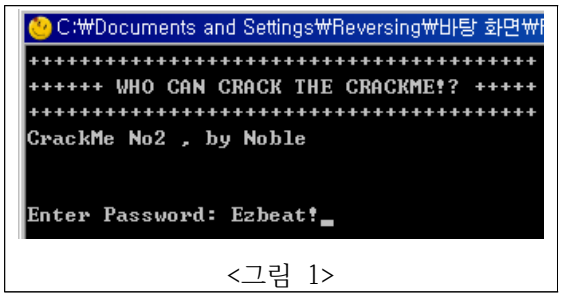
- Ezbeat -
풀어봅시다~!
레포트가 너무 많네요... 흑흑.. 짬짬히 ..~!

프로그램을 실행 시켜서 해본 결과

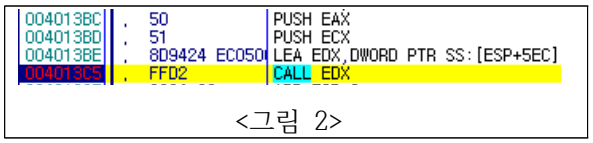


<그림 1>

패스워드를 치라네요..? 패스워드를 치고 엔터를 치니 프로그램이 종료가 되었습니다.

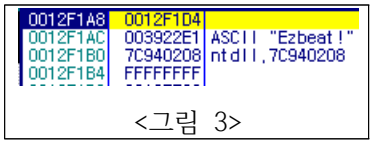
무슨 오류 창이나 메시지도 없이 그냥 종료되니 허무하군요 ^^;

차근차근 봐봅시다.
프로그램이 종료되는 함수 시점부터 살펴보았습니다. 해당 부분은



<그림 2>

위와 같은 부분이 되겠습니다. 함수를 호출 전에 위에서 EAX와 ECX를 PUSH해주고 있는데 EAX에는 제가 입력한 패스워드가 들어있었습니다.
스택 영역을 살펴보면



<그림 3>

뭐 이렇게 되어있군요.

이제 해당 함수 안으로 들어가 보겠습니다.
다 볼 필요는 없고 중요한 부분만 설명하고 끝내겠습니다.

```
0012F7BC    8B45 0C         MOV EAX,DWORD PTR SS:[EBP+C]              //[EBP+C]에는 아까 PUSH한 패스워드 존재
0012F7BF    0FBE08          MOVSX ECX,BYTE PTR DS:[EAX]               //한 글자 씩 꺼내서 비교
0012F7C2    83F9 43         CMP ECX,43                                 //첫 글자와 0x43비교
0012F7C5    0F85 F7000000   JNZ 0012F8C2          //다르면 점프 인데 점프해서 가보면 ExitProcess 존재
0012F7CB    8B45 0C         MOV EAX,DWORD PTR SS:[EBP+C]
0012F7CE    0FBE48 01       MOVSX ECX,BYTE PTR DS:[EAX+1]
0012F7D2    83F9 52         CMP ECX,52                                 //두 번째 글자와 0x52비교! 아래 쭉
                                                                        같은 루틴 이므로 설명은 생략하겠습니다.
0012F7D5    0F85 E7000000   JNZ 0012F8C2
0012F7DB    8B45 0C         MOV EAX,DWORD PTR SS:[EBP+C]
0012F7DE    0FBE48 02       MOVSX ECX,BYTE PTR DS:[EAX+2]
```

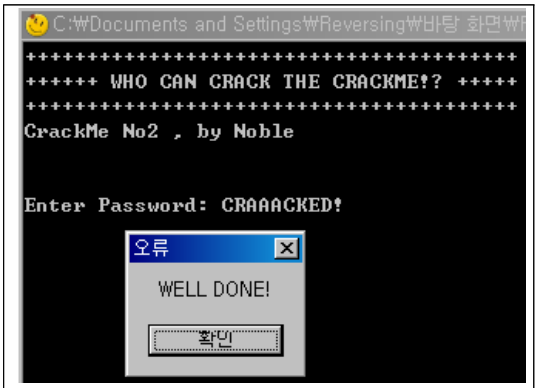
```
0012F7E2    83F9 41          CMP ECX,41
0012F7E5    0F85 D7000000    JNZ 0012F8C2
0012F7EB    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F7EE    0FBE48 03        MOVSX ECX,BYTE PTR DS:[EAX+3]
0012F7F2    83F9 41          CMP ECX,41
0012F7F5    0F85 C7000000    JNZ 0012F8C2
0012F7FB    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F7FE    0FBE48 04        MOVSX ECX,BYTE PTR DS:[EAX+4]
0012F802    83F9 41          CMP ECX,41
0012F805    0F85 B7000000    JNZ 0012F8C2
0012F80B    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F80E    0FBE48 05        MOVSX ECX,BYTE PTR DS:[EAX+5]
0012F812    83F9 43          CMP ECX,43
0012F815    0F85 A7000000    JNZ 0012F8C2
0012F81B    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F81E    0FBE48 06        MOVSX ECX,BYTE PTR DS:[EAX+6]
0012F822    83F9 4B          CMP ECX,4B
0012F825    0F85 97000000    JNZ 0012F8C2
0012F82B    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F82E    0FBE48 07        MOVSX ECX,BYTE PTR DS:[EAX+7]
0012F832    83F9 45          CMP ECX,45
0012F835    0F85 87000000    JNZ 0012F8C2
0012F83B    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F83E    0FBE48 08        MOVSX ECX,BYTE PTR DS:[EAX+8]
0012F842    83F9 44          CMP ECX,44
0012F845    75 7B            JNZ SHORT 0012F8C2
0012F847    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F84A    0FBE48 09        MOVSX ECX,BYTE PTR DS:[EAX+9]
0012F84E    83F9 21          CMP ECX,21
0012F851    75 6F            JNZ SHORT 0012F8C2
0012F853    8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F856    0FBE48 0A        MOVSX ECX,BYTE PTR DS:[EAX+A]
0012F85A    85C9             TEST ECX,ECX
0012F85C    74 13            JE SHORT 0012F871              //모든 문자가 맞다면 이 점프문을 실행
0012F85E    8BF4             MOV ESI,ESP
0012F860    6A 01            PUSH 1
0012F862    8B45 08          MOV EAX,DWORD PTR SS:[EBP+8]
0012F865    8B48 08          MOV ECX,DWORD PTR DS:[EAX+8]
0012F868    FFD1             CALL ECX
0012F86A    3BF4             CMP ESI,ESP
0012F86C    E8 83030000      CALL 0012FBF4
0012F871    C745 F4 0100000>MOV DWORD PTR SS:[EBP-C],1
0012F878    C645 E0 57       MOV BYTE PTR SS:[EBP-20],57
0012F87C    C645 E1 45       MOV BYTE PTR SS:[EBP-1F],45
0012F880    C645 E2 4C       MOV BYTE PTR SS:[EBP-1E],4C
0012F884    C645 E3 4C       MOV BYTE PTR SS:[EBP-1D],4C
0012F888    C645 E4 20       MOV BYTE PTR SS:[EBP-1C],20
0012F88C    C645 E5 44       MOV BYTE PTR SS:[EBP-1B],44
0012F890    C645 E6 4F       MOV BYTE PTR SS:[EBP-1A],4F
0012F894    C645 E7 4E       MOV BYTE PTR SS:[EBP-19],4E
0012F898    C645 E8 45       MOV BYTE PTR SS:[EBP-18],45
0012F89C    C645 E9 21       MOV BYTE PTR SS:[EBP-17],21
0012F8A0    33C0             XOR EAX,EAX
0012F8A2    8845 EA          MOV BYTE PTR SS:[EBP-16],AL
0012F8A5    8BF4             MOV ESI,ESP
0012F8A7    6A 00            PUSH 0
0012F8A9    6A 00            PUSH 0
0012F8AB    8D45 E0          LEA EAX,DWORD PTR SS:[EBP-20]
```

```
0012F8AE    50              PUSH EAX
0012F8AF    6A 00           PUSH 0
0012F8B1    8B4D 08         MOV ECX,DWORD PTR SS:[EBP+8]
0012F8B4    8B51 0C         MOV EDX,DWORD PTR DS:[ECX+C]
0012F8B7    FFD2            CALL EDX                        ;User32.MessageBoxA     Well Done출력!
0012F8B9    3BF4            CMP ESI,ESP
0012F8BB    E8 34030000     CALL 0012FBF4
0012F8C0    EB 13           JMP SHORT 0012F8D5
0012F8C2    8BF4            MOV ESI,ESP
0012F8C4    6A 01           PUSH 1
0012F8C6    8B45 08         MOV EAX,DWORD PTR SS:[EBP+8]
0012F8C9    8B48 08         MOV ECX,DWORD PTR DS:[EAX+8]
0012F8CC    FFD1            CALL ECX                            ; kernel32.ExitProcess
0012F8CE    3BF4            CMP ESI,ESP
```

결국 Hex 값으로

0x43	0x52	0x41	0x41	0x41	0x43	0x4B	0x45	0x44	0x21
C	R	A	A	A	C	K	E	D	!

답은 CRAAACKED! 가 됩니다.



<그림 4>

팝업창이 떠서 좋았는데 오류라고 되어 놀랬지만 WELL DONE!을 보고 다시 기뻐했네요 ^^