Crack me 문제 보고서

|  |  |
| --- | --- |
| 문제명 | CodeEngn Reverse2 L08(WarRock #3) |
| 파일명 | Advance 08.exe |
| 풀이자 | 서치원(2011003155) |
| 풀이 기간 | 2016/02/20 ~ 2017/02/26 |
| 문제 내용 | Serial이 5D88-53B4-52A87D27-1D0D-5B09 일때의 name을 구하시오.  \*name은 2자리이다. |

문제 풀이:

문제 파일을 실행 시 창 하나가 뜨게됩니다.



key 값이 정해져 있을 때 알맞은 name을 구하는 문제입니다.

PEiD로 확인한 결과 따로 Packing 된 것은 없었습니다.

먼저 정답 확인 위치를 찾기 위해 referenc text 기능을 사용해 봅니다.



확인한 결과 정답 출력 부분이 있어 그 부분으로 이동하였습니다.

위의 코드 부분을 보면 정답 확인 루틴이 하나의 메소드로 되어있습니다.

메소드의 루틴을 보면 입력한 name 값을 가져와 길이를 비교하는데 3자 이상 30자 아래이어야 합니다.

문제의 조건에서 name의 길이는 2자리였으므로 3자리 이상의 조건을 걸때의 값을 2로 바꿔줍니다.



위의 화면에서 EAX,3을 EAX,2로 바꿔주면 2자리 이름도 무사히 넘어가게 됩니다.





그 후, 진행해보면 45BB9B의 메소드 CALL 후에 EDX에 어떤 루틴으로 인해 생성된 키값이 불러와집니다. 그 후, EAX에 입력한 키값이 불러와지고 45BBA4 주소에서 CALL된 메소드의 결과에 따라 정답 여부를 가리므로 입력한 이름에 의해 계산된 키 값이 "5D88-53B4-52A87D27-1D0D-5B09"과 같아야 할것입니다.



이름을 통해 키값을 생성하는 루틴은 매우 길게 되어 있습니다.

전체 루틴은 4개의 작은 반복문을 돈 후 나머지 처리를 한 후 종료되는데 각 반복문에서는 입력된 이름을 한글자씩 읽으면서 계산해 5구간의 키 값 중 1,2,4,5번 구간의 키 값을 만들어내게 됩니다.

계산을 쉽게 하기 위해 첫번째 루틴에서 생성된 첫번째 구간만으로 알맞은 이름을 계산할 수 있는지 프로그램을 짜 테스트해봅니다.



첫번째 반복문의 구간입니다. ESI에 이름의 한 글자를 가져온 후, 여러 연산을 한 후, EDX에 저장합니다. 이렇게 입력한 이름의 모든 글자의 값으로 계산한 후, 키 값의 첫번째값에 ESI 값 중 후위 4자리를 대입합니다. 이 루틴을 C++ 프로그램으로 짠 후, 이름값 00 ~ zz 값을 전부 대입해봅니다.

전부 대입해서 계산한 결과 'C6' 값이 유일하게 키값의 첫번째 값 0x5D88을 생성하므로 이것이 알맞은 이름일것입니다.



정답을 확인한 화면입니다.