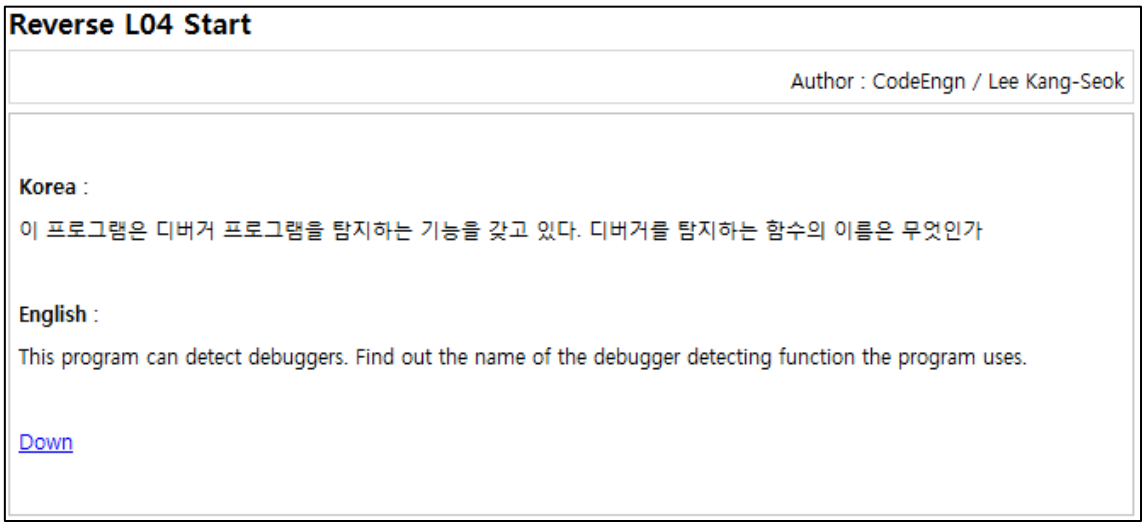
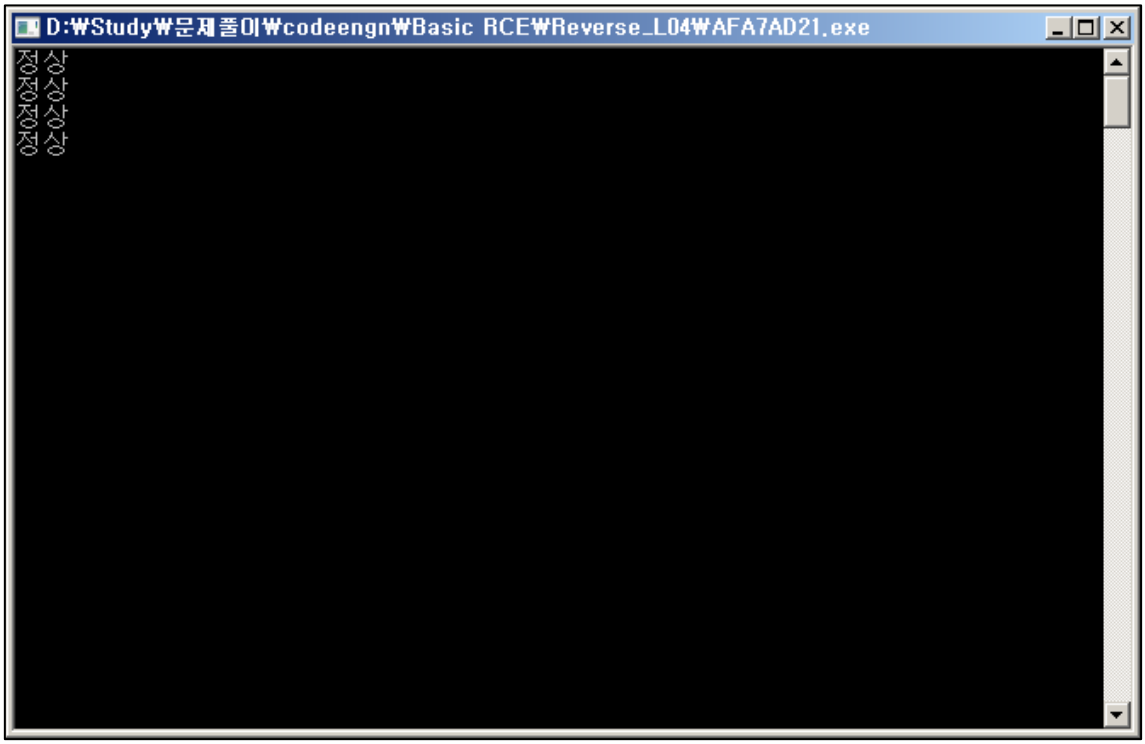
CodeEngn Reverse Challenge

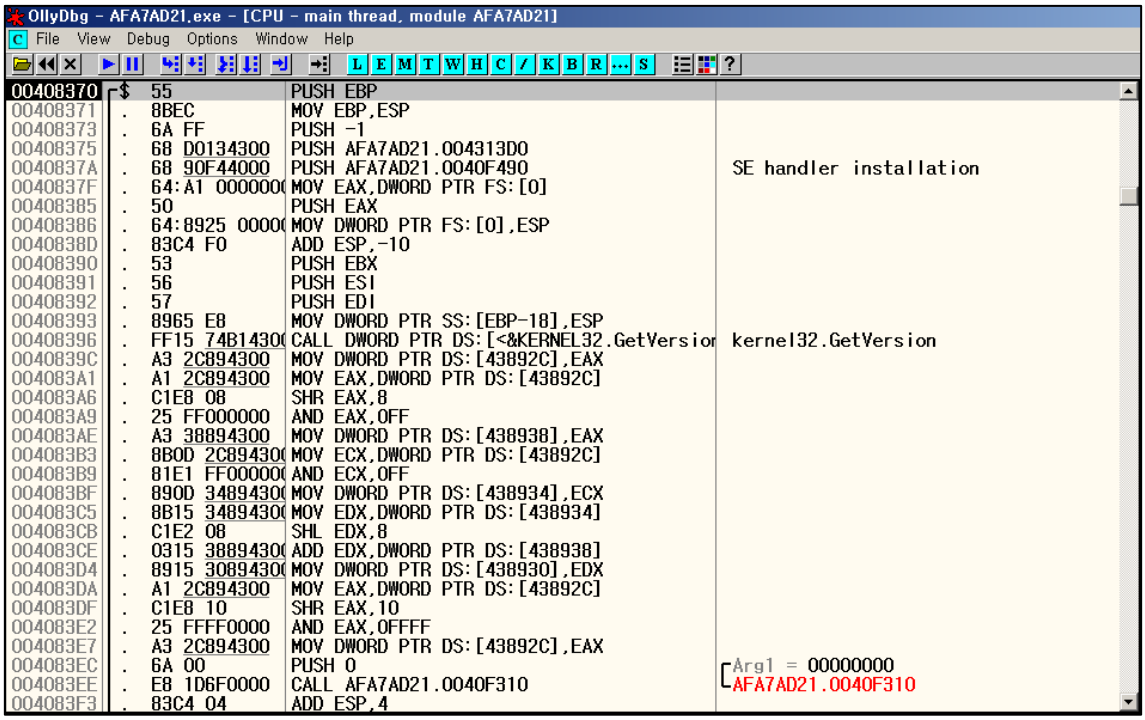
Basic RCE #4



디버거 탐지 함수 이름을 묻고 있습니다. 우선 프로그램을 독립으로 실행했을 때와 올리디버거에 붙여서 실행했을 때의 결과가 달랐습니다.



디버거에 붙여서 실행하지 않으면 위와 같이 정상메시지가 디버거에 붙여서 붙여서 실행했을 시 "디버깅 당함" 이라는 메시지가 나타났습니다.

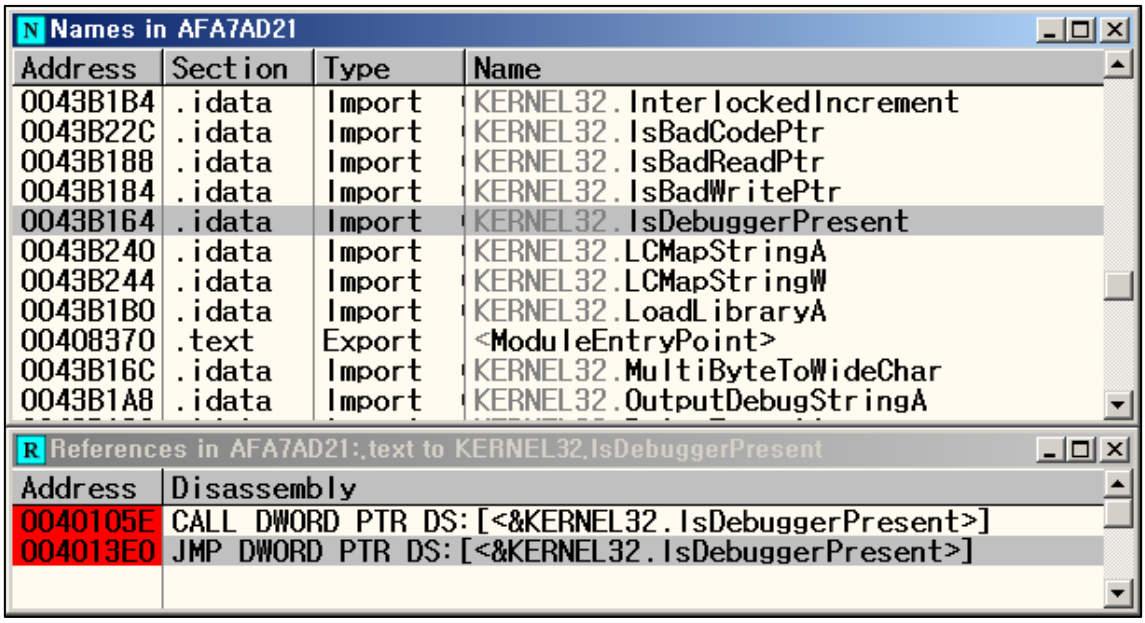


프로그램 동작이 단순한 것을 확인했기 때문에 step by step 접근으로 알아내려고 삽을 들었지만 결국 디버깅을 감지하는 함수를 찾을 수 없어 IDA 로 의사 코드를 얻었습니다.

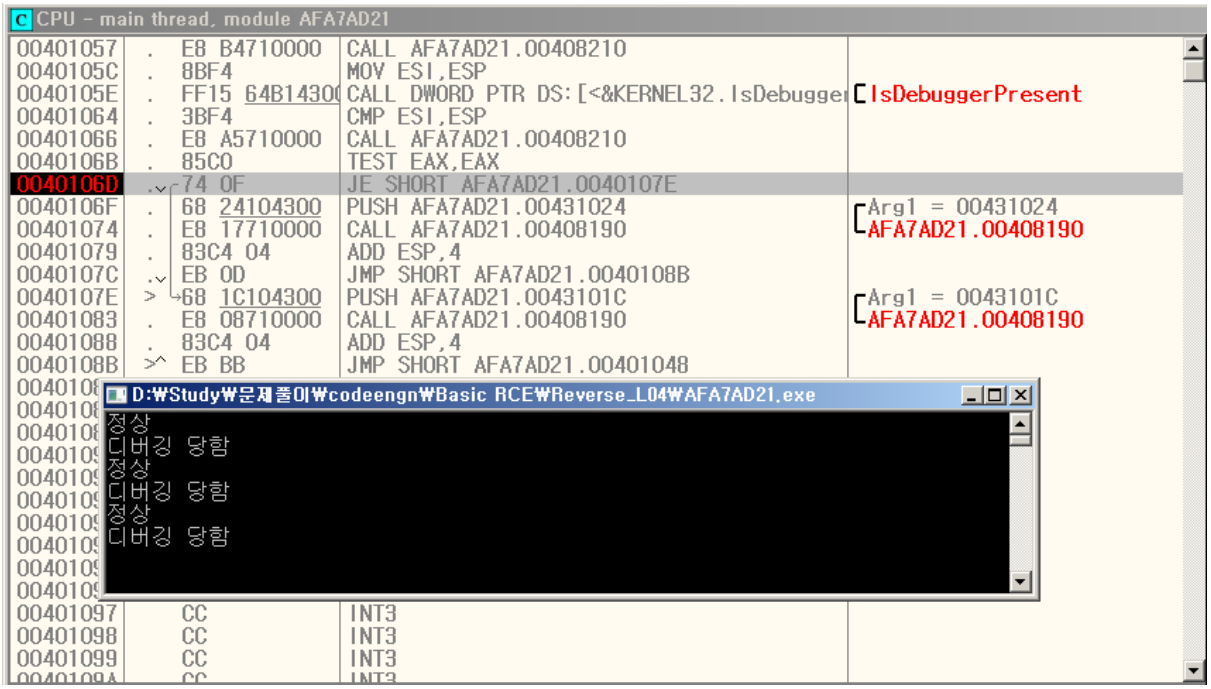
```
void __cdecl main_0()
{
  int v0; // eax@2
  int v1; // eax@2
  int v2; // [sp+0h] [bp-4Ch]@2
  char v3; // [sp+Ch] [bp-40h]@1
  int v4; // [sp+4Ch] [bp+0h]@2

  memset(&v3, -858993460, 0x40u);
  while ( 1 )
  {
    Sleep(0x3E8u);
    _chkesp(&v2 == &v2, v0, (unsigned int)&v4);
    v1 = IsDebuggerPresent();
    if ( _chkesp(&v2 == &v2, v1, (unsigned int)&v4) )
      printf("");
    else
      printf("정");
  }
}
```

의사코드는 예상했던 데로 단순했습니다. 한글이 깨져서 메시지 출력 부분이 제대로 표시되지 않지만 "정" 이라고 표시된 부분이 "정상"을 출력하는 것으로 예상할 수 있습니다. 그리고 눈에 띄는 IsDebuggerPresent 함수를 올리디버거로 확인해 보았습니다.



Ctrl + N 기능으로 심볼 이름 리스트에서 isDebuggerPresent 함수 이름을 확인 할 수 있었고 Find references to import 기능으로 심볼 이름이 참조되는 위치를 찾아 브레이크 포인트를 설정하고 분기시의 참조되는 플래그를 임의로 조작해서 "정상", "디버깅 당함" 메시지를 선택적으로 출력할 수 있었습니다.



명령어 JE를 JMP로 변경 후 바이너리를 새로 생성한 후 디버거에 붙여서 실행해보니 "정상" 메시지만 확인 될 뿐 디버깅 여부를 확인하지 못했습니다. 이번 문제의 정답은 isDebuggerPresent 입니다.

Team B10S letmeIn