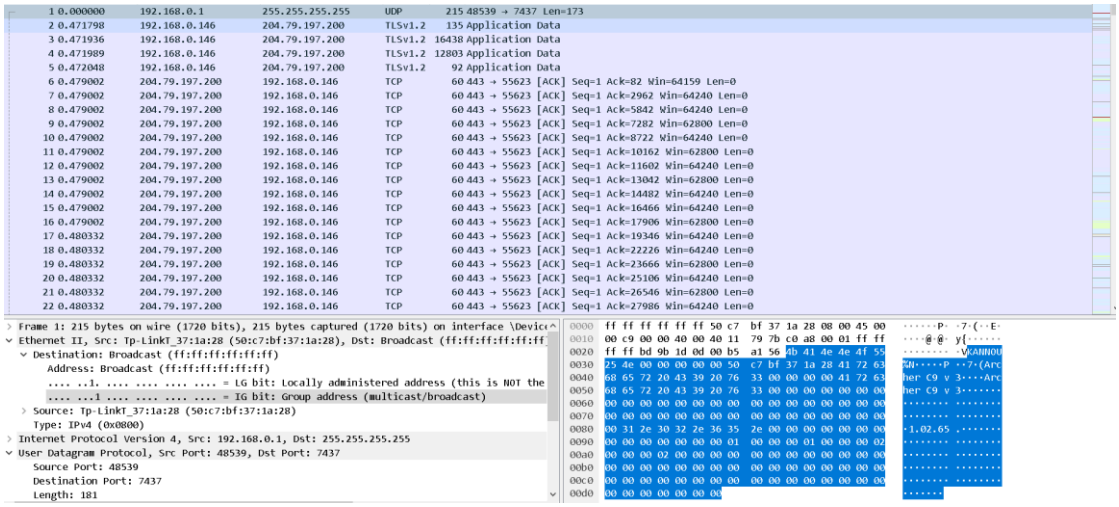
1. Suspicious Web



와이어 샤크로 파일을 열면 네트워크 패킷들이 보인다.

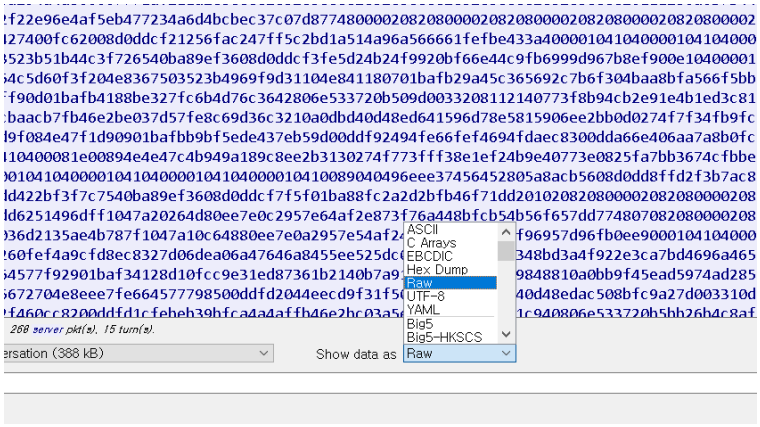
Ethernet · 11 | IPv4 · 32 | IPv6 · 1 | TCP · 27 | UDP · 51

Address A	Port A	Address B	Port B	Packets	Bytes	Stream ID	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
192.168.0.146	55688	3.38.95.245	80	321	396.256 KiB	15	47	7.064 KiB	274	389.191 KiB	3.678485	14.1847	3.983 KiB	219.499 KiB
192.168.0.146	55679	142.250.206.196	443	84	54.857 KiB	5	31	4.194 KiB	53	50.663 KiB	2.449616	0.3520	95.330 KiB	1.124 MiB
192.168.0.146	55623	204.79.197.200	443	24	30.079 KiB	0	5	28.830 KiB	19	1.249 KiB	0.471798	0.0245	9.178 MiB	407.146 KiB
192.168.0.146	55692	172.217.175.78	443	41	12.768 KiB	17	17	3.062 KiB	24	9.706 KiB	3.783532	5.1897	4.719 KiB	14.962 KiB
192.168.0.146	55678	172.217.25.173	443	30	12.593 KiB	4	12	3.698 KiB	18	8.895 KiB	2.442809	0.2448	120.881 KiB	290.728 KiB
192.168.0.146	55702	172.217.175.106	443	27	10.695 KiB	24	11	1.596 KiB	16	9.100 KiB	14.461794	0.2217	57.570 KiB	328.303 KiB
192.168.0.146	55694	64.233.188.188	5228	20	9.040 KiB	18	9	1.224 KiB	11	7.816 KiB	4.460841	0.2485	39.396 KiB	251.653 KiB
192.168.0.146	55684	172.217.161.206	443	14	9.027 KiB	11	6	923 바이트	8	8.126 KiB	2.792243	0.2646	27.249 KiB	245.654 KiB
192.168.0.146	55677	172.217.26.227	443	25	8.480 KiB	3	11	1.651 KiB	14	6.829 KiB	2.396449	0.2829	46.693 KiB	193.100 KiB
192.168.0.146	55700	172.217.175.106	443	13	7.013 KiB	21	6	923 바이트	7	6.111 KiB	9.167658	0.1959	36.809 KiB	249.567 KiB
192.168.0.146	55697	142.250.196.99	443	13	6.968 KiB	19	6	923 바이트	7	6.066 KiB	4.631694	0.1980	36.418 KiB	245.104 KiB
192.168.0.146	55681	172.217.161.202	443	13	6.960 KiB	7	6	923 바이트	7	6.059 KiB	2.614101	0.1999	36.065 KiB	242.417 KiB
192.168.0.146	55680	142.250.207.46	443	13	6.937 KiB	6	6	923 바이트	7	6.035 KiB	2.591618	0.1762	40.917 KiB	273.967 KiB
192.168.0.146	55701	211.115.106.203	80	9	1.277 KiB	22	5	711 바이트	4	597 바이트	9.214299	5.0195	1.106 KiB	951 바이트
192.168.0.146	55683	211.115.106.203	80	9	1.234 KiB	10	5	667 바이트	4	597 바이트	2.675964	5.0083	1.040 KiB	953 바이트
192.168.0.146	65444	3.114.202.64	443	12	1.004 바이트	13	6	544 바이트	6	460 바이트	3.259635	20.0542	217 바이트	183 바이트
192.168.0.146	62841	162.159.133.234	443	6	464 바이트	20	3	213 바이트	3	251 바이트	6.748288	15.9009	107 바이트	126 바이트
192.168.0.146	65400	20.198.118.190	443	3	379 바이트	23	2	151 바이트	1	228 바이트	12.058099	0.1682	7.013 KiB	10.589 KiB
173.223.227.33	443	192.168.0.146	55657	3	199 바이트	25	2	145 바이트	1	54 바이트	18.518583	0.0001		
192.168.0.146	55689	3.38.95.245	80	3	178 바이트	16	2	116 바이트	1	62 바이트	3.678829	0.0055	163.317 KiB	87.290 KiB
192.168.0.146	65331	146.66.152.39	443	2	170 바이트	26	1	110 바이트	1	60 바이트	19.662667	0.0536	16.028 KiB	8.742 KiB
211.115.106.203	80	192.168.0.146	55645	3	168 바이트	8	1	60 바이트	2	108 바이트	2.621153	6.5922	72 바이트	131 바이트
52.98.51.130	443	192.168.0.146	65089	2	148 바이트	1	1	94 바이트	1	54 바이트	1.000800	0.0001		
211.115.106.203	80	192.168.0.146	55656	2	114 바이트	12	1	60 바이트	1	54 바이트	2.982195	0.0001		
211.115.106.203	80	192.168.0.146	55658	2	114 바이트	14	1	60 바이트	1	54 바이트	3.507011	0.0001		
192.168.0.146	55627	13.89.178.26	443	1	54 바이트	2	1	54 바이트	0	0 바이트	1.314557	0.0000		
192.168.0.146	55508	211.115.106.75	80	1	54 바이트	9	1	54 바이트	0	0 바이트	2.675311	0.0000		

Statistic -> Conversations으로 들어가 TCP 통신을 패킷 Byte가 큰 순서대로 정렬하여 가장 큰 패킷을 Follow Stream을 통해 분석해 보았다.

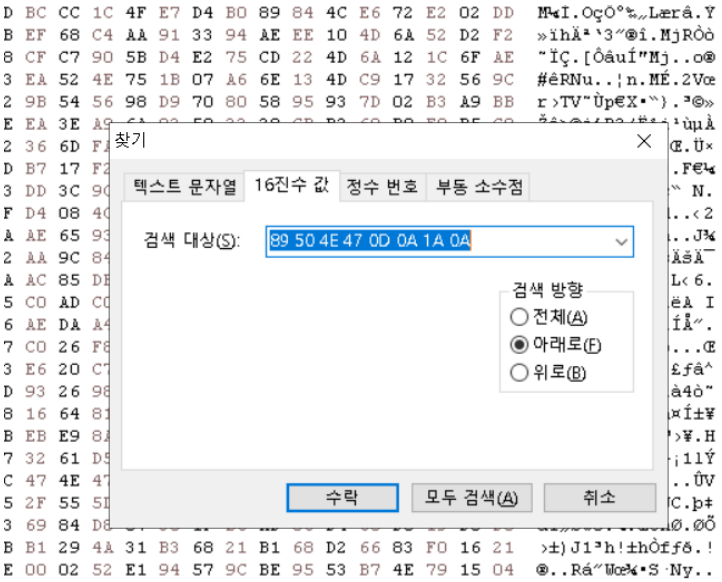
.PNG
.
...
IHDR.............8.y....sRGB..... .IDATx^...|........d.G@T.u"
...u..$n..p....am.U.U.`m..]..C;....!..=C&!d.{...@+%.s.}..I.'...@r..........s..>.@....@....@....@....@....@....@....@@... ....
....D.x.......@....@....@....@....@....@...t....@....@....@....@....@....@.....x..;....@....@....@....@....@....@....@...t....@...
....@....@....@....@....@...t....@....@....@....@....@....@....@....@.....x..;....@....@....@....@....@....@....@...t....@....@...
....@...t....@....@....@....@....@....@.....x..;....@....@....@....@....@....@...t....@....@....@....@....@....@.....x..;....
...@....@....@....@....@.....x..;....@....@....@....@....@....@...t....@....@....@....@....@....@....@.....x..;....@....@....@....@....
...@.....x..;....@....@....@....@....@....@....@...t....@....@....@....@....@....@....@.....x..;....@....@....@....@....@....@...t....@
..@....@....@....@....@....@...t....@....@....@....@....@....@....@.....x..;....@....@....@....@....@....@....@...t....@....@....@....@
..@....l.......hMi.....~5.....< ....|.
.=..H...08.P...t.2.fw..PH..a..........<.. ....(.....i..o.O%..j..p.?....o....:. ....ji..VaQ._KXT......h."a..C.....G.....5m.#
Dd.R....."..f.v...yu.........9$....)7\u..v....ffwq.!. ....(..{Dk...t.......{..t. ...5..).....T..K.....H}..`......R....
$'.;.....Q|vB..~......... t. .:PZ.t..)7..4..m...'@.. ....!%O.$.....QJ..ff...C:@....@....@....@....@....#@...Y.......{.o .]..n{.W.e.z
6....?..K$....nD..i..g.....@....@....@....@....@.......c.....0}GOy.,0.....Q.!.a.fL]`....6u.R;Ii.....A.....).\6V.........f...}.t.... ....
...A..2....l6B.nS7 .   ..2..!@w?.........<.........wH.. ....@p....1..t.s.........F...}W..i.l2....
H.d....X..n..
.t..O....|..{5..z!5....;....@....@....@....@....@ 8.%yy.....NG..   .....^.C~.....$..f..iO......n@j......=.. @w?..+.*.c....:.Xz........@
K.........t.s.....K.@w>.%.{.gL...F..m...vj(,.8.'..0..t.SH."G....W#Z...d..   t.#....@....@....@....@.....@In.7EdR.Z..V...../;z./....B.G.
....`..............F.........f.......M.%.....z &M..~.)W.J..s.jD...KV.W.wH.. ....@p.Jrs.!"...Q|vB..~..
)W.J.E{.."....}........@....@....@....@.....@qN.CJ.'..Q|vB..~.....?+. t..P.`.......'.m...vF.E^.%..1
.t.cH.:0..<..F..ee...C:@....@....@....@....@....#P.....y*8..g'.....FEw.)...A(....1..6.!@..............E...J._...%[.2..v.!. ....

내리다 보면 PNG 형식의 데이터가 있는 것을 확인할 수 있다.

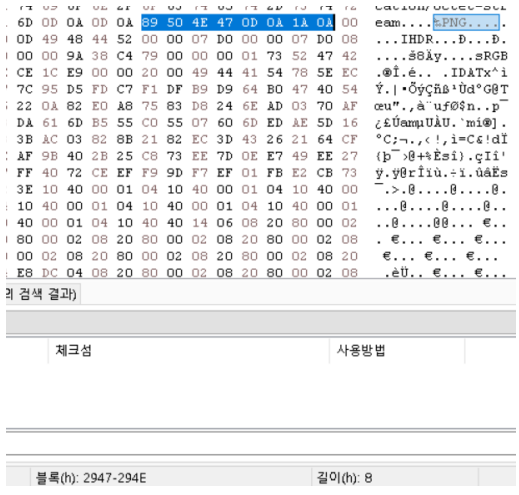


Raw 포맷으로 변경하여 모두 복사 한 뒤 HxD로 옮겼다.

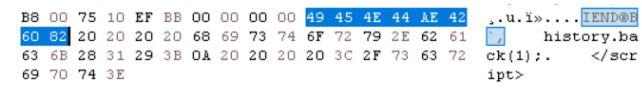
PNG파일 시그니처 : 89 50 4E 47 0D 0A 1A 0A ~ 49 45 4E 44 AE 42 60 82

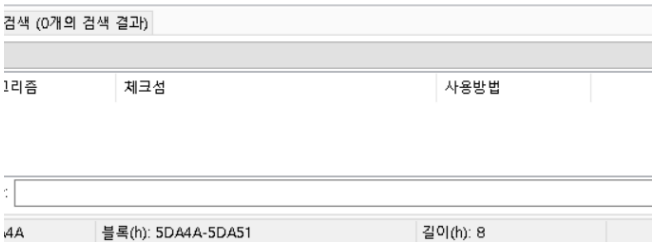


PNG파일을 찾기 위해 PNG 파일 시그니처를 찾는다.

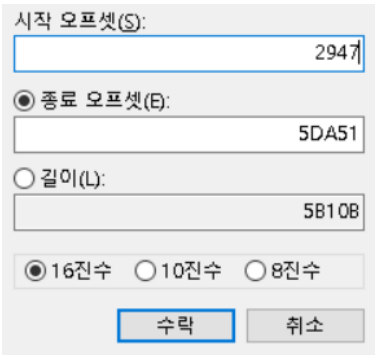


블록의 시작 위치(2947)를 기록해주고, 푸터 시그니처를 찾는다.

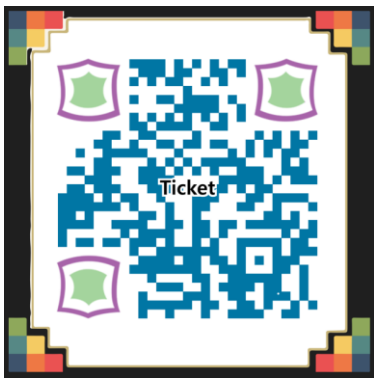




블록을 더블 클릭하여



아까 찾은 시작 위치와 푸터 시그니처의 종료 위치를 설정하고 나온 부분을 복사하여 새 파일로 붙여넣고, PNG 형식으로 저장하면

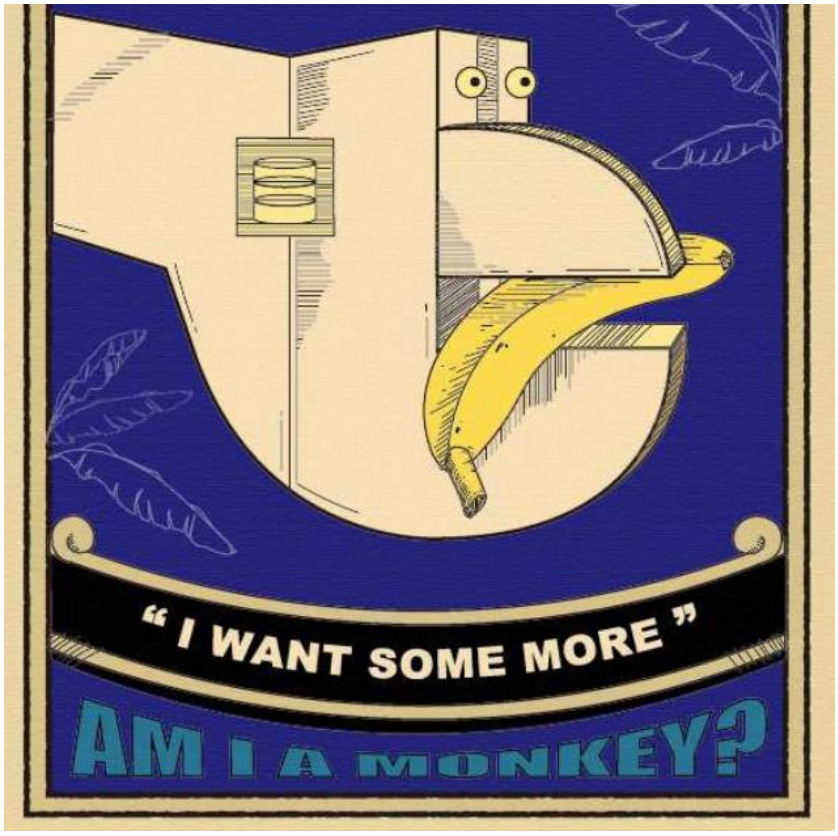


QR코드 이미지가 뜬다.

QR코드로 들어가면 링크가 flag이다.

IxC{S00_Ea2Y_sHa8K}

2. Suspicious PDF



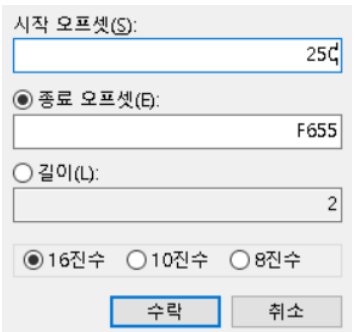
그림이 하나 있는 PDF파일이 주어진다.

```
63 6F 64 65 20 0A 3E 3E 0A 73 74 72 65 61 6D 0A   code .>>.stream
FF D8 FF DB 00 43 00 0F 0B 0C 0D 0C 0A 0F 0D 0C   ÿØÿÛ.C.........
0D 11 10 0F 12 17 26 19 17 15 15 17 2F 22 24 1C   ......&...../"$
26 38 31 3B 3A 37 31 36 35 3D 45 58 4B 3D 41 54   &81;:7165=EXK=A
42 35 36 4D 69 4E 54 5B 5E 63 64 63 3C 4A 6C 74   B56MiNT[^cdc<Jl
6C 60 73 58 61 63 5F FF DB 00 43 01 10 11 11 17   l`sXac_ÿÛ.C....
14 17 2D 19 19 2D 5F 3F 36 3F 5F 5F 5F 5F 5F 5F   ..-..-_?6?_____
```

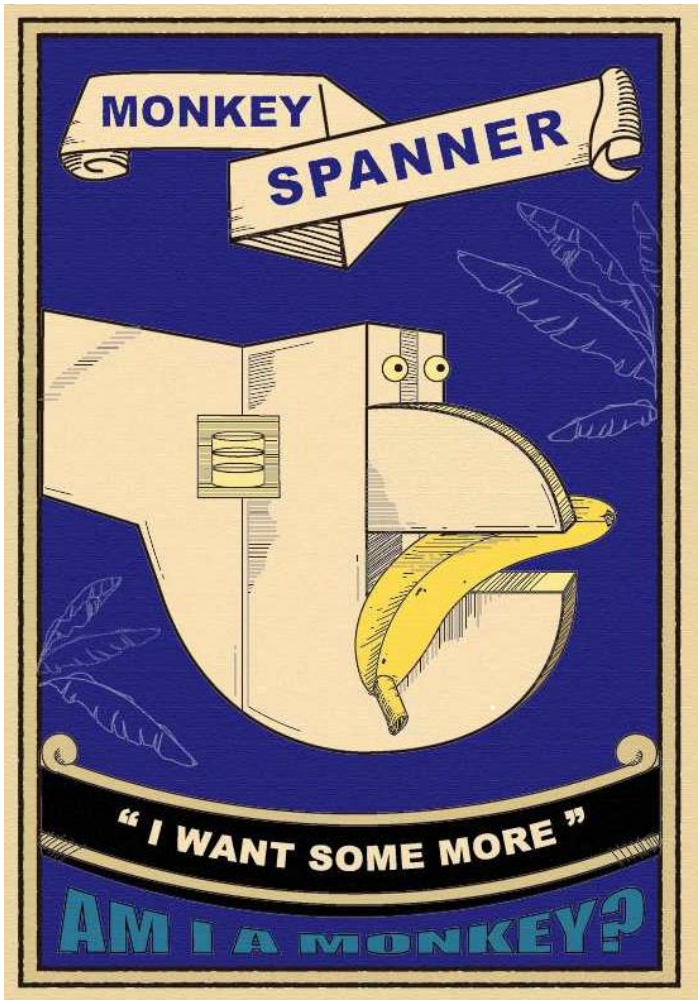
HxD로 옮기고 분석해보면 JPG 파일 시그니처가 보인다.

PDF 안에 JPG 파일을 숨겨 놓은 듯 하다.

JPG 파일 시그니처 : FF D8 FF E0 / FF D8 FF E1 / FF D8 FF DB ~ FF D9



Jpg 부분만 추출해서 jpg 형식으로 저장하면



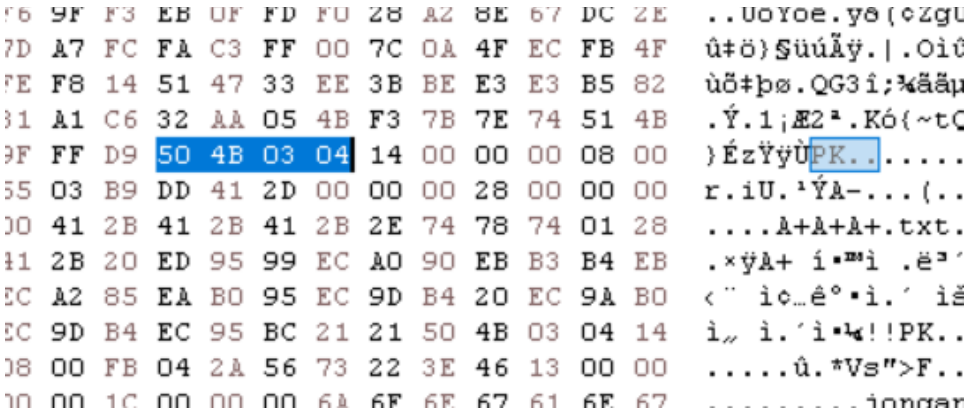
제목이 있는 완전한 그림이 뜨고 flag는 제목이다.

IxC{MONKEY_SPANNER}

3. Happy time with professor

	월	화	수	목	금
9	영화와문학 교407	인터넷과 법 교301	기초독일어 2 교101	화법과생활 교209A	6.25 전쟁사 교103A
10					
	생활일본어 1 교208	창업과법률 교101	통합영어 공2258-H.	일본문화 교204A	독일문화 산책 교105
11					
12	통합영어 공2258-H.	국가안보론 교103A	역사속호국 인물의리더십 교103A	법학개론 교201	법여성학 교207
1					
	기초독일어 2 교101	일본문화 교204A	법여성학 교207	생활일본어 1 교208	창업과법률 교101
2					
3	생활법률 교105	화법과생활 교209A	6.25 전쟁사 교103A	영화와문학 교407	인터넷과 법 교103A
4					
5	역사속호국 인물의리더십	법학개론 교201	독일문화 산책 교105	생활법률 교105	국가안보론 교103A

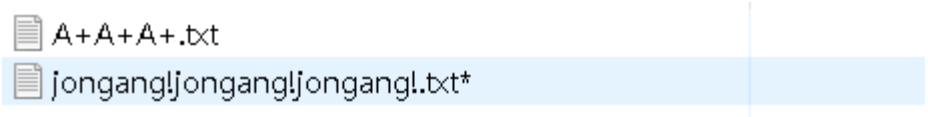
시간표가 들어있는 jpg 파일이 주어진다.



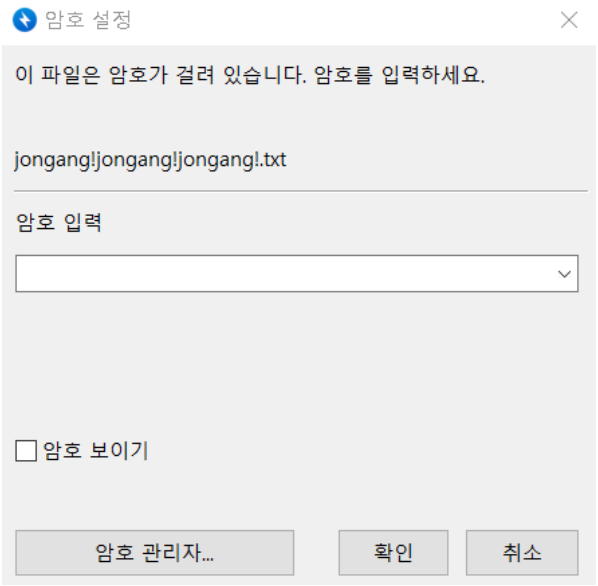
```
?6 9F F3 EB UF FD FU 28 A2 8E 67 DC 2E   ..UoYoe.y8(¢ZgU
7D A7 FC FA C3 FF 00 7C 0A 4F EC FB 4F   û‡ö}§üúÃÿ.|.Oìû
?E F8 14 51 47 33 EE 3B BE E3 E3 B5 82   ùõ‡þø.QG3î;¾ããµ
31 A1 C6 32 AA 05 4B F3 7B 7E 74 51 4B   .Ý.1¡Æ2ª.Kó{~tQ
9F FF D9 50 4B 03 04 14 00 00 00 08 00   }ÉzŸÿÙPK.......
55 03 B9 DD 41 2D 00 00 00 28 00 00 00   r.iU.¹ÝA-...(..
00 41 2B 41 2B 41 2B 2E 74 78 74 01 28   ....A+A+A+.txt.
41 2B 20 ED 95 99 EC A0 90 EB B3 B4 EB   .×ÿA+ í•™ì .ë³´
EC A2 85 EA B0 95 EC 9D B4 20 EC 9A B0   ‹¨ ì¢…ê°•ì.´ ìš
EC 9D B4 EC 95 BC 21 21 50 4B 03 04 14   ì„ ì.´ì•¼!!PK..
08 00 FB 04 2A 56 73 22 3E 46 13 00 00   .....û.*Vs">F..
00 00 1C 00 00 00 6A 6F 6E 67 61 6E 67   ..........jongar
```

HxD로 옮겨서 파일 시그니처 하나씩 다 찾다 보면 ZIP파일의 시그니처가 찾아진다.

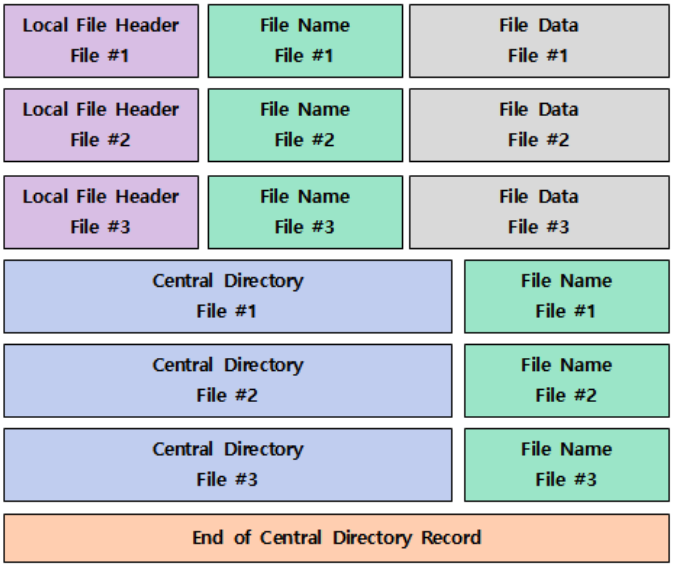
ZIP파일은 푸터 시그니처가 없어서 시그니처부터 파일의 끝까지 추출해서 zip파일로 저장하면



2개의 파일이 zip파일에 저장되어 있다.

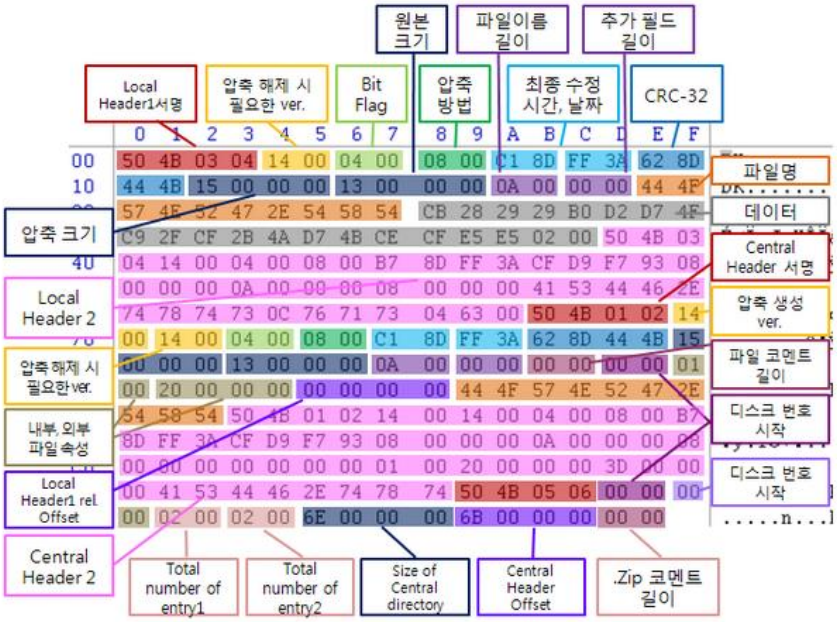


jongang!jongang!jongang!.txt 파일은 암호가 걸려있는 것 같은데 이걸 풀어야 하는 것 같다.



Zip 파일의 구조는 위와 같다.

일반적으로 Local File Header, Central Directory, End of central directory record 이렇게 3개의 파일 구조로 되어있고, zip파일의 내부 파일 하나 당 하나의 Local File Header, Central Directory 를 반드시 거쳐야 한다.

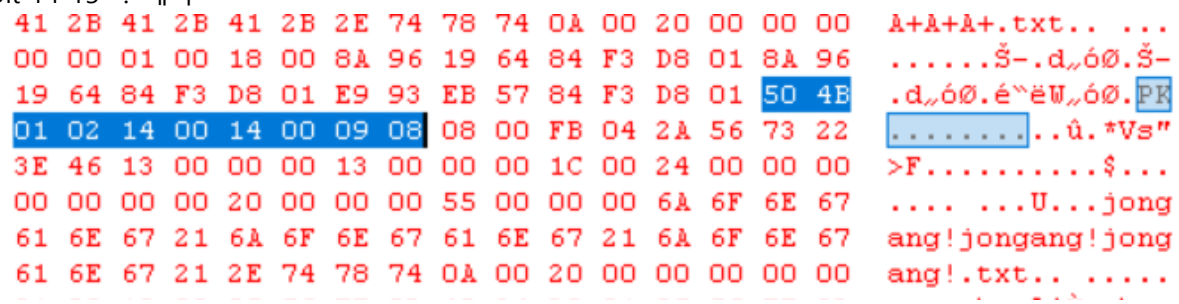


< Zip 파일의 내부 구조 분석 >

중요한 점은 Central Directory에서 파일의 암호화 정보를 저장하는 비트를 가지는 Flags 부분이 있다는 것이다.

Flags 부분은 2byte로 되어있으며 비트 별 식별자는 아래와 같다.

- 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 : 헤더 값을 마스크
- Bit 14-15 : 예약



jongang!jongang!jongang!.txt 파일의 central directory 부분을 읽어보면
flags 값은 09 08 이다.

- Signature: 50 4B 01 12
- Version: 14 00
- Vers. needed: 14 00
- Flags: 09 08

이를 리틀 엔디안 방식으로 읽으면 '08 09'가 되고, 비트로 풀면 아래와 같다.

Bit 위치	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
값	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	1

Bit 00 부분이 1 로 설정되어 있는 것으로 보아 암호화가 걸려있다는 것을 확인할 수 있고, 이 부분을 0 으로 바꿔주면 08 08 이 된다.
08 08 로 변경하여 암호화를 무력화하고 txt 파일만 추출하여 내용을 확인하면 flag 가 들어있다.

lxC{I_WANG_JONGANG}

참고 사이트 : https://bing-su-b.tistory.com/96