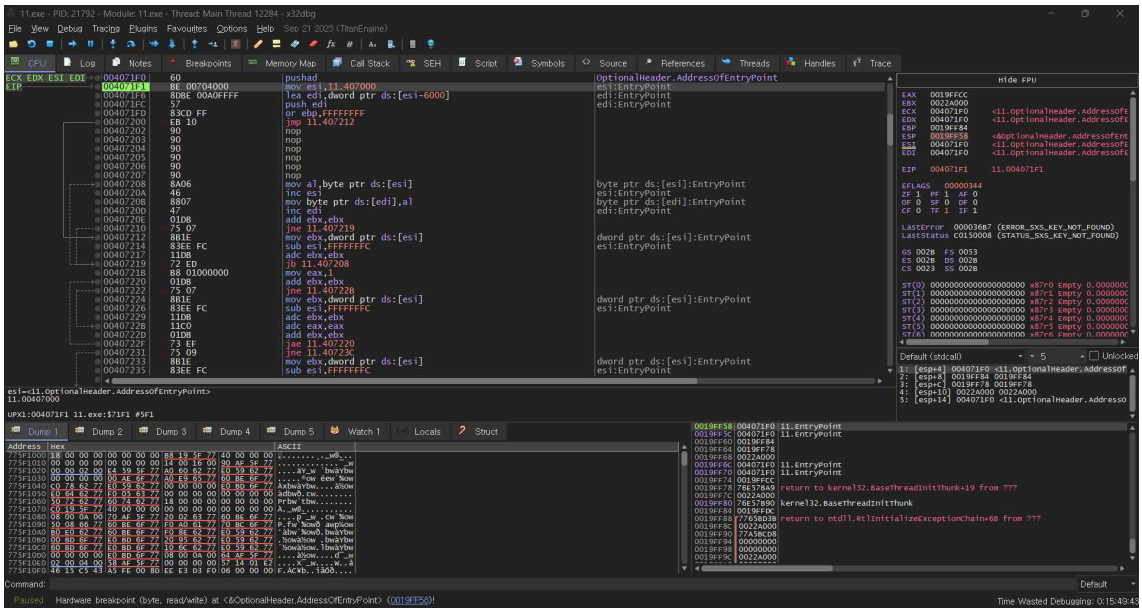
CodeEngn Basic RCE

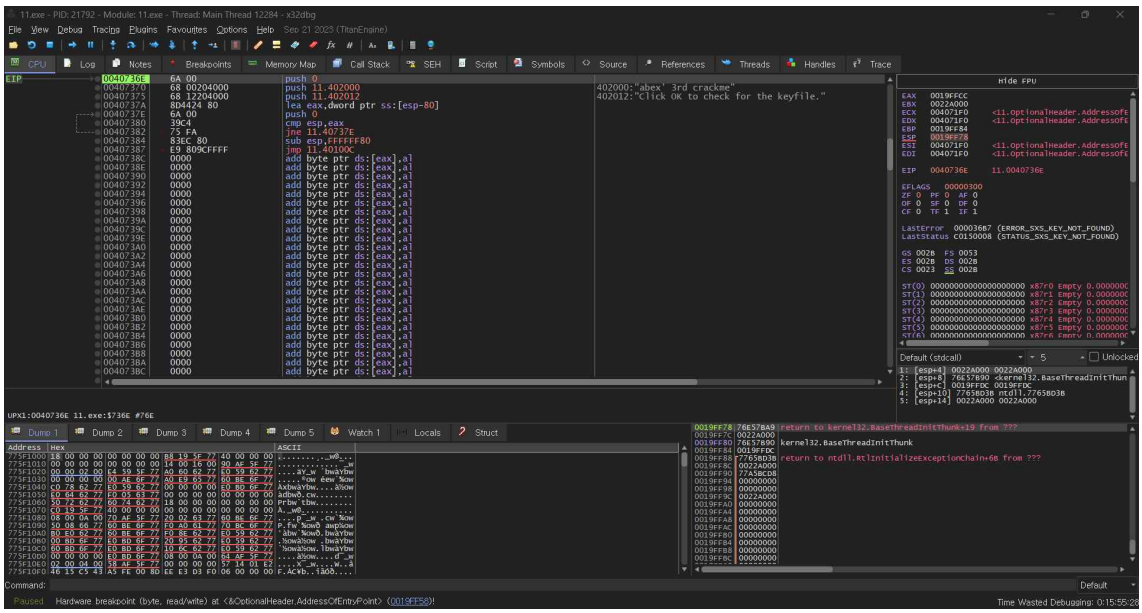
강 보 성

[11]

풀이 : pushad 명령어를 실행하고 ESP에 저장된 주소로 이동하여 popad 명령어 실행 이후의 위치를 찾는다. pushad ~ popad 사이의 과정은 패킹된 코드를 언패킹(원본 코드를 메모리상에 복구)하여 원본 코드를 받아오는 과정이다. pushad를 통해 레지스터의 값을 보존하고, 언패킹 후 popad를 통해 이전의 레지스터 값을 복원하여 원본 코드를 사용할 수 있게 한다.

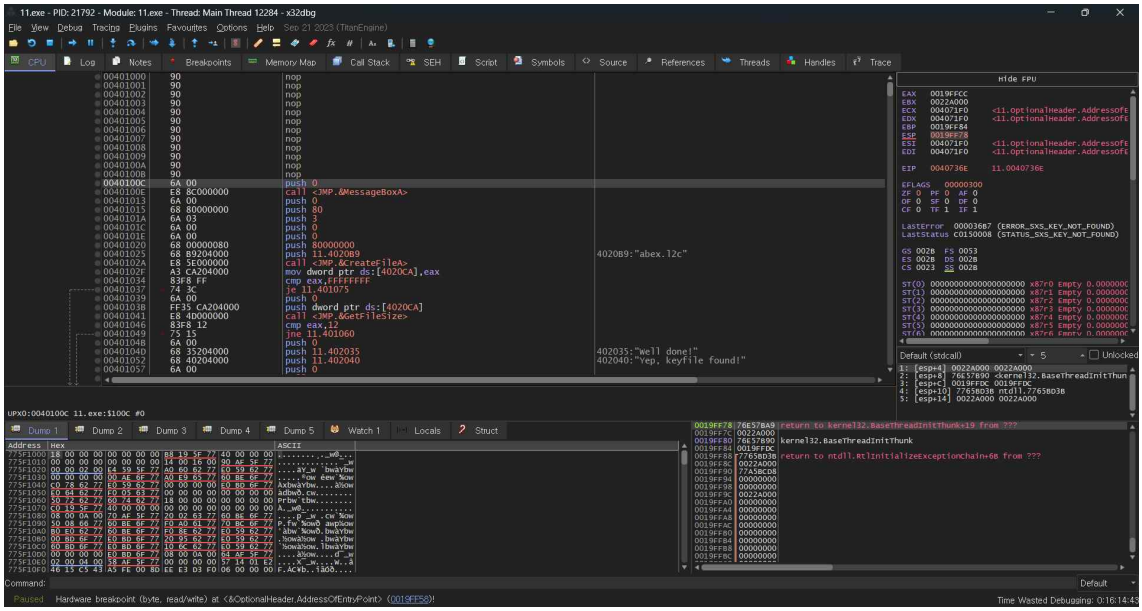


<pushad 명령어 실행 후 ESP에 레지스터 값이 저장된 모습>



<popad 이후 코드>

해당 위치에서 StolenByte로 의심되는 문자열 데이터를 발견할 수 있다.
또 아래에 원본 코드로 점프하는 코드가 보인다. 해당 코드의 주소를 따라가 보면 원본 코드를 발견할 수 있다.



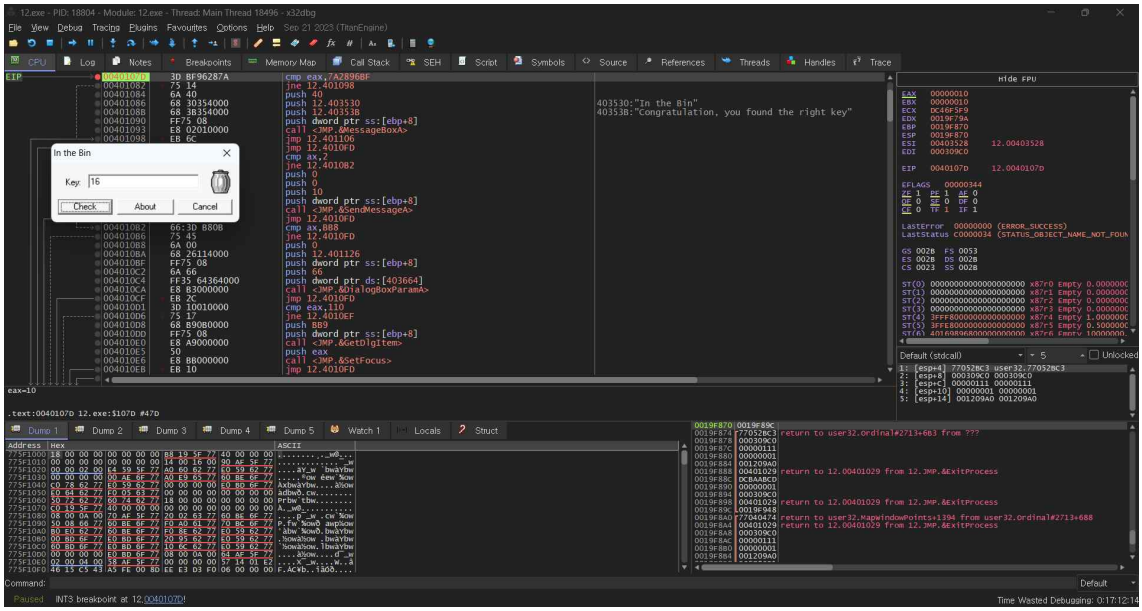
<원본 코드>

원본 코드를 보면 위쪽으로 의미 없는 값이 채워져 있음을 확인할 수 있다. 이것으로 원본 코드에 StolenByte 기법이 적용되어 있음을 알 수 있다. 동시에 원본 코드의 OEP가 00401000 임을 알 수 있다.

답 : 00401000 6A006800204000681220400O

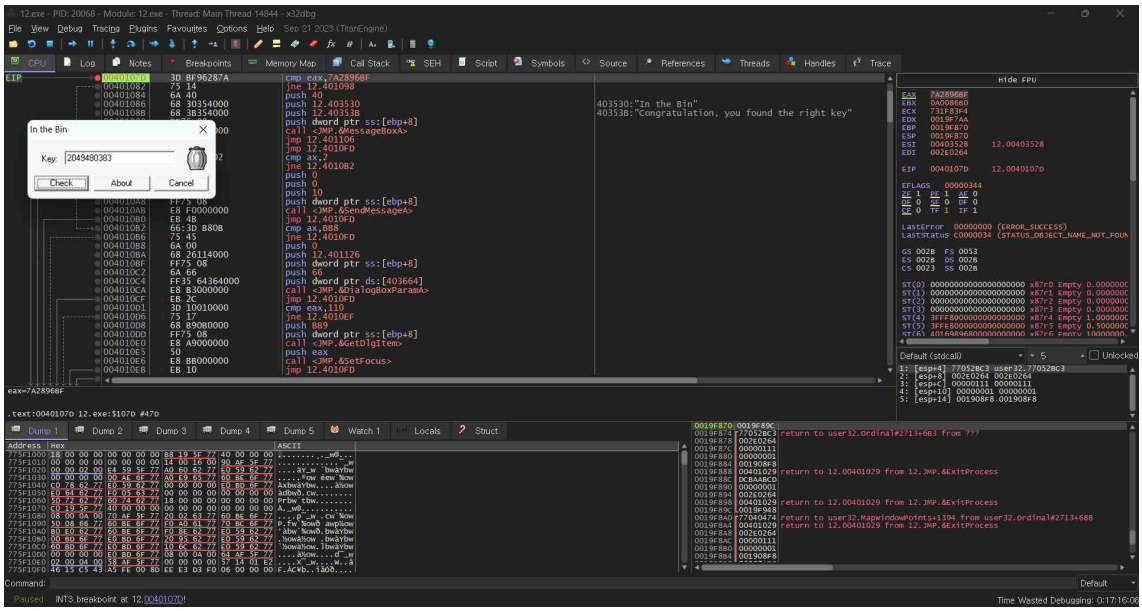
[12]

풀이 : 문자열 데이터를 분석하다 보면 성공 구문이 보인다. 그 위로 cmp 명령어가 보이는데 eax 값과 7A2896BF라는 값을 비교하여 같으면 성공구문을 출력하는 것으로 보인다. 실제로 cmp 명령어에 브레이크 포인트를 걸어두고 디버깅을 하면 입력값이 16진수로 변환되어 eax에 저장되는 것을 확인할 수 있다.



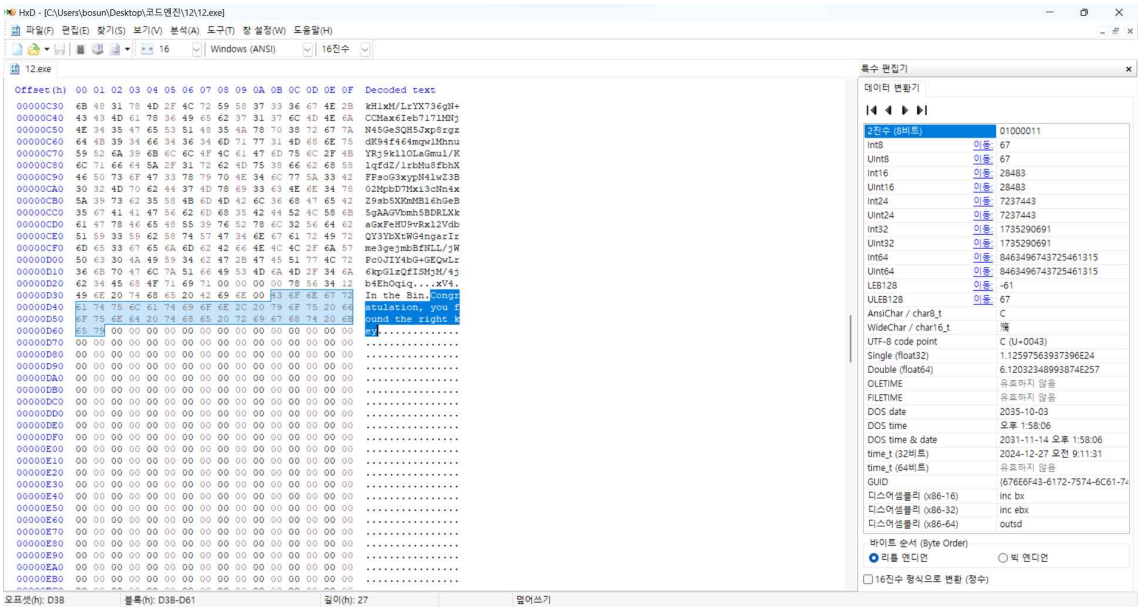
<16을 입력한 뒤 변화된 eax 값>

따라서 프로그램을 다시 실행하여 7A2896BF라는 값을 10진수로 변환한 값인 2049480383를 입력하면 성공 구문이 출력됨을 확인할 수 있다.

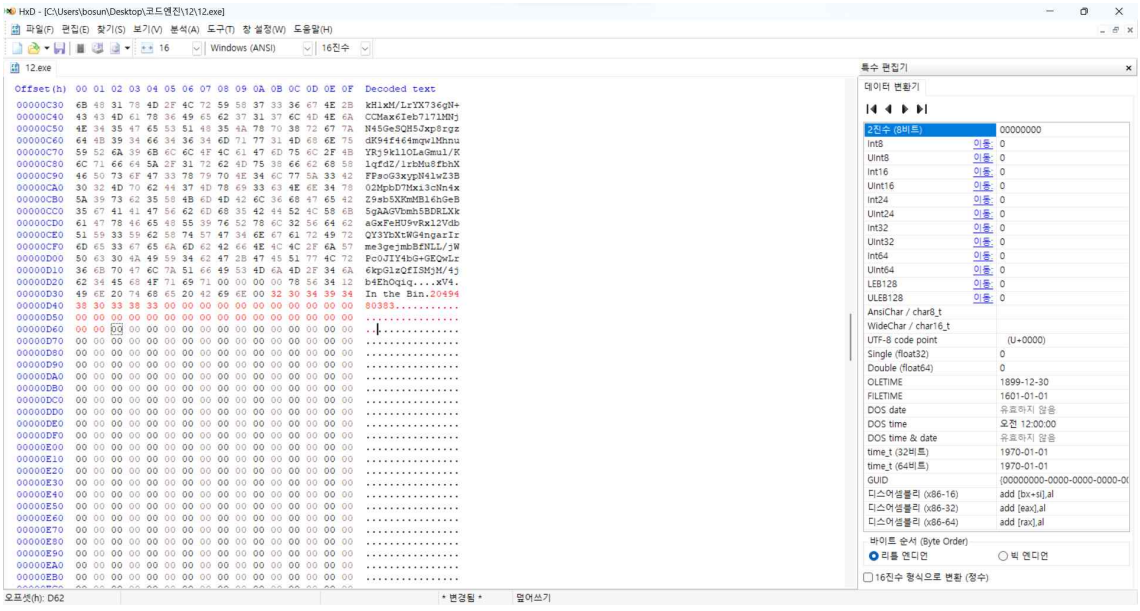


<2049480383을 입력한 뒤 변화된 eax 값>

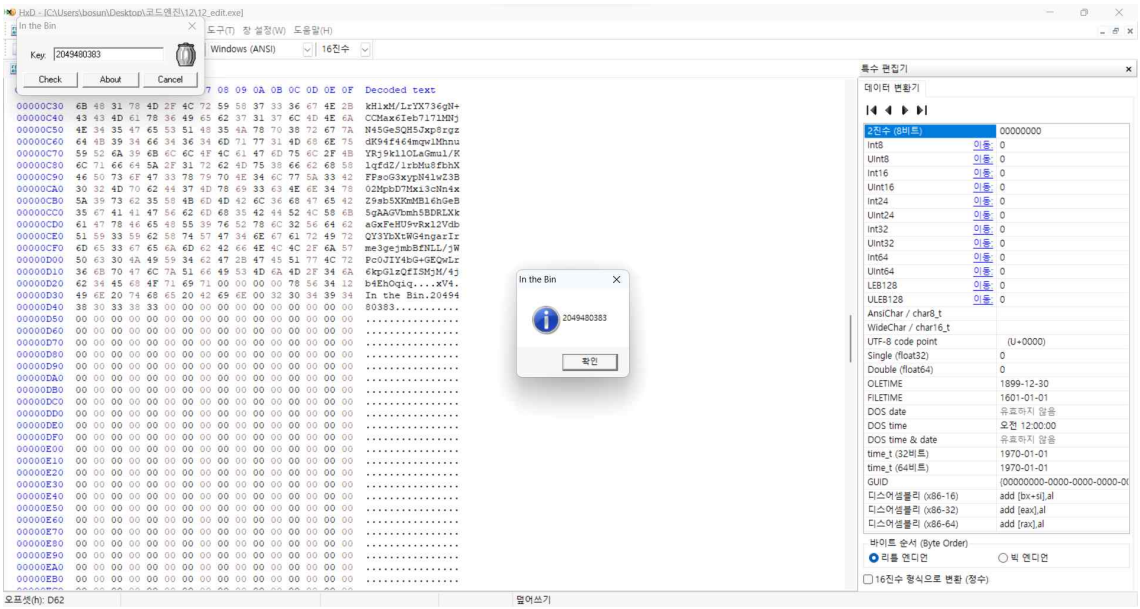
다음으로, Key 값을 성공메시지 대신 출력하기 위해서 헥스에디터를 사용하여 성공메시지 문자열을 2049480383으로 overwrite한다.



<성공메시지 영역 선택>



<overwrite(0D3B~0D45)>

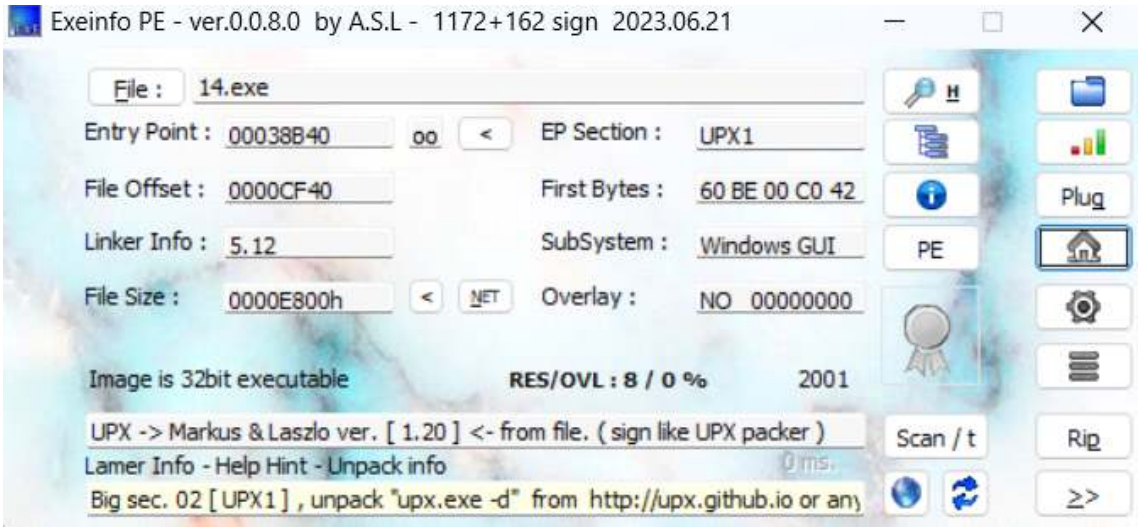


<성공메시지 대신 Key 값 출력>

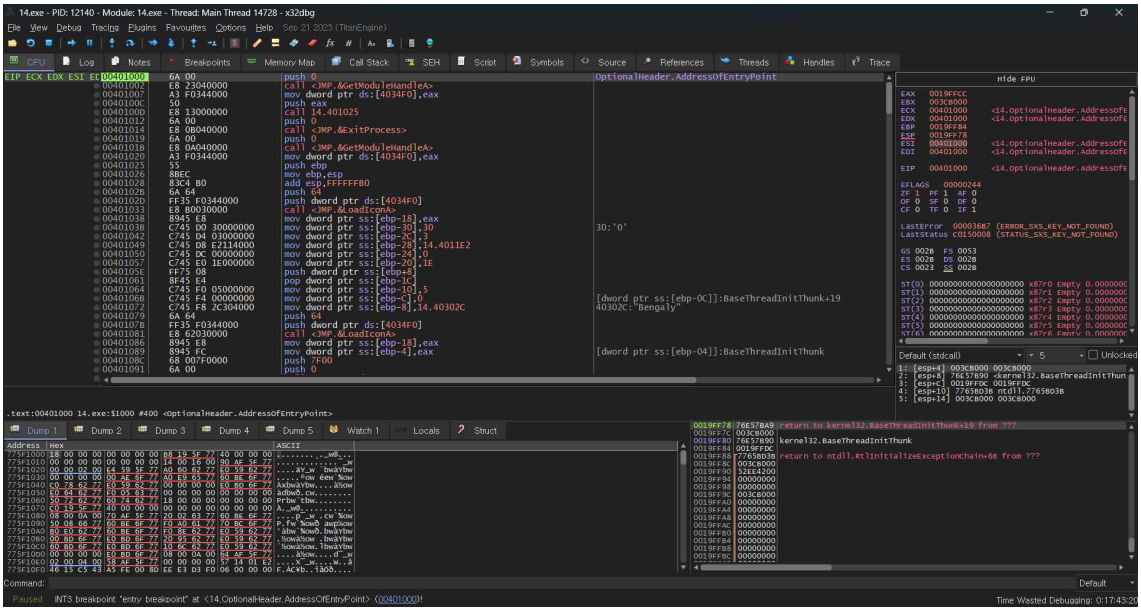
답 : 2049480383 0D3B 0D45

[14]

풀이 : 14.exe가 upx로 패킹되어있기 때문에 원활한 분석을 위해 언패킹한다.

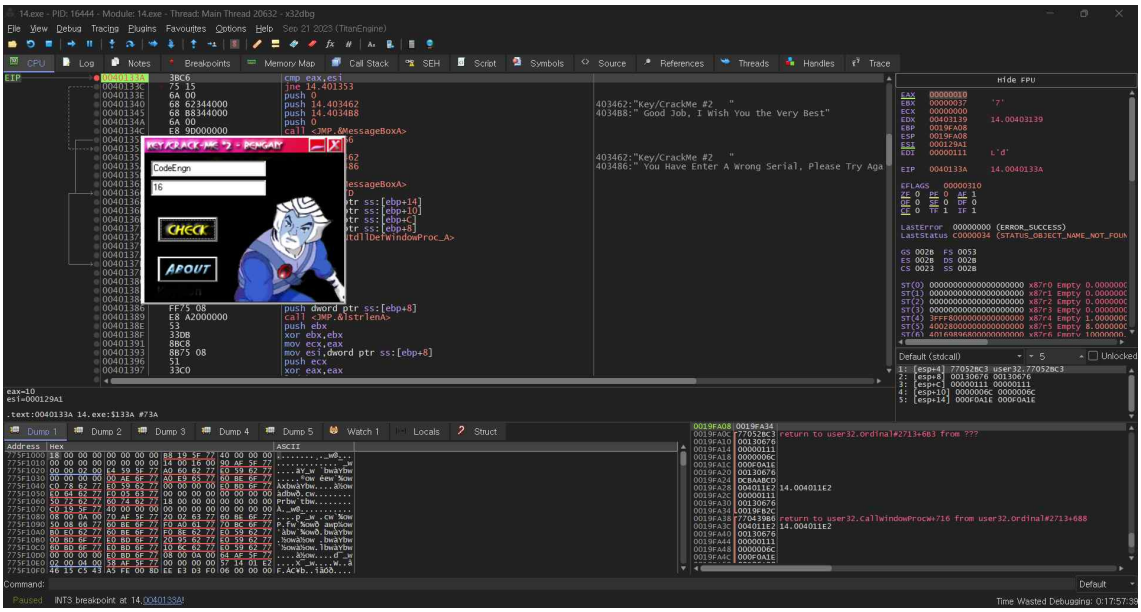


<Exeinfo PE에서 확인한 14.exe 정보>



<언패킹 후 14.exe 디버깅 화면>

그 후 문자열 데이터를 분석하다 보면 성공 메시지를 발견할 수 있는데, 성공 메시지 위로 eax와 esi를 비교하는 cmp 명령어가 보인다. 해당 명령어에 브레이크 포인트를 걸어두고 다시 실행하여 시리얼 값으로 16을 입력한다. 그러면 eax에 10이 저장됨과 esi에 129A1가 저장됨을 확인할 수 있다.



<시리얼 값으로 16을 입력 후, 변화된 eax 값과 esi 값>

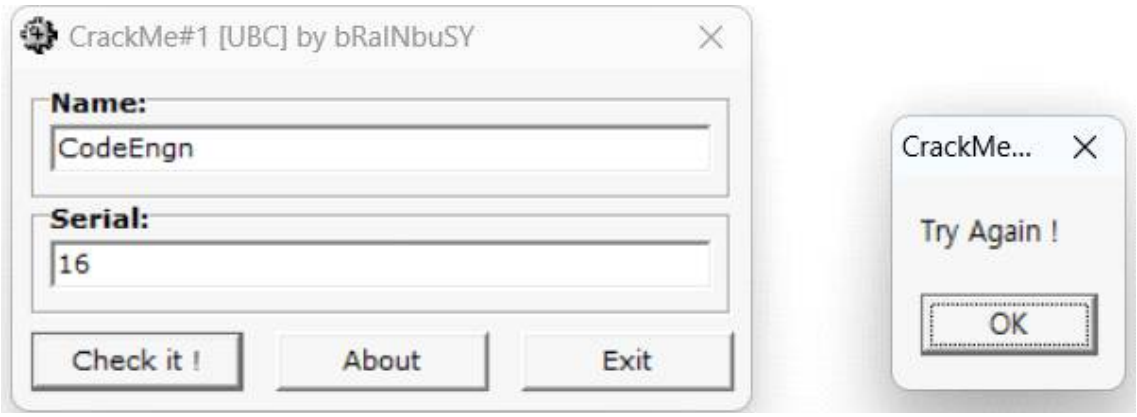
내가 입력한 값을 16진수로 변환시키고 129A1과 비교한 뒤 같으면 성공 메시지를 출력하는 것으로 추측된다. 따라서 프로그램을 다시 실행시키고 129A1을 10진수로 변환시킨 값인 76193을 시리얼값으로 입력해본다. 그러면 성공 메시지가 출력됨을 확인할 수 있다.

<성공 메시지 출력>

답 : 76193

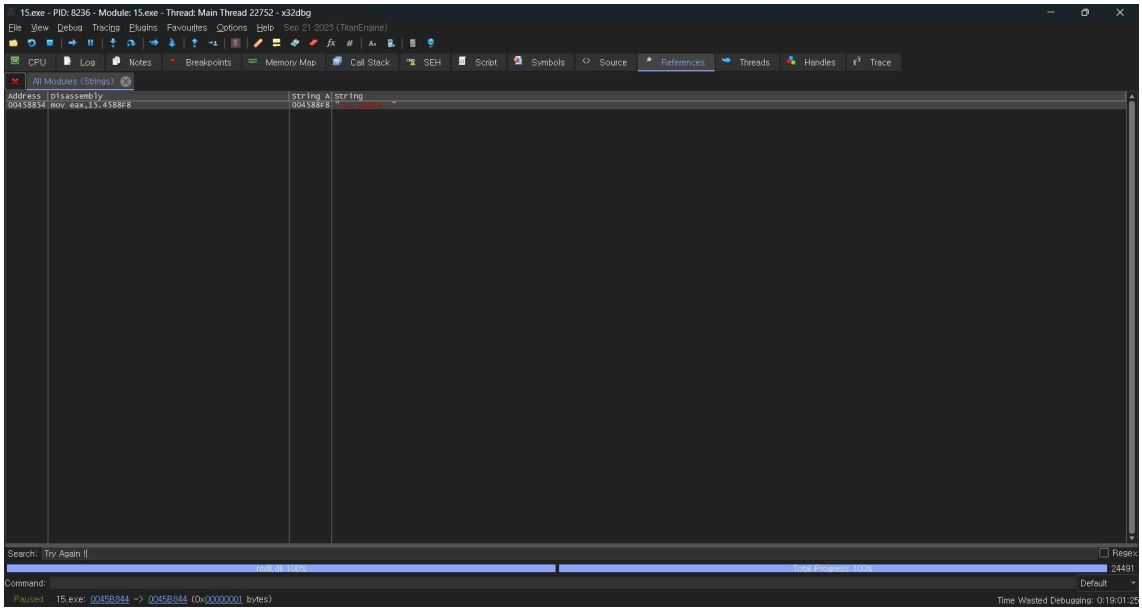
[15]

풀이 : 디버깅을 하기에 앞서 프로그램을 실행하여 임의의 값을 시리얼 값으로 입력해본다. 그러면 Try Again ! 이라는 메시지를 확인할 수 있다.



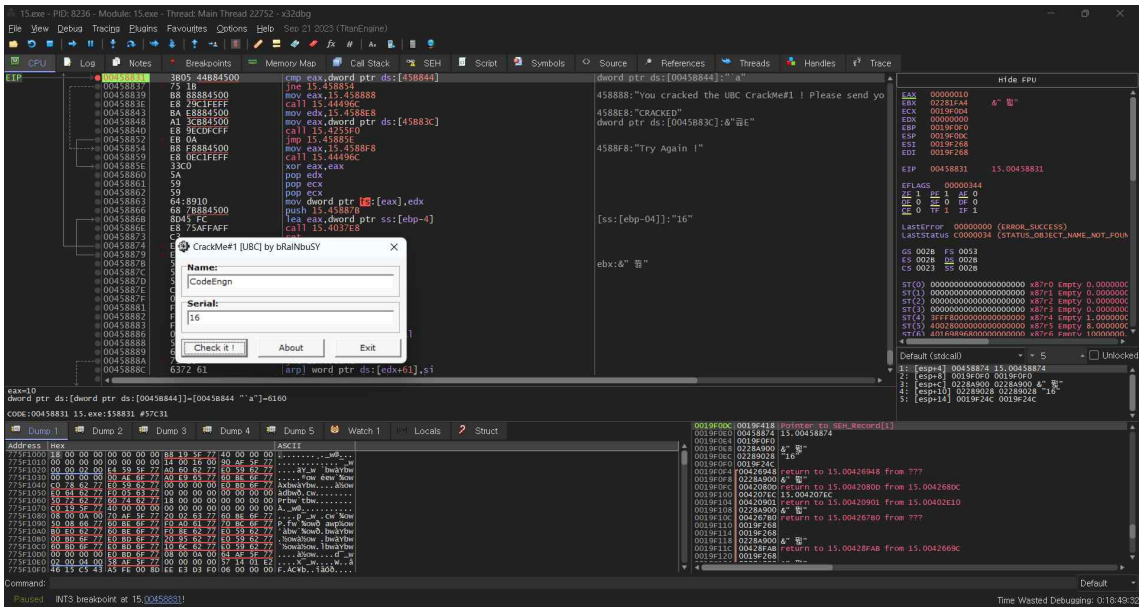
<프로그램 실행>

15.exe를 x64dbg로 열어서 Try Again ! 이라는 문자열을 검색하여 해당 주소로 이동한다. 그러면 성공 메시지를 확인할 수 있다.



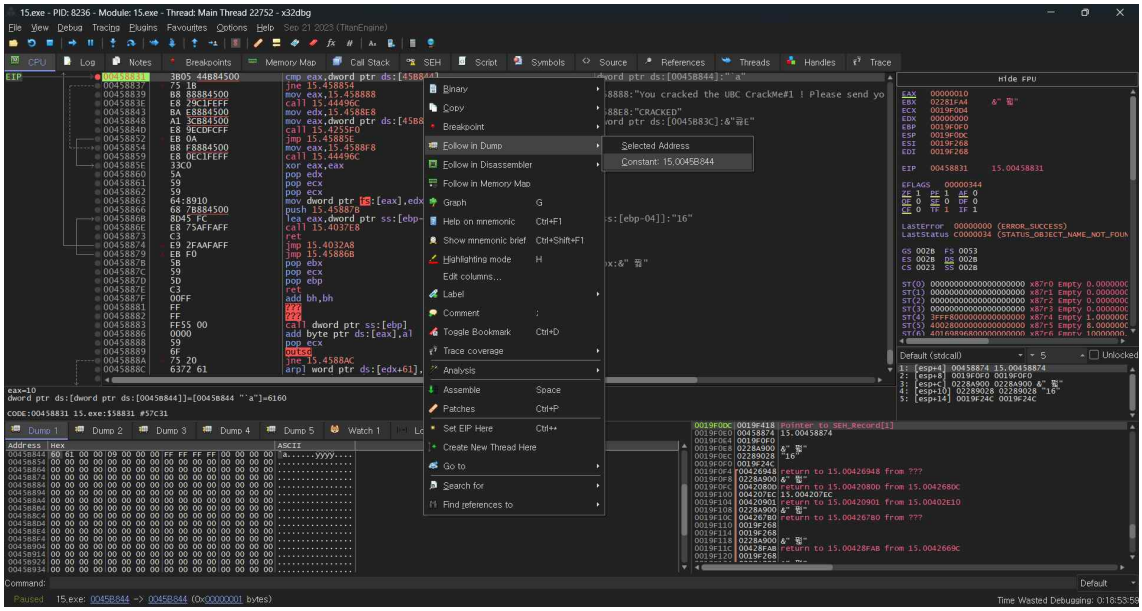
<“Try Again !” 문자열 검색>

또 성공 메시지 위 쪽으로 eax와 ds:[45B844] 데이터를 비교하는 cmp 명령어를 발견할 수 있다.*(**ds**란 세그먼트 레지스터 중 하나로 데이터 영역을 의미한다. 크기는 2바이트이다. **dword ptr**은 메모리에 있는 데이터를 나타낸다. 이때 나타내는 데이터의 크기는 4바이트이다.)* 해당 명령어에 브레이크 포인트를 걸어두고 다시 실행하여 시리얼 값으로 16을 입력한다. 그러면 eax에 10이 저장됨을 확인할 수 있다.

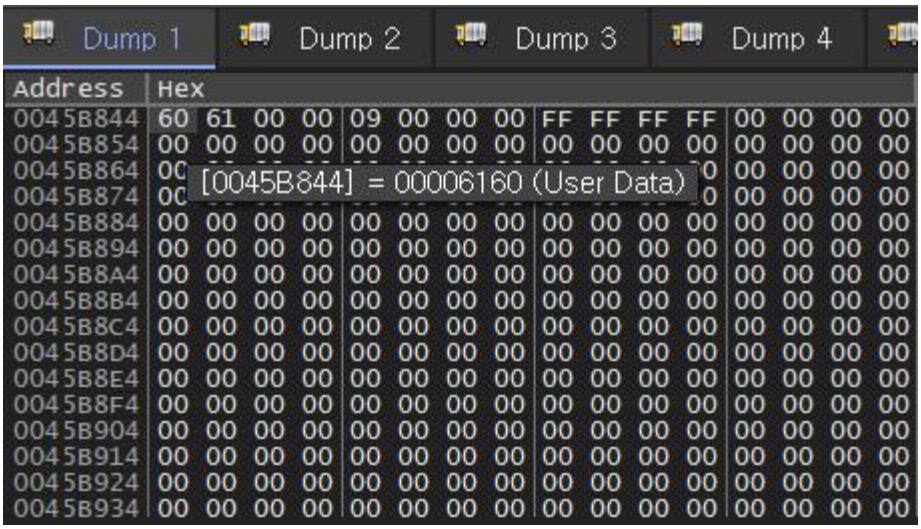


<시리얼 값으로 16을 입력 후, 변화된 eax 값>

내가 입력한 값을 16진수로 변환시키고 ds:[45B844] 데이터와 비교한 뒤 같으면 성공 메시지를 출력하는 것으로 추측된다. ds:[45B844] 데이터를 확인하기 위해 15.0045B844의 덤프를 따라간다. 그러면 해당 데이터가 0x6160임을 알 수 있다.

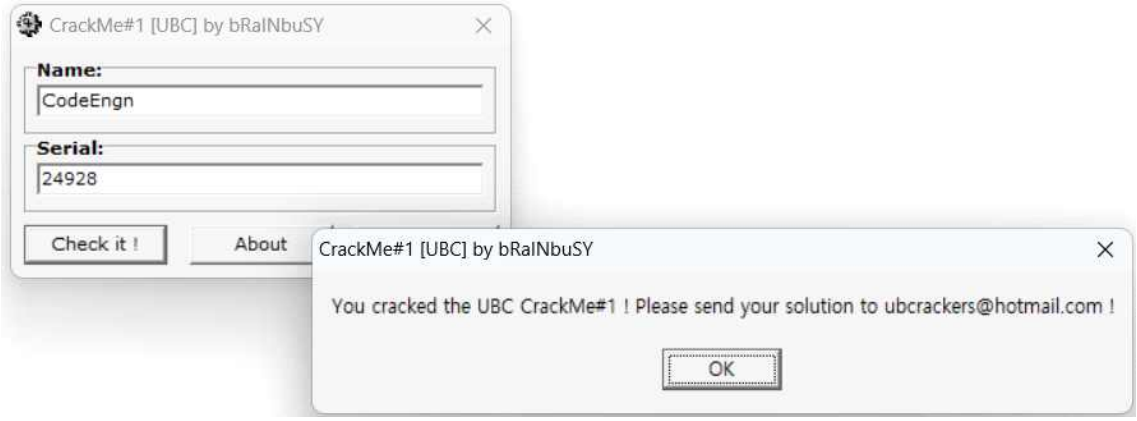


<ds:[45B844] 덤프 따라가기>



<ds:[45B844] 데이터 확인(리틀엔디안)>

0x6160을 10진수로 변환하여 시리얼 값으로 입력해보면 성공 메시지를 확인할 수 있다.



<성공 메시지 출력>

답 : 24928