Crack me 문제 보고서

|  |  |
| --- | --- |
| 문제명 | WarRock's Keygen 03 |
| 파일명 | WarRock 03.exe |
| 풀이자 | 서치원(2011003155) |
| 풀이 기간 | 2016/01/30 ~ 2017/02/05 |
| 문제 내용 | \* Key 값이 BEDA-2F56-BC4F4368-8A71-870B 일때 Name은 무엇인가  \* 조건을 name이 1자리만 받아도 되도록 코드를 패치해도 된다. |

문제 풀이:

문제 파일을 실행 시 창 하나가 뜨게됩니다.



이름과 key를 입력하는 칸이 있고 check하는 버튼이 있습니다.

check 버튼을 누를시 정답이 아니면 별다른 반응이 없습니다.

packing은 되어있지 않고 Delphi로 작성되었습니다.



ollydbg로 실행하고 referenced string으로 검색한 상황입니다. 정답을 표시하는 문자열이 있으므로 이동합니다.



이동하면 정답 확인 루틴이 확인 됩니다. 45B8AD에서 jump 분기가 있으므로 404C3C 함수의 결과에 따라 정답이 결정됨을 유추할 수 있습니다.

먼저 45B863까지는 이름의 길이 검사를 하는 것 같으므로 넘기고 45B875로 jump하는 것이 에러 처리에서 넘어가는 부분이므로 이 주소부터 분석합니다.

이 위치에서부터 진행을 하다보면 함수 45B850 에서 어느 문자열이 생성되어 저장됨을 확인할 수 있습니다. 즉, 이 함수 내부의 루틴에 따라 맞는 name을 알 수 있을 것입니다.



함수 내부로 들어온 상황입니다. 진행을 하다보면 45B89D 지점부터 name을 한글자씩 읽어서 일정 루틴을 돌아 각 글자마다 key값을 생성해냅니다.

이쪽의 루틴을 알기 쉽게 하기 위해 문제에서 조건이 name을 1자리만 받을 수 있게 처리하는 것이 걸려있습니다. 45B824 주소에서 name의 길이를 3자리 이상으로 받게 처리하고 있으므로 먼저 그 위치를 패치합니다.



name을 1글자만 받을 수 있게 된 상태에서 다시 serial을 생성하는 루틴으로 들어옵니다. 진행을 하면 45B8C0에서 EDX에 계산한 ESI 값을 저장하고 이후 루틴에서 사용하는데 이때 앞의 4자리 값만을 사용하게 됩니다. 입력하는 serial 값이

BEDA-2F56-BC4F4368-8A71-870B 이므로 위의 루틴 이후에 앞의 4자리 수 BEDA를 만드는 1글자를 찾으면 될것입니다. 글자 a 부터 0까지 돌아가면서 검증을 해보면 맞는 값은 "F"가 됨을 알 수 있습니다.



패치 후 정답을 확인한 화면입니다.