PNG

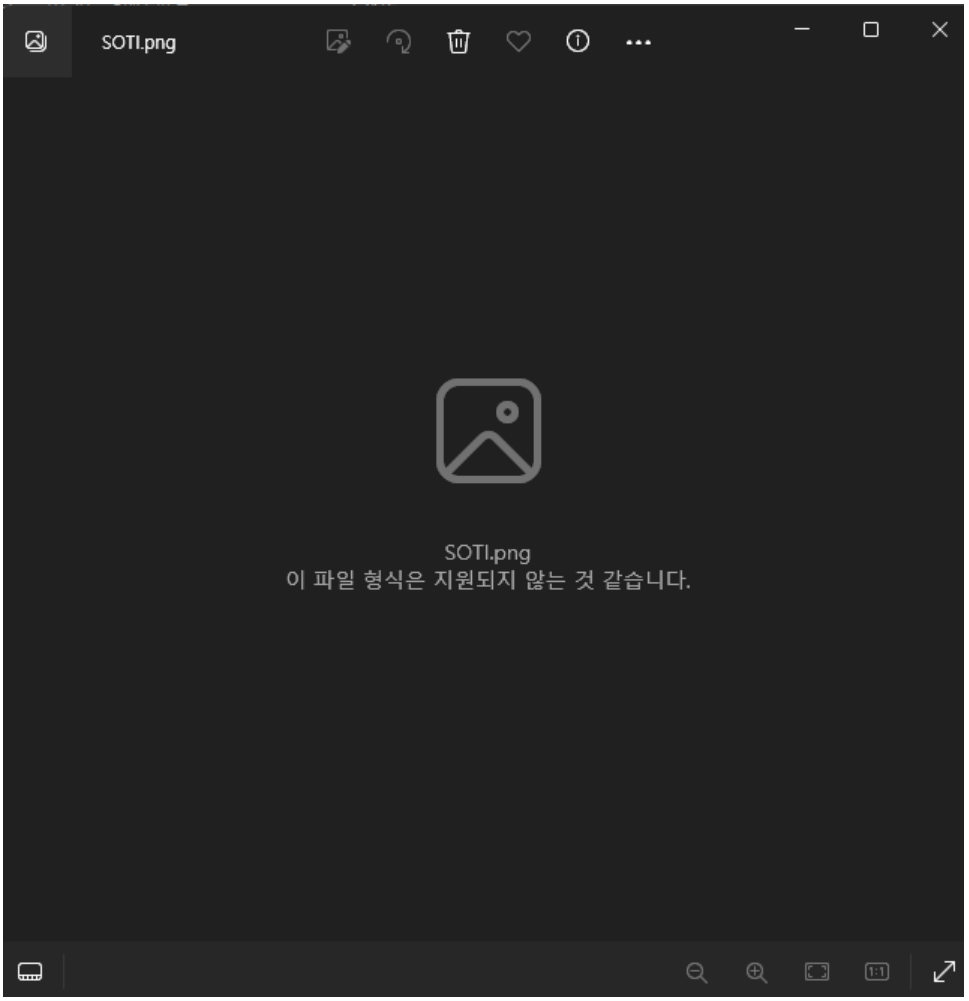
문제 분류	forensic
poc 작성자	김재환
문제 개발자	김재환

1. 문제

문제를 읽어보면 `PNG 파일을 잘 아는가?` 에 대해서 물어보고 있습니다.

- ~~어서 이 건방진 녀석을 혼내주러 갑시다.~~

soti.png 파일을 다운로드해서 보면, 파일이 깨져 보이지 않는 것을 볼 수 있습니다.



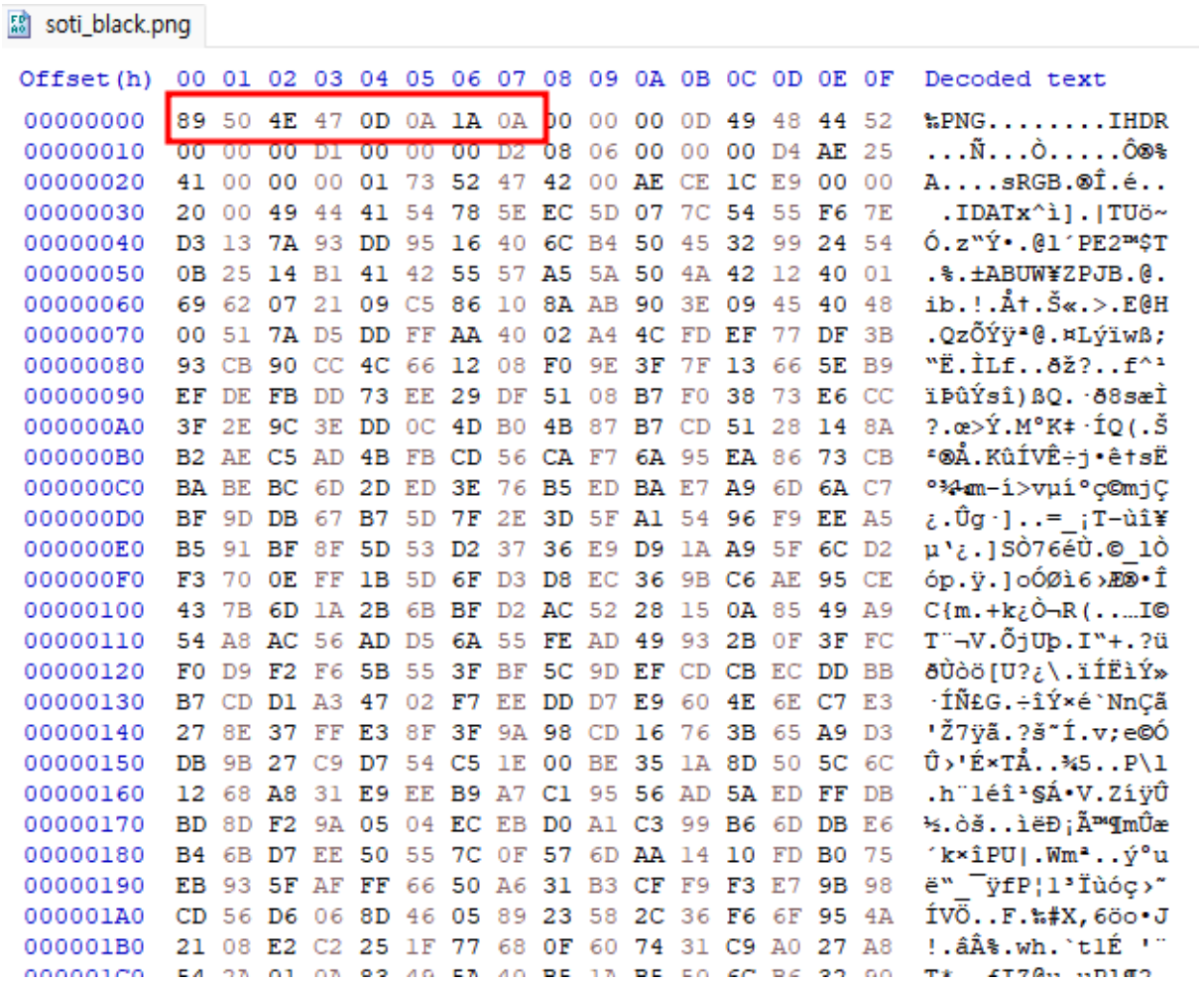
- 무언가 복구를 해야 할 것 같습니다.

2. 풀이

2.1 PNG 헤더 복구

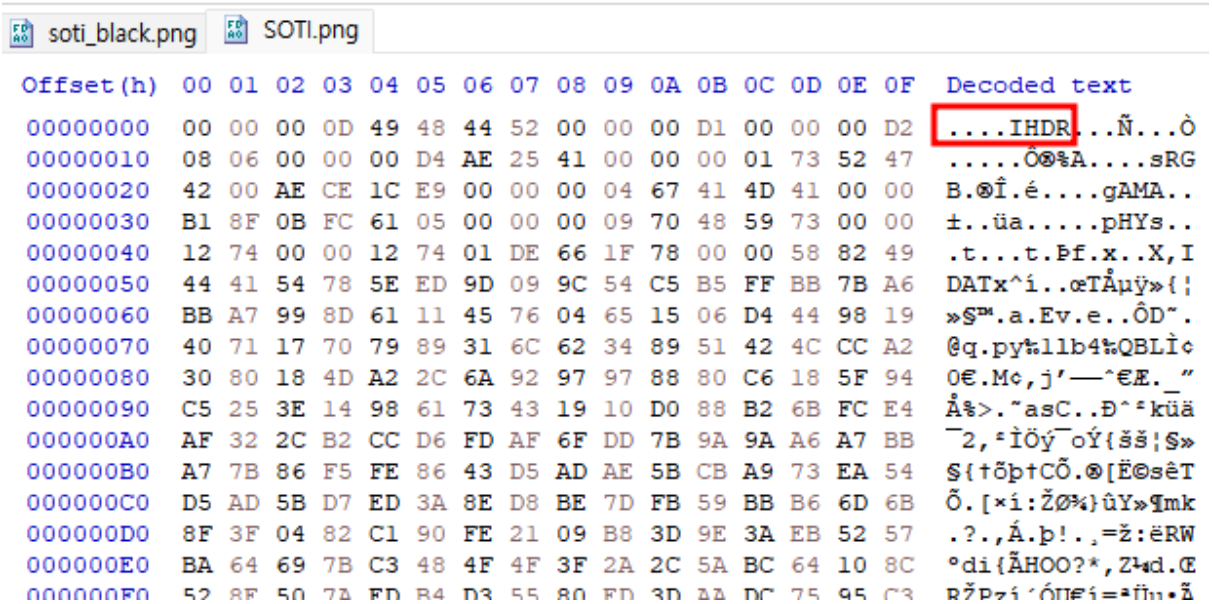
모든 파일에는 `시그니처` 라는 것이 존재합니다. 파일 최상단 헤더에 위치하며, 자신을 어떤 형태로 분석해야 할지 표시하는 문구라고 생각하면 됩니다.

일반적인 PNG 파일을 HxD를 이용하여, HEX값으로 열어 보겠습니다.



- `89 50 4E 0D 0A 1A 0A` 라는 헤더 값이 존재합니다.
- 우측에 디코드 된 문자열을 보면, 헤더 부분이 존재하는 것을 볼 수 있습니다.
- 그 뒤는 파일의 넓이, 포멧 형식 등 다양한 정보가 저장되어 있습니다.

그렇다면 우리가 분석해야 할 파일인 `soti.png`는 어떨까요?



- 이런... PNG 부분이 없어져 있군요

> 💡 근데 이게 PNG파일인줄 어떻게 아나요??

좋은 질문입니다. 이는 시그니처 뿐만이 아니라 파일을 구분하는 `청크`를 보면 됩니다

청크는 반드시 필요한 청크가 있고, 그 이외에는 따로 요구되거나, 특정 데이터를 저장하는데 사용됩니다.

PNG에서 반드시 포함되어야 하는 청크는 다음과 같습니다.

- `IHDR`
- `IDAT`
- `IEND`

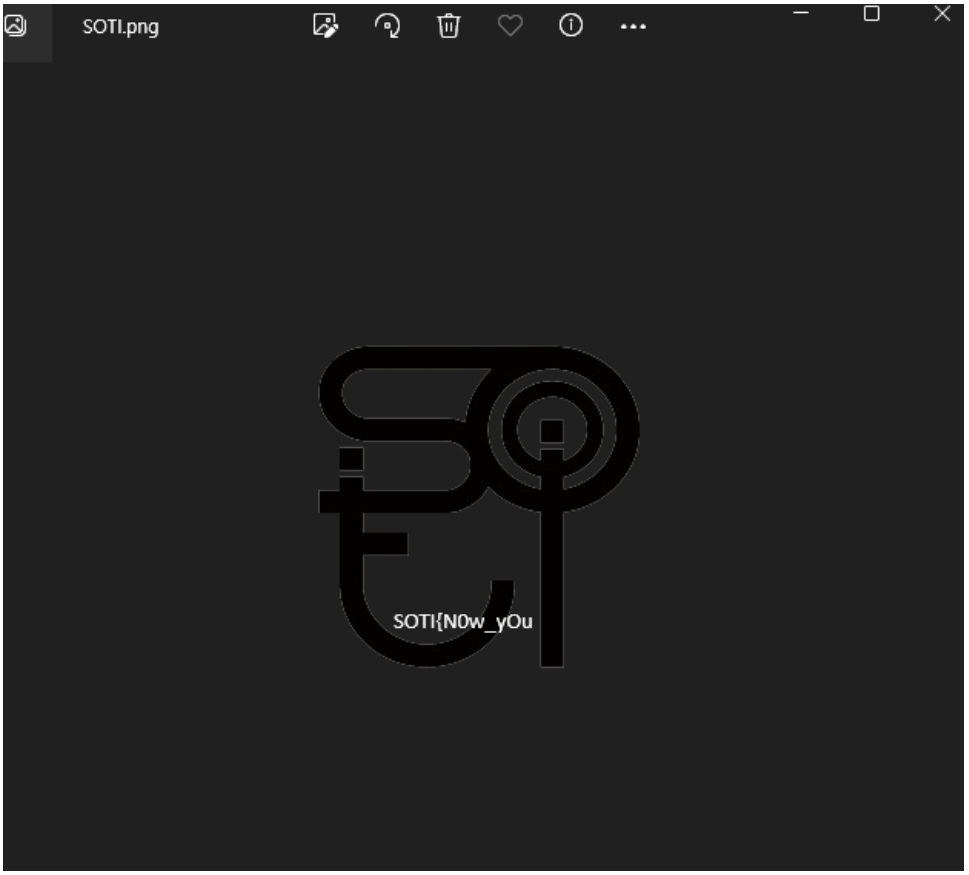
따라서 그림에 `IHDR`이 존재하기 때문에, PNG 파일임을 알 수 있습니다.

복구

PNG 헤더를 표기하는 시그니처를 손상된 파일에 덧붙여 보겠습니다.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	89	50	4E	47	0D	0A	1A	0A	00	00	00	0D	49	48	44	52	%PNG.....IHDR
00000010	00	00	00	D1	00	00	00	D2	08	06	00	00	00	D4	AE	25	...Ñ...Ò.....Ô®%
00000020	41	00	00	00	01	73	52	47	42	00	AE	CE	1C	E9	00	00	A....sRGB.®Î.é..
00000030	00	04	67	41	4D	41	00	00	B1	8F	0B	FC	61	05	00	00	..gAMA..±..üa...
00000040	00	09	70	48	59	73	00	00	12	74	00	00	12	74	01	DE	..pHYs...t...t.Þ
00000050	66	1F	78	00	00	58	82	49	44	41	54	78	5E	ED	9D	09	f.x..X,IDATx^í..
00000060	9C	54	C5	B5	FF	BB	7B	A6	BB	A7	99	8D	61	11	45	76	œTÅµÿ»{¦»§™.a.Ev
00000070	04	65	15	06	D4	44	98	19	40	71	17	70	79	89	31	6C	.e..ÔD˜.@q.py‰1l
00000080	62	34	89	51	42	4C	CC	A2	30	80	18	4D	A2	2C	6A	92	b4‰QBLÌ¢0€.M¢,j'
00000090	97	97	88	80	C6	18	5F	94	C5	25	3E	14	98	61	73	43	——ˆ€Æ._”Å%>.˜asC
000000A0	19	10	D0	88	B2	6B	FC	E4	AF	32	2C	B2	CC	D6	FD	AF	..Ðˆ²küä¯2,²ÌÖý¯
000000B0	6F	DD	7B	9A	9A	A6	A7	BB	A7	7B	86	F5	FE	86	43	D5	oÝ{šš¦§»§{†õþ†CÕ
000000C0	AD	AE	5B	CB	A9	73	EA	54	D5	AD	5B	D7	ED	3A	8E	D8	.®[Ë©sêTÕ.[×í:ŽØ

오~ 복구된 파일을 볼 수 있습니다.



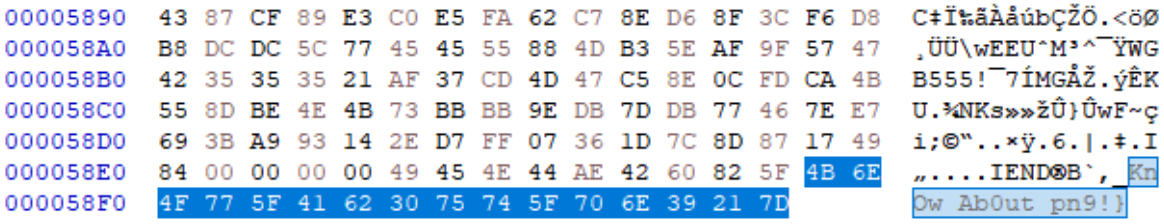
- 근데... FLAG가 반쪽이 잘려있습니다. 나머지 어딘가에 FLAG가 숨겨져 있는 듯 합니다.

2.2 반쪽짜리 FLAG 찾기(footer)

앞서 PNG 파일에는 여러가지 `청크` 가 존재한다고 했습니다. 여기서 `IEND` 는 PNG 데이터의 끝을 알리는 부분으로, 해당 부분에 데이터를 숨기는 `스테가노그래피` 기법이 존재합니다.

- 해당 섹션에 데이터를 숨기게 되면, PNG는 이를 표시하지 않기 때문입니다.

아니! 근데 해당 청크 이후에 무슨 데이터가 있네요



- 이로써 FLAG를 모두 찾게 되었습니다.