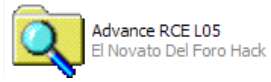
Advance RCE L05

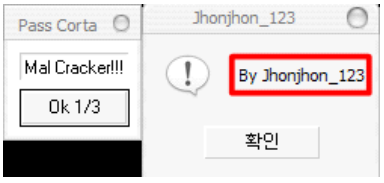
2010년 9월 21일 화요일

오전 1:22

파일 확인

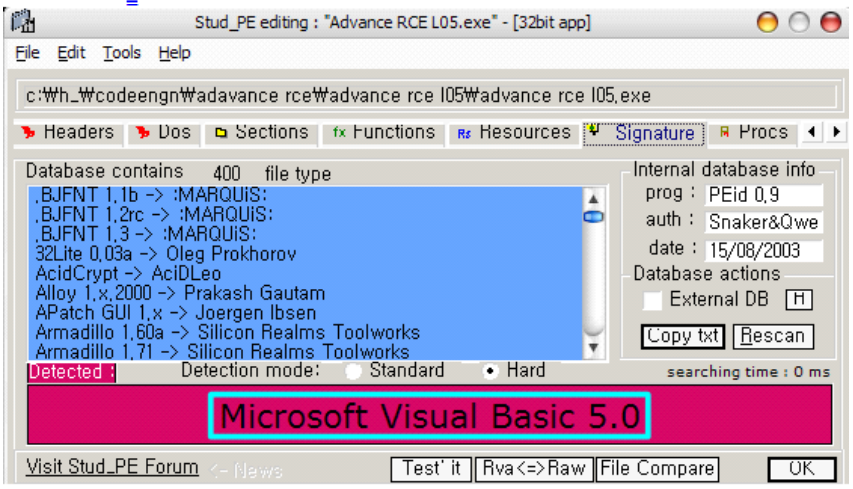


프로그램 실행



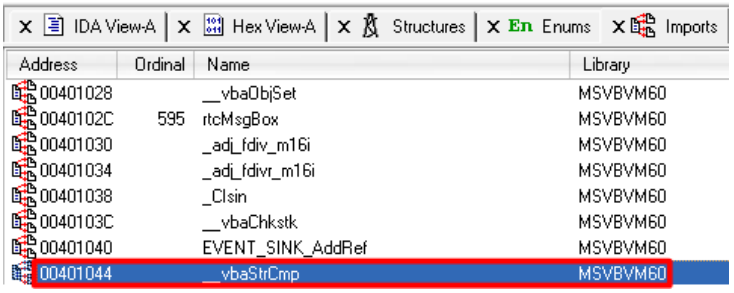
String 를 입력하여 맞지 않으면 " Mal Cracker!!! "라는 글과 " By Jhonjohn_123 " 이라는 MessageBox 를 호출한다.

With Stud_PE



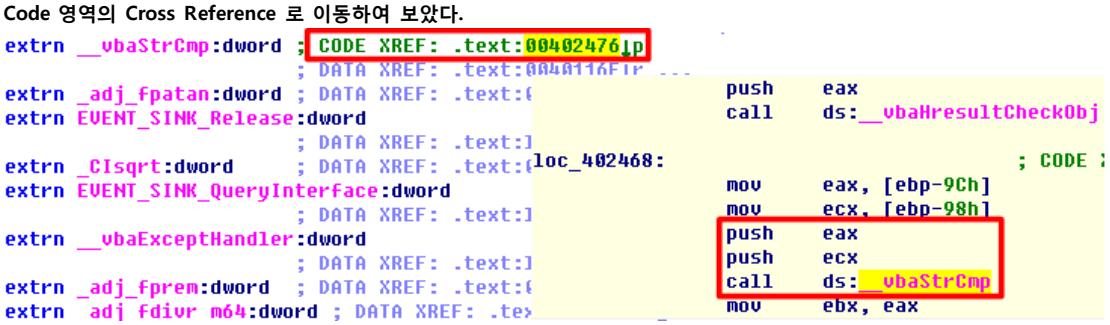
Visual Basic 5.0 으로 만들어진 프로그램이다.

With IDA Pro



Basic RCE L03과 같이 **vbaStrCmp** 라는 함수가 눈에 보였다.

Code 영역의 Cross Reference 로 이동하여 보았다.



Breakpoint 를 설정 후 Run 시켜 보았다.

```
.text:00402474 push    eax
.text:00402475 push    ecx
.text:00402476 call    ds:  vbaStrCmp
.text:0040247C mov     e|ecx=debug007:00159574
.text:0040247E lea     e|db  36h ; 6
.text:00402484 neg     e|db   0
.text:00402486 sbb     e|db  37h ; 7
.text:00402488 lea     e|db   0
.text:0040248E push    e|db  37h ; 7
.text:0040248F inc     e|db   0
.text:00402490 push    e|db  33h ; 3
.text:00402491 push    2|db   0
.text:00402493 neg     e|db  34h ; 4
.text:00402495 call    d|db   0
.text:0040249B lea     e|db  35h ; 5
.text:004024A1 lea     e|db   0
```

EAX 에는 사용자가 입력한 값이, ECX 에는 677345 가 들어가 있었다.

677345 입력 후 확인

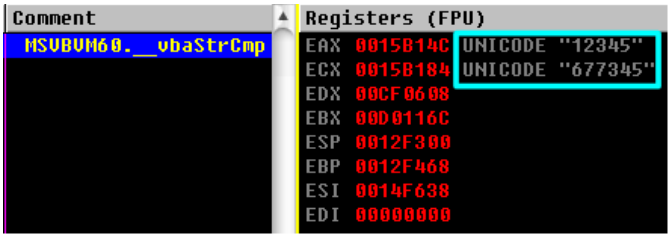
With OllyDbg

Jhonjhon 라는 문장을 All referenced text strings 를 통해 찾아 보았다.

```
00402468  > 8B85 64FFFFFF  MOV EAX,DWORD PTR SS:[EBP-9C]
0040246E  . 8B8D 68FFFFFF  MOV ECX,DWORD PTR SS:[EBP-98]
00402474  . 50             PUSH EAX
00402475  . 51             PUSH ECX
00402476  . FF15 44104000  CALL DWORD PTR DS:[<&MSVBVM60.__vbaStrCmp>]      MSVBVM60.__vbaStrCmp
0040247C  . 8BD8           MOV EBX,EAX
0040247E  . 8D95 68FFFFFF  LEA EDX,DWORD PTR SS:[EBP-98]
00402484  . F7DB           NEG EBX
00402486  . 1BDB           SBB EBX,EBX
00402488  . 8D85 64FFFFFF  LEA EAX,DWORD PTR SS:[EBP-9C]
0040248E  . 52             PUSH EDX
0040248F  . 43             INC EBX
00402490  . 50             PUSH EAX
00402491  . 6A 02          PUSH 2
00402493  . F7DB           NEG EBX
00402495  . FF15 7C104000  CALL DWORD PTR DS:[<&MSVBVM60.__vbaFreeStrList>]  MSVBVM60.__vbaFreeStrList
0040249B  . 8D8D 5CFFFFFF  LEA ECX,DWORD PTR SS:[EBP-A4]
004024A1  . 8D95 60FFFFFF  LEA EDX,DWORD PTR SS:[EBP-A0]
004024A7  . 51             PUSH ECX
004024A8  . 52             PUSH EDX
004024A9  . 6A 02          PUSH 2
004024AB  . FF15 18104000  CALL DWORD PTR DS:[<&MSVBVM60.__vbaFreeObjList>]  MSVBVM60.__vbaFreeObjList
004024B1  . 83C4 18        ADD ESP,18
004024B4  . 66:3BDF        CMP BX,DI
004024B7  .v0F84 57010000  JE Advance_.00402614
004024BD  . 8B35 8C104000  MOV ESI,DWORD PTR DS:[<&MSVBVM60.__vbaVarDup>]   MSVBVM60.__vbaVarDup
004024C3  . B9 0A000000    MOV ECX,0A
004024C8  . B8 04000280    MOV EAX,80020004
004024CD  . 898D 1CFFFFFF  MOV DWORD PTR SS:[EBP-E4],ECX
004024D3  . 898D 2CFFFFFF  MOV DWORD PTR SS:[EBP-D4],ECX
004024D9  . BB 08000000    MOV EBX,8
004024DE  . 8D95 ECFEFFFF  LEA EDX,DWORD PTR SS:[EBP-114]
004024E4  . 8D8D 3CFFFFFF  LEA ECX,DWORD PTR SS:[EBP-C4]
004024EA  . 8985 24FFFFFF  MOV DWORD PTR SS:[EBP-DC],EAX
004024F0  . 8985 34FFFFFF  MOV DWORD PTR SS:[EBP-CC],EAX
004024F6  . C785 F4FEFFFF  MOV DWORD PTR SS:[EBP-10C],Advance_.00401F80    UNICODE "Jhonjhon_123"
```

위에 Code 를 살펴본 결과 **vbaStrCmp** 함수가 보였다.

Breakpoint 설정 후 실행



사용자가 입력한 값과, 677345 를 비교 하며 004024B7 의 JMP 문을 통해 분기 하는 것을 확인 하였다.

※ 위의 00402000 영역대는 처음 Code 에서 아래로 기본 적인 설정 하는 도중 JMP 로 인해 불러 오게 되었다.

```
00401D88    00              DB 00
00401D89    00              DB 00
00401D8A    00              DB 00
00401D8B    00              DB 00
00401D8C  . 816C24 04 370   SUB DWORD PTR SS:[ESP+4],37
00401D94  .v E9 E7030000    JMP Advance_.00402180
00401D99    00              DB 00
00401D9A    00              DB 00
00401D9B    00              DB 00
00401D9C    00              DB 00
00401D9D    00              DB 00
00401D9E    00              DB 00
00401D9F    00              DB 00
00401DA0    2C304000        DD Advance_.0040302C
00401DA4    24214000        DD Advance_.00402124
```

보충 설명

위의 수행 과정을 살펴보면 모두 MSVBVM60 에서 함수를 호출해 온다.

이는 VB 파일의 동작 방식으로 **MSVBVM60.dll 이라는 VB 전용 엔진을 사용**하는 것 때문이다.

ex) MessageBox 호출 시 ,

VB 컴파일러는 MSVBVM60.dll!rtcMsgBox() 함수가 호출되도록 하고, 이 함수내부에서 user32.dll!MessageboxA()를 호출한다.

Code 처음 수행 시

```
Address   Hex dump          Disassembly                          Comment
004011BC  $-FF25 88104000   JMP DWORD PTR DS:[<&MSVBVM60.#100>]   MSVBVM60.ThunRTMain
004011C2    00              DB 00
004011C3    00              DB 00
004011C4  $ 68 B4174000     PUSH Advance_.004017B4
004011C9  . E8 EEFFFFFF     CALL <JMP.&MSVBVM60.#100>
```

RT_MainStruct 구조체 주소인 004017B4 를 Parameter 로 넘기고 VB 엔진 (MSVBVM60) 의 메인 함수인 ThunkRTMain 호출

- **VB 파일의 기본 startup 코드** 이다.
- 직접 ThunkRTMain 으로 가는것이 아니라, JMP 명령을 통해 가는 **간접 호출 방법 (VC ++, VB 에서 사용) 을 사용**한다.

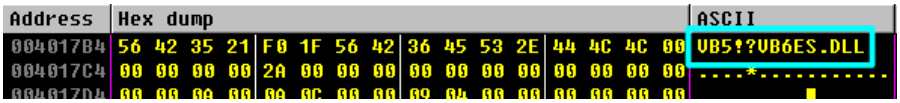
Step Into (F7) 을 통해 Code 확인

```
Address   Hex dump          Disassembly                          Comment
733735A4  55                PUSH EBP
733735A5  8BEC              MOV EBP,ESP
733735A7  6A FF             PUSH -1
733735A9  68 D0973873       PUSH MSVBVM60.733897D0
733735AE  68 FDBA4573       PUSH MSVBVM60.7345BAFD
733735B3  64:A1 00000000    MOV EAX,DWORD PTR FS:[0]
733735B9  50                PUSH EAX
733735BA  64:8925 0000000(  MOV DWORD PTR FS:[0],ESP
733735C1  51                PUSH ECX
733735C2  51                PUSH ECX
733735C3  83EC 4C           SUB ESP,4C
733735C6  53                PUSH EBX
733735C7  56                PUSH ESI
733735C8  57                PUSH EDI
733735C9  8965 E8           MOV DWORD PTR SS:[EBP-18],ESP
733735CC  8B75 08           MOV ESI,DWORD PTR SS:[EBP+8]
733735CF  8935 70E84773     MOV DWORD PTR DS:[7347E870],ESI
733735D5  8365 FC 00        AND DWORD PTR SS:[EBP-4],0
733735D9  8D45 A0           LEA EAX,DWORD PTR SS:[EBP-60]
733735DC  50                PUSH EAX
733735DD  FF15 A0103773     CALL DWORD PTR DS:[<&KERNEL32.GetStartu kernel32.GetStartupInfoA
733735E3  0FB745 D0         MOVZX EAX,WORD PTR SS:[EBP-30]
733735E7  A3 6CE84773       MOV DWORD PTR DS:[7347E86C],EAX
733735EC  FF35 D8E74773     PUSH DWORD PTR DS:[7347E7D8]         Advance_.00400000
```

MSVBVM60.dll 이 Loading 된 주소로 **메모리 주소가 완전히 틀려진다.**

RT_MainStruct 구조체

ThunRTMain() 함수의 Parameter 인 구조체 로써, 위에서는 **004017B4 주소에 RT_MainStruct 가 존재**한다.



RT_MainStruct 구조체의 멤버는 또 다른 구조체 함수들의 주소를 가지고 있다.

- 즉, **VB 엔진은 Parameter 로 넘어온 RT_MainStruct 구조체를 가지고 실행 시 필요한 모든 정보를 얻는다.**

답

677345