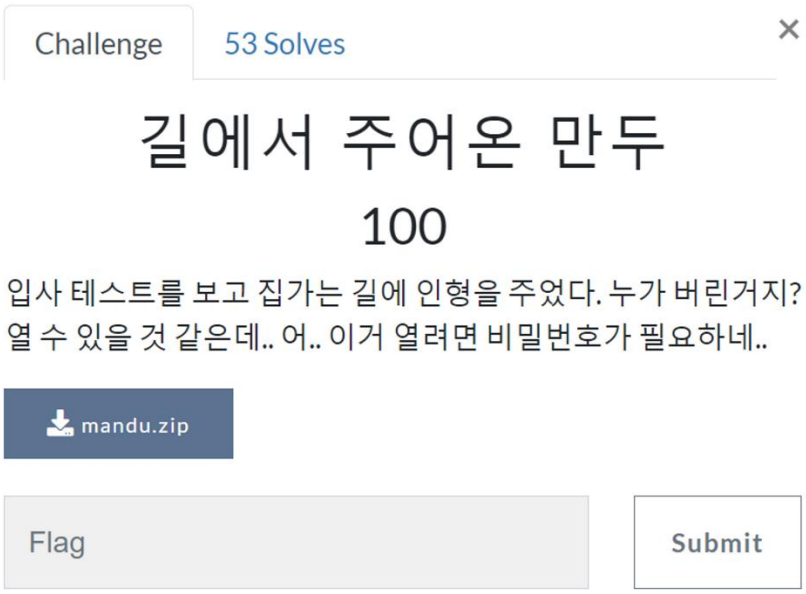
[N0Named] 길에서 주워온 만두 Writeup



문제는 다음과 같다. 일단 Openstego를 사용하는 문제이고, 문제에서 '열려면 비밀번호가 필요하다' 고 하였으므로 주어진 파일에서 비밀번호를 알아낸 후 openstego에서 그 비밀번호를 사용해 extract data를 하여 풀이하는 방법일 듯하다.

문제에서 제공하는 big.png 이미지는 이것이다. 일단 이 이미지를 헥스 에디터에 넣어보았다.

big.png

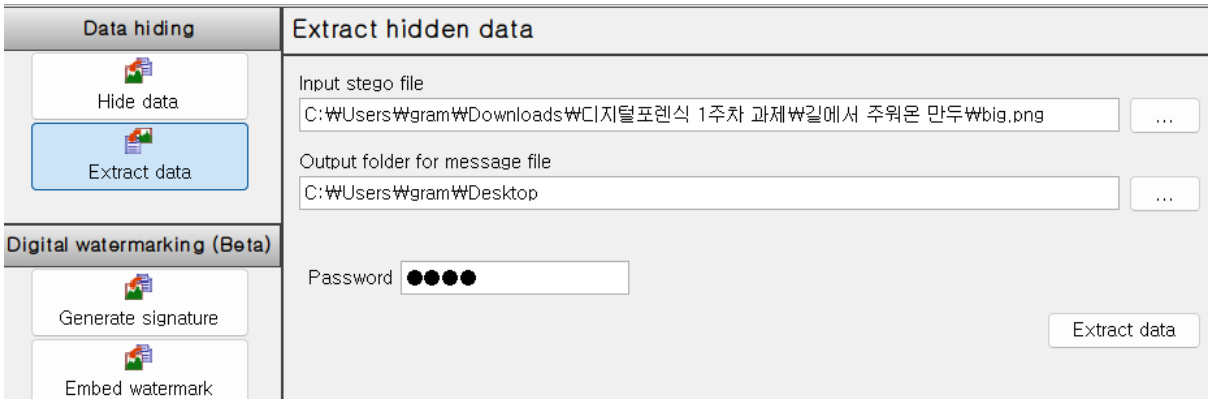
```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text
00000000   89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52  %PNG........IHDR
00000010   00 00 03 AB 00 00 03 E1 08 02 00 00 00 90 43 AE  ...«...á......C®
00000020   D8 00 00 80 00 49 44 41 54 78 DA 7C 9D 0B 92 5C  Ø..€.IDATxÚ|..'\
```

일단 헤더 시그니처는 제대로 들어가 있다. 그러면 푸터 시그니처를 확인해보자.

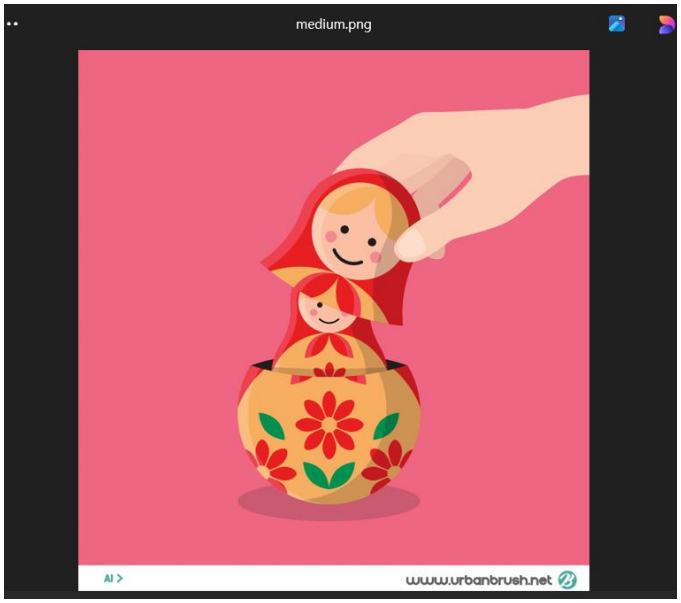
```
00090060   EA 63 A9 B8 B1 BD E1 00 00 00 00 49 45 4E 44 AE  êc©¸±½á....IEND®
00090070   42 60 82 50 41 53 53 3A 31 32 33 34              B`,PASS:1234
```

푸터 시그니처를 확인해보니 여기에 비밀번호가 숨겨져 있음을 확인할 수 있었다.

그럼 이제 이 비밀번호를 사용해서 openstego로 데이터를 추출해보자.



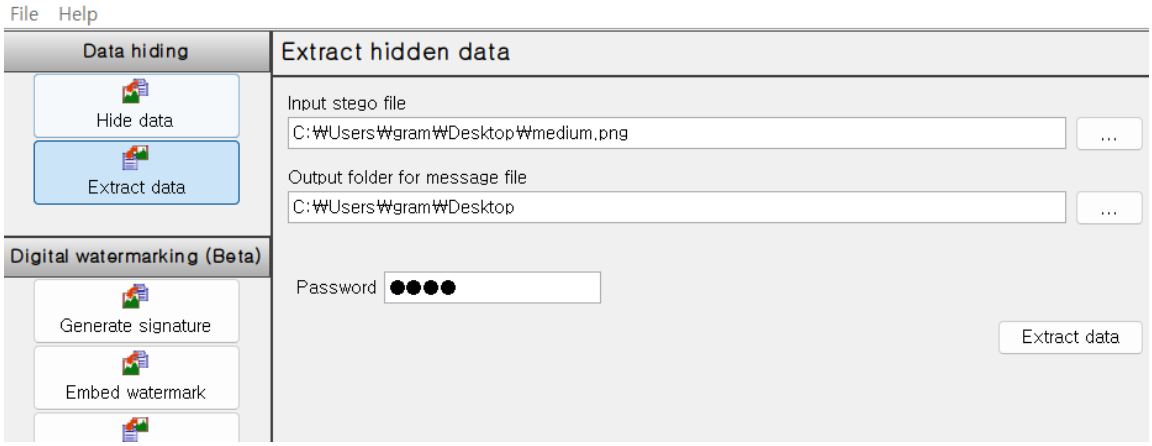
이렇게 해서 extract data를 한다.



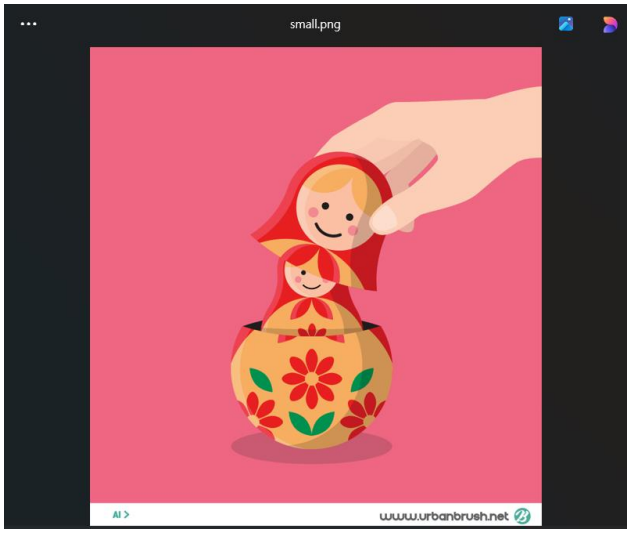
그러면 medium.png 파일이 추출된다.

문제에서 제공한 이미지에 있는 그림은 마트료시카이다. 이 문제도 마트료시카와 같이 계속해서 파일에서 데이터를 추출해내는 과정을 반복하여 가장 안쪽에 있는 플래그를 찾아내는 형식이라고 추측하였다.

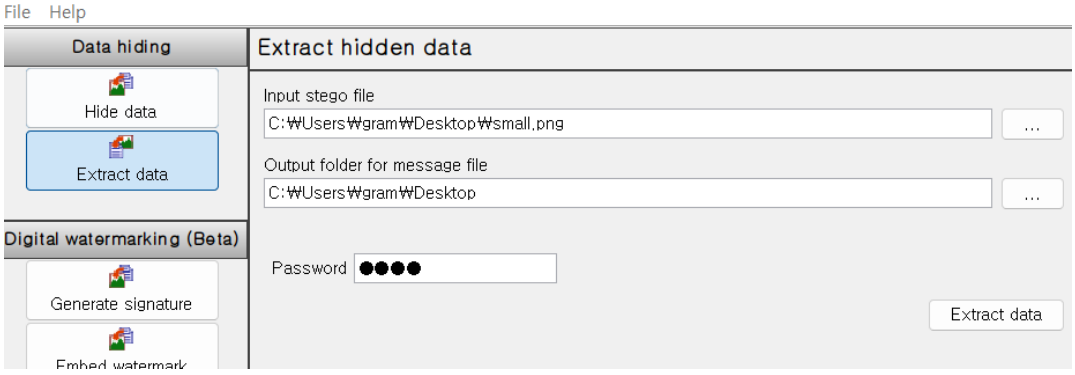
그럼 이제 medium.png 파일을 다시 openstego에 넣고 추출해보겠다. 혹시 몰라 이 파일 또한 헥스 에디터에 넣어보았지만 유의미한 값이 보이지 않아 비밀번호는 1234로 통일했다.

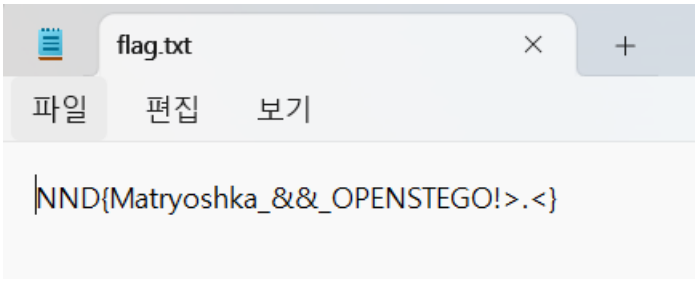


역시 small.png 파일이 추출된다.



다시 small.png 파일을 openstego에 넣고 추출한다.





이번에는 flag.txt 파일이 추출되었다. 이렇게 플래그를 확인할 수 있다.