201 – Log and Found

Team Information

Team Name : ForensicGPT

Team Member : Eungchang Lee, Donghyun HA, Hyunwoo Shin, Jinhyeok Son

Email Address : forensicgpt@googlegroups.com

Instructions

Description Kate is a server administrator at a fashion design company and recently underwent an internal audit within the company due to an incident where design files stored on the server were leaked. The company has requested a digital forensics analysis of the server's volume to resolve this issue. Please provide the analysis results for each question.

Target	Hash (MD5)
draft_server.001	4e6354ddcf52c2f0e436c60f2c5878ac

Questions

1) List the original and changed file names of the renamed files. (50 points)
2) List the file names of the deleted files. (50 points)
3) Provide the deleted time for each file. (100 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:	HxD	Publisher:	Maël Hörz
Version:	2.5.0.0		
URL:	https://www.mh-nexus.de		

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

Name:	ARIN	Publisher:	Seonho Lee
Version:	Beta		
URL:	https://github.com/horensic/ARIN		

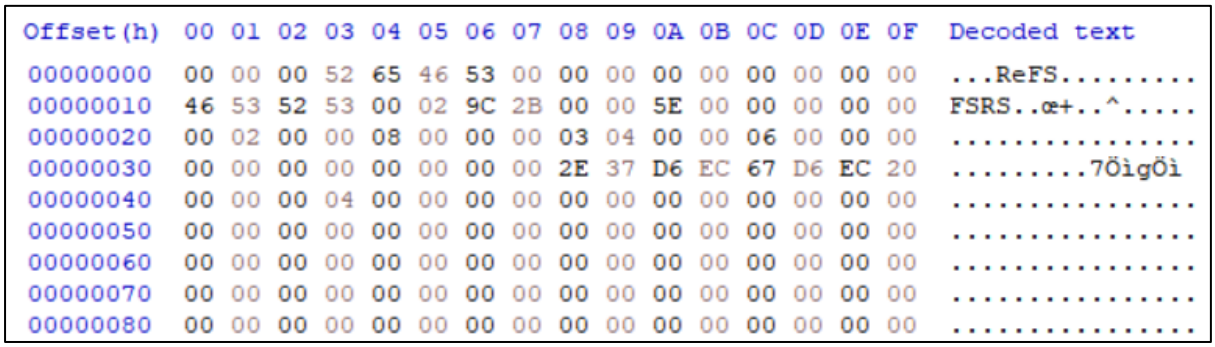
Name:	Dcode	Publisher:	Digital Detective
Version:	5.5		
URL:	https://www.digital-detective.net/dcode/		

Step-by-step methodology:

[그림 1] draft_server.001 hash

다운로드 받은 증거 파일의 해시 값을 산출하여 무결성을 확인하였습니다.

1) List the original and changed file names of the renamed files. (50 points)

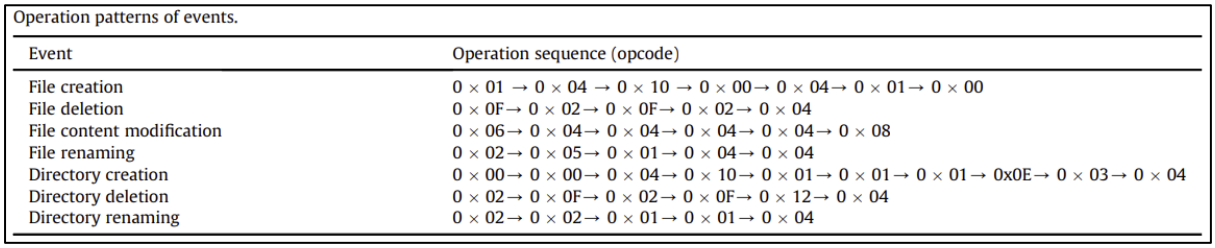


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	00	00	00	52	65	46	53	00	00	00	00	00	00	00	00	00	...ReFS.....
00000010	46	53	52	53	00	02	9C	2B	00	00	5E	00	00	00	00	00	FSRS..œ+..^.....
00000020	00	02	00	00	08	00	00	00	03	04	00	00	06	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	2E	37	D6	EC	67	D6	EC	20	.....7ÖìgÖì
00000040	00	00	00	04	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

[그림 2] 분석 대상 파일의 파일 시스템 확인

먼저, HxD를 통해 분석 대상 파일의 파일 시스템이 ReFS임을 확인하였습니다. 그리고, 참고 문헌[1]를 통해 ReFS에 존재하는 Logfile이 Windows NTFS에 존재하는 $Logfile이 다르다는 것을 확인할 수 있었습니다.

그리고, 해당 참고문헌에서 보이고 있는 다음 그림을 통해 File renaming과 File deletion 시에 발생하는 트랜잭션 로그의 로그 레코드 구조 내에 기록되는 opcode를 확인하였습니다.



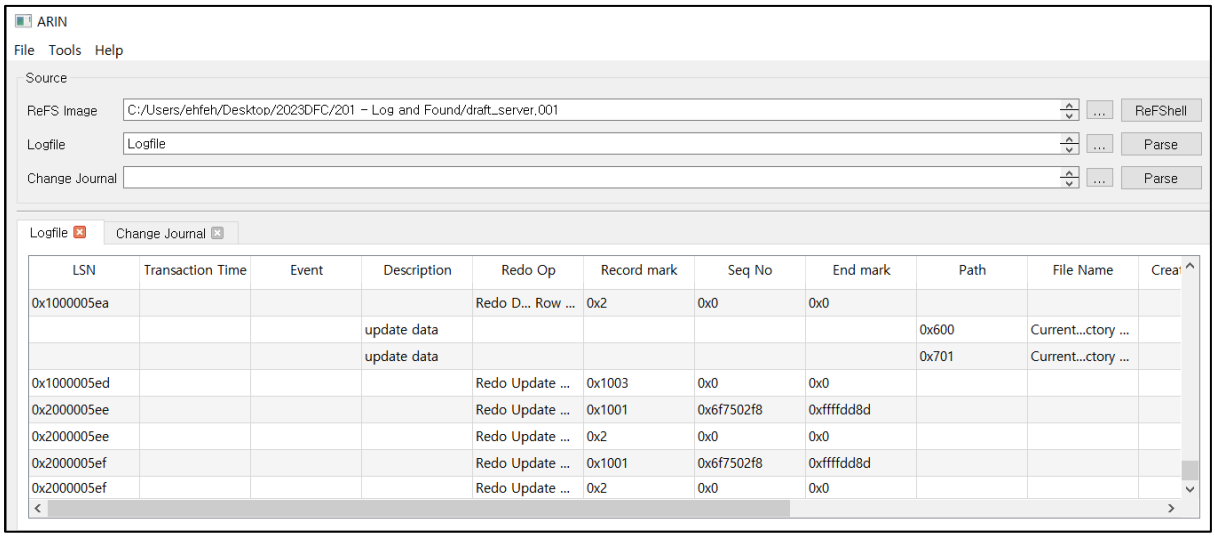
Operation patterns of events.

Event	Operation sequence (opcode)
File creation	0 × 01 → 0 × 04 → 0 × 10 → 0 × 00→ 0 × 04→ 0 × 01→ 0 × 00
File deletion	0 × 0F→ 0 × 02→ 0 × 0F→ 0 × 02→ 0 × 04
File content modification	0 × 06→ 0 × 04→ 0 × 04→ 0 × 04→ 0 × 04→ 0 × 08
File renaming	0 × 02→ 0 × 05→ 0 × 01→ 0 × 04→ 0 × 04
Directory creation	0 × 00→ 0 × 00→ 0 × 04→ 0 × 10→ 0 × 01→ 0 × 01→ 0 × 01→ 0x0E→ 0 × 03→ 0 × 04
Directory deletion	0 × 02→ 0 × 0F→ 0 × 02→ 0 × 0F→ 0 × 12→ 0 × 04
Directory renaming	0 × 02→ 0 × 02→ 0 × 01→ 0 × 01→ 0 × 04

[그림 3] Operation patterns of events

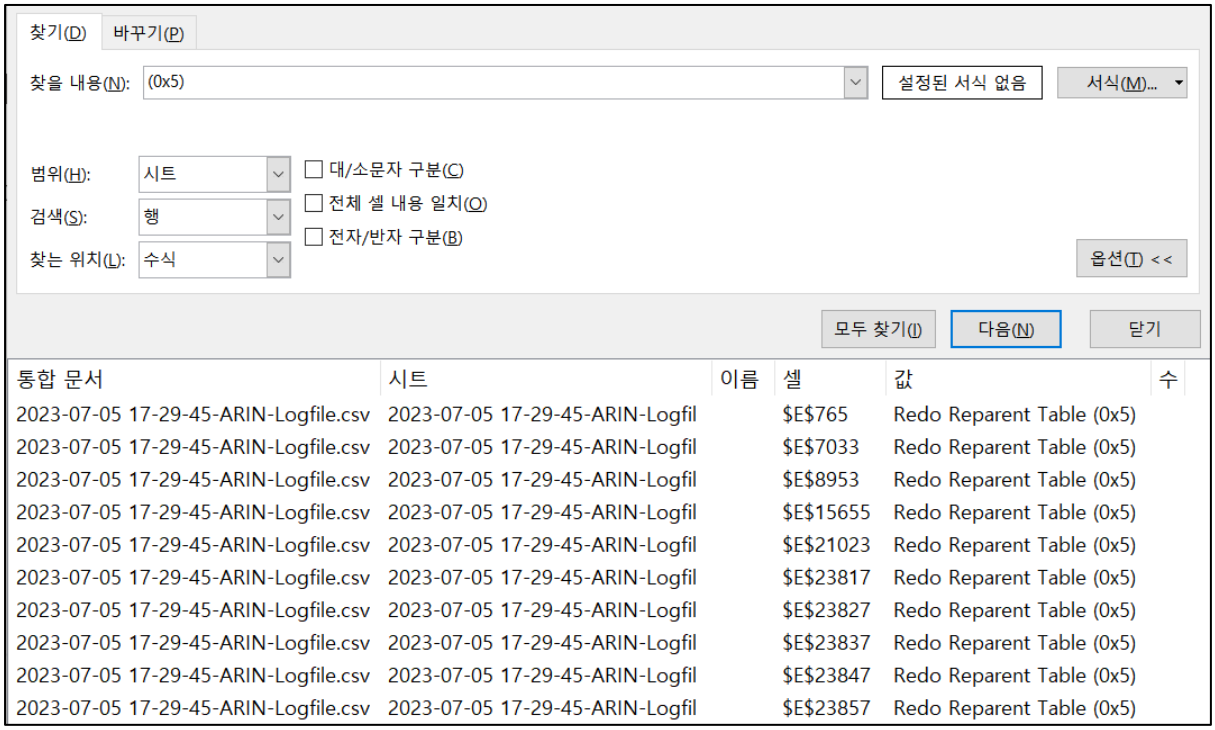
ReFS에서 File renaming에 관한 opcode sequence는 0x02 -> 0x05 -> 0x01 -> 0x04 -> 0x04 임을 위 그림에서 확인하였습니다.

[1] https://www.sciencedirect.com/science/article/pii/S2666281721000342



[그림 4] Analysis Logfile in ReFS using ARIN

HxD를 통해 로우 레벨로 분석을 수행하기 전, 본 참고 문헌에서 사용된 도구 ARIN을 통해 분석에 참고할 csv파일을 뽑아내었습니다. 해당 도구로는 완벽한 파싱이 되지 않아 참고 자료로 활용하였습니다.



[그림 5] Searching 0x5 opcode

Renaming과 관련된 opcode sequence중 구별되는 opcode인 0x5를 토대로 검색을 진행한 결과, 10개의 값이 산출되었습니다.

0x10000003d				Redo Delete Row (0x2)	0x1001	0x6eb0a140	0xffffdd8d		
0x10000003d				Redo Reparent Table (0x5)	0x0	0x6eb0a238	0xffffdd8d		
0x10000003d				Redo Insert Row (0x1)	0x2	0x0	0x0		
			update data					0x600	Current Directory Index
			update data					0x600	1008023_B.jpg

[그림 6] file renaming에 관한 opcode sequence 흔적 확인

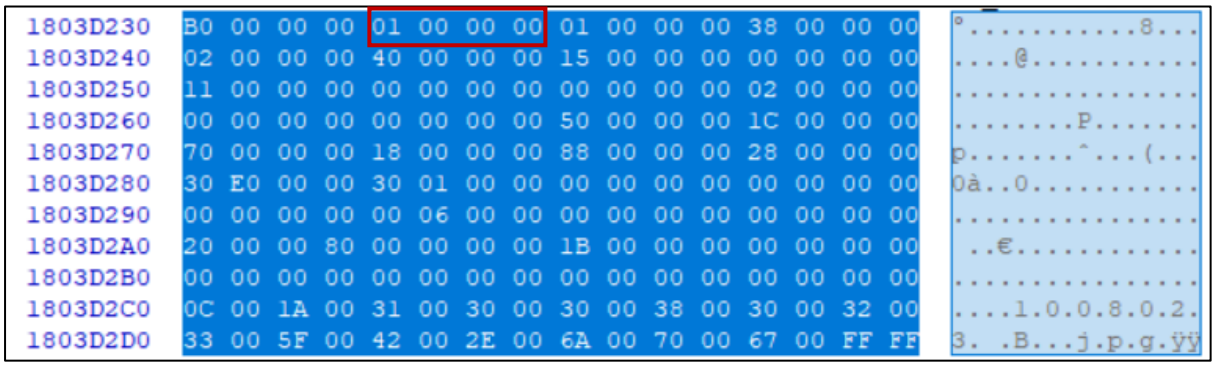
하나씩 살펴보면 Redo Delete Row(0x2) -> Redo Reparent Table(0x5) -> Redo Insert Row(0x1)의 opcode sequence를 가지고, 0x600에 해당되는 루트 디렉토리에서 update data를 진행하고 1008023_B.jpg에 대한 update data를 진행하는 것을 알 수 있습니다.

이러한 흔적과 참고 문헌을 기반으로 HxD 상에서 로우 레벨로 살펴보면 다음과 같이 기록되는 것을 확인할 수 있습니다.

- **0x02 -> 0x05**

[그림 7] File renaming에 대한 log entry의 opcode 추적 - 1

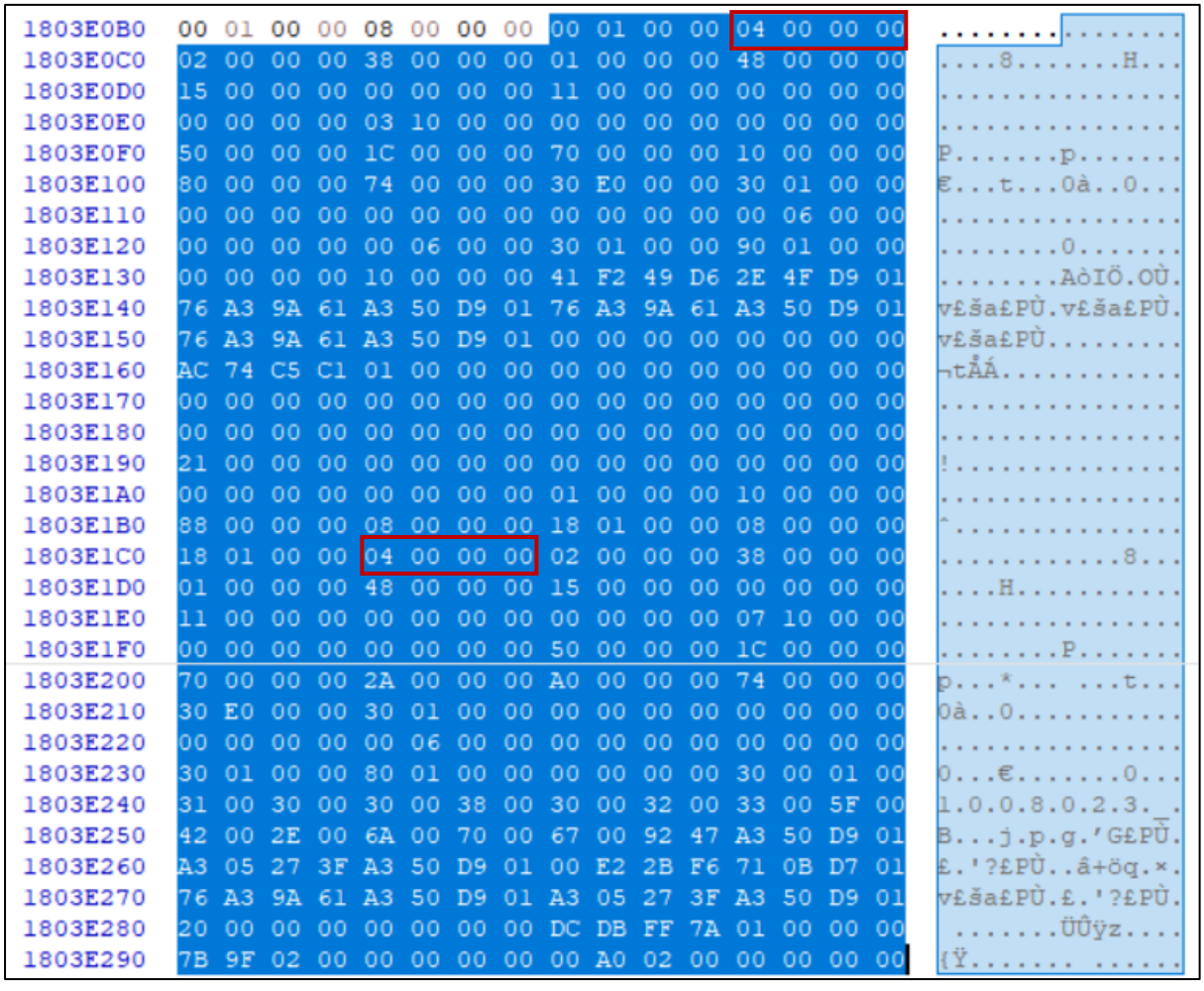
- **0x01**



[그림 8] File renaming에 대한 log entry의 opcode 추적 - 2

- **0x04(0x600 디렉토리) -> 0x04(1008023_B.jpg)**

ARIN 도구에서 살펴본 update data는 실제로 0x04 opcode에 해당하는 Redo Update with Root Operation이 진행된 것을 알 수 있습니다.



[그림 9] File renaming에 대한 log entry의 opcode 추적 - 3

이러한 구조를 가지는 renaming 된 파일들을 식별한 결과, 다음과 같이 표로 정리해볼 수 있습니다.

[표 1] 원본 파일과 이름이 변경된 파일 리스트

Original File -> Renaming File
1008022_B.jpg -> 1008023_B.jpg
1012200_B.jpg → 1012201_B.jpg
1012377_B.jpg → 1012378_B.jpg
1014097_B.jpg → 1014098_B.jpg
1014394_B.jpg → 1014398_B.jpg
1015100_B.jpg → 1015101_B.jpg
1013282_B.jpg → 1013283_B.jpg
1012211_B.jpg → 1012210_B.jpg
1012343_B.jpg → 1012341_B.jpg
1014179_B.jpg → 1014180_B.jpg

2) List the file names of the deleted files. (50 points)

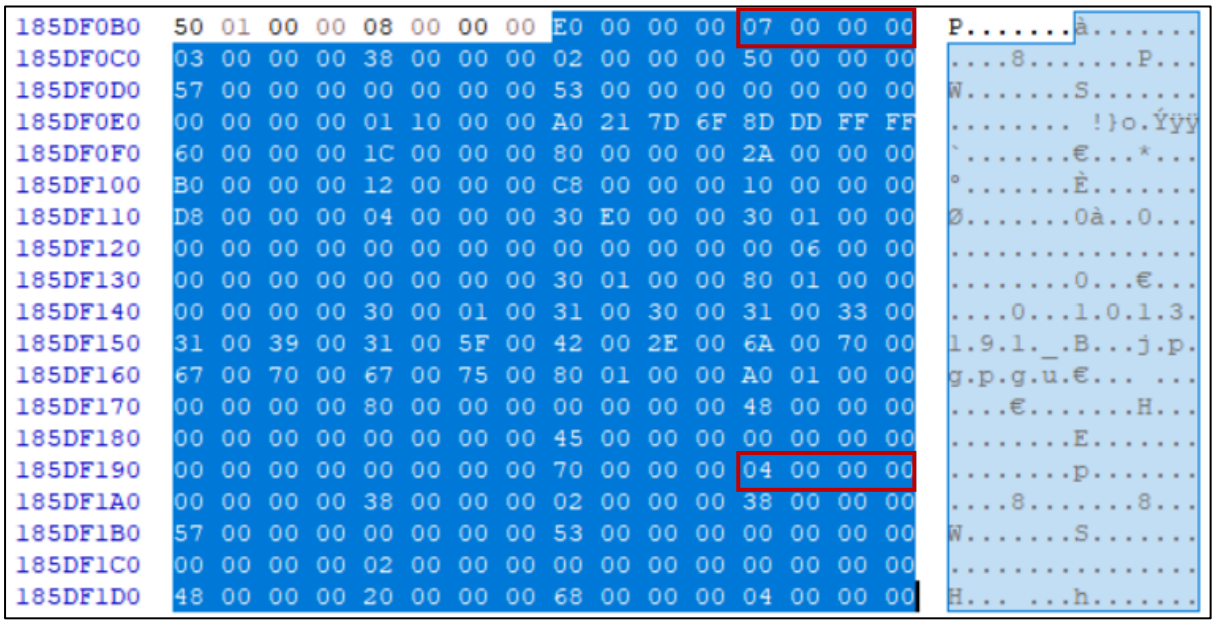
File renaming과 달리, ReFS에서 삭제된 파일의 opcode sequence는 [그림 3]과 같이 **0x0F -> 0x02 -> 0x0F -> 0x02 -> 0x04**의 순서를 가지고 있습니다.

- 0x0F : Redo Delete Table
- 0x02 : Redo Delete Row
- 0x04 : Redo Update Data with Root

HxD 상에서 살펴본 삭제된 파일에 대한 트랜잭션 로그에 대한 로그 레코드 구조는 다음과 같았습니다.

- **0x07 -> 0x04**

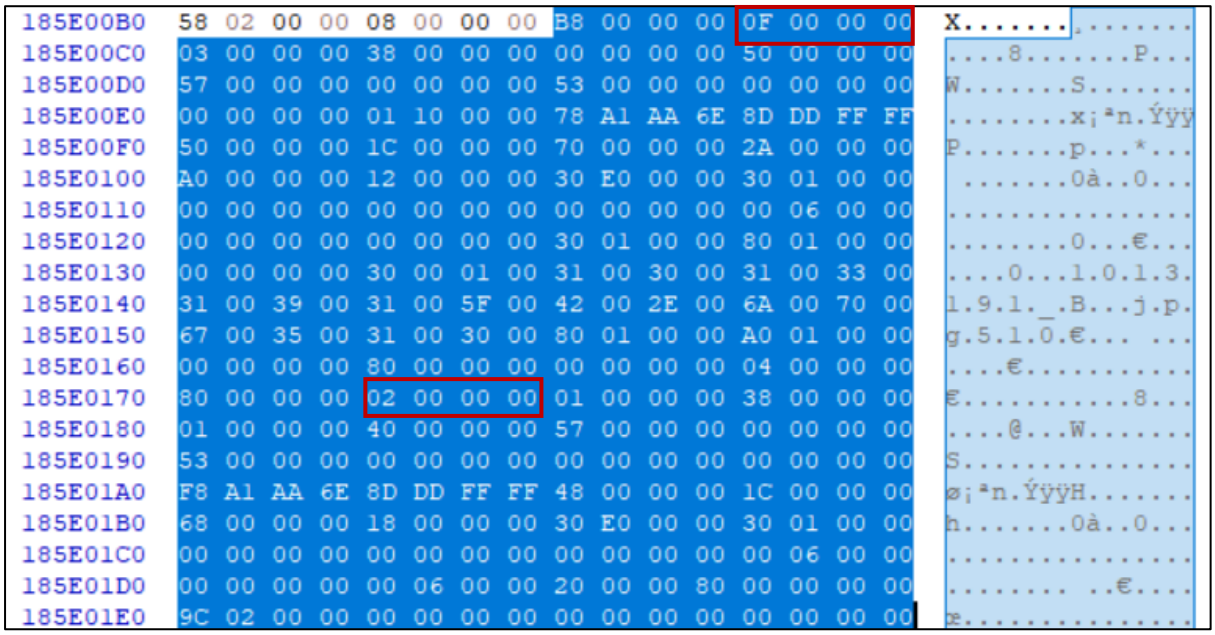
삭제의 경우 0x07(Redo Free)가 발생한다는 것을 다른 참고 문헌[2]을 통해 알 수 있었습니다.



[그림 10] File deletion에 대한 log entry의 opcode 추적 - 1

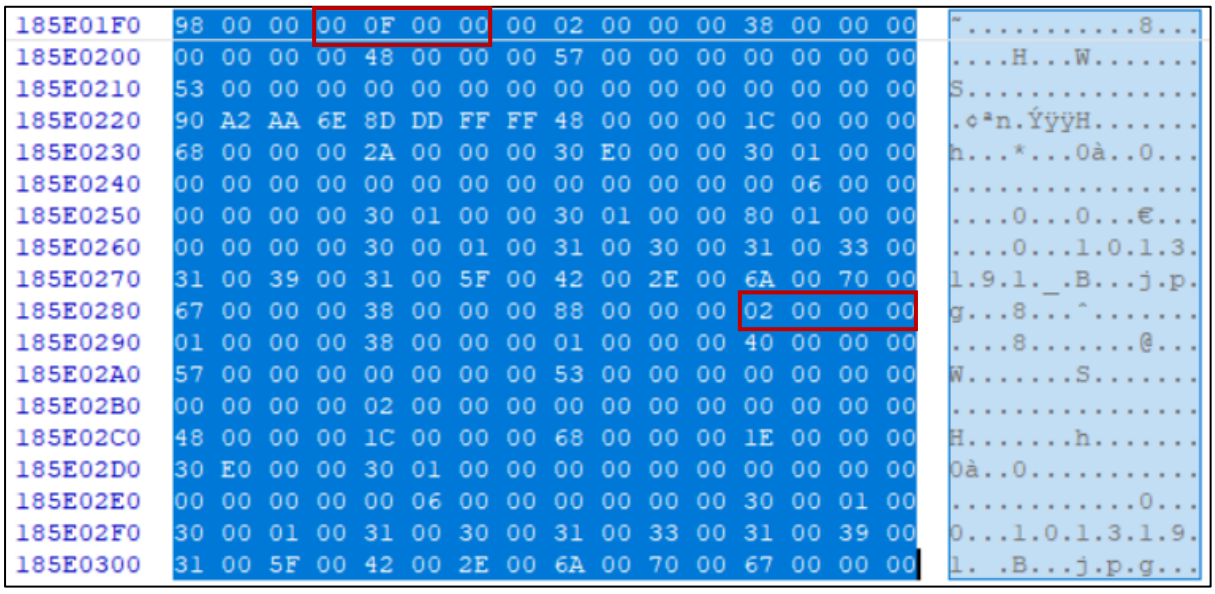
[2] https://www.kci.go.kr/kciportal/ci/sereArticleSearch/ciSereArtiView.kci?sereArticleSearchBean.artiId=ART002513085

- **0x0F -> 0x02**



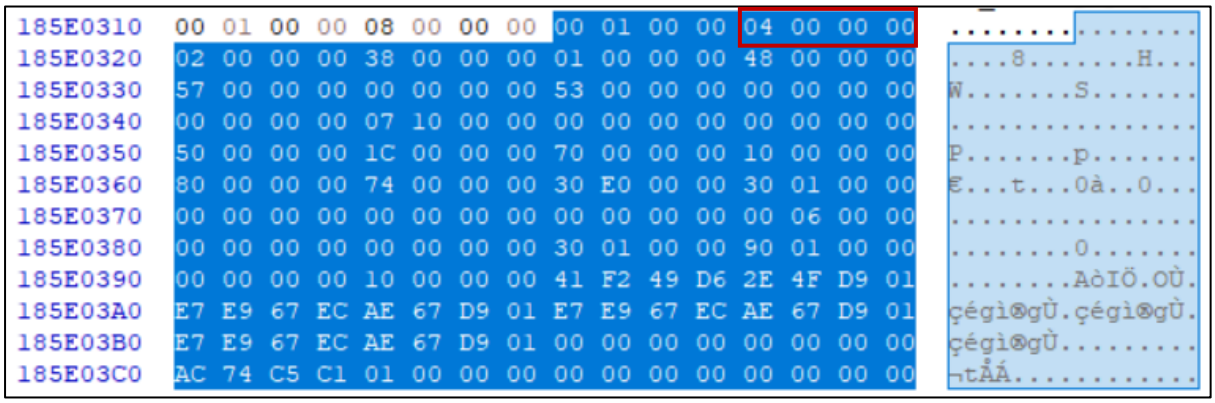
[그림 11] File deletion에 대한 log entry의 opcode 추적 - 2

- **0x0F -> 0x02**



[그림 12] File deletion에 대한 log entry의 opcode 추적 - 3

- **0x04**



[그림 13] File deletion에 대한 log entry의 opcode 추적 - 4

이러한 구조를 가지는 삭제된 파일들을 식별한 결과, 다음과 같이 표로 정리해볼 수 있습니다.

[표 2] 삭제된 파일의 파일 이름 리스트

Deleted File Name
1013191_B.jpg
1012353_B.jpg
1008103_B.jpg
1013029_B.jpg
1014381_B.jpg

3) Provide the deleted time for each file. (100 points)

참고문헌 2에서 확인한 파일 삭제에 대한 삭제 시각은 삭제된 파일의 부모 디렉터리의 메타데이터 수정시간으로 알 수 있음을 다음 그림과 같이 확인하였습니다.

Case	Transaction Time	Description
파일 생성	파일 생성시간	File Record 내 타임스탬프 중 생성시간
파일 삭제	부모 디렉터리 메타데이터 수정시간	삭제된 파일의 부모 디렉터리의 메타데이터 수정시간
파일 내용 수정	파일 수정시간	File Record 내 타임스탬프 중 수정시간
파일 이름 변경	부모 디렉터리 메타데이터 수정시간	이름이 변경된 파일의 부모 디렉터리의 메타데이터 수정시간
디렉터리 생성	디렉터리 생성시간	Directory Record 내 타임스탬프 중 생성시간
디렉터리 삭제	부모 디렉터리 메타데이터 수정시간	삭제된 디렉터리의 부모 디렉터리의 메타데이터 수정시간
디렉터리 이름 변경	부모 디렉터리 메타데이터 수정시간	이름이 변경된 디렉터리의 부모 디렉터리의 메타데이터 수정시간

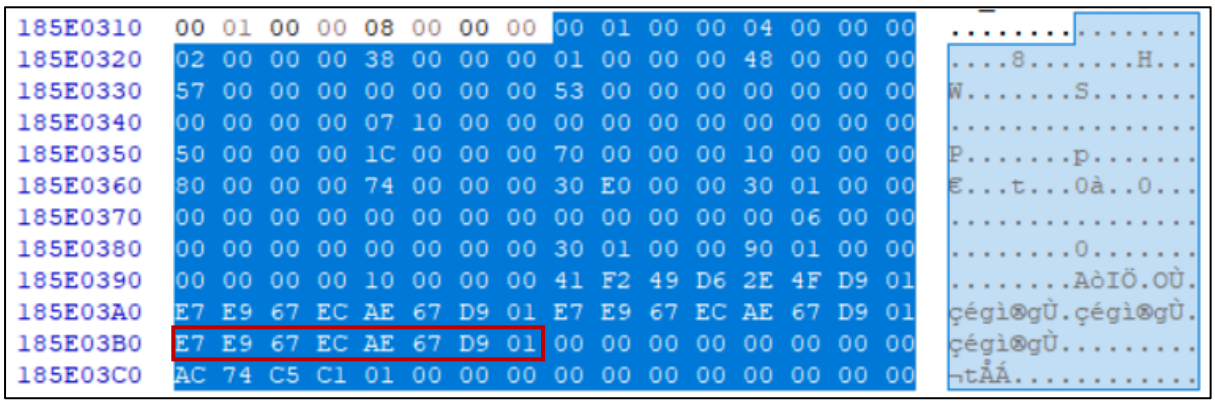
[그림 14] 트랜잭션 시간 값이 기록되는 경우

또한, File deletion event에 대한 삭제 시각은 5번째 record에 Change Time을 확인하는 것으로 다음 그림과 같이 참고문헌 1에서 참고할 수 있었습니다.

Event	Event time
File creation	Creation time in the 5th record
File deletion	Changed time in the 5th record
File content modification	Modified time in the 4th record
File renaming	Changed time in the 4th record
Directory creation	Creation time in the 3rd record
Directory deletion	Changed time in the 6th record
Directory renaming	Changed time in the 5th record

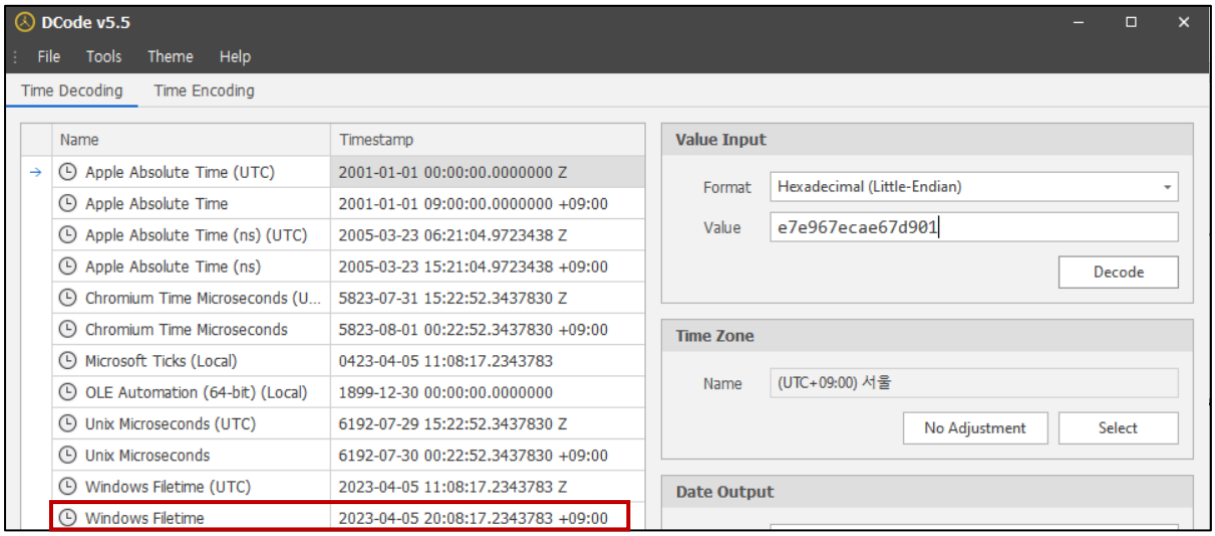
[그림 15] 이벤트 시간 식별 방법

따라서, 삭제된 파일에 대해 5번째 record인 0x04 opcode가 기록된 record를 살펴보았습니다.



[그림 16] opcode 0x04를 가지는 record에서 change time 확인

0x04 opcode를 가진 부모 디렉터리를 나타내는 record에서 Change Time은 4번째에 기록되는 것을 알 수 있습니다.



[그림 17] Hex로 표현된 시간 값 변환 수행

따라서, 위 2번에서 확인한 삭제된 파일들에 대한 삭제 시각은 다음과 같이 표로 정리해볼 수 있습니다.

[표 3] 삭제된 파일의 삭제 시각 확인

Deleted File	Deleted time
1013191_B.jpg	2023-04-05 20:08:17.2343783 (UTC + 09:00)
1012353_B.jpg	2023-04-05 20:08:43.7762854 (UTC + 09:00)
1008103_B.jpg	2023-04-05 20:09:13.4040759 (UTC + 09:00)
1013029_B.jpg	2023-04-05 20:09.38.4102339 (UTC + 09:00)
1014381_B.jpg	2023-04-05 20:10:06.9314698 (UTC + 09:00)