 2. NTFS Log Tracker v1.7 에서 추출된 DB 읽기 (LogFile TABLE) 3. 삭제 파일의 [Event] 컬럼이 "Non-Resident"일 경우 4. 삭제 파일의 [Detail] 컬럼에서 파일 데이터가 저장된 클러스터 offsets 을 획득 5. 획득한 클러스터 offsets 으로 이동하여 넉넉하게 파일 데이터를

출력하여, 사용자의 판단으로 파일 데이터를 확인

● 실행 결과

```
<<=====WEBBROWSER FILE DOWNLOAD DISCOVERY=====*>
[*] ZoneId=3
[*] ReferrerUrl=https://www.dropbox.com/
[*] HostUrl=https://ucb5f39162e0b5fc8fd4e6146dff.dl.dropboxusercontent.com/cd/0/get/Bjb2iT0uEik9CjcBS6vzevXk9VWiWo0gljqn_9u_lekFD5dtpyjZraeHuVjNPnT_KIaf4_JQg3ZAGj9\xd8Pk5905u2D60WZdgSABCimtKTPB2pHx0X1mXkVtr9hhICrpdbqcogx2hvXXSIBhoXLRJapobk3yRqUzP90CAgZQ5K1zQ-0KPumGxZ-FtssUfAL0YTgkI/file
[---]

<<=====ERASER DISCOVERY=====*>
[*] filename : 2_2_5_b.txt
[*] detail
[-->] LSN      EventTime      Event      Detail     FullPath
[-->] 4455267  2022-04-05 14:34:11 Renaming File  2_2_5_b.txt -> `7a}w377k[P      \Area2\Job2\`7a}w377k[P
[-->] 4455580  2022-04-05 14:34:11 Renaming File  `7a}w377k[P -> ZuTeo4zuR`q      \Area2\Job2\ZuTeo4zuR`q
[-->] 4455885  2022-04-05 14:34:11 Renaming File  ZuTeo4zuR`q -> iB(jRcfcrK=      \Area2\Job2\iB(jRcfcrK=
[-->] 4456200  2022-04-05 14:34:11 Renaming File  iB(jRcfcrK= -> 7pvuDpZg!_Y      \Area2\Job2\7pvuDpZg!_Y
[-->] 4456516  2022-04-05 14:34:11 Renaming File  7pvuDpZg!_Y -> S3vzt70Bi=3      \Area2\Job2\S3vzt70Bi=3
[-->] 4456821  2022-04-05 14:34:11 Renaming File  S3vzt70Bi=3 -> /}ykHG-bpa!      \Area2\Job2\}/}ykHG-bpa!
[-->] 4457134  2022-04-05 14:34:11 Renaming File  /}ykHG-bpa! -> +w78CQ}=-D1      \Area2\Job2\+w78CQ}=-D1
[-->] 4209520  2022-03-15 16:23:03 File Deletion      \Area8\Job1\8_1_a.txt
[---]

[*] filename : 2_2_5_a.txt
[*] detail
[-->] LSN      EventTime      Event      Detail     FullPath
[-->] 4458164  2022-04-05 14:34:11 Renaming File  2_2_5_a.txt -> 3bG[Ja{iczA      \Area2\Job2\3bG[Ja{iczA
[-->] 4459101  2022-04-05 14:34:11 Renaming File  3bG[Ja{iczA -> VF+i1YD}ege      \Area2\Job2\VF+i1YD}ege
[-->] 4459721  2022-04-05 14:34:11 Renaming File  VF+i1YD}ege -> YX3XOAFk12Y      \Area2\Job2\YX3XOAFk12Y
[-->] 4460026  2022-04-05 14:34:11 Renaming File  YX3XOAFk12Y -> UpPsjG97iv!      \Area2\Job2\UpPsjG97iv!
[-->] 4209520  2022-03-15 16:23:03 File Deletion      \Area8\Job1\8_1_a.txt
[---]

<<=====FILE RECOVERY=====*>
[*] filename : 2_2_5_b.txt
[---]
[*] filename : 2_2_5_a.txt
[*] File Candidate by Cluster offset 919726
b'EZC\x02\x04\x03\x08EasyCrypt 2.4 By Copynull@nate.comE0B4EA5DD471A4C34AX\x0c\x9aw@\xb39\x00G\xbcM\x03\x0e\xd7i\xbey\xdc\xcdE\xcdk\x8a\xd5u\xf2\xaa\xb3\xe1\xaa\xc6t(\x0b\x89\xd6\xaa\xcf\x84|\xf9"(\xe5\xf1".\x90m\x08\x99\xb5;\xf0\x80\x9a^@\xeu\xd9\xaa\xe9\x92u3\xc6\xc50X\x9F\x95\x82u\x9c\xaa2TH\xeb\x9a\xaa1\x1f\x90f2\x9e\xb0\xd1\xd0\xb4j\xeaR\xb2\x9d\xdeBpvA\xad\xcb\x9a\xf6\xff@\x98\xaa"\x04\x90F\x92\x86N.A(\xa4D\xd7\xf4H\x16t\xb38\x8a\xaa2\x17\x15\xb3\xaa3\x9er\x06y\xdd\x15: c\x897\x1c\x1e\x03\xee\x1bj\xb9\xb5\xe80\xf8\x0e;Q\x9a\xc8[\x22\x15\xd5\xcc\xad\x9efIh\x1c\xee3\xaa\xe7d\xf7h\xfb\xb8"\xb3\xee"\x2\xdf\xf6\xaa\xfe\t]:\xea\xf8c\x19_\x07p\xbc\x95\xed\xaa\xeaB\xdb\n7i+\xe71Y\xde\xaa\xdcS\xf8\xb1\xb6\xf9\x968n\x19\xab\x1c@_\x00\x92\xcf\xb1\xb4\x12\xc9\xbd0\x91\xc6\xc3\xbf;\xedj\xd5\xc4\xb5,\x2\xfe\x14\x0e\x13h&q\xc1\xb2w\x14\xb2?\xdc7\x8a\xafK\x8b\xd6\x8eRn\x7f\xe1\xad\|u\xc5m\xfe\xcc\x97\xf3,\xbc\xd9\xc07i40\xaeFH\xfc\x7f\xe7d\xadvJ#\xbd\x17\x82\x84q\x01\xef\x18\xf9y\x16\x86\xb4=\xa1\x10\x1b7\xcb\x0f\xe70\xe8w\xd1\x88\x1cmd\xcb\t\xc3\x12u\x8e?\x9e9'\x17\xb6\xcc\x8c\xfc\d\xe13\xedH"\x01,\xd1I\xf8\xd5\xaa\x1e\x85\xb6p\xbb\xb8\xe60\xf0k\xad(\xbdc7p\x01m9\|\xf1\x2}\x7\xd4\xed8\xf2\xb1;\xfe\xc9\x8d\xd4\xf9w\x94\xaf\xe7\xf1Re\xe6\x9c\x15\x9bZf\x90'\x9d\x07\xea\xcaLT~\x0f\xac\xbd\xee\xb7\xea\xecUS\x14\xe7\x16\xce\xfb\xb5 0\xca\x8aG\xe6:\x03t\x93y\xaa\xd8L\x8b\x08:\xdb\x93\'\xbff\xb3\x940\xd2\xd8\x96\x94\xfa5\xdb\xadQ\xb8\x07\x1a\xd5\xbd3d\taHc'
```

● 실행 코드

```
import sqlite3
import re
import binascii

con = sqlite3.connect('402_2022-08-01_00-50-29.db')
cursor = con.cursor()

cursor.execute('SELECT * FROM Logfile')
data= cursor.fetchall()
```

```

#LSN EventTime Event Detail FileNameFullPath CreateTime ModifiedTime
MFT_ModifiedTime AccessTime RedoInfo TargetVCN MFT_ClusterIndex
# 0      1      2      3      4      5      6      7
8          9     10     11     12

#[1] dropbox 흔적 발견
print("\n\n<=====WEBBROWSER FILE DOWNLOAD
DISCOVERY=====>\n")
startLSN = None
lastLSN = None
Found = False

# NTFS Log Tracker v1.7에서 추출된 DB-Logfile TABLE
for row in data :
    Detail = row[3]

    # [Detail] 컬럼에서 ".crdownload" 발견하면, 첫 LSN(startLSN) 값 저장
    if ".crdownload" in Detail :
        Found = True
        startLSN = row[0]

    # 다음 열부터 [Redoinfo]가 "Update Resident Value" 발견하면, 마지막
    LSN(lastLSN) 값 저장
    if Found :
        RedoInfo = row[10]
        if RedoInfo == "Update Resident Value" :
            lastLSN = row[0]
            break

# USB.dd에서 추출한 "$LogFile"를 읽어오기
with open("$LogFile", 'rb') as log : log_data = log.read()

# "$LogFile"에서 [ZoneTransfer] 발견
for m in re.finditer(b'\x5a\x6f\x6e\x65\x49\x64\x3d', log_data):
    start = m.start()
    ZoneTransfer = str(log_data[start:start+500])[2:-1][:
str(log_data[start:start+500])[2:-1].find('#)].replace(r'\r\n', '\r\n')
    LSN = str(log_data[start-168:start-168+4])[2:-1]
    LSN_Offset = str()
    for hex in LSN.split(r'\x'):
        if len(hex) == 0 :
            continue
        elif len(hex) == 2:
            LSN_Offset += hex
        else:
            LSN_Offset += hex[:2]
            for alpha in hex[2:] :
                LSN_Offset += str(alpha.encode("utf-8")).hex()
    LSN_Offset = LSN_Offset[-2:] + LSN_Offset[-4:-2] + LSN_Offset[2:4] +
    LSN_Offset[0:2]
    zone_identified_LSN = int('0x'+LSN_Offset, 16)

    # ZoneTransfer 가 저장된 LSN 0이 startLSN 과 lastLSN의 사이에 존재한다면,
    웹브라우저로 다운로드한 파일(.crdownload)의 ZONEID임을 증명

```

```

if startLSN < zone_identified_LSN and zone_identified_LSN < lastLSN :
    if 'ZoneId=3' in ZoneTransfer:
        for ENTER in ZoneTransfer.split('\r\n'):
            print("[*] " + ENTER)
print("[-----]")
#[2] 완전 삭제 흔적 발견
print("\n\n<=====ERASER DISCOVERY=====>")
idx = 0
eraser_case = dict()
file_deletion = list()

# NTFS Log Tracker v1.7에서 추출된 DB-Logfile TABLE
for row in data:
    Event = row[2]

    # [Event] 컬럼에서 "File Deletion" 발견
    if Event == "File Deletion" :
        file_deletion.append(row)
        MFT_ModifiedTime = row[8]
        CreateTime = row[6]

        # [MFT_ModifiedTime] 컬럼 값이 [CreateTime]보다 빠를 경우
        if MFT_ModifiedTime > CreateTime :
            filename = row[4]
            re_row_list = list()

            # 동일한 [filename]의 Logfile 중에서 [Event]가 "Renaming File"
            발견하면, 완전 삭제(Eraser) 흔적 증명
            for re_row in list(reversed(data)) :
                re_filename = re_row[4]
                re_Event = re_row[2]
                if re_filename == filename and re_Event == "Renaming File" :
                    filename = re.findall('(.+?) ->', re_row[3])[0]
                    re_row_list.append(re_row)
                    continue
                if re_row_list :
                    eraser_case[filename] = list(reversed(re_row_list))

# 출력 구문
for key, value in eraser_case.items() :
    print("[*] filename : " + key)
    print("[*] detail")
    print("[-->] LSN EventTime Event")
    Detail FullPath")
    for log in value:
        print("[-->] ", end='')


```

```

        print(log[0], log[1], log[2], log[3], log[5], sep='\t')
        print("[-->]", end='')
        print(file_deletion[idx][0], file_deletion[idx][1],
file_deletion[idx][2], file_deletion[idx][3], file_deletion[idx][5],
sep='\t')
        print("[-----]")
-----])

#[3] 파일 복구
print("\n\n<<=====FILE RECOVERY=====>>")

# USB.dd 를 읽어오기
with open('USB.dd', 'rb') as usb : usb_data = usb.read().hex()

# [1] 완전 삭제 도구 흔적에서 찾았던 삭제 파일(2_2_5_a.txt/2_2_5_b.txt)을
기준으로
for deletion_filename in eraser_case.keys() :
    count = int()
    print("[*] filename : " + deletion_filename)

# NTFS Log Tracker v1.7 에서 추출된 DB-Logfile TABLE
for row in data :
    Event = row[2]
    filename = row[4]

    # 삭제 파일의 [Event] 컬럼이 "Non-Resident"일 경우
    if deletion_filename == filename and "Non-Resident File" in Event :
        count+=1
        LSN = row[0]
        Detail = row[3]

        # 삭제 파일의 [Detail] 컬럼에서 파일 데이터가 저장된 클러스터
offsets 을 획득
        cluster_offset = int(re.findall('Data Runs\((in Volume\):
(.+?)\(', Detail)[0]) * 8 * 512 * 2
        print("[*] File Candidate by Cluster offset "+re.findall('Data
Runs\((in Volume\): (.+?)\(', Detail)[0])

        # 획득한 클러스터 offsets 으로 이동하여 낙낙하게 파일 데이터를
출력하여, 사용자의 판단으로 파일 데이터를 확인

print(bytes.fromhex(usb_data[cluster_offset:cluster_offset+1100]))
        print("[-----]")
-----]

if not count:

```

```
print("[-  
-----]  
")
```