

## Advance RCE L02

2010년 9월 21일 화요일

오전 1:22

### 파일 확인



Advance RCE L02

### 프로그램 실행

```
C:\WH_\\CodeEngn\\Advance RCE\\Advance RCE L02\\Advance RCE L02.exe
+++++ WHO CAN CRACK THE CRACKME!? +++++
+++++ CrackMe No2 , by Noble

Enter Password: 123456
```

Password 를 입력받아, 틀릴 시에 창이 꺼지도록 만들어진 프로그램이다.

### With Ollydbg

Address	Hex dump	Disassembly	Comment
004012B5	. 68 30424100	PUSH Advance_.00414230	ASCII "Enter Password: "
004012BA	. 68 40844100	PUSH Advance_.00418440	
004012BF	. AA	STOS BYTE PTR ES:[EDI]	
004012C0	. E8 BB0A0000	CALL Advance_.00401D80	
004012C5	. 8D8C24 F80300	LEA ECX,DWORD PTR SS:[ESP+3F8]	
004012CC	. 51	PUSH ECX	
004012CD	. 68 D0844100	PUSH Advance_.004184D0	
004012D2	. E8 39000000	CALL Advance_.00402010	
004012D7	. 83C4 10	ADD ESP,10	
004012DA	. 8D9424 880700	LEA EDX,DWORD PTR SS:[ESP+788]	

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

실행 후, Step Over ( F8 ) 로 Debugging

Address	Hex dump	Disassembly	Comment
004012B5	. 68 30424100	PUSH Advance_.00414230	ASCII "Enter Password: "
004012BA	. 68 40844100	PUSH Advance_.00418440	
004012BF	. AA	STOS BYTE PTR ES:[EDI]	
004012C0	. E8 BB0A0000	CALL Advance_.00401D80	
004012C5	. 8D8C24 F80300	LEA ECX,DWORD PTR SS:[ESP+3F8]	
004012CC	. 51	PUSH ECX	
004012CD	. 68 D0844100	PUSH Advance_.004184D0	
004012D2	. E8 39000000	CALL Advance_.00402010	
004012D7			
004012DA			
004012E1			
004012E6			
004012EB			
004012ED			
004012EF			
004012F0			
004012F5			

```
C:\WH_\\CodeEngn\\Advance RCE\\Advance RCE L02\\Advance RCE L02.exe
+++++ WHO CAN CRACK THE CRACKME!? +++++
+++++ CrackMe No2 , by Noble

Enter Password:
```

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

Address	Hex dump	Disassembly	Registers (FPU)
004013C5	. FFD2	CALL EDI	EAX 00382779 ASCII "1234"
004013C7	. 83C4 08	ADD ESP,8	ECX 0012F1D4
004013CA	. E8 07EA0000	CALL Advance_.0040FDD6	EDX 0012F794 ASCII "U영곡?"
004013CF	. 8B4424 18	MOV EAX,DWORD PTR SS:[ESP+18]	EBX 0012FF7C
004013D3	. 3BC7	CMP EAX,EDI	ESP 0012F1A8
004013D5	. 74 10	JE SHORT Advance_.004013F4	EBP 003837CD
004013D7	. 8D48 FF	LEA ECX,DWORD PTR DS:[EAX-1]	ESI 0012FBC8
004013DA	. 8A40 FF	MOV AL,BYTE PTR DS:[EAX-1]	EDI 00000000

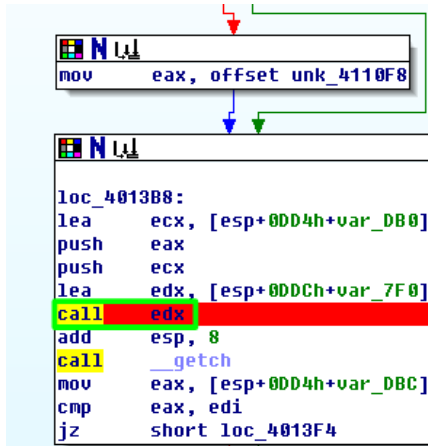
Call EDI 이후 Program 이 종료 되는 것을 확인 및 입력한 값을 변수로 받는 것을 확인

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

Address	Hex dump	Disassembly
0012F7BF	0FB008	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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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	0FB048 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.

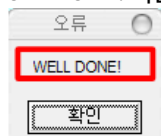
```

Stack[00000014]:0012F7A0 lea edi, [ebp-0E4h]
Stack[00000014]:0012F7A6 mov ecx, 39h
Stack[00000014]:0012F7AB mov eax, 0CCCCCCCCh
Stack[00000014]:0012F7B0 rep stosd
Stack[00000014]:0012F7B2 mov eax, dword ptr ds:loc_406008
Stack[00000014]:0012F7B7 xor eax, ebp
Stack[00000014]:0012F7B9 mov [ebp-4], eax
Stack[00000014]:0012F7BC mov eax, [ebp+0Ch]
Stack[00000014]:0012F7BF movsx ecx, byte ptr [eax]
Stack[00000014]:0012F7C2 cmp ecx, 4Ah
Stack[00000014]:0012F7C5 jnz loc_121
Stack[00000014]:0012F7CB mov eax, [
Stack[00000014]:0012F7CE movsx ecx, bl
Stack[00000014]:0012F7D2 cmp ecx, 5
Stack[00000014]:0012F7D5 jnz loc_121
Stack[00000014]:0012F7DB mov eax, [
Stack[00000014]:0012F7DE movsx ecx, bl
Stack[00000014]:0012F7E2 cmp ecx, 4
Stack[00000014]:0012F7E5 jnz loc_121
Stack[00000014]:0012F7EB mov eax, [
Stack[00000014]:0012F7EE movsx ecx, bl
Stack[00000014]:0012F7F2 cmp ecx, 4
Stack[00000014]:0012F7F5 jnz loc_121
Stack[00000014]:0012F7FB mov eax, [
Stack[00000014]:0012F7FE movsx ecx, bl
Stack[00000014]:0012F802 cmp ecx, 4
Stack[00000014]:0012F805 jnz loc_121

```

아까 ASCII 값이라고 예상 했던 값들을 하나하나 변환 하여 보았다.

- CRAAAKED! 라는 값이 나오게 되었고, 이를 입력하여 보았다.



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

IDA View-EIP

```

F58]:0012F871 loc_12F871:
F58]:0012F871 mov dword ptr [ebp-0Ch], 1
F58]:0012F878 mov byte ptr [ebp-20h], 'W'
F58]:0012F87C mov byte ptr [ebp-1Fh], 'E'
F58]:0012F880 mov byte ptr [ebp-1Eh], 'L'
F58]:0012F884 mov byte ptr [ebp-1Dh], 'L'
F58]:0012F888 mov byte ptr [ebp-1Ch], ' '
F58]:0012F88C mov byte ptr [ebp-1Bh], 'D'
F58]:0012F890 mov byte ptr [ebp-1Ah], ' '
F58]:0012F894 mov byte ptr [ebp-19h], 'N'
F58]:0012F898 mov byte ptr [ebp-18h], 'E'
F58]:0012F89C mov byte ptr [ebp-17h], '!'
F58]:0012F8A0 xor eax, eax
F58]:0012F8A2 mov [ebp-16h], al
F58]:0012F8A5 mov esi, esp
F58]:0012F8A7 push 0
F58]:0012F8A9 push 0
F58]:0012F8AB lea eax, [ebp-20h]
F58]:0012F8AE push eax
F58]:0012F8AF push 0
F58]:0012F8B1 mov ecx, [ebp+8]
F58]:0012F8B4 mov edx, [ecx+0Ch]
F58]:0012F8B7 call edx

```

General registers

```

EAX 0012F180 Stack[00000F58]:0012F180
EBX 0012FF7C Stack[00000F58]:0012FF7C
ECX 0012F1D4 Stack[00000F58]:0012F1D4
EDX 77D307EA user32.dll:user32 MessageBoxA
ESI 0012F0B0 Stack[00000F58]:0012F0B0
EDI 0012F1A0 Stack[00000F58]:0012F1A0
EBP 0012F1A0 Stack[00000F58]:0012F1A0
ESP 0012F0A0 Stack[00000F58]:0012F0A0
EIP 0012F8B7 Stack[00000F58]:0012F8B7
EFL 00000246

```

Modules

Path	E
C:\H_\CodeEngn\Advance RCE\Advance RCE L02\Ad...	0040C
C:\WINDOWS\system32\lpk.dll	6234C
C:\WINDOWS\system32\usp10.dll	73F8C
C:\WINDOWS\system32\imm32.dll	762FC

Threads

Decimal	Hex	State
---------	-----	-------

답  
CRAAAKED!