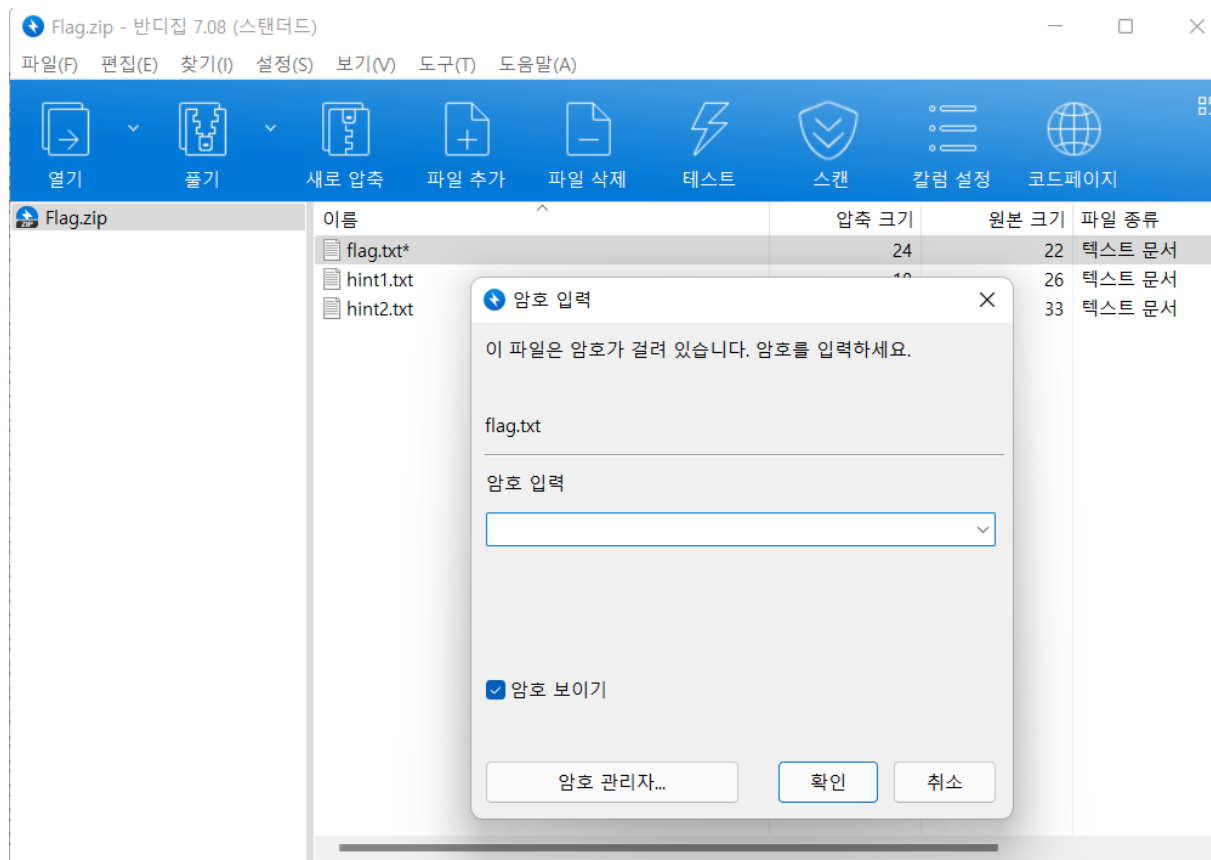
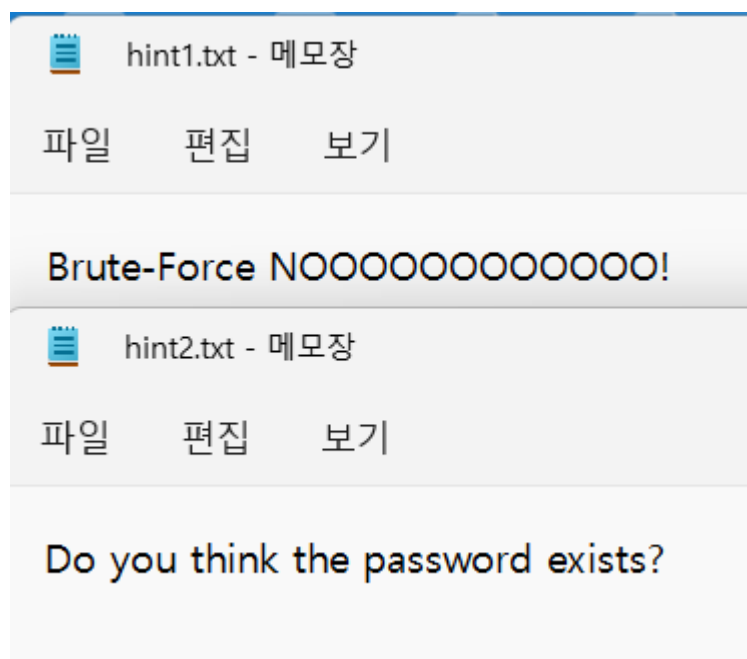


[Forensic] Secret Document



이럴수가... 압축을 푸는 것도 못하게 한다. 안에 flag파일이 있던데 그것만 확인하면 되는 문제인가 보다



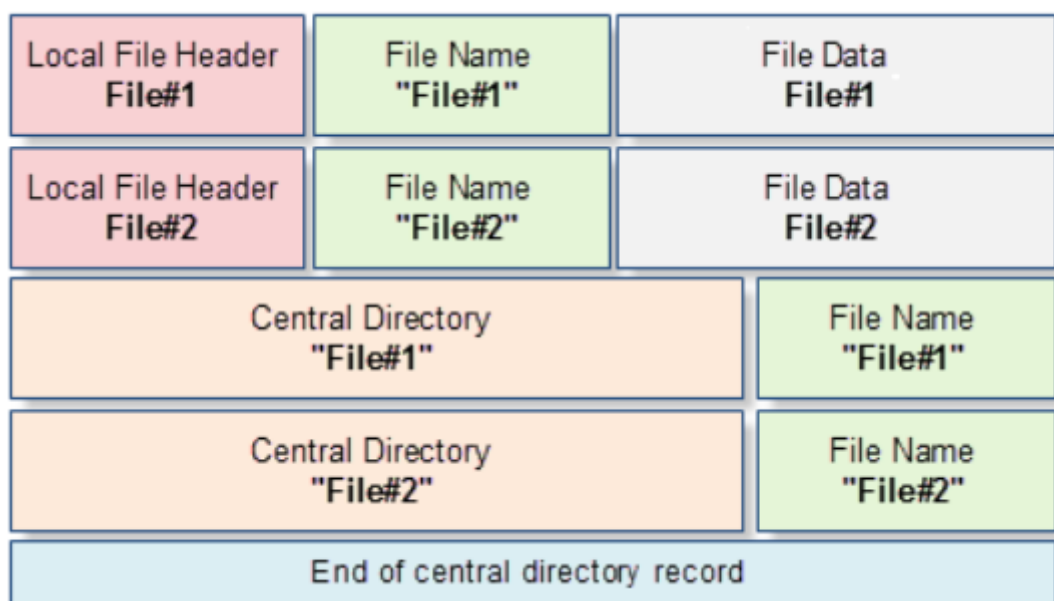
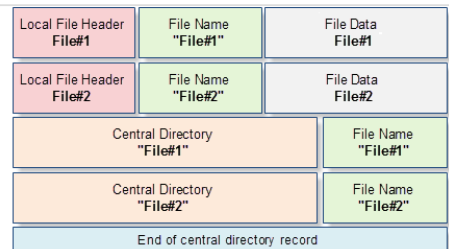
hint1,2의 내용은 이러하다. 무차별 대입은 하지 말라고 한다.

문제를 풀기위해 우선 zip 파일 구조를 먼저 알아봤다.

ZIP File Format

(1) ZIP 파일 포맷 ZIP 파일 형식은 데이터를 압축하고 보관하기 위한 파일 형식임. ZIP 파일은 하나 혹은 여러개의 파일들을 그 크기를 줄여 압축하고 하나로 묶어 저장함. ZIP 파일 형식에서는 다양한 종류

 <https://koromoon.blogspot.com/2020/02/zip-file-format.html>



zip 파일은 일반적으로 위의 구조를 따른다.

Local File Header: 압축파일에 대한 기본 정보들이 포함된다.

- 압축 전후 파일 크기
- 파일 수정시간
- CRC-32 체크섬
- 파일 이름의 지역 포인터
- 압축 해제시 필요한 아카이브 버전

File Name:

압축된 파일 이름 형식에 대한 임의 길이와 바이트 순서를 나타낸다.

File Data:

임의 길이의 바이트 배열 형태로 압축된 파일 콘텐츠

Central Directory:

Local File Header의 확장된 데이터 뷰를 제공한다.

파일 속성, 구조에 대한 로컬 기준을 가진다 ← ?

```
50 4B 03 04 14 00 00 00 08 00 7A 73 6D 4D 66 43 PK.....zsmMfC
B4 D0 18 00 00 00 16 00 00 00 08 00 00 00 66 6C 'D.....f1
61 67 2E 74 78 74 F3 48 4C CE 76 0E 71 AB F6 33 ag.txtóHLÍv.q«ö3
C8 33 8E B7 74 50 51 29 37 28 4A A9 05 00 50 4B È3Ž·tPQ)7(J@..PK
```

hdx로 열어본 파일은 이런 식으로 분리할 수 있을 것이다.

```
50 4B 03 04 14 00 00 00 08 00 7A 73 6D 4D 66 43 PK.....zsmMfC
B4 D0 18 00 00 00 16 00 00 00 08 00 00 00 66 6C 'D.....f1
61 67 2E 74 78 74 F3 48 4C CE 76 0E 71 AB F6 33 ag.txtóHLÍv.q«ö3
C8 33 8E B7 74 50 51 29 37 28 4A A9 05 00 50 4B È3Ž·tPQ)7(J@..PK
```

flag.txt의 local header file, file name, file Data

```
C8 33 8E B7 74 50 51 29 37 28 4A A9 05 00 50 4B È3Ž·tPQ)7(J@..PK
03 04 14 00 00 00 08 00 9D 73 6D 4D EB 65 B2 E0 ..smMëe²à
12 00 00 00 1A 00 00 00 09 00 00 00 68 69 6E 74 .....hint
31 2E 74 78 74 73 2A 2A 2D 49 D5 75 CB 2F 4A 4E l.txts**-IÖuË/JN
55 F0 F3 47 02 8A 00 50 4B 03 04 14 00 00 00 08 UðóG.Š.PK.....
```

hint1.txt의 local header file, file name, file Data

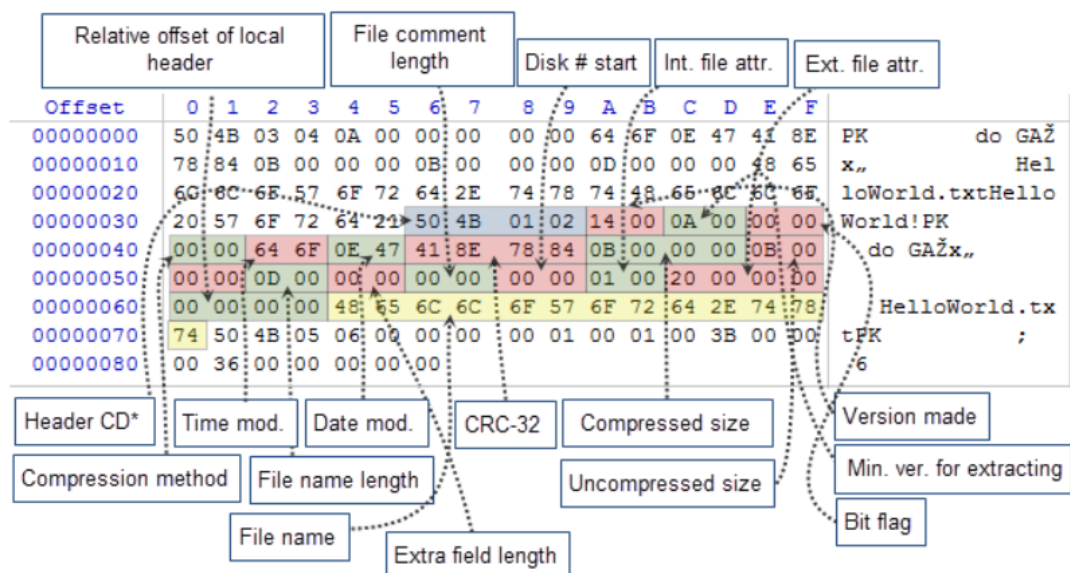
```
55 F0 F3 47 02 8A 00 50 4B 03 04 14 00 00 00 08 UðóG.Š.PK.....
00 B6 73 6D 4D DA 0D 40 E1 21 00 00 00 21 00 00 .qsmMÚ.á!....!..
00 09 00 00 00 68 69 6E 74 32 2E 74 78 74 73 C9 .....hint2.txtsË
57 A8 CC 2F 55 28 C9 C8 CC CB 06 92 A9 0A 05 89 W"i/U(ÉÈiË.'@...%
C5 C5 E5 F9 45 29 0A A9 15 99 C5 25 C5 F6 00 50 ÅÅÅùE) .@.ÅÅÅö.P
```

hint2.txt의 local header file, file name, file Data

```
C5 C5 E5 F9 45 29 0A A9 15 99 C5 25 C5 F6 00 50 ÅÅÅùE) .@.ÅÅÅö.P
4B 01 02 14 00 14 00 09 08 08 00 7A 73 6D 4D 66 K.....zsmMf
43 B4 D0 18 00 00 00 16 00 00 00 08 00 24 00 00 C'D.....$..
00 00 00 00 00 20 00 00 00 00 00 00 00 66 6C 61 .....fla
67 2E 74 78 74 0A 00 20 00 00 00 00 00 01 00 18 g.txt..
00 ED D8 7B 9E 11 7B D4 01 C4 E4 F2 15 11 7B D4 .iø{ž.{Ô.Ääò..{Ô
01 C4 E4 F2 15 11 7B D4 01 50 4B 01 02 14 00 14 .Ääò..{Ô.PK.....
```

flag.txt⁹ | central directory

(4) Central Directory 구조



Flags	2 바이트	바이트 식별자
		Bit 00 : 암호화된 파일
		Bit 01 : 압축 옵션
		Bit 02 : 압축 옵션
		Bit 03 : 데이터 기술자 (data descriptor)
		Bit 04 : 강화된 디플레이션 (deflation)
		Bit 05 : 압축된 패치 데이터
		Bit 06 : 강력한 암호화
		Bit 07-10 : 사용하지 않음
		Bit 11 : 언어 인코딩
		Bit 12 : 예약
		Bit 13 : 헤더값을 마스크
		Bit 14-15 : 예약


central directory의 bit flag 부분을 보면 암호화와 관련된 내용이 들어있는 거 같다...

```
C5 C5 E5 F9 45 29 0A A9 15 99 C5 25 C5 F6 00 50  ÅÅåùE).@.ÅÅö.P
4B 01 02 14 00 14 00 09 08 08 00 7A 73 6D 4D 66  K.....zsmMf
43 B4 D0 18 00 00 00 16 00 00 00 08 00 24 00 00  C'D.....$. .
```

bit flag 부분은 다음과 같다. 여기를 바꿔주는 문제인거 같은데 사실 잘 몰라서 그냥 암호가 걸려있지 않은 파일을 새로 만들어서 비교했다.

암호가 걸려있지 않은 파일은 bit flag가 00 00이었고 그래서 따라 바꿔주니....

```
C5 C5 E5 F9 45 29 0A A9 15 99 C5 25 C5 F6 00 50 AAaüE) .@. ¢A&Aö.P
4B 01 02 14 00 14 00 00 00 08 00 7A 73 6D 4D 66 K.....[.zsmMf
43 B4 D0 18 00 00 00 16 00 00 00 08 00 24 00 00 C'D.....$..
.. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

이름	수정한 날짜	유형
 flag.txt	2018-11-13 오후 2:27	텍스트 문서
 hint1.txt	2018-11-13 오후 2:28	텍스트 문서
 hint2.txt	2018-11-13 오후 2:29	텍스트 문서

우아악 풀어버렸다~!

정확한 원리를 찾아보니...

HEX	809
DEC	2,057
OCT	4 011
BIN	1000 0000 1001

해당 두 바이트를 16비트로 바꿔서 해석해야 하는 거였다^^ 생각해보니 프로그램에선 바이트로 보여주고 블로그에는 분형 bit라고 나와있는데 바꿔 해석할 생각을 못한 내가 바보였다^^;

해당 bit의 0번째 index가 1로 활성화 되어있기 때문에 암호화가 된 것!

해당 부분을 0으로 바꿔주면 문제는 풀린다.