20191235 이형규 2주차 CTF문제 풀이 보고서

콩고 문제

Q. 패스워드를 찾아라!

A.

문제를 보면 Password is ?이라는 문구와 함께 빠르게 알파벳이 지나가게 된다. 처음에는 소스 페이지에 저장된 값들이 지나가는 것으로 생각하였으나 후에 서버에 저장된 값들을 받아온다는 것을 알아낸 뒤에 네트워크를 확인해본 결과 받아오는 값을 확인할 수 있었습니다.

답은 FLAG{a7981201c48d0ece288afd01ca43c55b} 임을 확인할 수 있습니다.

말리 문제

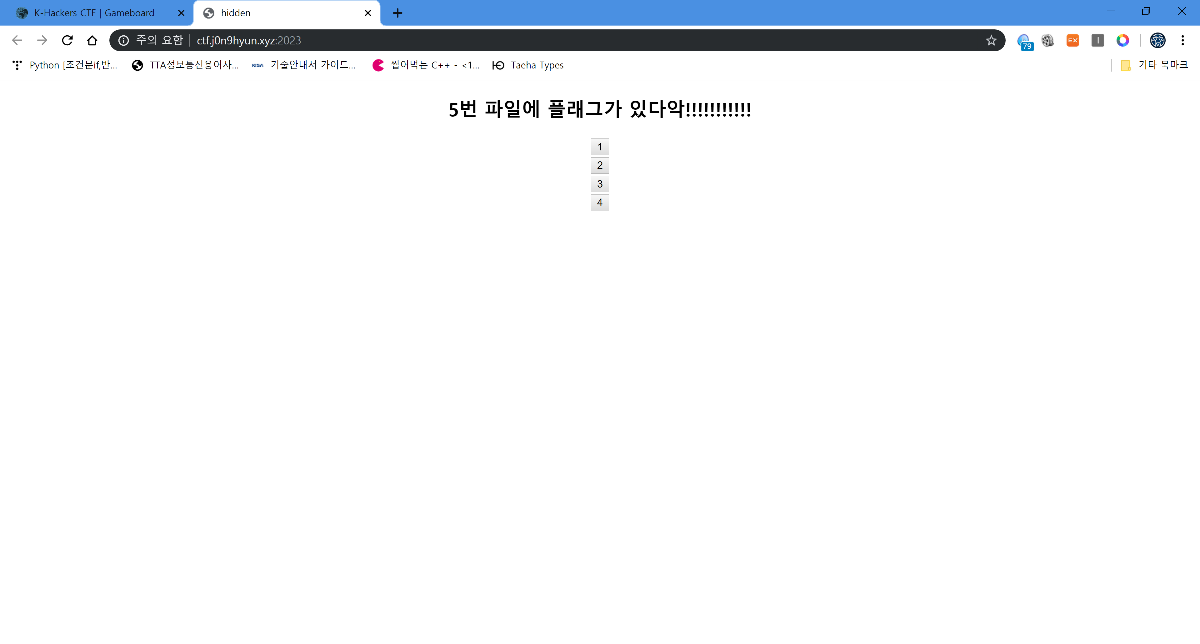
Q. 특정 소프트웨어를 실행할 때 자동으로 활성화되는 광고 프로그램으로 이를 통해 악성코드가 유포되거나 사용자의 동의 없이 인터넷 기록을 수집할 수 있는 것은?

A. 답은 애드웨어로 이를 base64로 인코딩한 값이 정답이다.

몽골 문제

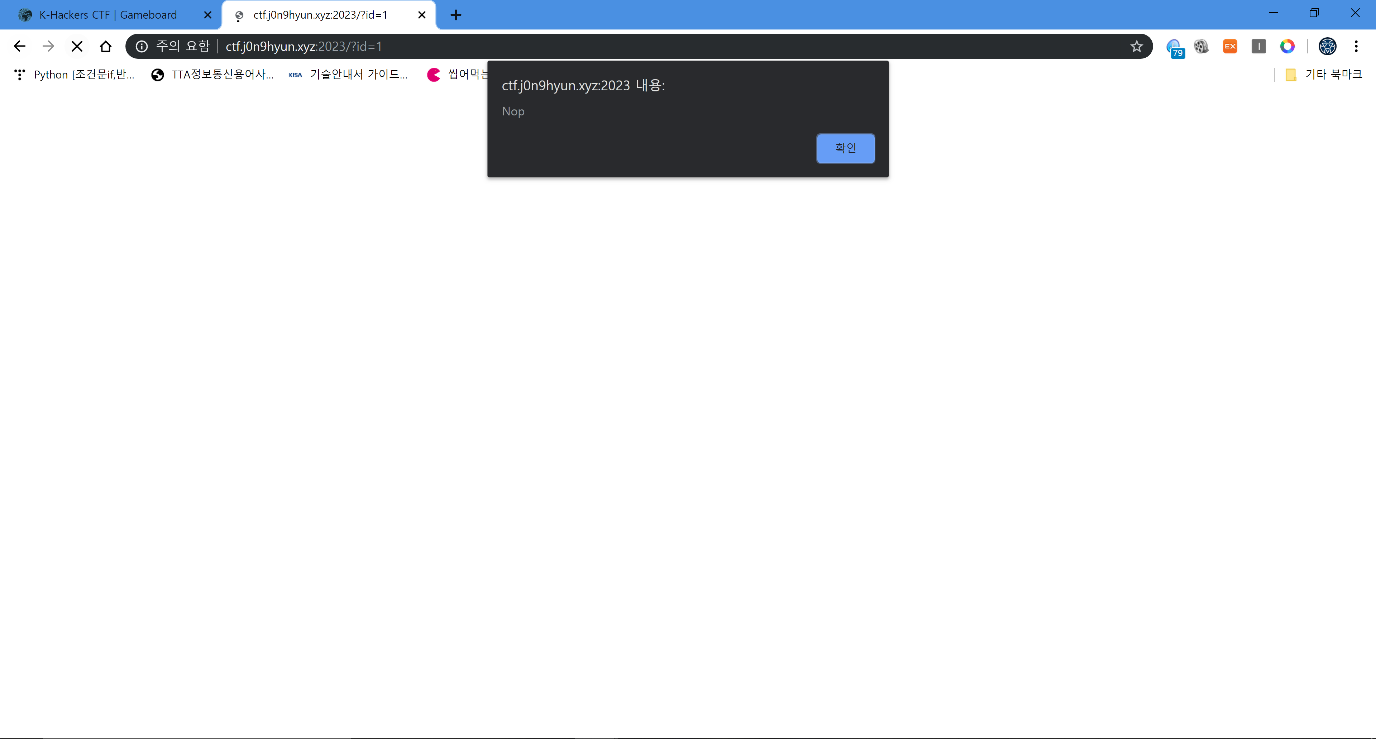
Q. 플래그를 찾아라.

A

. 

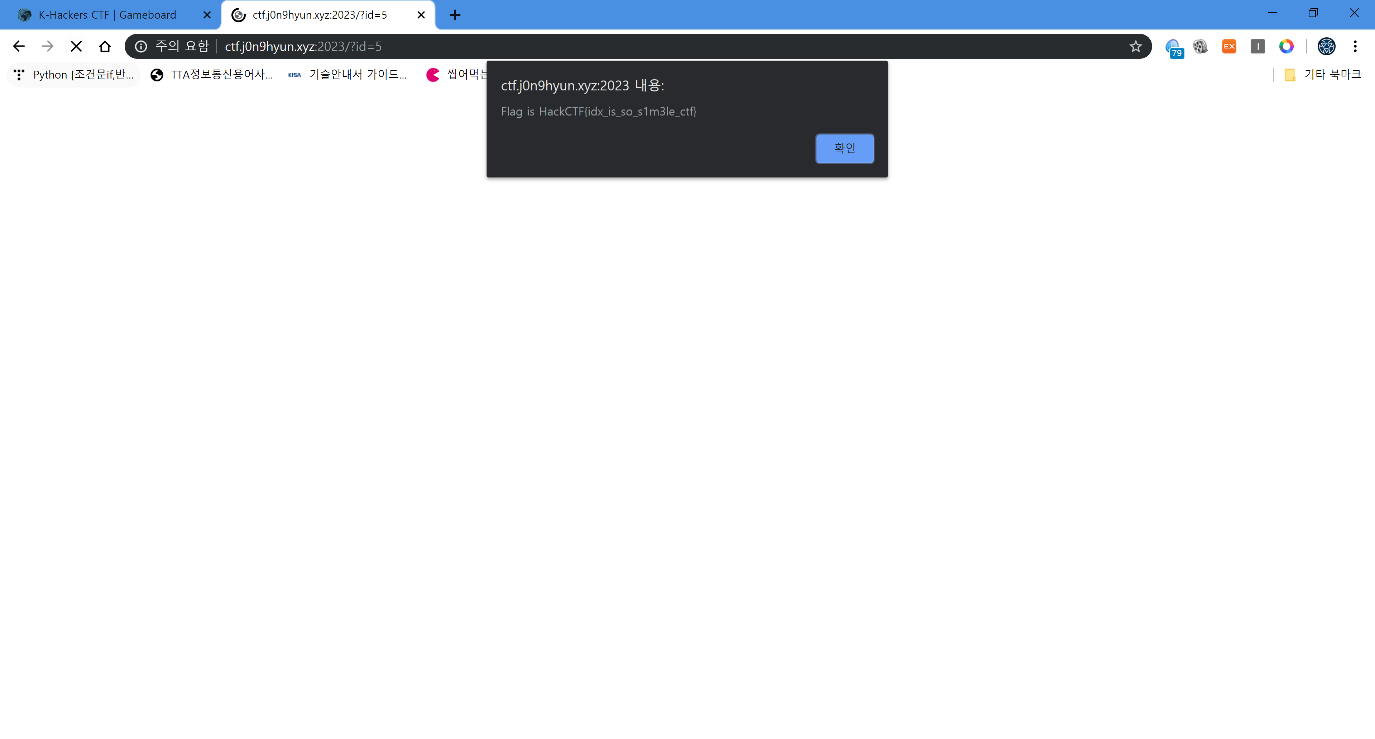
페이지에 접근하면 다음과 같은 페이지를 볼수가 있는데 5번째 파일에 플래그가 있다고 한다,

그런데 5번째 페이지가 숨겨져있음을 확인할수 있는데,



1번에 들어가면 <http://ctf.j0n9hyun.xyz:2023/?id=1>로 이동을 하며 Nop이라는 에코를 보내는데,

링크 뒤의 id=1이 수상하여 1번부터 4번까지 들어가 봤을 때, 뒤의 숫자가 페이지 숫자라는 것을 확인할수 있기 때문에 id=5로 접근을 하게 되면,



다음과 같은 에코를 통해 플래그 값을 확인할 수가 있다. 정답은 idx\_is\_so\_s1m3le\_ctf가 되겠다.

페루 문제

Q. 웹사이트 취약점 공격의 하나로, 사용자가 자신의 의자와는 무관하게 공격자가 의도한 행위(수정, 삭제, 등록 등)를 특정 웹사이트에 요청하게 하는 공격은?

A. 정답은 CSRF로 Cross-Site Request Forgery이다. 답의 필터링이 맨 앞 글자만 대문자로 하라고 하였으니 base64로 Cross-site request forgery 되겠다.

에콰도르 문제

Q. 웹 애플리케이션의 뒷단에 있는 Database에 질의하는 과정 사이에 일반적인 값 외에 악의적인 의도를 갖는 구문을 삽입하여 공격자가 원하는 구문을 실행하는 기법.

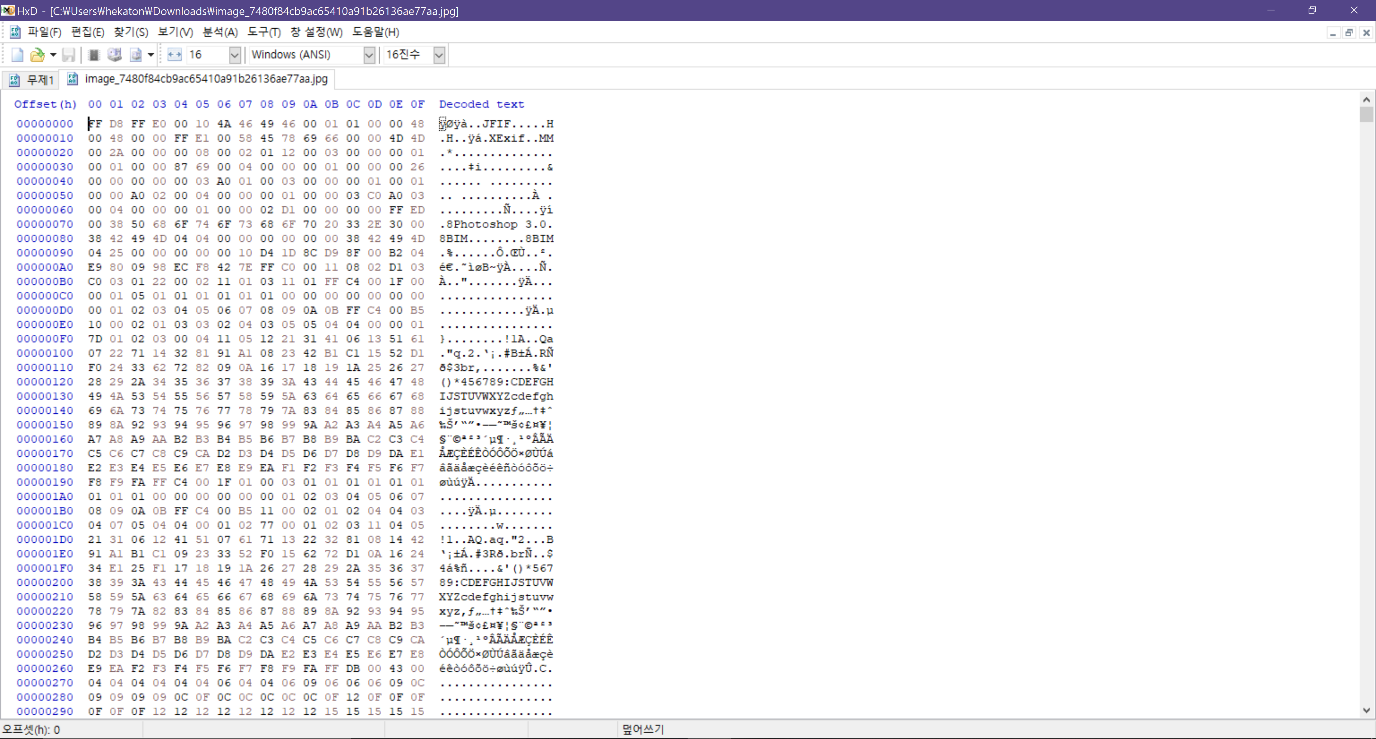
A. 정답은 SQL Injection이다. SQL 인젝션은 사용자가 입력한 데이터를 제대로 필터링 또는 탈출탈출 못하였을 때 발생하는 보안 취약점이기 때문에 약간의 필터링을 통해 이런 공격은 막아낼수 있다.

나이지리아 문제

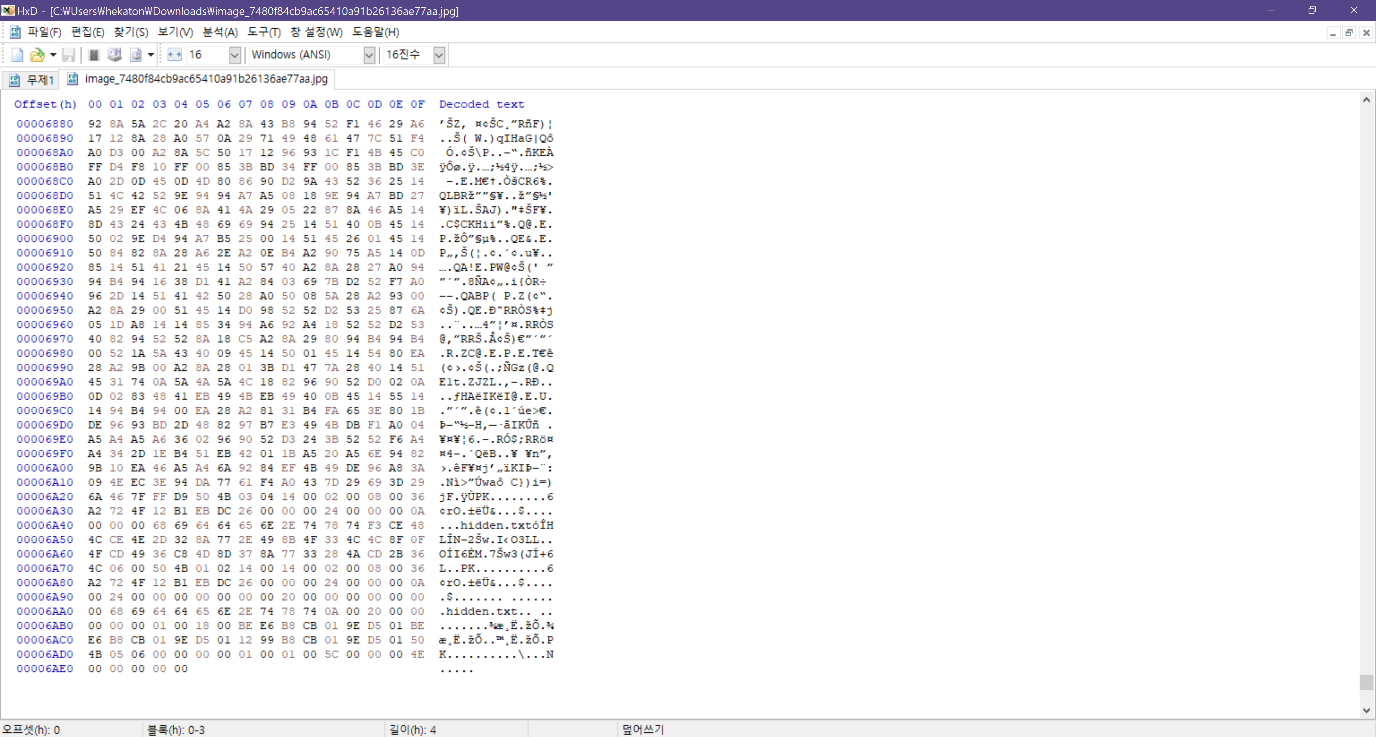
Q. 플래그를 찾아라

A. 처음 jpg파일을 다운받으면 단순한 그림밖에 없는 것을 확인할 수 있는데, jpg파일을 압축할때 불필요한 픽셀들을 제거하는 방식으로 더 나은 화질을 유지한다고 한다. 이 과정에서 jpeg파일을 인코딩 할 수가 있는데, 이 과정에서 플래그를 숨긴 것이라고 생각할 수 있다.

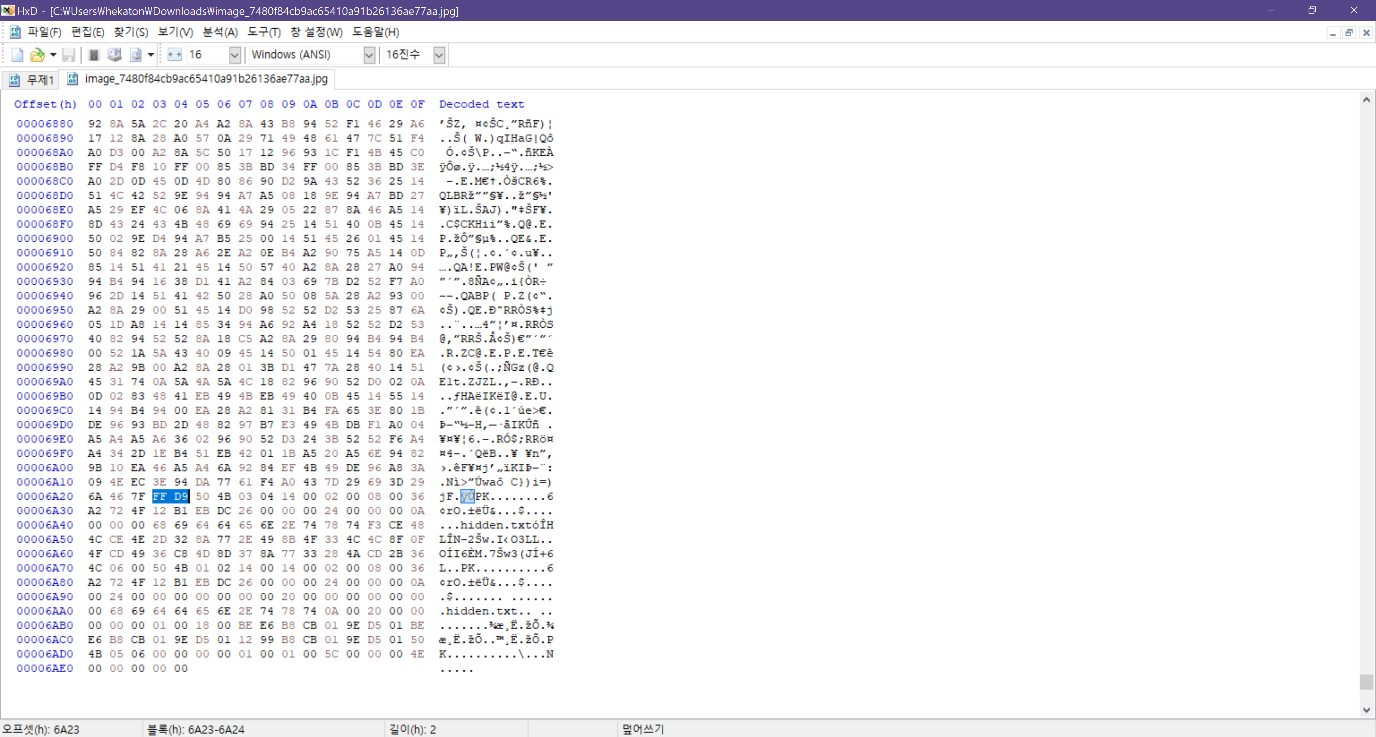
디스크 편집기 기능과 메모리 편집 가능을 도와주는 HxD를 통해 jpg파일을 뜯어 보면



다음과 같이 파일 내부를 볼 수가 있는데, 파일의 맨 처음에 붙는 코드인 ÿØÿà 부터 쭉 내려가다 보면 다음의 파일이 어떤 프로그램으로 편집되었는지 등에 대한 정보를 내가 참고한 페이지인 <https://nightohl.tistory.com/entry/JPEG-%ED%97%A4%EB%8D%94-%EA%B5%AC%EC%A1%B0>에서 헤더 구조를 비교하면서 읽어가다 보면



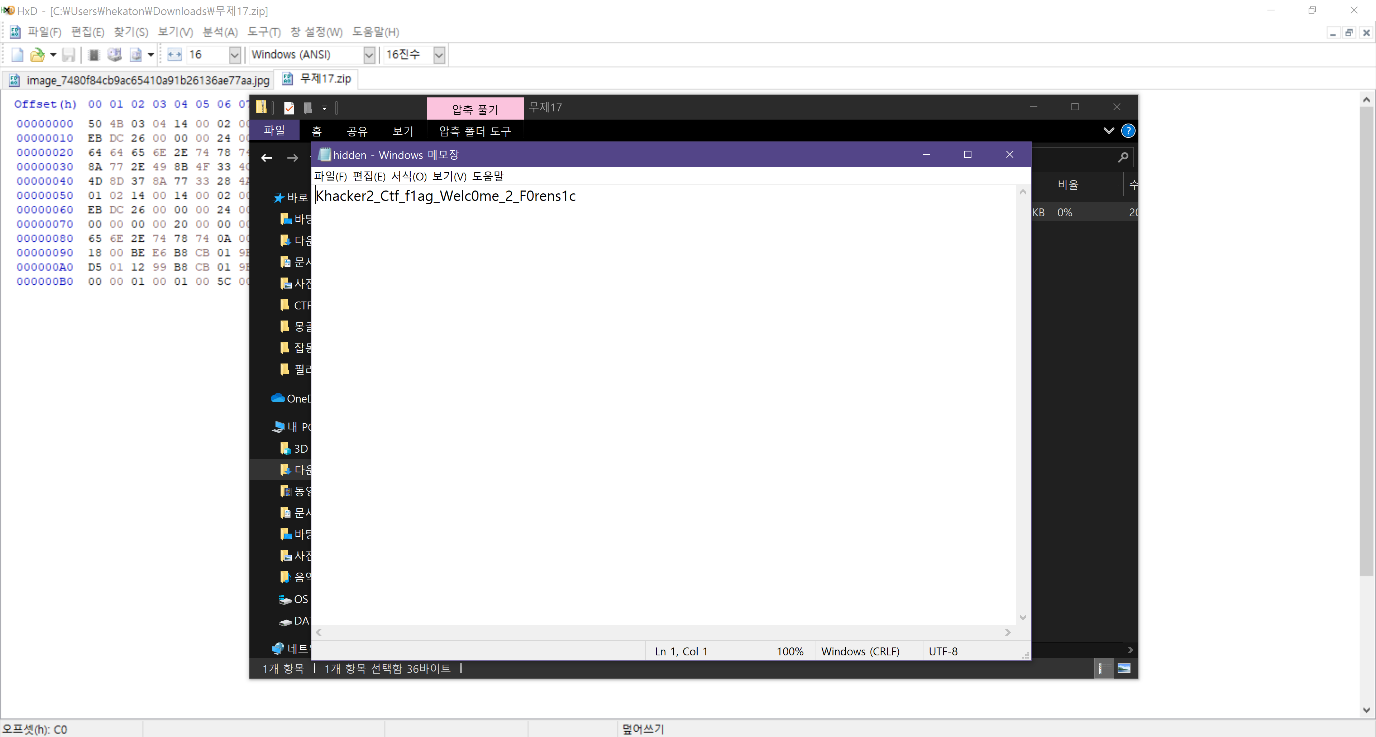
파일의 맨 끝에 hidden.txt 라는 구문이 있음을 확인할 수 있다.



파일의 끝을 나타내는 FF D9 뒤의 부분에 있는 코드가 내가 찾는 숨겨진 값이라고 생각되기 때문에 FF D9뒤의 부분을 끌어다가



붙여넣기 한 후에 파일 시그니처 값을 확인하자. 50 4B 03 04 는 zip으로 해당 파일 포멧을 확인 하였기 때문에 다음의 값을 알아낼 수 있다. 값을 저장할 때 뒤에 .zip을 붙이고 저장을 하게 되면 저장된 파일 안에 메모장 파일이 하나 생성되는데, 그 안의 값이 바로 zip포멧으로 디코딩 된 값이다. 그 값은 내가 찾으려는 값과 같다.

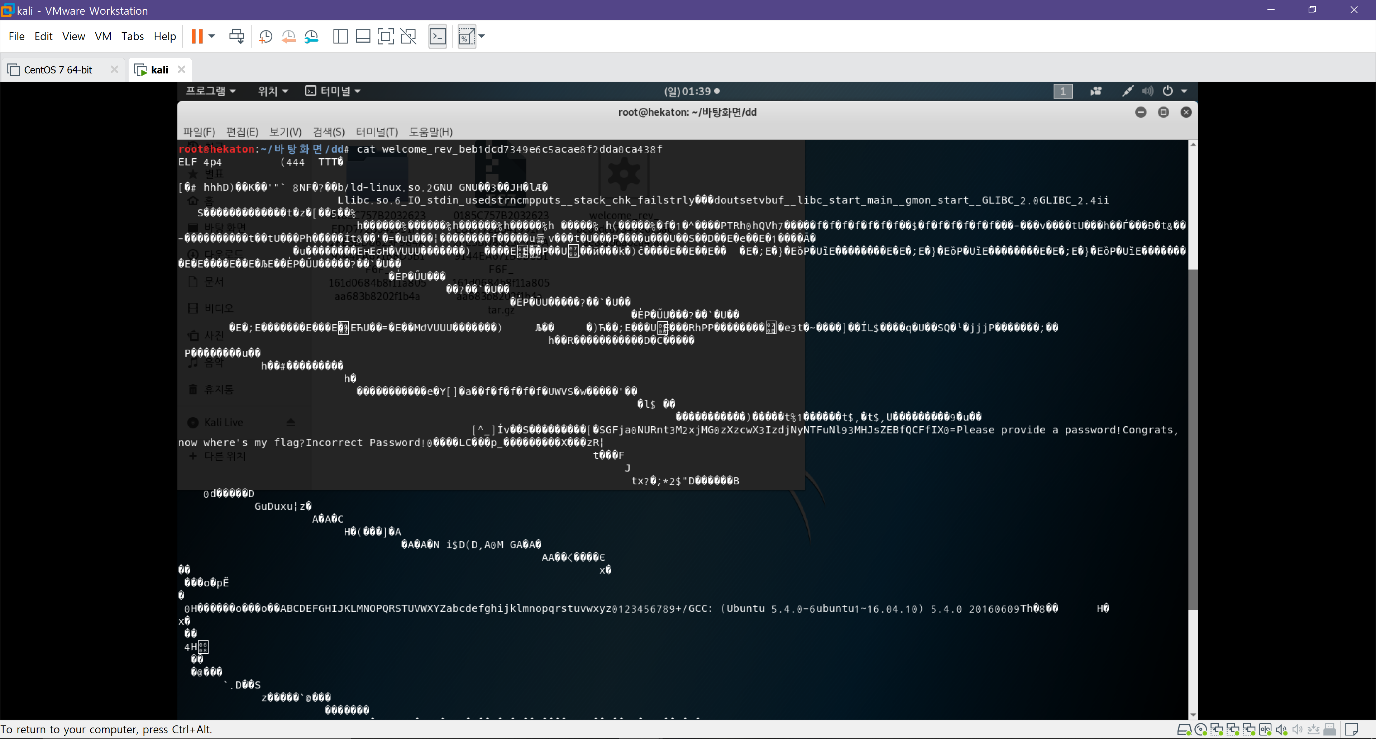


이렇게 답을 알아낼 수가 있다.

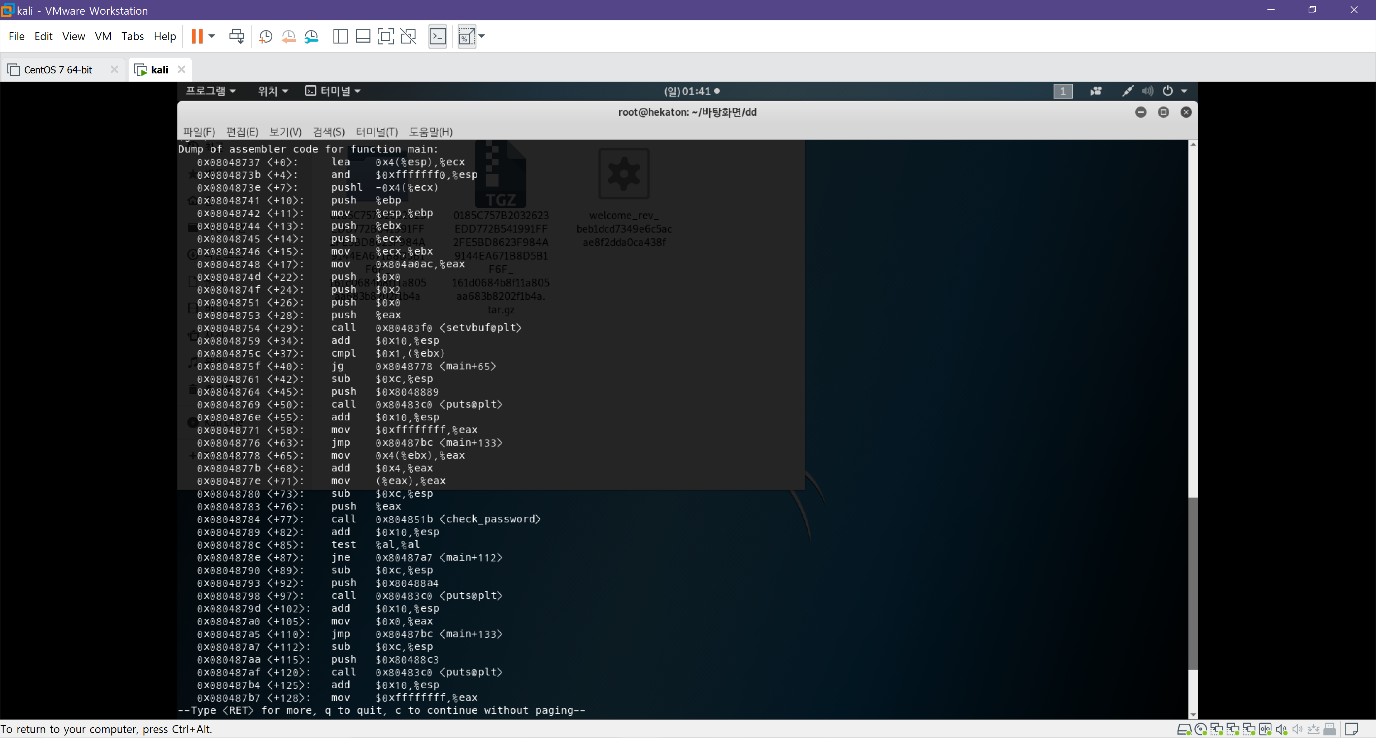
앙고라 문제

Q. 플래그를 찾아라

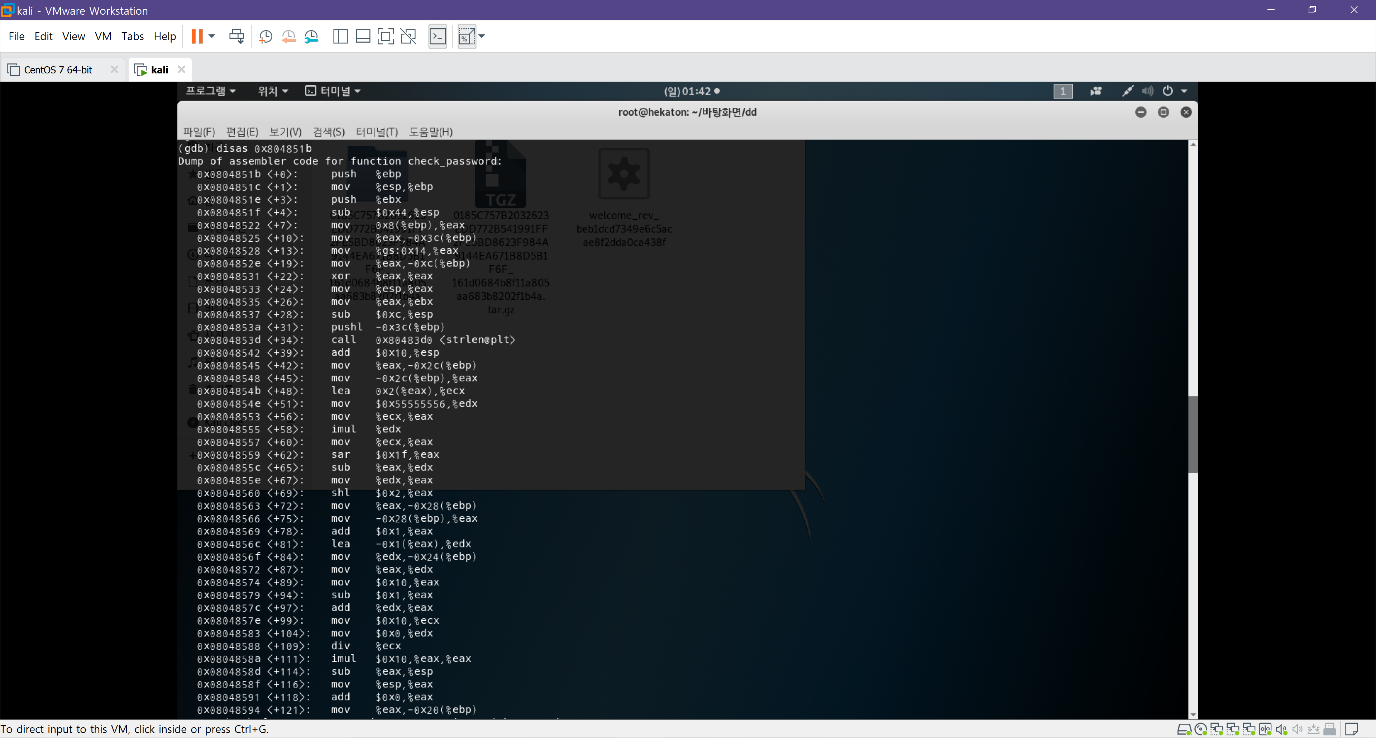
A. 처음 밑에 링크 되어있는 파일을 리눅스 환경에서 다운받으면 실행이 되지 않기 때문에 리눅스 자체에서 지원하는 디버거를 통해 내부에 어떤 일이 일어나고 있는지 알아볼 것이다.



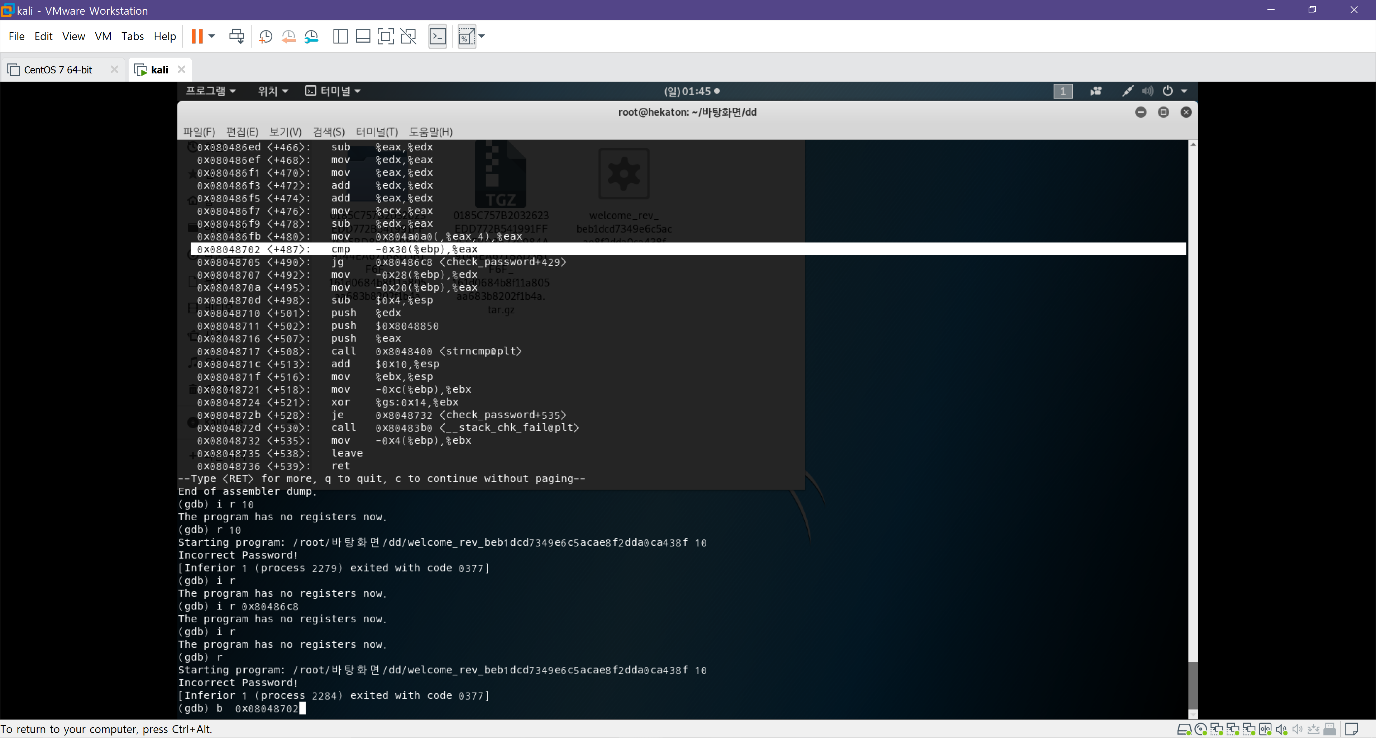
일단 처음으로 리눅스 자체에서 지원하는 cat 명령어로 내부를 본 것인데, 아무것도 알아볼 수가 없다. 그렇다면 내부에서 일어나는 일을 어떻게 확인 해야 할까 – gdb 디버깅 툴을 이용하자.



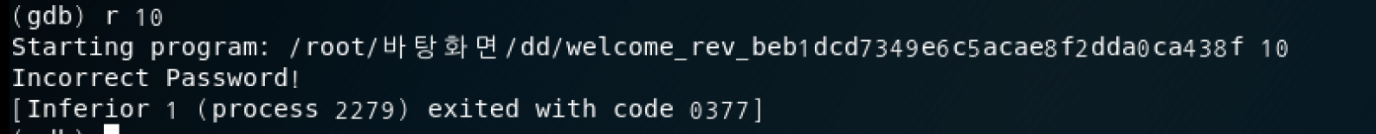
먼저 디버깅할 파일이 있는 위치로 이동한후 disas main으로 이동을 한 후 무엇이 있나 보다보면



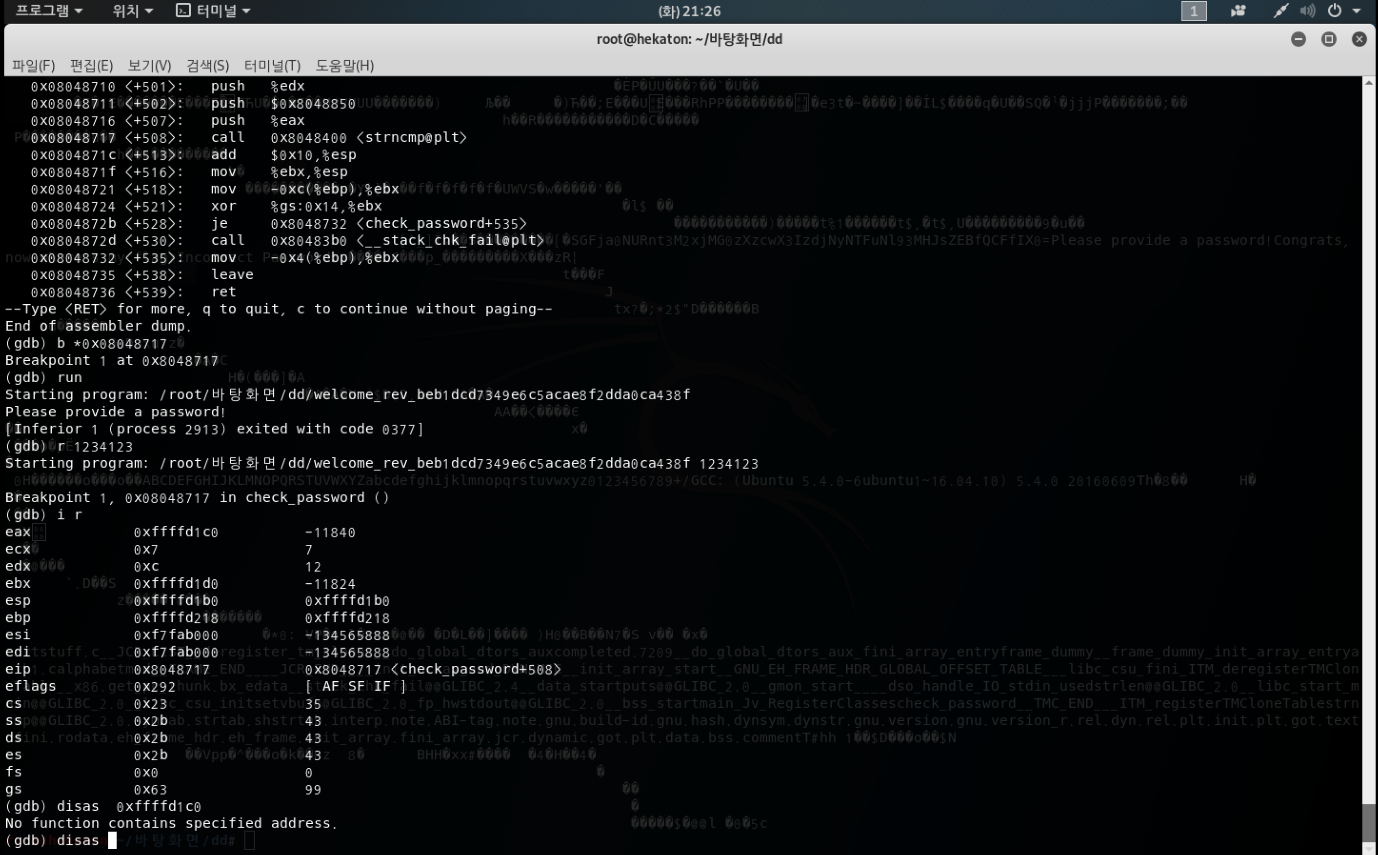
Check password 라 적혀 있는 부분을 찾을 수 있는데, 여기를 disas 로 이동하면 내부 코드를 볼수 있다.



그리고 브레이크 포인트를 걸어 실행을 하면



그냥 실행되어 버리는 것으로 보아 브레이크 포인트를 잘못 잡은 것 같다, 다른 곳에 브레이크 포인트를 걸고 실행을 다시 해보니

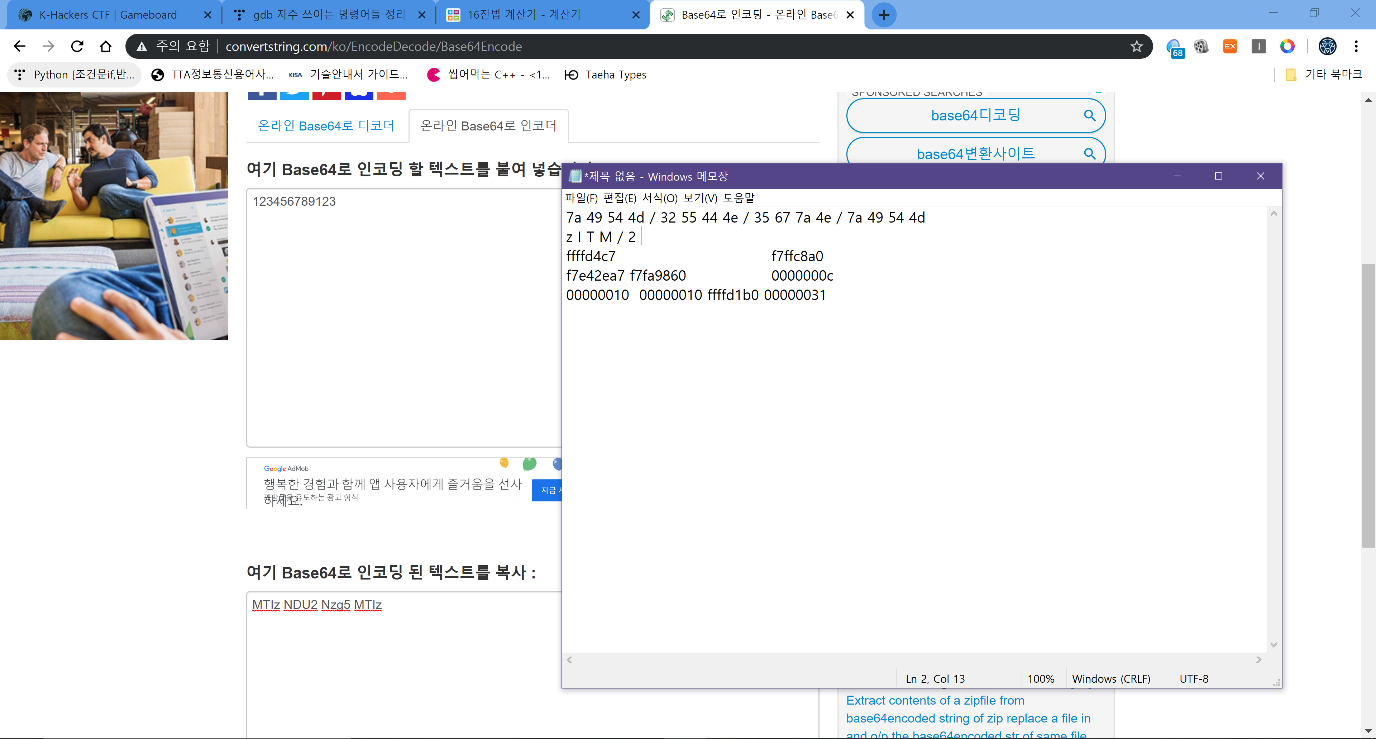


제대로 입력한값이 들어간 것을 확인했고



비교가 되는 정답값을 확인하기 위해 값을 다르게 넣어 실행을 해보았지만 숫자가 달라지는 것을 확인했다.

eax내부를 확인해본 결과 내가 입력한 값이 암호화 되어 들어가는 것을 밑의 사진을 통해 확인할수 있었다.



이 다음 정답 값을 찾아야 하지만 아무리 뒤져봐도 답이 나오지 않아 (내가 못 찾아서) 더 이상의 진척이 없어 지쳐버렸다.