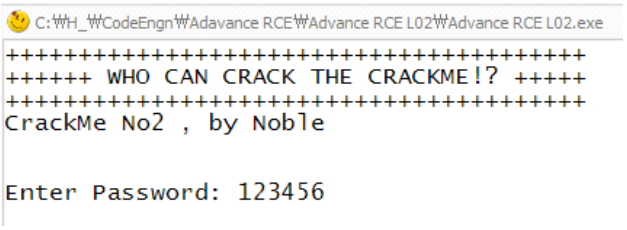
Advance RCE L02

2010년 9월 21일 화요일

오전 1:22

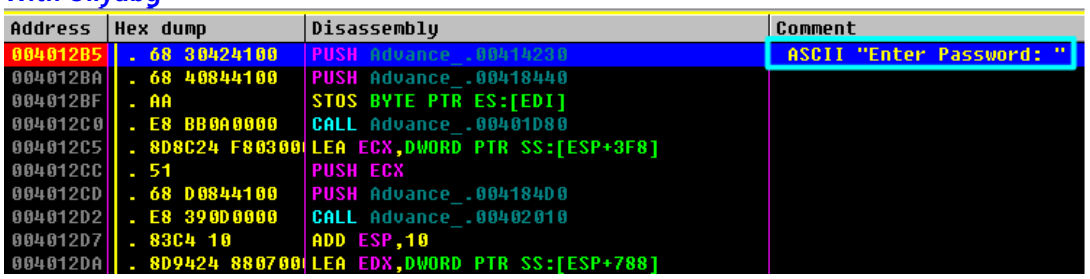
파일 확인

프로그램 실행



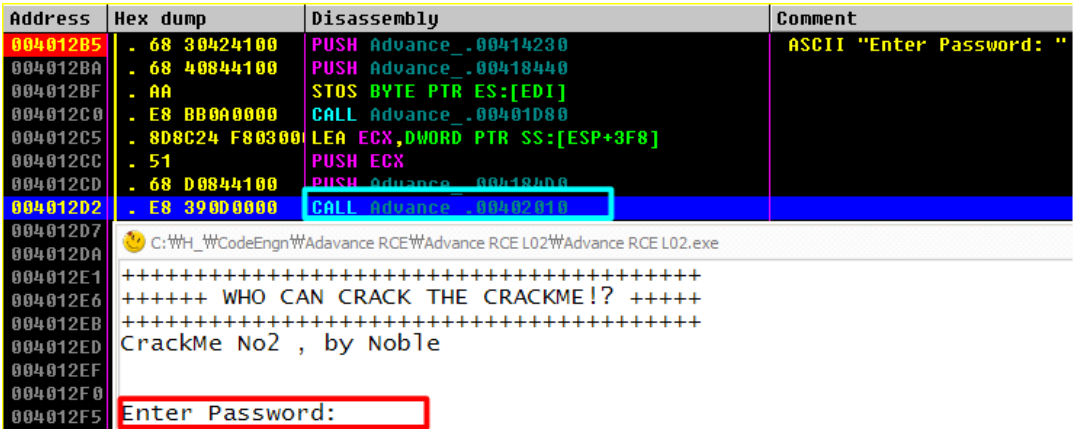
Password 를 입력받아, 틀릴 시에 창이 꺼지도록 만들어진 프로그램이다.

With Ollydbg



프로그램 실행시에 나온 구문인 **"Enter Password : " 를 All referenced text strings 를 통해 찾아가서 Breakpoint 를 설정**하였다.

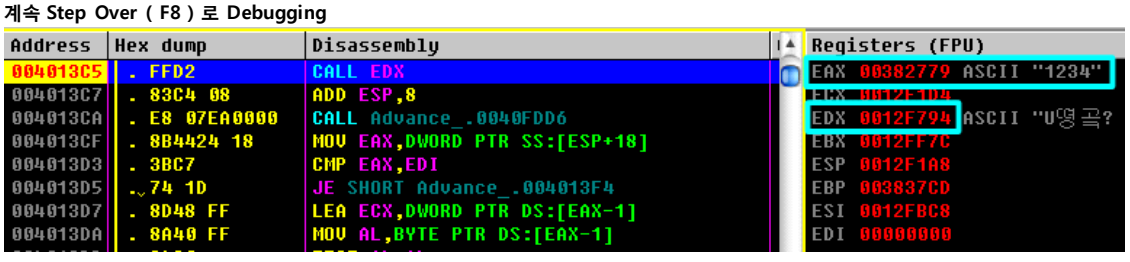
실행 후, Step Over (F8) 로 Debugging



Call 00402010 에서 사용자의 입력을 기다리고 있으므로, 입력 함수 임을 알 수 있었다.

- **Call 00401D80 은 출력함수** 일 것이다.

계속 Step Over (F8) 로 Debugging



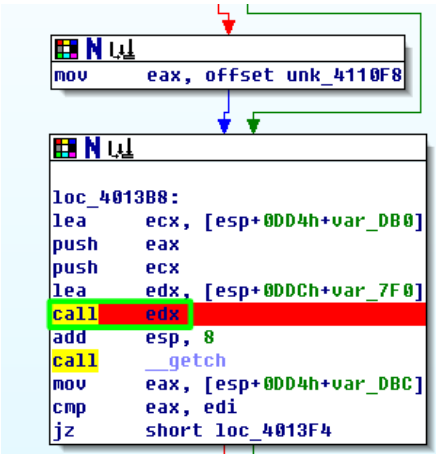
Call EDX 이후 Program 이 종료 되는 것을 확인 및 입력한 값을 변수로 받는 것을 확인

Step Into (F7) 로 EDX (0012F794) 지점 Code 확인

```
Address   Hex dump          Disassembly
0012F7BF  0FBE08            MOVSX ECX,BYTE PTR DS:[EAX]
0012F7C2  83F9 43           CMP ECX,43
0012F7C5  0F85 F7000000     JNZ 0012F8C2
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
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
```

ASCII 문자로 보이는 값과 입력한 값을 한글자씩 비교하는 Code 이다.

With IDA Pro



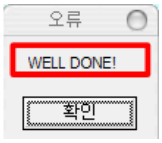
Call EDX 부분에 BreakPoint 를 설정한 후, IDA 에서 지원하는 Debugging 모드를 실행 시켜보았다.

```
Stack[00000D14]:0012F7A0 lea     edi, [ebp-0E4h]
Stack[00000D14]:0012F7A6 mov     ecx, 39h
Stack[00000D14]:0012F7AB mov     eax, 0CCCCCCCCh
Stack[00000D14]:0012F7B0 rep stosd
Stack[00000D14]:0012F7B2 mov     eax, dword ptr ds:loc_406008
Stack[00000D14]:0012F7B7 xor     eax, ebp
Stack[00000D14]:0012F7B9 mov     [ebp-4], eax
Stack[00000D14]:0012F7BC mov     eax, [ebp+0Ch]
Stack[00000D14]:0012F7BF movsx   ecx, byte ptr [eax]
Stack[00000D14]:0012F7C2 cmp     ecx, 43h
Stack[00000D14]:0012F7C5 jnz     loc_12
Stack[00000D14]:0012F7CB mov     eax, [
Stack[00000D14]:0012F7CE movsx   ecx, b
Stack[00000D14]:0012F7D2 cmp     ecx, 5
Stack[00000D14]:0012F7D5 jnz     loc_12
Stack[00000D14]:0012F7DB mov     eax, [
Stack[00000D14]:0012F7DE movsx   ecx, b
Stack[00000D14]:0012F7E2 cmp     ecx, 4
Stack[00000D14]:0012F7E5 jnz     loc_12
Stack[00000D14]:0012F7EB mov     eax, [
Stack[00000D14]:0012F7EE movsx   ecx, b
Stack[00000D14]:0012F7F2 cmp     ecx, 4
Stack[00000D14]:0012F7F5 jnz     loc_12
Stack[00000D14]:0012F7FB mov     eax, [
Stack[00000D14]:0012F7FE movsx   ecx, b
Stack[00000D14]:0012F802 cmp     ecx, 4
Stack[00000D14]:0012F805 jnz     loc_12
```

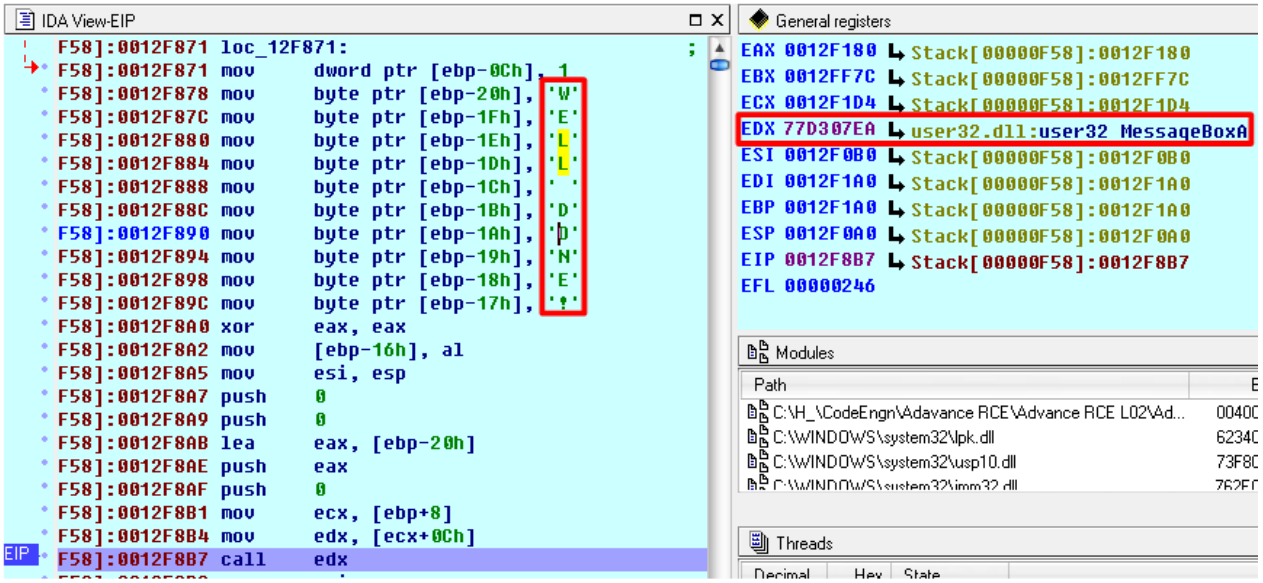
Use standard symbolic constant
67 H
103o
1000011b B
'C' R
Manual... Alt+F1
Create function... P
Undefine U
Synchronize with
Jump to IP
Set IP Ctrl+N
Run to cursor F4
Add breakpoint F2

아까 **ASCII 값이라고 예상 했던 값들을 하나하나 변환** 하여 보았다.

- **CRAAACKED! 라는 값이 나오게 되었고, 이를 입력**하여 보았다.



비교 후 **JMP 하는 곳을 찾아 본 결과 WELL DONE 이라는 구문을 찾을 수 있었고 MessageBoxA 함수를 호출하는 것을 확인**하였다.



답

CRAAACKED!