

2022 국민대학교 정보보안암호수학과 동아리 연합 암호경진대회

4번 문제 – 디지털 포렌식 답안

20192243 이용진, 20192233 박진철

[문제1] 덤프 뜯 'FaS_USB.img' 이미지에서 모든 파일을 추출하시오

문제와 함께 주어진 'FaS_USB.img' 이미지에서 파일들을 추출하였다.

이름	수정된 날짜	유형	크기
[1]_SEED_Algorithm_Specification_korean_M.0	2022-10-17 오후 1:42	0 파일	287KB
20220723_FaS_20192243_이용진.0	2022-10-17 오후 1:42	0 파일	877KB
FaS 동아리 가입 신청서.hwp	2022-10-17 오전 11:05	응용 프로그램	110KB
FAT32_20182222박세준.0	2022-10-17 오후 1:41	0 파일	1,063KB
Hive_Ransomware_Integrated_Decryption_Tool.0	2022-10-17 오후 1:41	0 파일	54KB
Hive_랜섬웨어_통합_복구도구_사용_매뉴얼.0	2022-10-17 오후 1:41	0 파일	785KB
Lu.0	2022-10-17 오후 1:41	0 파일	4,165KB
Mario_Voice.0	2022-10-17 오후 1:41	0 파일	9,026KB
Ragnar_Ransomware_Decryption_Tool.0	2022-10-17 오후 1:41	0 파일	5,886KB
Ragnar_랜섬웨어_복구도구_사용_매뉴얼.0	2022-10-17 오후 1:41	0 파일	468KB
뽕이.0	2022-10-17 오후 1:41	0 파일	95KB
미국_연방형사소송규칙_원문본.0	2022-10-17 오후 1:40	0 파일	2,136KB
시스템_로그_조사.0	2022-10-17 오후 1:40	0 파일	16KB
우리서버강아지는머리3개커버로스.0	2022-10-17 오후 1:40	0 파일	1,589KB
잉리_PPT.0	2022-10-17 오후 1:40	0 파일	37,956KB
제출용.0	2022-10-17 오후 1:40	0 파일	1,677KB
진짜_급한_경우에만_열기.txt	2022-10-16 오전 3:30	텍스트 문서	1KB
키.0	2022-10-17 오후 1:40	0 파일	579KB
팀워크클래스_찐찐찐막.0	2022-10-17 오후 1:40	0 파일	5,106KB
해석학에서.0	2022-10-17 오후 1:39	0 파일	18KB
해시함수_SHA256_소스코드_활용_매뉴얼.0	2022-10-17 오후 1:39	0 파일	934KB

텍스트파일, 암호화 실행파일을 제외한 총 19개의 파일 존재함을 알 수 있다.

[문제2] 추출한 모든 파일이 각각 몇 번째 섹터에 있는지 풀이과정을 상세히 작성하시오

파일 시스템은 데이터를 효과적으로 관리하기 위해 파일을 체계적으로 기록하는 방식이다.

문제에서 쓰이는 운영시스템인 windows는 주로 FAT32를 사용한다.

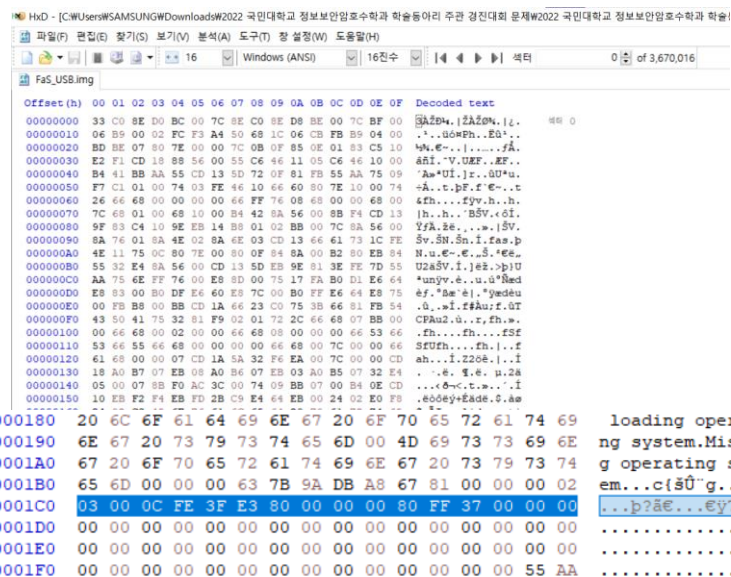
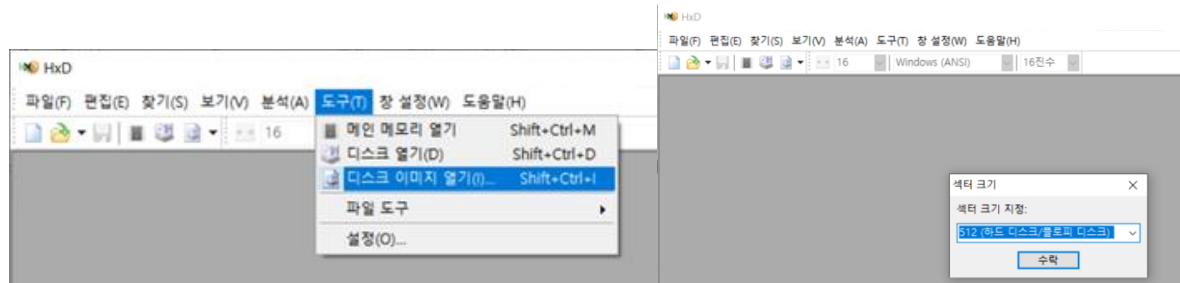
(섹터: 컴퓨터가 주소지정을 할 수 있는 최소의 단위 저장공간)

클러스터: 섹터를 여러 개로 묶은 단위(8개의 섹터가 클러스터 1개)

FAT는 예약영역, FAT영역(#1 FAT, #2 FAT), 데이터 영역으로 나누어져 있고 파일의 이름, 크기 등의 정보는 데이터 영역에 저장되어 있다.

따라서 이미지의 데이터 영역을 확인하면 파일의 정보를 알 수 있다.

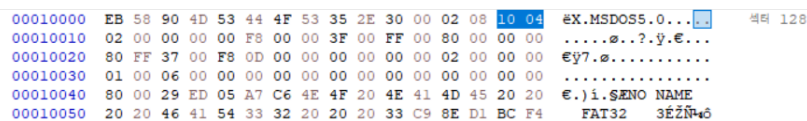
HxD를 이용하여 이미지파일의 hex값을 확인하였다.



0x1c6은 파티션 시작주소를 의미한다.

0x80=128이므로 128섹터로 이동한다.

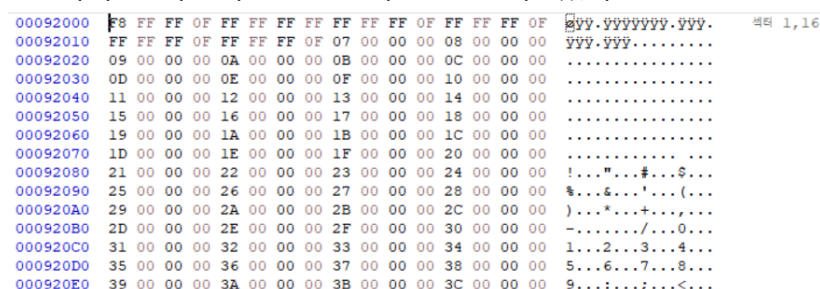
128섹터는 예약 영역의 시작부분이다.



1000E~1000F는 예약영역의 크기를 의미한다.

10 04->0x0410=1040

128+1040=1168섹터로 이동하면 FAT영역으로 갈 수 있다.



```

0000FF00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00010000 EB 58 90 4D 53 44 4F 53 35 2E 30 00 02 08 10 04 eX.MSDOS5.0.... 섹터 128
00010010 02 00 00 00 00 F8 00 00 3F 00 FF 00 80 00 00 00 .....ø..?.ÿ.ë...
00010020 80 FF 37 00 F8 0D 00 00 00 00 00 00 02 00 00 00 eÿ7.8.....
00010030 01 00 06 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
00010040 80 00 29 ED 05 A7 C6 4E 4F 20 4E 41 4D 45 20 20 e.)i.$ENO NAME
00010050 20 20 46 41 54 33 32 20 20 20 33 C9 8E D1 BC F4 FAT32 3ÉŽÑ+6

```

10024~10027은 FAT영역의 크기를 의미한다.

F8 0D 00 00->0x0DF8=3576

1168+3576=4744, 4744+3576=8320섹터로 이동하면 데이터 영역으로 갈 수 있다.

```

00410000 42 20 00 49 00 6E 00 66 00 6F 00 0F 00 72 72 00 B .I.n.f.o...rr. 섹터 8,320
00410010 6D 00 61 00 74 00 69 00 6F 00 00 00 6E 00 00 00 m.a.t.i.o...n...
00410020 01 53 00 79 00 73 00 74 00 65 00 0F 00 72 6D 00 .S.y.s.t.e...rm.
00410030 20 00 56 00 6F 00 6C 00 75 00 00 00 6D 00 65 00 .V.o.l.u...m.e.
00410040 53 59 53 54 45 4D 7E 31 20 20 20 16 00 96 F0 6E SYSTEM~1 ..-8n
00410050 51 55 51 55 00 00 F1 6E 51 55 03 00 00 00 00 00 QUQU...ñQU.....
00410060 43 6F 00 6E 00 5F 00 54 00 6F 00 0F 00 E2 6F 00 Co.n...T.o...ão.
00410070 6C 00 2E 00 30 00 00 00 FF FF 00 00 FF FF FF FF l...0...ÿÿ..ÿÿÿÿ
00410080 02 77 00 61 00 72 00 65 00 5F 00 0F 00 E2 44 00 .w.a.r.e..._..ãD.
00410090 65 00 63 00 72 00 79 00 70 00 00 00 74 00 69 00 e.c.r.y.p...t.i.
004100A0 01 52 00 61 00 67 00 6E 00 61 00 0F 00 E2 72 00 .R.a.g.n.a...â.r.
004100B0 5F 00 52 00 61 00 6E 00 73 00 00 00 6F 00 6D 00 _R.a.n.s...O.m.
004100C0 52 41 47 4E 41 52 7E 31 30 20 20 20 00 45 0E 6F RAGNAR~10 .E.o
004100D0 51 55 51 55 00 00 29 6D 51 55 06 00 29 F4 5B 00 QUQU...)mQU...)ô[.
004100E0 42 6C AD C4 B3 6C AD 5F 00 AC C0 0F 00 02 A9 C6 B1.Ã'1...-Ã...@E
004100F0 5F 00 E4 B9 74 B2 BC C5 2E 00 00 00 30 00 00 00 _..ä't...ã....0...
00410100 01 52 00 61 00 67 00 6E 00 61 00 0F 00 02 72 00 .R.a.g.n.a...r.
00410110 5F 00 9C B7 2C C1 E8 C6 B4 C5 00 00 5F 00 F5 BC _..æ'ÃæE'Ã..._..ô*
00410120 52 41 47 4E 41 52 7E 32 30 20 20 20 00 06 0F 6F RAGNAR~20 ...o
00410130 51 55 51 55 00 00 26 6D 51 55 C6 05 17 4F 07 00 QUQU...&mQU...O...
00410140 41 B1 B6 74 C7 2E 00 30 00 00 00 0F 00 F4 FF FF A±tÇ...0....ôÿÿ
00410150 FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF ÿÿÿÿÿÿÿÿÿÿ..ÿÿÿÿ

```

데이터 영역에서 모든 파일과 디렉토리는 '디렉토리 엔트리'로 표현된다.

루트 디렉토리의 구조 :

Root Directory															
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
File Name								Extension		Attr	Reserved		Create Time		
Created Data	Last accessed Date		Starting Cluster High		Last Written Time		Last Written Date		Starting Cluster Low		File Size				

이름이 길 경우(8글자 이상) LFN 엔트리에 이름이 저장되며, 루트 디렉토리에는 이름이 5글자+ ~1형태로 저장된다.

LFN 구조 :

LFN Entry																
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
Seq Num	Name 1 (Unicode)										Attr	Rev	Checksum			
Name 2 (Unicode)										Reserved		Name 3 (Unicode)				

루트 디렉토리에 작성된 클러스터 시작부분과 파일 크기를 이용해 파일의 위치를 찾을 수 있다.

따라서 각 파일의 루트 디렉토리를 찾아서 몇 번째 섹터에 있는지 확인 가능하다.

Starting Cluster High + Starting Cluster Low = x라 하자.

8320섹터가 2번 클러스터의 시작위치이기 때문에, x-2클러스터를 이동해야한다.

1개의 클러스터는 8개의 섹터로 구성되어 있기 때문에 (x-2)*8 섹터를 이동해야 한다.

따라서 8320+(x-2)*8섹터가 파일이 있는 위치이다.

파일 크기를 통해서 어느 섹터까지 있는지도 확인할 수 있다.

1) 해시함수_SHA256_소스코드_활용_매뉴얼.0

해시함->c7 d8 bd c3 c7 d4->루트 디렉토리에서 이름이 c7 d8 bd c3 c7 d4 + 7E 31로 구성

```

00410400 C7 D8 BC AE C7 D0 7E 31 30 20 20 00 8E 13 6F C0 A0 10 2E 00 8,322
00410410 51 55 51 55 00 00 FA 6C 51 55 3F 36 00 46 00 00 QUQU...QU?6.F..
00410420 42 A4 C2 54 CF DC B4 5F 00 5C D6 0F 00 17 A9 C6 B A T I U ' . \ O ... @ E
00410430 5F 00 E4 B9 74 B2 BC C5 2E 00 00 00 30 00 00 00 . a + t = i a . . . . 0 . .
00410440 01 74 D5 DC C2 68 D5 18 C2 5F 00 0F 00 17 53 00 . t O U A n O . A . . . . S .
00410450 48 00 41 00 32 00 35 00 36 00 00 00 5F 00 8C C1 H . A . 2 . 5 . 6 . . . . G A
00410460 C7 D8 BD C3 C7 D4 7E 31 30 20 20 20 00 90 13 6F C0 A0 10 2E 00 90 13 6F
00410470 51 55 51 55 00 00 F6 6C 51 55 44 36 64 97 0E 00 QUQU...QUUD6d-..

```

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 44 36->0x3644=13892

8320+(13892-2)*8=119440섹터에 위치

파일 크기: 64 97 0E 00->0x0E9764=956260

956260/512=1867.6..

119440+(1867.6...-1)=121306.6...

=>119440~121307섹터에 위치

2)해석학에서.0

해석학->c7 d8 bc ae c7 d0->루트 디렉토리에서 이름이 c7 d8 bc ae c7 d0 + 7E 31로 구성

섹터 8,322

$8320 + (12465 - 2) * 8 = 108024$ 섹터에 위치

파일 크기: 00 0A 09 00->0x090A00=592384

592384/512=1157

108024+(1157-1)=109180

=>108024~109180섹터에 위치

5)제출용.0

제출용->c1 a6 c3 e2 bf eb->루트 디렉토리 이름이 c1 a6 c3 e2 bf eb로 구성

004102D0	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
004102E0	C1 A6 C3 E2 BF EB 20 20 30 20 20 20 00 2A 13 6F	A;Aaē 0 .*.c
004102F0	51 55 51 55 00 00 0E 6D 51 55 0C 2F 12 33 1A 00	QUQU...mQU./..3..
00410300	42 2E 00 74 00 78 00 74 00 00 00 0F 00 4E FF FF	B..t.x.t.....Nyy

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 0C 2F->0x2F0C=12044

8320+(12044-2)*8=104656섹터에 위치

파일 크기: 12 33 1A 00->0x1A3312=1717010

1717010/512=3353.5...

104656+(3353.5...-1)=108008.5...

=>104656~108009섹터에 위치

6)잉리_PPT.0

잉리-> c0 d7 b8 ae

00410290	2E 00 30 00 00 00 FF FF FF FF 00 00 FF FF FF FF	..Ö...YYYY..YYYY
004102A0	C0 D7 B8 AE 5F 50 50 54 30 20 20 20 00 7D 0F 6F	A×,® PPT0 .}.c
004102B0	51 55 51 55 00 00 12 6D 51 55 FB 09 DE 0F 51 02	QUQU...mQUû.P.Q.
004102C0	41 1C C8 9C CD A9 C6 2E 00 30 00 0F 00 50 00 00	A.Ëœï®E..0...P..

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: FB 09->0x09FB=2555

8320+(2555-2)*8=28744섹터에 위치

파일 크기: DE 0F 51 02->0x02510FDE=38866910

38866910/512=75911.9...

28744+(75911.9...-1)=104654.9...

=>28744~104655섹터에 위치

7)우리서버강아지는머리3개커버로스.0

우리서->bf ec b8 ae bc ad-> 루트 디렉토리에서 이름이 bf ec b8 ae bc ad + 7E 31로 구성

00410230	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
00410240	01 B0 C6 AC B9 1C C1 84 BC 15 AC 0F 00 9B 44 C5	.°E¬¹.Á„¼.¬...>DÅ
00410250	C0 C9 94 B2 38 BA AC B9 33 00 00 00 1C AC E4 CE	ÀE"°8°¬¹3....¬äî
00410260	BF EC B8 AE BC AD 7E 31 30 20 20 20 00 71 0F 6F	¿ì,®¼.~10 .q.c
00410270	51 55 51 55 00 00 16 6D 51 55 6D 08 83 D2 18 00	QUQU...mQUm.fÔ..
00410280	41 89 C7 AC B9 5F 00 50 00 50 00 0F 00 36 54 00	AhÇ¬¹ .P.P...6T.

=> 21256~25526섹터에 위치

10)뚱이.0

뚱이-> b6 d7 c0 cc

```
00410150 FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF yyyyyyyyyyyy..yyyy
00410160 B6 D7 C0 CC 20 20 20 20 30 20 20 20 00 2D 0F 6F 1xAI 0 .-.c
00410170 51 55 51 55 00 00 22 6D 51 55 3B 06 05 7B 01 00 QUQU.."mQU;..{..
00410180 42 38 BB F8 BC 2E 00 30 00 00 00 0F 00 FA FF FF B8»e4..0.....úÿÿ
```

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 3B 06->0x063B=1595

$8320 + (1595 - 2) * 8 = 21064$ 섹터에 위치

파일 크기: 05 7B 01 00->0x017B05=97029

$97029 / 512 = 189.5...$

$21064 + (189.5... - 1) = 21252.5...$

=>21064~21253섹터에 위치

11)Ragnar_랜섬웨어_복구도구_사용_매뉴얼.0

앞의 6글자가 같은 파일이 존재하므로, LFN을 이용해 루트 디렉토리를 찾아야한다.

```
004100E0 42 6C AD C4 B3 6C AD 5F 00 AC C0 0F 00 02 A9 C6 B1.Ä'1. .-Ä...@E
004100F0 5F 00 E4 B9 74 B2 BC C5 2E 00 00 00 30 00 00 00 .ä't:4Ä....0...
00410100 01 52 00 61 00 67 00 6E 00 61 00 0F 00 02 72 00 .R.a.g.n.a....r.
00410110 5F 00 9C B7 2C C1 E8 C6 B4 C5 00 00 5F 00 F5 BC .æ',ÄèE'Ä...ô4
00410120 52 41 47 4E 41 52 7E 32 30 20 20 20 00 06 0F 6F RAGNAR~20 ...c
00410130 51 55 51 55 00 00 26 6D 51 55 C6 05 17 4F 07 00 QUQU..)mQUE..O..
00410140 41 B1 B6 74 C7 2E 00 30 00 00 00 0F 00 F4 FF FF A±tç..0.....ôÿÿ
```

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: C6 05->0x05C6=1478

$8320 + (1478 - 2) * 8 = 20128$ 섹터에 위치

파일 크기: 17 4F 07 00->0x074F17=478999

$478999 / 512 = 935.5...$

$20128 + (935.5... - 1) = 21062.5...$

=>20128~21063섹터에 위치

12)Ragnar_Ransomware_Decryption_Tool.0

파일과 비슷한 이름의 LFN을 찾으면 루트 디렉토리를 찾을 수 있다.

```
00410080 02 77 00 61 00 72 00 65 00 5F 00 0F 00 E2 44 00 .w.a.r.e._...âD.
00410090 65 00 63 00 72 00 79 00 70 00 00 00 74 00 69 00 e.c.r.y.p...t.i.
004100A0 01 52 00 61 00 67 00 6E 00 61 00 0F 00 E2 72 00 .R.a.g.n.a...âr.
004100B0 5F 00 52 00 61 00 6E 00 73 00 00 00 6F 00 6D 00 .R.a.n.s...o.m.
004100C0 52 41 47 4E 41 52 7E 31 30 20 20 20 00 45 0E 6F RAGNAR~10 .E.c
004100D0 51 55 51 55 00 00 29 6D 51 55 06 00 29 F4 5B 00 QUQU..)mQU...)ô[.
004100E0 42 6C AD C4 B3 6C AD 5F 00 AC C0 0F 00 02 A9 C6 B1.Ä'1. .-Ä...@E
```

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 06 00->0x06=6

8320+(6-2)*8=8352섹터에 위치

파일 크기: 29 F4 5B 00->0x5BF429=6026281

6026281/512=11770.0...

8352+(11770.0...-1)=20121.0...

=>8352~20122섹터에 위치

13)Mario_Voice.0

004108C0	41 4D 00 61 00 72 00 69 00 6F 00 0F 00 46 5F 00	AM.a.r.i.o...F_.
004108D0	56 00 6F 00 69 00 63 00 65 00 00 00 2E 00 30 00	V.o.i.c.e.....0.
004108E0	4D 41 52 49 4F 5F 7E 31 30 20 20 20 00 6D 14 6F	MARIO_~10 .m.d
004108F0	51 55 51 55 00 00 2D 6D 51 55 5D 3E EC 05 8D 00	QUQU...-mQU]>i...
00410900	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 5D 3E->0x3E5D=15965

8320+(15965-2)*8=136024섹터에 위치

파일 크기: EC 05 8D 00->0x8D05EC=9242092

9242092/512=18050.9...

136024+(18050.9...-1)=154073.9...

=>136024~154074섹터에 위치

14)Lu.0

00410890	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY...YYYY
004108A0	4C 55 20 20 20 20 20 20 30 20 20 20 00 22 14 6F	LU 0 ."..d
004108B0	51 55 51 55 00 00 31 6D 51 55 4B 3A 48 10 41 00	QUQU...lmQUK:H.A.
004108C0	41 4D 00 61 00 72 00 69 00 6F 00 0F 00 46 5F 00	AM.a.r.i.o...F_.

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 4B 3A->0x3A4B=14923

8320+(14923-2)*8=127688섹터에 위치

파일 크기: 48 10 41 00->0x411048=4264008

4264008/512=8328.1...

127688+(8328.1...-1)=136015.1...

=>127688~136016섹터에 위치

15)Hive_랜섬웨어_통합_복구도구_사용_매뉴얼.0

00410820	42 F5 BC 6C AD C4 B3 6C AD 5F 00 0F 00 57 AC C0	Bô4l.Ä³l. ...W-À
00410830	A9 C6 5F 00 E4 B9 74 B2 BC C5 00 00 2E 00 30 00	@E_.ä³t³Ä....0.
00410840	01 48 00 69 00 76 00 65 00 5F 00 0F 00 57 9C B7	.H.i.v.e. ...Wœ-
00410850	2C C1 E8 C6 B4 C5 5F 00 B5 D1 00 00 69 D5 5F 00	,ÄèÆ'Ä .µÑ...iÖ .
00410860	48 49 56 45 5F 7E 31 20 30 20 20 20 00 1C 14 6F	HIVE_~1 0 ...d
00410870	51 55 51 55 00 00 34 6D 51 55 86 39 87 42 0C 00	QUQU...4mQU+9+B..

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 86 39->0x3986=14726

18)20220723_FaS_20192243_이용진.0

Starting Cluster High: 00 00->0x00=0
Starting Cluster Low: 76 37->0x3776=14198
 $8320 + (14198 - 2) * 8 = 121888$ 섹터에 위치
파일 크기: F5 B0 0D 00->0x0DB0F5=897269
 $897269 / 512 = 1752.4...$
 $121888 + (1752.4... - 1) * 512 = 123639.4...$
=> 121888 ~ 123640 섹터에 위치

00410490	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
004104A0	03 63 00 61 00 74 00 69 00 6F 00 0F 00 4E 6E 00	.c.a.t.i.o...Nn
004104B0	5F 00 6B 00 6F 00 72 00 65 00 00 00 61 00 6E 00	_k.o.r.e...a.n.
004104C0	02 72 00 69 00 74 00 68 00 6D 00 0F 00 4E 5F 00	.r.i.t.h.m...N_
004104D0	53 00 70 00 65 00 63 00 69 00 00 00 66 00 69 00	S.p.e.c.i...f.i.
004104E0	01 5B 00 31 00 5D 00 5F 00 53 00 0F 00 4E 45 00	.[l.]_..S...NE.
004104F0	45 00 44 00 5F 00 41 00 6C 00 00 00 67 00 6F 00	E.D._SE1...g.o.
00410500	5F 31 5F 5F 53 45 7E 31 30 20 20 20 00 B7 13 6F	[_ SE~10 .g
00410510	51 55 51 55 00 00 47 6D 51 55 2E 37 07 7B 04 00	QUQU...GmQU.7.{..
00410520	43 30 00 00 00 FF FF FF FF FF FF FF 0F 00 DF FF FF	CO...YYYYYY..BY

Starting Cluster High: 00 00->0x00=0
Starting Cluster Low: 2E 37->0x372E=14126
 $8320+(14126-2)*8=121312$ 섹터에 위치
파일 크기: 07 7B 04 00->0x047B07=293639
 $293639/512=573.5...$
 $121312+(573.5...-1)=121884.5...$
=>121312~121885섹터에 위치

00410660	05 20 00 20 00 20 00 20 00 20 00 0F 00 A7 20 00 \$.
00410670	20 00 20 00 20 00 20 00 20 00 00 00 20 00 20 00
00410680	04 20 00 20 00 20 00 20 00 20 00 0F 00 A7 20 00 \$.
00410690	20 00 20 00 20 00 20 00 20 00 00 00 20 00 20 00
004106A0	03 20 00 20 00 20 00 20 00 20 00 0F 00 A7 20 00 \$.
004106B0	20 00 20 00 20 00 20 00 20 00 00 00 20 00 20 00
004106C0	02 1C C1 2E 00 68 00 77 00 70 00 0F 00 A7 20 00	. . Á . h . w . p . . \$.
004106D0	20 00 20 00 20 00 20 00 20 00 00 00 20 00 20 00
004106E0	01 46 00 61 00 53 00 20 00 D9 B3 0F 00 A7 44 C5	. F . a . S . . Û . . . \$ DÄ
004106F0	AC B9 20 00 00 AC 85 C7 20 00 00 00 E0 C2 AD CC	- Ç . . ä Ä . i
00410700	46 41 53 B5 BF 7E 31 20 45 58 45 20 0D 14 6F	FASpç~1 EXE . . .
00410710	51 55 51 55 00 00 B4 58 51 55 52 38 0F B4 01 00	QUQU . . XQURS . . .

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: 52 38->0x3852=14418

8320+(14418-2)*8=123648섹터에 위치

파일 크기: 0F B4 01 00->0x01B40F=111631

111631/512=218.0...

123648+(218.0...-1)=123865.0...

=>123648~123866섹터에 위치

21) 진짜_급한_경우에만_열기.txt

진짜_-> c1 f8 c2 a5 5f

```
004102D0 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF YYYYYYYYYY..YYYY
004102E0 C1 A6 C3 E2 BF EB 20 20 30 20 20 20 00 2A 13 6F Á;Ääë 0 .*o
004102F0 51 55 51 55 00 00 0E 6D 51 55 0C 2F 12 33 1A 00 QUQU...mQU./3..
00410300 42 2E 00 74 00 78 00 74 00 00 00 0F 00 4E FF FF B..t.x.t....Nÿÿ
00410310 FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF YYYYYYYYYY..YYYY
00410320 01 C4 C9 DC C9 5F 00 09 AE 5C D5 0F 00 4E 5F 00 .ÄEÜE ..@Ö.N.
00410330 BD AC B0 C6 D0 C5 CC B9 5F 00 00 00 F4 C5 30 AE ¼°EDÄî¹ ...öÄö
00410340 C1 F8 C2 A5 5F 7E 31 20 54 58 54 20 00 36 13 6F ÄöÄW ~1 TXT .6.c
00410350 51 55 51 55 00 00 D0 1B 50 55 B0 30 24 01 00 00 QUQU..Ð.PU°0$...
```

Starting Cluster High: 00 00->0x00=0

Starting Cluster Low: B0 30->0x30B0=12464

8320+(12464-2)*8=108016섹터에 위치

파일 크기: 24 01 00 00->0x0124=292

=>108016섹터에 위치

[문제3] 감염된 파일의 감염시간을 초 단위까지 작성하시오

감염이 되는 순간 파일은 수정이 된다.

따라서 파일이 수정된 시간을 알아내면 감염시간을 알 수 있다.

파일은 MS-DOS에서 사용된 시간 저장형식으로 날짜와 시간을 각각 2바이트로 저장한다.

날짜(H)								날짜(L)								시간(H)								시간(L)							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Y	Y	Y	Y	Y	Y	Y	Y	M	M	M	M	D	D	D	D	h	h	h	h	h	h	m	m	m	m	m	m	s	s	s	s

Y	1980년부터 지난 연도
M	1월=1, 2월=2, ..., 12월 =12
D	Day, 1~31일
h	Hour, 0~23시
m	Minute, 0~59분
s	Second, 2초 단위로 0~29의 값이 저장

파일 수정 시간을 2진수로 변환하여 날짜와 시간을 계산하면 수정시간이 나오게된다.

1)해시함수_SHA256_소스코드_활용_매뉴얼.0

```
00410450 48 00 41 00 32 00 35 00 36 00 00 00 5F 00 8C C1 H.A.2.S.6... .GA
00410460 C7 D8 BD C3 C7 D4 7E 31 30 20 20 20 00 90 13 6F 00~AQ0~10 ...c
00410470 51 55 51 55 00 00 F6 6C 51 55 44 36 64 97 0E 00 QUQU...61QUd6d...
```

Last Written Time: F6 6C

6C F6을 2진수로 변환->0110 1/100 111/1 0110

시간: 01101(13시), 분: 100111(39분), 초: 10110(22*2=44)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 39분 44초

2)해석학에서.0

```
004103F0 30 00 00 00 FF FF FF FF FF FF 00 00 FF FF FF FF 0...YYYYYY...YYY
00410400 C7 D8 BC AE C7 D0 7E 31 30 20 20 20 00 8E 13 6F 00~AQ0~10 ...c
00410410 51 55 51 55 00 00 FA 6C 51 55 3F 36 00 46 00 00 QUQU...61QU?6.F..
00410420 42 A4 C2 54 CF DC B4 5F 00 5C D6 0F 00 17 A9 C6 B=ATiU'_.\O...@E
```

Last Written Time: FA 6C

6C FA를 2진수로 변환->0110 1/100 111/1 1010

시간: 01101(13시), 분: 100111(39분), 초: 11010(26*2=52)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 39분 52초

3)팀팀클래스_찐찐찐막.0

```
004103B0 10 CC 10 CC 10 CC C9 B9 2E 00 00 00 30 00 00 00 .i.i.iE^....0..
004103C0 C6 C0 C6 C0 C5 AC 7E 31 30 20 20 20 00 3D 13 6F EAEAA~10 ...c
004103D0 51 55 51 55 00 00 00 6D 51 55 42 31 75 C4 4F 00 QUQU...mQUBluAO.
004103E0 41 74 D5 1D C1 59 D5 D0 C5 1C C1 0F 00 C0 2E 00 AtO.AYOD.A.A..
```

Last Written Time: 00 6D

6D 00를 2진수로 변환->0110 1/101 000/0 0000

시간: 01101(13시), 분: 101000(40분), 초: 00000(0*2=0)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 00초

4)키.0

00410370	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
00410380	C5 B0 20 20 20 20 20 20 30 20 20 00 38 13 6F	A° 0 .8.c
00410390	51 55 51 55 00 00 07 6D 51 55 B1 30 00 0A 09 00	QUQU...mQU±0....
004103A0	41 00 D3 00 D3 74 D0 98 B7 A4 C2 0F 00 89 5F 00	A.Ó.ÓtÐ~·«Ä..%_.

Last Written Time: 07 6D

6D 07를 2진수로 변환->0110 1/101 000/0 0111

시간: 01101(13시), 분: 101000(40분), 초: 00111(7*2=14)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 14초

5)제출용.0

004102D0	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
004102E0	C1 A6 C3 E2 BF EB 20 20 30 20 20 00 2A 13 6F	Ä;ÄÄzë 0 .*.c
004102F0	51 55 51 55 00 00 0E 6D 51 55 0C 2F 12 33 1A 00	QUQU...mQU./..3..

Last Written Time: 0E 6D

6D 0E를 2진수로 변환->0110 1/101 000/0 1110

시간: 01101(13시), 분: 101000(40분), 초: 01110(14*2=28)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 28초

6)잉리_PPT.0

00410290	2E 00 30 00 00 00 FF FF FF FF 00 00 FF FF FF FF	..Ö...YYYY..YYYY
004102A0	C0 D7 B8 AE 5F 50 50 54 30 20 20 00 7D 0F 6F	Ä*,@ PPT0 .}.c
004102B0	51 55 51 55 00 00 12 6D 51 55 FB 09 DE 0F 51 02	QUQU...mQUâ.ß.Q.

Last Written Time: 12 6D

6D 12를 2진수로 변환->0110 1/101 000/1 0010

시간: 01101(13시), 분: 101000(40분), 초: 10010(18*2=36)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 36초

7)우리서버강아지는머리3개커버로스.0

00410250	C0 C9 94 B2 38 BA AC B9 33 00 00 00 1C AC E4 CE	ÀÉ"°8°¬³3....¬äî
00410260	BF EC B8 AE BC AD 7E 31 30 20 20 20 00 71 0F 6F	ÿi,®¼.~10 .q.ç
00410270	51 55 51 55 00 00 16 6D 51 55 6D 08 83 D2 18 00	QUQU...mQUm.fÔ..

Last Written Time: 16 6D

6D 16를 2진수로 변환->0110 1/101 000/1 0110

시간: 01101(13시), 분: 101000(40분), 초: 10110(22*2=44)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 44초

8)시스템_로그_조사.0

004101F0	5F 00 70 C8 AC C0 2E 00 30 00 00 00 00 00 FF FF	.pE-À..0.....ÿÿ
00410200	BD C3 BD BA C5 DB 7E 31 30 20 20 20 00 52 0F 6F	ÀÀ°ÀÜ~10 .R.ç
00410210	51 55 51 55 00 00 1A 6D 51 55 69 08 00 40 00 00	QUQU...mQUI..@..
00410220	42 84 BC 5C B8 A4 C2 2E 00 30 00 0F 00 9B 00 00	B,,¼\,ÀÀ..0....>..

Last Written Time: 1A 6D

6D 1A를 2진수로 변환->0110 1/101 000/1 1010

시간: 01101(13시), 분: 101000(40분), 초: 11010(26*2=52)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 52초

9)미국_연방형사소송규칙_원문본.0

004101B0	AC C0 8C C1 A1 C1 DC AD 59 CE 00 00 5F 00 D0 C6	~ACA;AU.YI...ÐE
004101C0	B9 CC B1 B9 5F 7E 31 20 30 20 20 20 00 30 0F 6F	°I±³~1 0 .0.ç
004101D0	51 55 51 55 00 00 1C 6D 51 55 53 06 A2 5C 21 00	QUQU...mQUS.ç\!.
004101E0	41 DC C2 A4 C2 5C D1 5F 00 5C B8 0F 00 A4 F8 AD	AÜÀÀ\Ñ_.\...øø.

Last Written Time: 1C 6D

6D 1C를 2진수로 변환->0110 1/101 000/1 1100

시간: 01101(13시), 분: 101000(40분), 초: 11100(28*2=56)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 40분 56초

10)뚱이.0

00410150	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYVY
00410160	B6 D7 C0 CC 20 20 20 20 30 20 20 00 2D 0F 6F	h*AI 0 .-.c
00410170	51 55 51 55 00 00 22 6D 51 55 3B 06 05 7B 01 00	QUQU.. "mQU;...{..

Last Written Time: 22 6D

6D 22를 2진수로 변환->0110 1/101 001/0 0010

시간: 01101(13시), 분: 101001(41분), 초: 00010(2*2=4)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 41분 04초

11)Ragnar_랜섬웨어_복구도구_사용_매뉴얼.0

00410110	5F 00 9C B7 2C C1 E8 C6 B4 C5 00 00 5F 00 F5 BC	.æ.,ÀèE'Ä...ô
00410120	52 41 47 4E 41 52 7E 32 30 20 20 00 06 0F 6F	RAGNAR~20 ...c
00410130	51 55 51 55 00 00 26 6D 51 55 C6 05 17 4F 07 00	QUQU..&mQUË..O..

Last Written Time: 26 6D

6D 26를 2진수로 변환->0110 1/101 001/0 0110

시간: 01101(13시), 분: 101001(41분), 초: 00110(6*2=12)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 41분 12초

12)Ragnar_Ransomware_Decryption_Tool.0

004100B0	5F 00 52 00 61 00 6E 00 73 00 00 00 6F 00 6D 00	.R.a.n.s...o.m.
004100C0	52 41 47 4E 41 52 7E 31 30 20 20 00 45 0E 6F	RAGNAR~10 .E.c
004100D0	51 55 51 55 00 00 29 6D 51 55 06 00 29 F4 5B 00	QUQU..)mQU..)ô[.

Last Written Time: 29 6D

6D 29를 2진수로 변환->0110 1/101 001/0 1001

시간: 01101(13시), 분: 101001(41분), 초: 01001(9*2=18)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 41분 18초

13)Mario_Voice.0

004108D0	56 00 6F 00 69 00 63 00 65 00 00 00 2E 00 30 00	V.o.i.c.e.....0.
004108E0	4D 41 52 49 4F 5F 7E 31 30 20 20 00 6D 14 6F	MARIO_~10 .m.c
004108F0	51 55 51 55 00 00 2D 6D 51 55 5D 3E EC 05 8D 00	QUQU..-mQU]>i...

Last Written Time: 2D 6D

6D 2D를 2진수로 변환->0110 1/101 001/0 1101

시간: 01101(13시), 분: 101001(41분), 초: 01101(13*2=26)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=> 감염시간: 2022년 10월 17일 13시 41분 26초

14)Lu.0

00410890	FF FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	YYYYYYYYYY..YYYY
004108A0	4C 55 20 20 20 20 20 20 30 20 20 00 22 14 6F	LU 0 ."c
004108B0	51 55 51 55 00 00 31 6D 51 55 4B 3A 48 10 41 00	QUQU..lmQUK:H.A.

Last Written Time: 31 6D

6D 31를 2진수로 변환->0110 1/101 001/1 0001

시간: 01101(13시), 분: 101001(41분), 초: 10001(17*2=34)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=> 감염시간: 2022년 10월 17일 13시 41분 34초

15)Hive_랜섬웨어_통합_복구도구_사용_매뉴얼.0

00410850	2C C1 E8 C6 B4 C5 5F 00 B5 D1 00 00 69 D5 5F 00	,AëE'Ä .pN..iÖ .
00410860	48 49 56 45 5F 7E 31 20 30 20 20 00 1C 14 6F	HIVE ~1 0 ...c
00410870	51 55 51 55 00 00 34 6D 51 55 86 39 87 42 0C 00	QUQU..4mQU+9#B..

Last Written Time: 34 6D

6D 34를 2진수로 변환->0110 1/101 001/1 0100

시간: 01101(13시), 분: 101001(41분), 초: 10100(20*2=40)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=> 감염시간: 2022년 10월 17일 13시 41분 40초

16)Hive_Ransomware_Integrated_Decryption_Tool.0

004107F0	61 00 6E 00 73 00 6F 00 6D 00 00 00 77 00 61 00	a.n.s.o.m...w.a.
00410800	48 49 56 45 5F 52 7E 31 30 20 20 00 19 14 6F	HIVE_R~10 ...c
00410810	51 55 51 55 00 00 38 6D 51 55 78 39 0D D4 00 00	QUQU..8mQUx9.Ö..

Last Written Time: 38 6D

6D 38를 2진수로 변환->0110 1/101 001/1 1000

시간: 01101(13시), 분: 101001(41분), 초: 11000(24*2=48)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 41분 48초

17)FAT32_20182222박세준.0

00410750	32 00 30 00 31 00 38 00 32 00 00 00 32 00 32 00	2.0.1.8.2...2.2.
00410760	46 41 54 33 32 5F 7E 31 30 20 20 20 00 11 14 6F	FAT32_~10 ...c
00410770	51 55 51 55 00 00 3C 6D 51 55 6E 38 A4 9A 10 00	QUQU...<mQU8wš..

Last Written Time: 3C 6D

6D 3C를 2진수로 변환->0110 1/101 001/1 1100

시간: 01101(13시), 분: 101001(41분), 초: 11100(28*2=56)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 41분 56초

18)20220723_FaS_20192243_이용진.0

00410570	32 00 33 00 5F 00 46 00 61 00 00 00 53 00 5F 00	2.3. .F.a...S. .
00410580	32 30 32 32 30 37 7E 31 30 20 20 20 00 06 14 6F	202207~10 ...c
00410590	51 55 51 55 00 00 43 6D 51 55 76 37 F5 B0 0D 00	QUQU...CmQUv78°..

Last Written Time: 43 6D

6D 43를 2진수로 변환->0110 1/101 010/0 0011

시간: 01101(13시), 분: 101010(42분), 초: 00011(3*2=6)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=>감염시간: 2022년 10월 17일 13시 42분 06초

19)[1]_SEED_Algorithm_Specification_korean_M.0

004104F0	45 00 44 00 5F 00 41 00 6C 00 00 00 67 00 6F 00	E.D. .A.1...g.o.
00410500	5F 31 5F 5F 53 45 7E 31 30 20 20 20 00 B7 13 6F	1__SE~10 ...c
00410510	51 55 51 55 00 00 47 6D 51 55 2E 37 07 7B 04 00	QUQU...GmQU.7.{..

Last Written Time: 47 6D

6D 47를 2진수로 변환->0110 1/101 010/0 0111

시간: 01101(13시), 분: 101010(42분), 초: 00111(7*2=14)

Last Written Date: 51 55

55 51을 2진수로 변환->0101 010/1 010/1 0001

년: 0101010(42)->1980(기준)+42=2022년, 월: 1010(10월), 일: 10001(17일)

=> 감염시간: 2022년 10월 17일 13시 42분 14초

[문제4] 실행파일이 어떠한 방식으로 암호화하였는지 사용 알고리즘과 분석 과정을 상세히 작성하시오

디버깅 프로그램인 기드라를 이용해서 실행파일을 분석하였다.

```
2 int __cdecl _main(int _Argc, char **_Argv, char **_Env)
3
4 {
5     int local_14;
6
7     __main();
8     local_14 = 0;
9     do {
10         _menu(local_14);
11         local_14 = local_14 + 1;
12     } while( true );
13 }
```

main함수에서, 실행된 횟수를 넣는 menu함수가 반복된다는 것을 알 수 있다.

```
2 void __cdecl _menu(int param_1)
3
4 {
5     bool bVar1;
6     int iVar2;
7     size_t _NewSize;
8     undefined3 extraout_var;
9     char local_22d [500];
10    char local_39;
11    char *local_38 [4];
12    undefined *local_28;
13    undefined *local_24;
14    undefined *local_20;
15    undefined *local_1c;
16    char *local_18;
17    int local_14;
18    int local_10;
19
20    local_10 = 1;
21    local_38 [0] = "pptx";
22    local_38 [1] = &DAT_00405193;
23    local_38 [2] = &DAT_00405197;
24    local_38 [3] = &DAT_0040519b;
25    local_28 = &DAT_0040519f;
26    local_24 = &DAT_004051a3;
27    local_20 = &DAT_004051a7;
28    local_1c = &DAT_004051ab;
29    if (param_1 == 0) {
30        _puts(&DAT_004051b0);
31    }
32    do {
33        if (local_10 != 1) {
34            return;
35        }
36        _printf(&DAT_004051e8);
37        _scanf("%s", local_22d);
38        local_18 = (char *)0x0;
39        local_18 = _getExt(local_22d);
40        for (local_14 = 0; local_14 < 8; local_14 = local_14 + 1) {
41            iVar2 = _strcmp(local_18, local_38 [local_14]);
42            if (iVar2 == 0) {
43                local_10 = 0;
44                break;
45            }
46        }
47        if (local_10 == 0) {
48            _printf(&DAT_0040522d);
49            _scanf("%s", &local_39);
50            if ((local_39 == 'Y') || (local_39 == 'y')) {
51                _printf(&DAT_00405242, local_22d);
52                _NewSize = _strlen(local_22d);
53                _realloc(local_22d, _NewSize);
54                bVar1 = _mcp(local_22d, param_1);
55                local_10 = CONCAT31(extraout_var, bVar1);
56                Sleep(2000);
57                _system("cls");
58            }
59            else {
60                _puts(&DAT_00405257);
61                Sleep(2000);
62                _system("cls");
63                local_10 = 1;
64            }
65        }
66    } while( true );
67 }
```

local_38~local_1c에는 확장자명이 저장되어 있다.(pdf, png, hwp, jpg, mp3, mp4, zip)

```

if ((local_39 == 'Y') || (local_39 == 'y')) {
    _printf(&DAT_00405242,local_22d);
    _NewSize = _strlen(local_22d);
    _realloc(local_22d,_NewSize);
    bVar1 = _mcp(local_22d,param_1);
    local_10 = CONCAT31(extraout_var,bVar1);
    Sleep(2000);
    _system("cls");
}

```

위의 부분을 통해 파일 암호화를 실행하면 mcp함수를 통해 암호화가 됨을 알 수 있다.

mcp함수에는 감염시킬 파일, 실행횟수를 집어넣는다.

```

2 bool __cdecl _mcp(char *param_1,int param_2)
3
4 {
5     bool bVar1;
6     char local_120 [260];
7     void *local_1c;
8     size_t local_18;
9     FILE *local_14;
10    undefined4 local_10;
11
12    local_10 = 0;
13    local_14 = _fopen(param_1,"rb");
14    bVar1 = local_14 != (FILE *)0x0;
15    if (bVar1) {
16        _fseek(local_14,0,2);
17        local_18 = _ftell(local_14);
18        local_1c = _malloc(local_18 + 1);
19        _memset(local_1c,0,local_18 + 1);
20        _fseek(local_14,0,0);
21        _fread(local_1c,local_18,1,local_14);
22        _fclose(local_14);
23        _sie((int)local_1c,local_18,param_2);
24        _remove(param_1);
25        _cet(param_1,&DAT_00405188);
26        _rename(local_120,param_1);
27        local_14 = _fopen(param_1,"wb");
28        _fwrite(local_1c,1,local_18,local_14);
29        _fclose(local_14);
30        _free(local_1c);
31    }
32    else {
33        _printf(&DAT_00405158);
34        Sleep(2000);
35        _system("cls");
36    }
37    return !bVar1;
38}

```

mcp함수에서 파일을 분석하여 감염시킨다.

```

23    _sie((int)local_1c,local_18,param_2);
24    _remove(param_1);
25    _cet(param_1,&DAT_00405188);

```

파일과 실행된 횟수를 받아서 파일을 sie함수를 이용해 암호화하고, cet함수를 이용해 확장자명을 .0으로 바꾼다는 것을 알 수 있다.


```

1
2 void __cdecl sie(int param_1,int param_2,int param_3)
3
4 {
5     int local_10;
6     int local_c;
7     int local_8;
8
9     if (param_3 % 3 == 0) {
10         for (local_8 = 0; local_8 < param_2; local_8 = local_8 + 1) {
11             *(char *) (param_1 + local_8) = *(char *) (param_1 + local_8) + '\x01';
12         }
13     }
14     if (param_3 % 3 == 1) {
15         for (local_c = 0; local_c < param_2; local_c = local_c + 1) {
16             *(char *) (param_1 + local_c) = *(char *) (param_1 + local_c) + '\x02';
17         }
18     }
19     if (param_3 % 3 == 2) {
20         for (local_10 = 0; local_10 < param_2; local_10 = local_10 + 1) {
21             *(char *) (param_1 + local_10) = *(char *) (param_1 + local_10) + '\x03';
22         }
23     }
24     return;
25 }
26

```

파일이 들어온 순서를 알려주는 param_3로 인해 계산이 바뀐다.

param_3이 3n이라면(n은 자연수)->param_1의 요소들을 1만큼 더한다.

param_3이 3n+1이라면(n은 자연수)->param_1의 요소들을 2만큼 더한다.

param_3이 3n+2이라면(n은 자연수)->param_1의 요소들을 3만큼 더한다.

=>따라서 이 암호화 실행 알고리즘은 sie함수를 통해 파일이 들어온 순서대로 hex값을 1, 2, 3만큼 더하는 계산이 반복되는 프로그램이라는 사실을 알 수 있다.

[문제5] 분석한 내용을 바탕으로 복호틀을 만들어 암호화된 모든 파일을 복호화하고, 제작한 복호틀의 과정을 상세히 설명하시오

파일이 들어온 순서대로 파일의 hex값을 1, 2, 3 만큼 빼다면 잠겨있는 파일들을 복호화 할 수 있다.

또한 파일 시그니처를 통해 파일의 원래 확장자명을 알 수 있다.

파이썬을 이용하여 복호틀을 만들었다.

```

import sys
import os.path

def Invsie(file_hex, time):

    if time%3==0:
        if file_hex==0:
            result=255
        else:
            result=file_hex-1

    if time%3==1:
        if file_hex==0:
            result=254
        elif file_hex==1:
            result=255
        else:
            result=file_hex-2

    if time%3==2:
        if file_hex==0:
            result=253
        elif file_hex==1:
            result=254
        elif file_hex==2:
            result=255
        else:
            result=file_hex-3

    return result

```

Invsie함수는 파일의 값을 알려주는 file_hex와 몇번째로 실행되는지 알려주는 time을 받아온다.

time이 3n이라면(n은 자연수)->파일의 hex값에서 1만큼 뺀 값을 리턴한다.

time이 3n+1이라면(n은 자연수)->파일의 hex값에서 2만큼 뺀 값을 리턴한다.

time이 3n+2이라면(n은 자연수)->파일의 hex값에서 3만큼 뺀 값을 리턴한다.

```

def Invcet(f, h):
    f_name=os.path.splitext(f)

    if h[0:4]==[37, 80, 68, 70]:
        os.rename(f, f_name[0]+'pdf')

    elif h[0:8]==[137, 80, 78, 71, 13, 10, 26, 10]:
        os.rename(f, f_name[0]+'png')

    elif h[0:8]==[208, 207, 17, 224, 161, 177, 26, 225]:
        os.rename(f, f_name[0]+'hwp')

    elif h[0:3]== [255, 216, 255]:
        os.rename(f, f_name[0]+'jpg')

    elif h[0:3]==[73, 68, 51]:
        os.rename(f, f_name[0]+'mp3')

    elif h[0:8]==[0, 0, 0, 24, 102, 116, 121, 112]:
        os.rename(f, f_name[0]+'mp4')

    elif h[0:4]==[80, 75, 3, 4]:
        if h[4:8]==[20, 0, 6, 0]:
            os.rename(f, f_name[0]+'pptx')
        else:
            os.rename(f, f_name[0]+'zip')

    else:
        print("확장자 오류!")

```

Invcet함수는 파일이름(f)과 파일의 hex값(h)을 받아온다.

f_name에 파일이름과 파일확장자명을 나누어 저장한다.

파일에는 특정 확장자임을 알려주는 파일 시그니처가 존재한다

파일 시그니처는 파일의 가장 처음에 위치하는 특정 바이트들로 파일 포맷을 구분하기 위해 사용된다.

암호화 알고리즘을 통해 이 알고리즘에 쓰이는 포맷은 pdf, png, hwp, jpg, mp3, mp4, zip임을 알 수 있다.

따라서 알고리즘에 쓰인 포맷의 파일 시그니처를 찾아 파일에 맞는 확장자를 찾을 수 있다.

```

lfile=["해시함수_SHA256_소스코드_활용_매뉴얼.0",
      "해석학에서.0",
      "팀워크클래스_핀핀핀막.0",
      "키.0",
      "제출용.0",
      "잉리_PPT.0",
      "우리서버강아지는머리3개커버로스.0",
      "시스템_로그_조사.0",
      "미국_연방형사소송규칙_원문본.0",
      "똥이.0",
      "Ragnar_랜섬웨어_복구도구_사용_매뉴얼.0",
      "Ragnar_Ransomware_Decryption_Tool.0",
      "Mario_Voice.0",
      "Lu.0",
      "Hive_랜섬웨어_통합_복구도구_사용_매뉴얼.0",
      "Hive_Ransomware_Integrated_Decryption_Tool.0",
      "FAT32_20182222박세준.0",
      "20220723_FaS_20192243_이용진.0",
      "[1]_SEED_Algorithm_Specification_korean_M.0"]

D_hex=[]
for i in range(len(lfile)):
    with open(lfile[i], mode='r+b') as file:
        binary=file.read()

        for j in range(len(binary)):
            result=Invsie(binary[j], i)
            D_hex.append(result)

        file.seek(0)
        file.write(bytes(D_hex))

Invcet(lfile[i], D_hex)
D_hex=[]

```

lfile에는 감염된 순서대로 파일명을 저장한다.

lfile[i]를 열고 그것을 file이라는 이름으로 지정했다.

binary에서 파일을 읽는다.























파일의 hex값이 저장되어있는 binary를 Invsie함수에 넣고, 나온 result값을 D_hex에 저장한다.

D_hex의 값을 file에 넣어 복호화를 한다.

Invcet에 복호화한 파일과 파일 hex값을 넣어 파일에 맞는 확장자로 바꾸어준다.

이 과정을 모든 파일에 감염된 순서대로 수행하면 복호화가 완료된다.

[문제6] 복호화한 파일들과 문제의 정답들을 zip으로 압축하여 다음 email로 보내세요

 [1]_SEED_Algorithm_Specification_korean_M.pdf	2022-11-16 오전 10:34	Chrome HTML Do...	287KB
 20220723_FaS_20192243_이용진.pdf	2022-11-16 오전 10:34	Chrome HTML Do...	877KB
 FaS 동아리 가입 신청서.hwp	2022-10-17 오전 11:05	용용 프로그램	110KB
 FAT32_20182222박세준.pptx	2022-11-16 오전 10:34	Microsoft PowerP...	1,063KB
 Hive_Ransomware_Integrated_Decryption_Tool.zip	2022-11-16 오전 10:34	ALZip ZIP File	54KB
 Hive_랜섬웨어_통합_복구도구_사용_매뉴얼.pdf	2022-11-16 오전 10:34	Chrome HTML Do...	785KB
 Lu.mp4	2022-11-16 오전 10:34	MP4 - MPEG-4 동...	4,165KB
 Mario_Voice.mp4	2022-11-16 오전 10:34	MP4 - MPEG-4 동...	9,026KB
 prob4.py	2022-11-16 오전 10:30	Python File	3KB
 Ragnar_Ransomware_Decryption_Tool.zip	2022-11-16 오전 10:34	ALZip ZIP File	5,886KB
 Ragnar_랜섬웨어_복구도구_사용_매뉴얼.pdf	2022-11-16 오전 10:34	Chrome HTML Do...	468KB
 똥이.png	2022-11-16 오전 10:34	PNG 파일	95KB
 미국_연방형사소송규칙_원문본.pdf	2022-11-16 오전 10:34	Chrome HTML Do...	2,136KB
 시스템_로그_조사.hwp	2022-11-16 오전 10:33	한컴오피스 한글 2...	16KB
 우리서버강아지는머리3개커버로스.png	2022-11-16 오전 10:33	PNG 파일	1,589KB
 잉리_PPT.pptx	2022-11-16 오전 10:33	Microsoft PowerP...	37,956KB
 제출용.png	2022-11-16 오전 10:32	PNG 파일	1,677KB
 진짜_급한_경우에만_열기.txt	2022-10-16 오전 3:30	텍스트 문서	1KB
 키.hwp	2022-11-16 오전 10:32	한컴오피스 한글 2...	579KB
 팀워크클래스_핀핀핀막.pptx	2022-11-16 오전 10:32	Microsoft PowerP...	5,106KB
 해석학에서.hwp	2022-11-16 오전 10:32	한컴오피스 한글 2...	18KB
 해시함수_SHA256_소스코드_활용_매뉴얼.pdf	2022-11-16 오전 10:32	Chrome HTML Do...	934KB

만든 복호툴을 돌려 파일 복호화를 완료한다.

