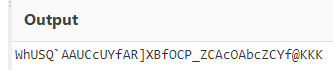
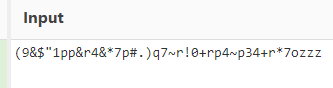
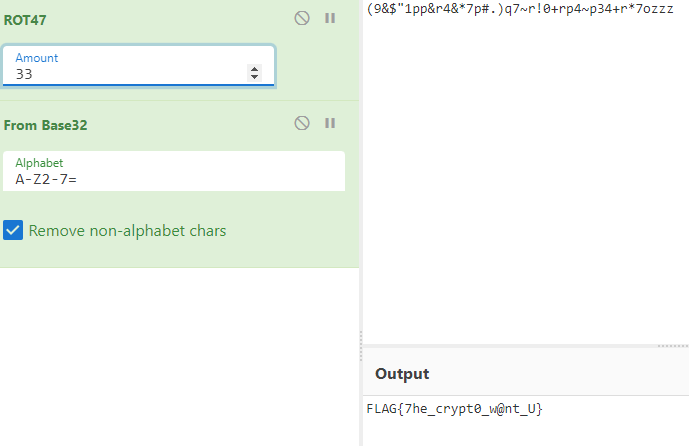
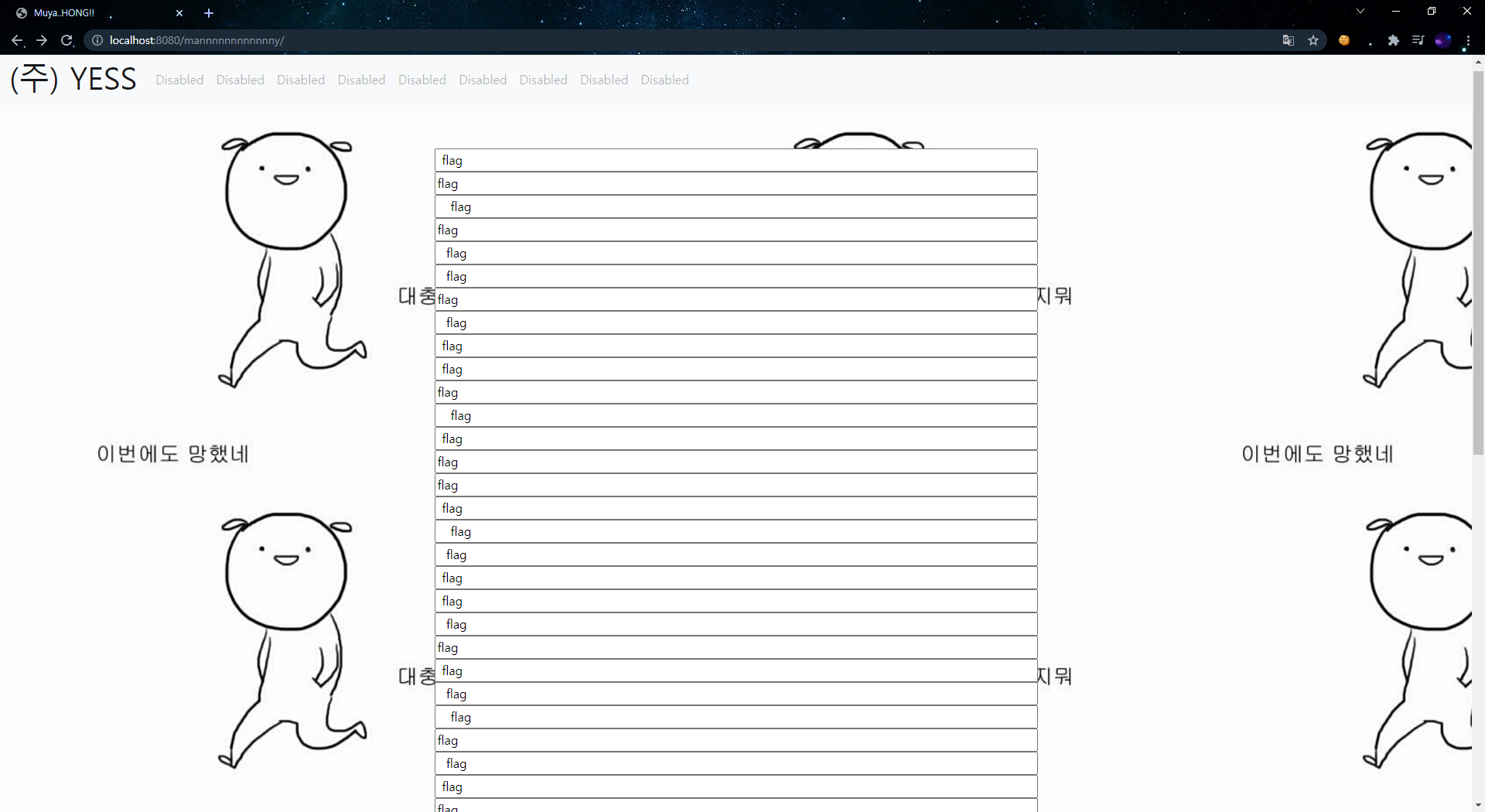
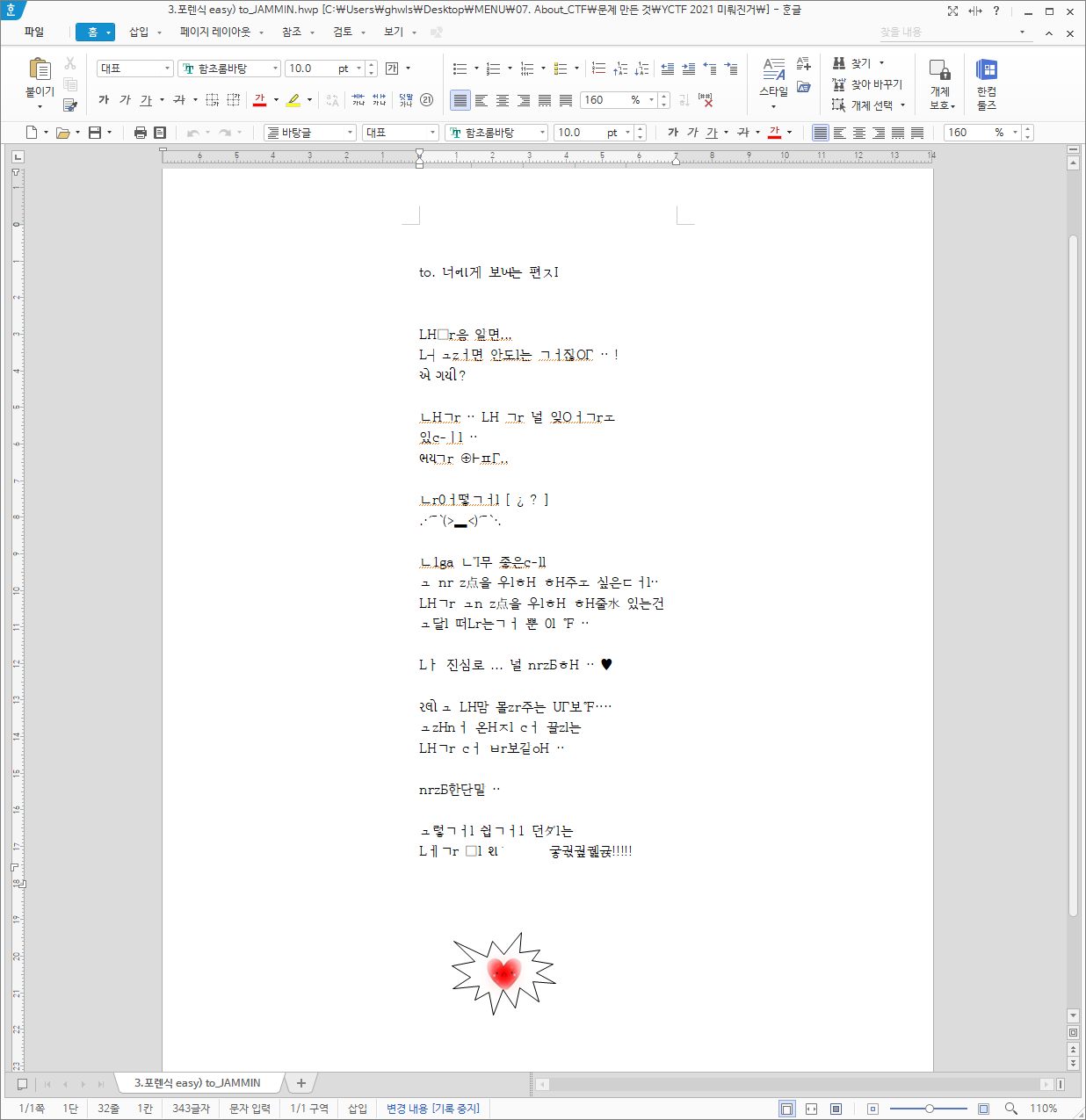
1. MICS easy) YCTF CRYPTO  
   ----------------------------------------------  
   (9&$"1pp&r4&\*7p#.)q7~r!0+rp4~p34+r\*7ozzz  
   ----------------------------------------------  
     
   1-1.   
   맨 뒤 글자가 세번 반복되는 것이므로 글자 치환암호인 ROT를 예상하며,   
   또 특수문자까지 변경된 것으로 보아 ROT 47으로 유추한다.  
     
   1-2. ROT 47을 돌려보았을 때 비슷한 다른 값이 나오는 것을 확인  
     
     
     
   1-3. 양식을 보고 Base 암호화 방식으로 추정하여 여러 값을 넣는다.  
   Base32 암호화에 반응하는 것을 알아챘지만 값은 여전히 답이 없다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
     
   1-4. ROT 47의 값을 33으로 조정하여 플래그를 찾았다.  
   
2. 텍스트이(가) 표시된 사진

   자동 생성된 설명3.웹 easy) mannnnnnnnnnnny  
   문제 사이트에 접속을 하면 다음과 같이 나타난다.  
     
     
   2-1. 많은 파라미터 값이 보이며 마지막에 다음과 같이 “GO !” 버튼이 보인다.  
     
     
   2-2. 이 버튼을 누르면  
   flag글자만 나온다.  
   다른 값을 넣고 버튼을 눌러보자.   
     
   2-3. 44, 12, 12 라는 값이 나왔다  
   이로써 12번째, 44번째 입력 란에  
   입력된 값이 출력되는 것을 확인했다.   
     
   2-4 44번째에 ls / 12번째에 ls -al 을 입력했지만, 그대로 출력이 된다.  
   window 명령프롬프트 명령어를 입력해보자  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   44번째 입력란에서 실행되는 것을 확인했다.  
   2-5. Ffllaaggggg 라는 곳이 디렉토리 이므로 cd 로 확인해보니 flag.txt파일이 있다.  
   cmd 명령어인 type .\ffllaaggggg\flag.txt  
   를 입력하면 다음과 같이 나온다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
   2-6.  
   해당 문자열을 가져와 암호화를 푼다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   암호 형태를 보니 base64에서 자바 난독화를 확인하여 raw inflate 를 추가하여 플래그를 획득하였다.
3. 포렌식 easy) to\_JAMMIN  
   해당 문제는 문서파일이 있다. 파일을 열면 다음과 같이 편지가 나온다.  
     
     
   3-1. 첨부파일에 대해 포렌식해본다.  
   (이때, 우클릭하여 그림파일로 저장을 하지 않아야한다.)  
   방법은 다음과 같다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   문서정보를 클릭한다.  
   3-2. 그림정보로 이동하여 삽입된 그림을 확인한 뒤, 저장하기를 클릭한다.  
   텍스트이(가) 표시된 사진

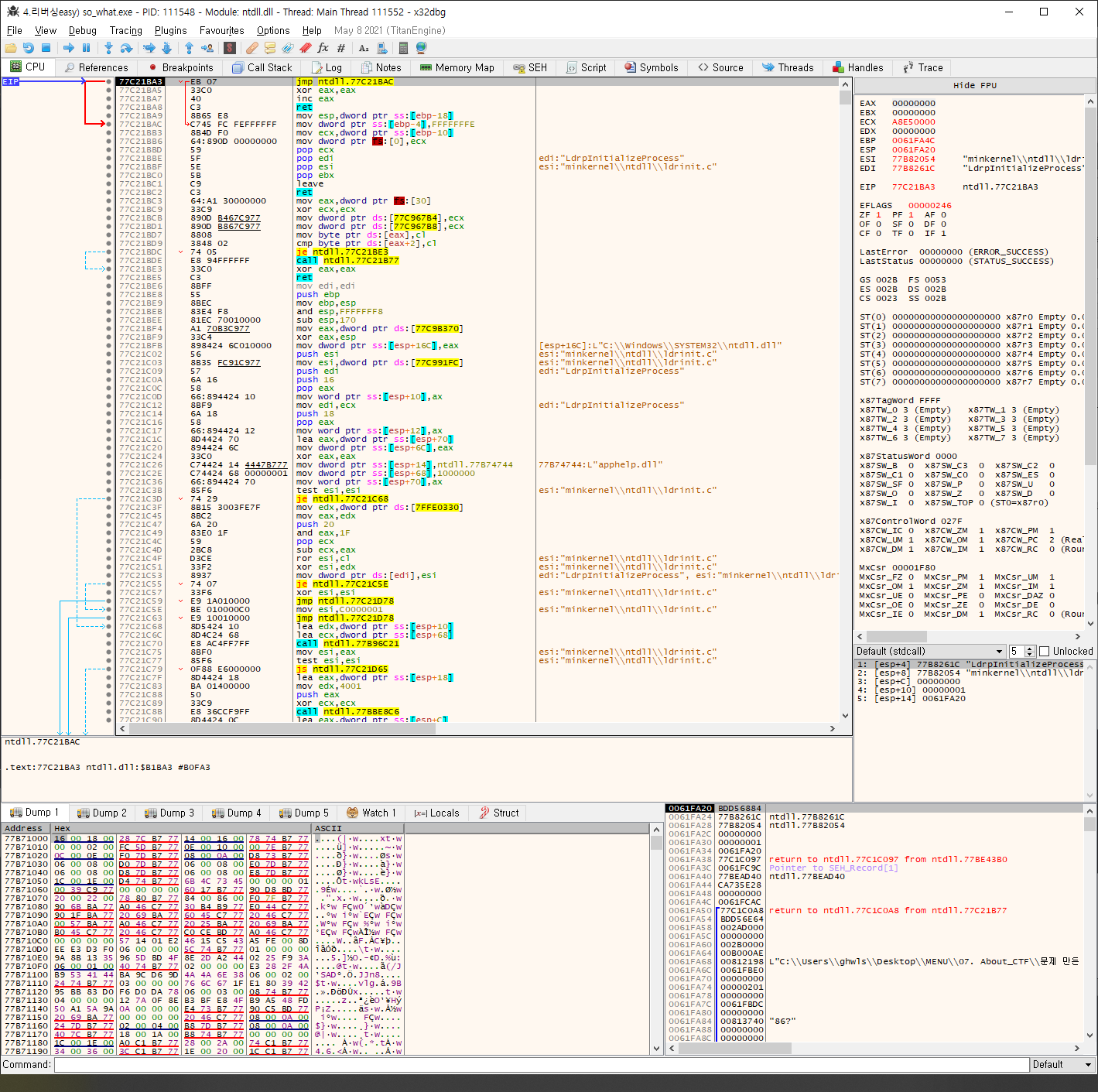
   자동 생성된 설명  
     
   3-3. 확인을 눌러 저장시킨 뒤 해당 디렉토리로 이동하여 저장된 그림파일을 OpenStego로 연다.  
   텍스트이(가) 표시된 사진

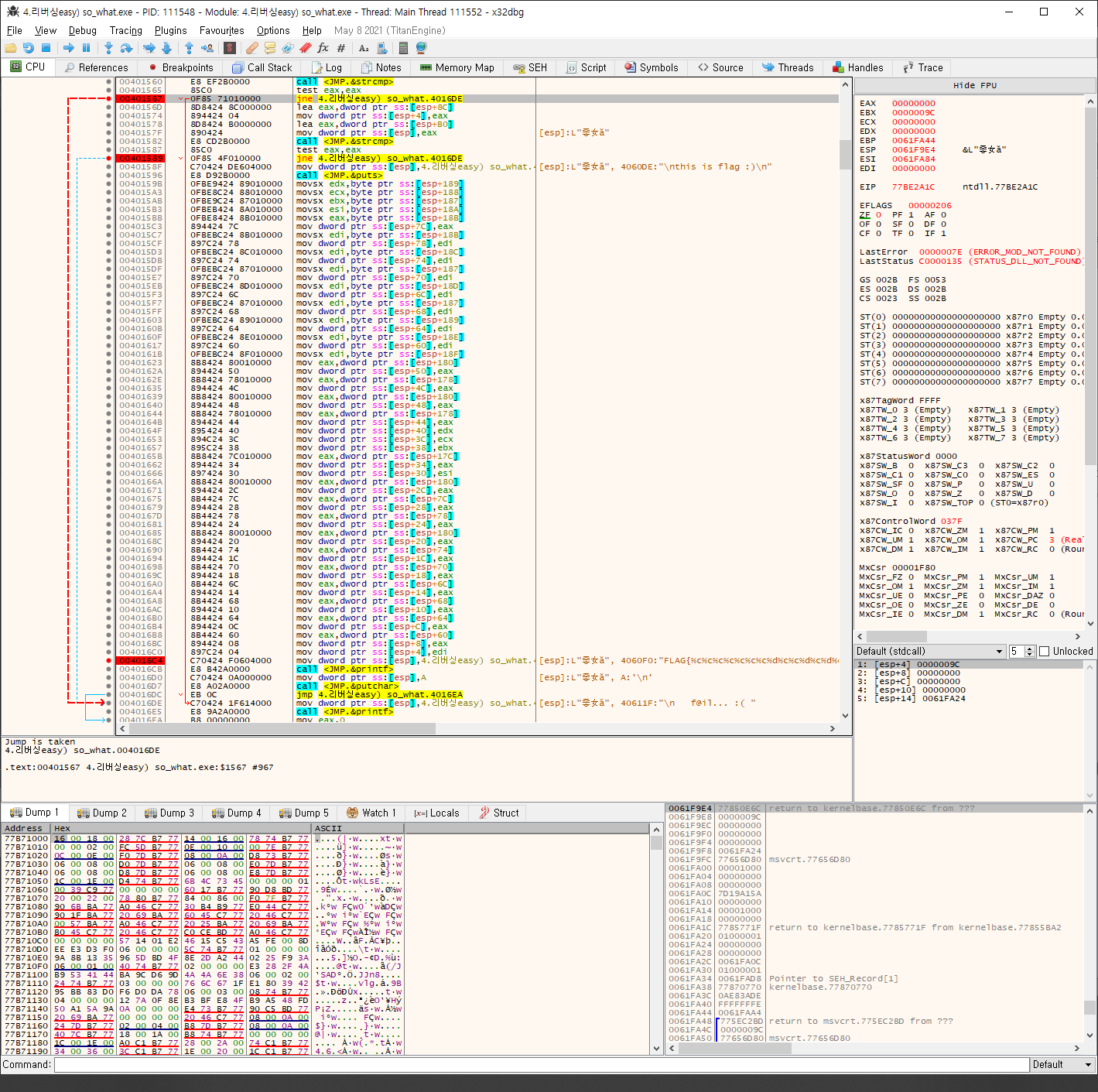
   자동 생성된 설명  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
   3-4Extract Hidden Data 를 클릭하여 나오는 flag.txt파일을 확인한다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
   텍스트이(가) 표시된 사진

   자동 생성된 설명 플래그를 확인했다.
4. 리버싱easy) so\_what  
   실행 시, 다음과 같이 나온다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
   4-1. 디버거 프로그램을 통해 실행순서를 조작하면 된다.  
     
     
   4-2. 실행하다보면 다음과 같이 문자열을 찾을 수 있다.  
     
   이때, 플래그의 값은 문자를 호출하여 가져오기 때문에 보이지 않는다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
     
     
   4-2. 이동해보면 다음과 같이 문자열을 검증하는 곳을 찾을 수 있다.  
     
     
   4-3. 해당 구간을 레지스터 값을 변경하여 통과하면 플레그가 출력된다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명
5. 리버싱easy) dead or alive  
   프로그램을 실행하면 문자가 출력됨과 동시에 444분후에 로그오프된다는 메시지가 뜬다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
   5-1. 디버거를 통해 확인하면 안티디버거 프로그램이 실행되며 종료되게 된다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명  
     
   5-2. 디버거 내에 숨어있는 함정을 jmp 문을 작성하여 원하는 주소로 이동하여 플래그를 획득한 것을 확인했다.  
   텍스트이(가) 표시된 사진

   자동 생성된 설명