Crack me 문제 보고서

|  |  |
| --- | --- |
| 문제명 | CodeEngn Reverse2 L09 |
| 파일명 | Advance 09.exe |
| 풀이자 | 서치원(2011003155) |
| 풀이 기간 | 2016/02/20 ~ 2017/02/26 |
| 문제 내용 | 비밀번호를 구하시오 |

문제 풀이:

문제 파일을 실행 시 창 하나가 뜨게됩니다.



콘솔 창에서 이름과 비밀번호를 입력하는 문구가 나오고 입력하면 정답을 계산하게 됩니다. 문제와는 달리 이름도 입력받기에 이름의 값이 비밀번호와 관련이 있을 경우 임의의 이름을 정해 진행합니다.

따로 packing은 되어 있지 않습니다.



진행을 하다보면 211A40 주소에서 CALL을 하고 입력창이 뜬 후, 2개의 값을 입력 받고 바로 정답 계산을 하므로 이 메소드의 내부에서 정답 처리를 하는 것 같습니다. 내부로 진입해봅니다.



내부로 진입하면 이름과 비밀번호를 입력받는 부분이 보입니다.

2개의 값을 입력 후에는 9713FF에서 971000 메소드로 진입합니다.



메소드로 진입 후 아래 코드를 보면 "DonaldDuck"이란 단어가 이름으로 입력한 "asdf"와 같이 각각 ECX, EAX 레지스터에 등록된 후, 밑에서 비교되는 것을 볼 수 있습니다. 비교되어 다르면 오답으로 점프하므로 이름은 "DonaldDuck"이어야 할 것입니다.



이름을 "DonaldDuck"으로 입력한 결과 오답이 나왔습니다. 비밀번호 쪽에서 틀렸을 가능성이 있으니 비밀번호를 사용하는 구간이 있는지를 검사해봅니다.



먼저 정답 출력 구간을 보면 마지막으로 2번의 검사를 하게 되는데 EBX 레지스터 값이 0이면 오답으로 넘어가고 ESP+0xB 주소의 값이 0이면 오답으로 넘어갑니다.



문제는 두 번째 검사에서 검사시의 스택 값 + 0xB의 값은 무조건 0의 값을 가지기 때문에 어떤 값을 넣어도 이 검사에서 오답으로 넘어가게 됩니다. 따라서 이 검사는 NOP 처리를 해야할 것입니다.



다시 위쪽의 코드를 보면 먼저 반복문이 보이게 되는데 ECX에 12A31F8값을 넣고 입력한 이름과 비교해 값이 같으면 점프를 하여 EAX 값이 변경되지 않은 상태에서 SETE BL를 실행하므로 EBX에 값이 1이 들어가 위의 검사에서 첫 번째 검사를 통과할 수 있습니다.



또 13A104B 지점에서 EAX에 특정 값을 불러와 비교하는데 위의 0x4D2는 비밀번호를 1234로 입력했을시 나오는 16진수 값으로 여기서 비밀번호 비교를 하게 되는것으로 보입니다. 즉, 정답 중 가능한 값은(두번째 검사가 NOP 처리가 되었을 경우) 이름이 0x013A331F8을 가지면 비밀번호와 상관없이 정답으로 넘어가게 되고 이름이 "DonaldDuck"이고 비밀번호가 0x0088228F를 가져도 정답이 될 것입니다.