202 – Why is not it playing?

Team Information

Team Name : ForensicGPT

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

Email Address : forensicgpt@googlegroups.com

Instructions

Description The memory card installed in the video recording device is damaged, so only some of the data in the memory can be obtained. The encoding information and video structure of file are found in the remaining data, but the image is not confirmed. The video recording device consists of a file system including a security element that can play and back up videos only through the manufacturer's exclusive viewer. The video, voice, and time information are recorded in this file, and one book is recorded in the video file.

Target	Hash (MD5)
final-final-problem.img	317492066299DE92BA2A2D718785160A

Questions

1) Submit the title and edition information of the book recorded in the video. (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:	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:	Visual Studio code	Publisher:	Microsoft
Version:	1.80.2		
URL:	https://code.visualstudio.com/download		

Name:	ffplay	Publisher:	FFmpeg developers
Version:	6.0		
URL:	https://ffmpeg.org/download.html		

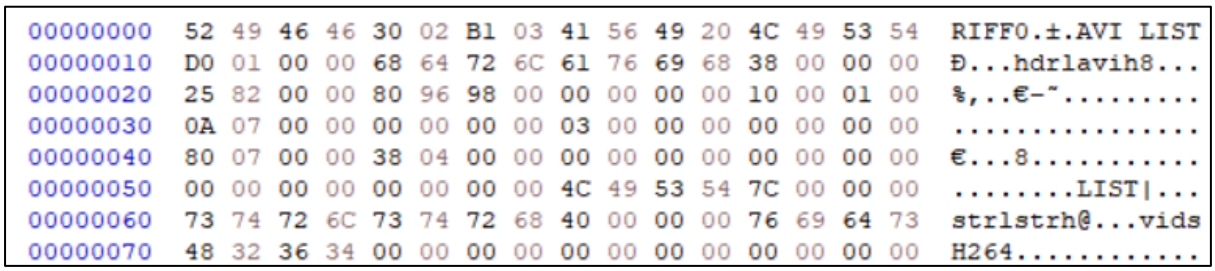
Name:	kmplayer	Publisher:	PandoraTV
Version:	4.2.2.68		
URL:	https://www.kmplayer.com/kr/home		

Step-by-step methodology:

[그림 1] 해시 값 확인

분석에 앞서, 주어진 파일에 대한 해시 값을 산출하여 MD5 해시 값이 일치함을 확인하였습니다.

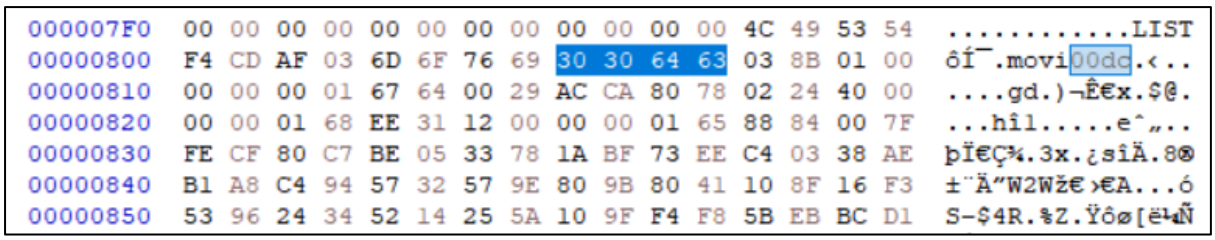
1) Submit the title and edition information of the book recorded in the video. (200 points)



```
00000000  52 49 46 46 30 02 B1 03 41 56 49 20 4C 49 53 54  RIFF0.±.AVI LIST
00000010  D0 01 00 00 68 64 72 6C 61 76 69 68 38 00 00 00  Ð...hdrlavih8...
00000020  25 82 00 00 80 96 98 00 00 00 00 00 10 00 01 00  %,..€–˜.........
00000030  0A 07 00 00 00 00 00 00 03 00 00 00 00 00 00 00  ................
00000040  80 07 00 00 38 04 00 00 00 00 00 00 00 00 00 00  €...8...........
00000050  00 00 00 00 00 00 00 00 4C 49 53 54 7C 00 00 00  ........LIST|...
00000060  73 74 72 6C 73 74 72 68 40 00 00 00 76 69 64 73  strlstrh@...vids
00000070  48 32 36 34 00 00 00 00 00 00 00 00 00 00 00 00  H264............
```

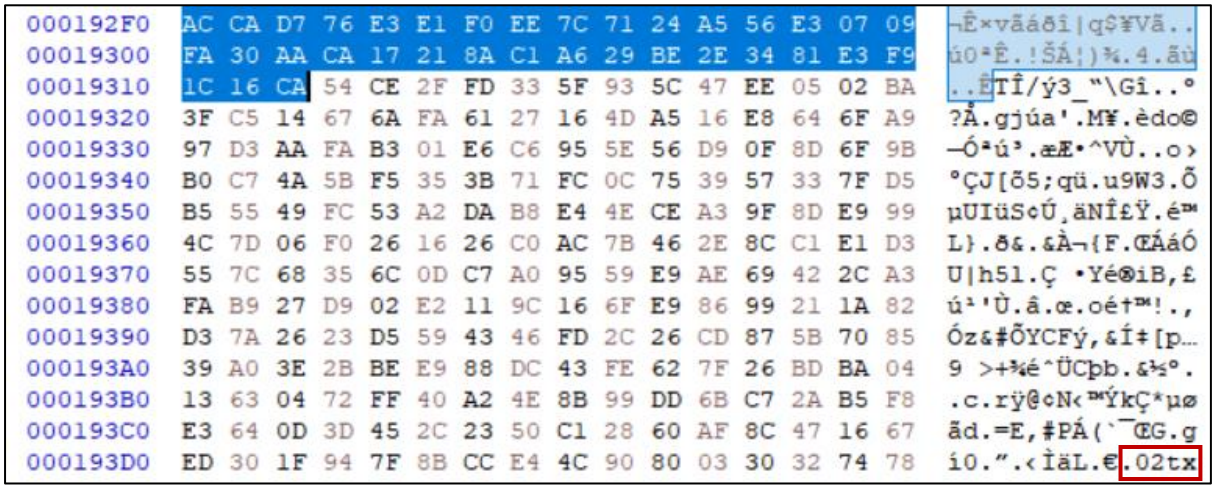
[그림 2] 헤더 구조(일부)

우선 주어진 파일은 RIFF 헤더 시그니처를 가진 AVI 파일 구조를 가지고 있습니다.



```
000007F0  00 00 00 00 00 00 00 00 00 00 00 00 4C 49 53 54  ............LIST
00000800  F4 CD AF 03 6D 6F 76 69 30 30 64 63 03 8B 01 00  ôÍ¯.movi00dc.‹..
00000810  00 00 00 01 67 64 00 29 AC CA 80 78 02 24 40 00  ....gd.)¬Ê€x.$@.
00000820  00 00 01 68 EE 31 12 00 00 00 01 65 88 84 00 7F  ...hî1.....eˆ„..
00000830  FE CF 80 C7 BE 05 33 78 1A BF 73 EE C4 03 38 AE  þÏ€Ç¾.3x.¿sîÄ.8®
00000840  B1 A8 C4 94 57 32 57 9E 80 9B 80 41 10 8F 16 F3  ±¨Ä"W2Wž€›€A...ó
00000850  53 96 24 34 52 14 25 5A 10 9F F4 F8 5B EB BC D1  S–$4R.%Z.Ÿôø[ë¼Ñ
```

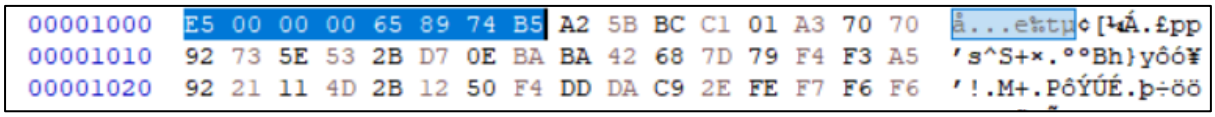
[그림 3] movi chunk의 00dc video data



```
000192F0  AC CA D7 76 E3 E1 F0 EE 7C 71 24 A5 56 E3 07 09  ¬Ê×vãáðî|q$¥Vã..
00019300  FA 30 AA CA 17 21 8A C1 A6 29 BE 2E 34 81 E3 F9  ú0ªÊ.!ŠÁ¦)¾.4.ãù
00019310  1C 16 CA 54 CE 2F FD 33 5F 93 5C 47 EE 05 02 BA  ..ÊTÎ/ý3_"\Gî..º
00019320  3F C5 14 67 6A FA 61 27 16 4D A5 16 E8 64 6F A9  ?Å.gjúa'.M¥.èdo©
00019330  97 D3 AA FA B3 01 E6 C6 95 5E 56 D9 0F 8D 6F 9B  —Óªú³.æÆ•^VÙ..o›
00019340  B0 C7 4A 5B F5 35 3B 71 FC 0C 75 39 57 33 7F D5  °ÇJ[õ5;qü.u9W3.Õ
00019350  B5 55 49 FC 53 A2 DA B8 E4 4E CE A3 9F 8D E9 99  µUIüS¢Ú¸äNΣŸ.é™
00019360  4C 7D 06 F0 26 16 26 C0 AC 7B 46 2E 8C C1 E1 D3  L}.ð&.&À¬{F.ŒÁáÓ
00019370  55 7C 68 35 6C 0D C7 A0 95 59 E9 AE 69 42 2C A3  U|h5l.Ç •Yé®iB,£
00019380  FA B9 27 D9 02 E2 11 9C 16 6F E9 86 99 21 1A 82  ú¹'Ù.â.œ.oé†™!.‚
00019390  D3 7A 26 23 D5 59 43 46 FD 2C 26 CD 87 5B 70 85  Óz&#ÕYCFý,&Í‡[p…
000193A0  39 A0 3E 2B BE E9 88 DC 43 FE 62 7F 26 BD BA 04  9 >+¾éˆÜCþb.&½º.
000193B0  13 63 04 72 FF 40 A2 4E 8B 99 DD 6B C7 2A B5 F8  .c.rÿ@¢N‹™ÝkÇ*µø
000193C0  E3 64 0D 3D 45 2C 23 50 C1 28 60 AF 8C 47 16 67  ãd.=E,#PÁ(`¯ŒG.g
000193D0  ED 30 1F 94 7F 8B CC E4 4C 90 80 03 30 32 74 78  í0.”.‹ÌäL.€.02tx
```

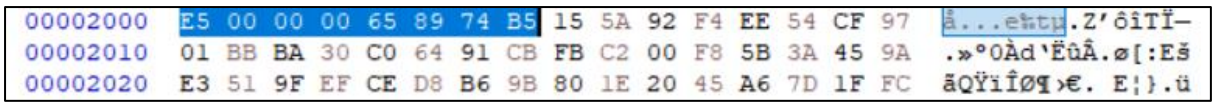
[그림 4] 사이즈 miss

[그림 3]과 [그림 4]를 보면 movi chunk에서 video data를 갖는 첫번째 00dc를 살펴보면, data size가 0x018B03인 것을 알 수 있습니다. 하지만, 헤더에서도 알 수 있듯이 해당 파일이 가지는 H264 codec는 SPS, PPS, IDR ... 와 같은 구조를 가지기 때문에, 해당 사이즈를 따라 SPS 데이터 시작 오프셋인 0x810부터 사이즈를 조회해보면 02tx 영역 이전까지 사이즈가 올바르게 매칭되지 않는 모습을 확인할 수 있습니다. 또한, 사이즈 이후 0x19313 오프셋부터 .이전까지 데이터의 사이즈가 0xC8(=200 bytes)임을 알 수 있습니다.



00001000	E5	00	00	00	65	89	74	B5	A2	5B	BC	C1	01	A3	70	70
00001010	92	73	5E	53	2B	D7	0E	BA	BA	42	68	7D	79	F4	F3	A5
00001020	92	21	11	4D	2B	12	50	F4	DD	DA	C9	2E	FE	F7	F6	F6

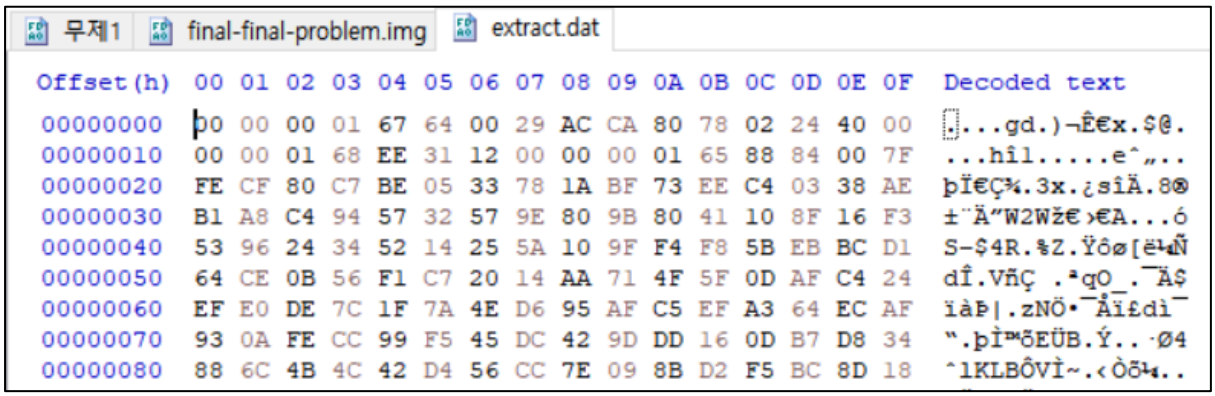
[그림 5] 특정 8 bytes 데이터 식별 - 1



00002000	E5	00	00	00	65	89	74	B5	15	5A	92	F4	EE	54	CF	97
00002010	01	BB	BA	30	C0	64	91	CB	FB	C2	00	F8	5B	3A	45	9A
00002020	E3	51	9F	EF	CE	D8	B6	9B	80	1E	20	45	A6	7D	1F	FC

[그림 6] 특정 8 bytes 데이터 식별 - 2

앞서 살펴본 하나의 00dc 데이터 내에는 반복되는 특정 8 bytes 데이터가 존재했습니다. 해당 데이터의 개수는 총 25개로 이는, 25 * 8 = 200(=0xC8)로 size miss가 0xC8만큼 존재하는 것과 일치했습니다. 이는, 다른 00dc 데이터도 동일하게 특정 8 bytes data를 삭제해주면 video data size가 딱 맞게 일치함을 확인하였습니다.



무제1 | final-final-problem.img | extract.dat

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	00	00	00	01	67	64	00	29	AC	CA	80	78	02	24	40	00	....gd.)¬Ê€x.$@.
00000010	00	00	01	68	EE	31	12	00	00	00	01	65	88	84	00	7F	...hî1.....e^„..
00000020	FE	CF	80	C7	BE	05	33	78	1A	BF	73	EE	C4	03	38	AE	þÏ€Ç¾.3x.¿sîÄ.8®
00000030	B1	A8	C4	94	57	32	57	9E	80	9B	80	41	10	8F	16	F3	±¨Ä"W2Wž€›€A...ó
00000040	53	96	24	34	52	14	25	5A	10	9F	F4	F8	5B	EB	BC	D1	S–$4R.%Z.Ÿôø[ë¼Ñ
00000050	64	CE	0B	56	F1	C7	20	14	AA	71	4F	5F	0D	AF	C4	24	dÎ.VñÇ .ªqO_.¯Ä$
00000060	EF	E0	DE	7C	1F	7A	4E	D6	95	AF	C5	EF	A3	64	EC	AF	ïàÞ|.zNÖ•¯Åï£dì¯
00000070	93	0A	FE	CC	99	F5	45	DC	42	9D	DD	16	0D	B7	D8	34	"..þÌ™õEÜB.Ý..·Ø4
00000080	88	6C	4B	4C	42	D4	56	CC	7E	09	8B	D2	F5	BC	8D	18	ˆlKLBÔVÌ~.‹Òõ¼..

[그림 7] 00dc video frame data추출

결과적으로 여러 video data중 하나의 00dc video frame data를 추출하였고, 특정 8 bytes를 모두 제거한 뒤 SPS부터 오도록하여 extract.dat으로 저장 후 ffplay를 통해 확인하였습니다.

```
C:₩Users₩ehfeh₩Desktop₩2023DFC₩202 - Why is not it playing>ffplay.exe extract.dat
ffplay version 6.0-full_build-www.gyan.dev Copyright (c) 2003-2023 the FFmpeg developers
  built with gcc 12.2.0 (Rev10, Built by MSYS2 project)
  configuration: --enable-gpl --enable-version3 --enable-static --disable-w32threads --disab
onfig --enable-iconv --enable-gnutls --enable-libxml2 --enable-gmp --enable-bzlib --enable-l
ble-zlib --enable-librist --enable-libsrt --enable-libssh --enable-libzmq --enable-avisynth
libcaca --enable-sdl2 --enable-libaribb24 --enable-libdav1d --enable-libdavs2 --enable-libua
le-libravle --enable-libsvtav1 --enable-libwebp --enable-libx264 --enable-libx265 --enable-l
nable-libaom --enable-libjxl --enable-libopenjpeg --enable-libvpx --enable-mediafoundation --
0r --enable-libfreetype --enable-libfribidi --enable-liblensfun --enable-libvidstab --enable
enable-amf --enable-cuda-llvm --enable-cuvid --enable-ffnvcodec --enable-nvdec --enable-nven
dxva2 --enable-libvpl --enable-libshaderc --enable-vulkan --enable-libplacebo --enable-openc
libgme --enable-libmodplug --enable-libopenmpt --enable-libopencore-amrwb --enable-libmp3lam
-libtheora --enable-libtwolame --enable-libvo-amrwbenc --enable-libilbc --enable-libgsm --ena
ble-libopus --enable-libspeex --enable-libvorbis --enable-ladspa --enable-libbs2b --enable-l
-enable-librubberband --enable-libsoxr --enable-chromaprint
  libavutil      58.  2.100 / 58.  2.100
  libavcodec     60.  3.100 / 60.  3.100
  libavformat    60.  3.100 / 60.  3.100
  libavdevice    60.  1.100 / 60.  1.100
  libavfilter     9.  3.100 /  9.  3.100
  libswscale      7.  1.100 /  7.  1.100
  libswresample   4. 10.100 /  4. 10.100
  libpostproc    57.  1.100 / 57.  1.100
Input #0, h264, from 'extract.dat':0KB vq=    0KB sq=    0B f=0/0
  Duration: N/A, bitrate: N/A
  Stream #0:0: Video: h264 (High), yuv420p(progressive), 1920x1088, 25 tbr, 1200k tbn
    nan M-V:    nan fd=   0 aq=    0KB vq=    0KB sq=    0B f=0/0
```

[그림 8] ffplay를 통해 추출한 프레임 데이터 확인

[그림 9] 식별된 책 제목과 edition 정보

[그림 10] 정상적으로 재생되는 가공 파일

또한, 특정 반복 8 bytes data를 주어진 파일에서 모두 제거하고 kmplayer를 통해 재생해본 결과 정상적으로 영상이 출력됨을 확인할 수 있었습니다.

따라서, 책 제목은 **Introduction to Spectroscopy**이며 edition은 **Fourth Edition**임이 확인되었습니다.