REPORT

Basic RCE Level 3

이름 : 신 민 구
제 출 일 : 2018.06.06

1. **문제 분석 및 실행**

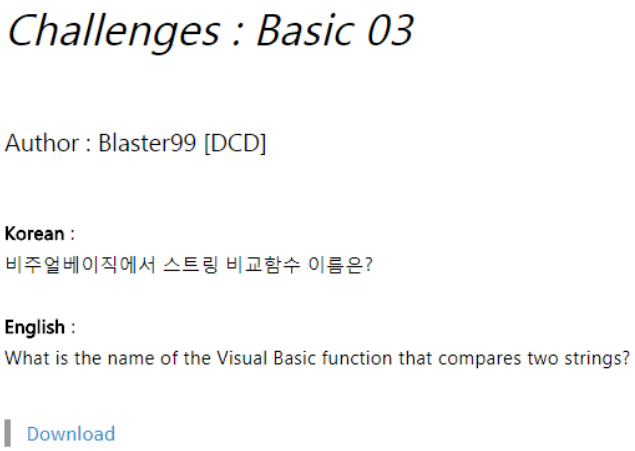


그림 1.1 Basic 03

Visual basic에서 스트링 비교함수 이름을 찾는 문제이다. 프로그램을 다운 받아보자.

그리고 프로그램을 실행 시켰더니 다음과 같은 오류 메시지 박스가 나오면서 EP(EntryPoint)가 이상한 곳에서 나온다. EP란 프로그램을 실행 시켰을 때 처음 위치를 말한다.

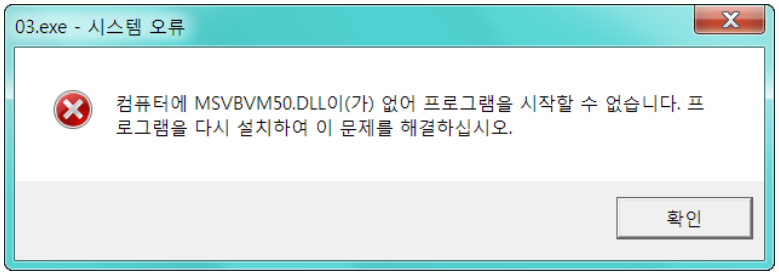


그림 1.2 실행 시 오류 메시지 박스

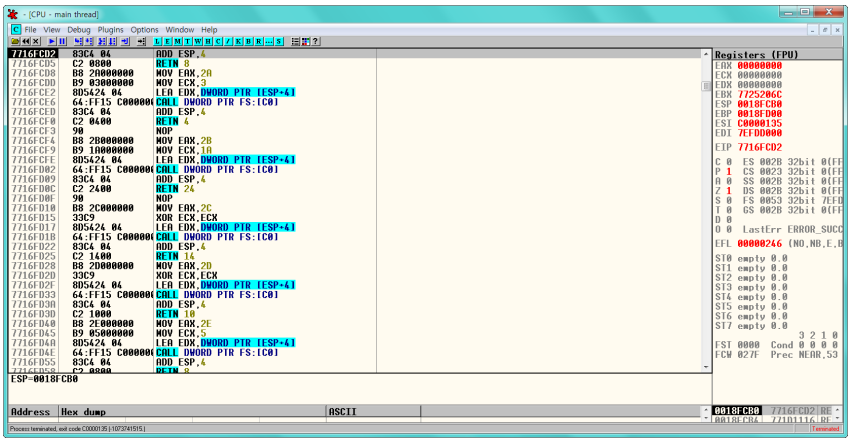


그림1.3 오류 후 EP

위의 오류 메시지 박스를 보니 MSVBVM50.DLL이라는 프로그램이 없어서 시작할 수 없다고 한다. 그러면 MSVBVM50.DLL을 찾아서 같은 공간 안에 붙여주어야 한다. 여기서 MSVBVM50.DLL은 Windows 운영 시스템을 위해 Microsoft에서 개발한 MSDN Disc 2455와 관련된 DLL 파일 유형이다. 이 파일을 찾아서 넣어주면 정상 실행이 될 것 같다.

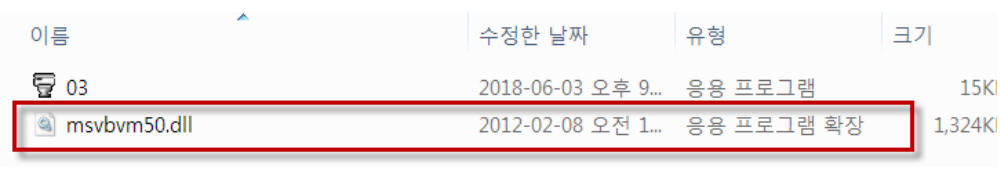


그림 1.4 MSVBVM50.DLL

위의 그림과 같이 문제 프로그램과 MSVBVM50.DLL을 한 공간에 넣어서 실행 시켜주어야 한다. OllyDbg를 통하여 프로그램을 열기 전에 일단 실행을 시켜서 무슨 프로그램인지를 봐야 한다.

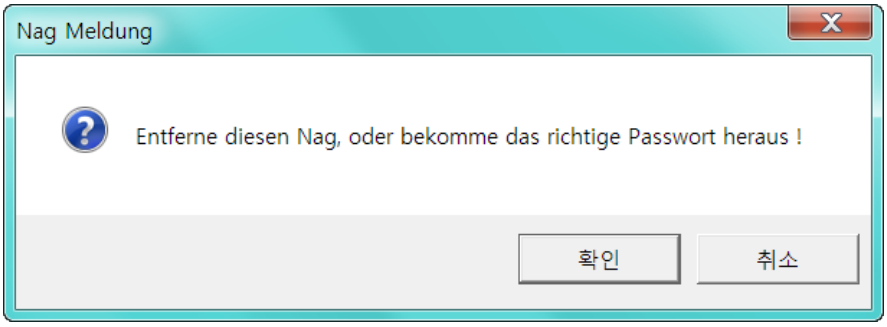


그림 1.5 실행 시 메시지 박스

실행을 시켰더니 다음과 같은 메시지 박스가 실행된다. [확인] 버튼을 눌러보자.

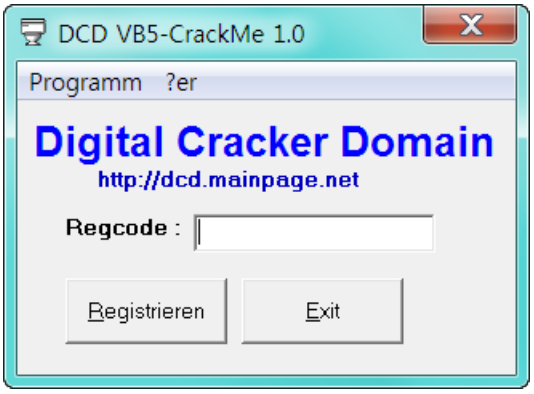


그림 1.6 에디터 박스

다음과 같이 에디터 박스가 뜨는데 Regcode를 보아하니 어떤 코드를 입력하는 프로그램이다. 여기에 아무 값이나 넣어보자.

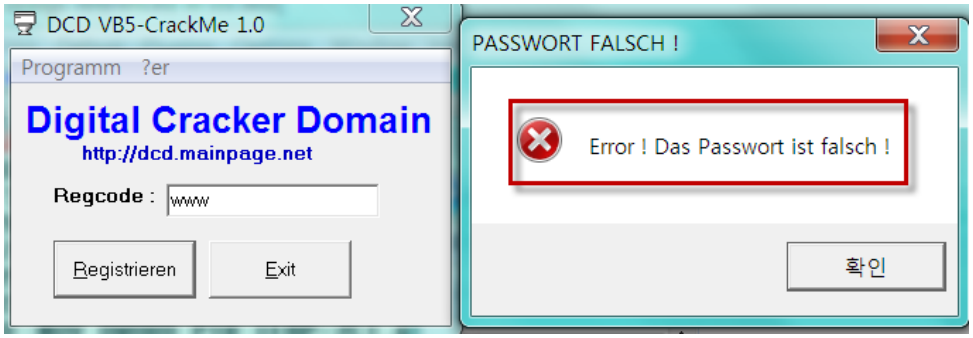


그림 1.7 에러 메시지 박스

즉 값을 입력을 하면 어떤 값과 비교하여 맞으면 옳은 메시지 박스를 틀리면 에러 메시지 박스를 실행시킬 것 같다고 예측 할 수 있다. 그리고 어떤 문자열을 비교하여 확인한다는 것 생각해보면 문자열 비교 함수가 있을 것이라고 예측할 수 있다. 이제 OllyDbg를 열어서 확인 해보자.

2. OllyDbg 분석

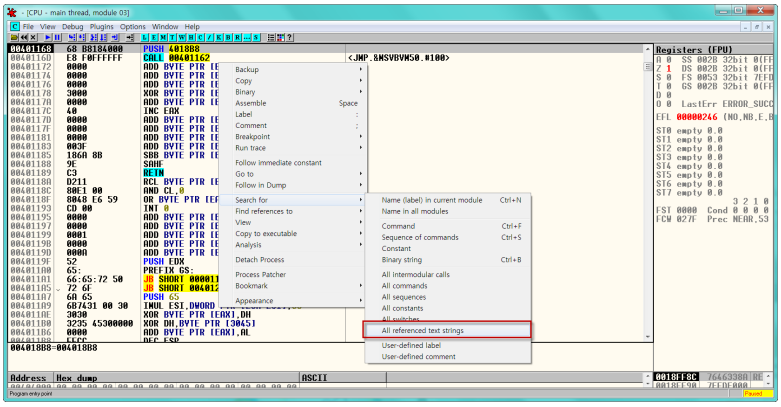
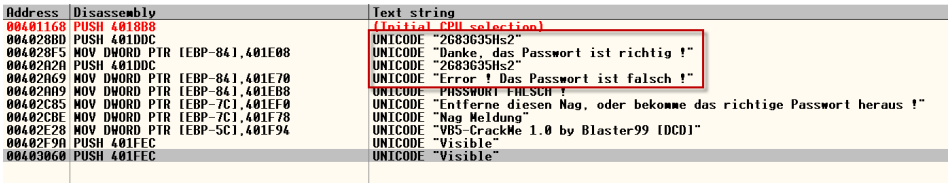


그림 2.1 search for

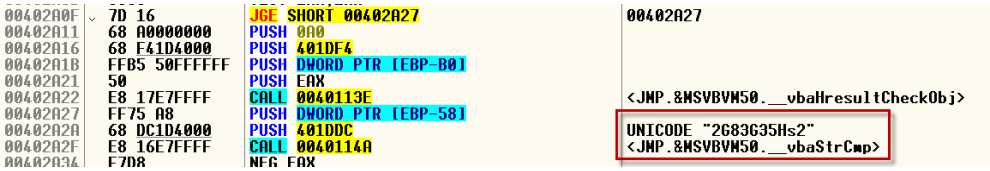
위의 EP주소를 보니 처음에 실행 시켰을 때와 다른 EP주소이다. MSVBVM50.DLL파일로 제대로 찾아간 것이다. 그림 2.20에서 보면 값을 잘못 입력하였을 때 메시지 박스에서 뜨는 저 문자열이 있을 것이라고 예측할 수 있다. 그래서 [우측 마우스 버튼 – Search for – all referenced text strings] 메뉴를 통하여 찾아보면 될 것이다.



Address	Disassembly	Text string
00401168	PUSH 4018B8	(Initial CPU selection)
004028BD	PUSH 401DDC	UNICODE "2G83G35Hs2"
004028F5	MOV DWORD PTR [EBP-84],401E08	UNICODE "Danke, das Passwort ist richtig !"
00402A2A	PUSH 401DDC	UNICODE "2G83G35Hs2"
00402A69	MOV DWORD PTR [EBP-84],401E70	UNICODE "Error ! Das Passwort ist falsch !"
00402A89	MOV DWORD PTR [EBP-84],401EB8	UNICODE "PASSWORT FALSCH !"
00402C85	MOV DWORD PTR [EBP-7C],401EF0	UNICODE "Entferne diesen Nag, oder bekomme das richtige Passwort heraus !"
00402CBE	MOV DWORD PTR [EBP-7C],401F78	UNICODE "Nag Meldung"
00402E28	MOV DWORD PTR [EBP-5C],401F94	UNICODE "VB5-CrackMe 1.0 by Blaster99 [DCD]"
00402F9B	PUSH 401FEC	UNICODE "Visible"
00403060	PUSH 401FEC	UNICODE "Visible"

그림 2.2 all referenced text strings

예상과 같이 문자열을 찾을 수 있었다. 더블 클릭하면 해당하는 주소로 바로 넘어 갈 수 있다.



```
00402A0F   7D 16          JGE SHORT 00402A27
00402A11   68 A0000000    PUSH 0A0
00402A16   68 F41D4000    PUSH 401DF4
00402A1B   FFB5 50FFFFFF  PUSH DWORD PTR [EBP-B0]
00402A21   50             PUSH EAX
00402A22   E8 17E7FFFF    CALL 0040113E
00402A27   FF75 A8        PUSH DWORD PTR [EBP-58]
00402A2A   68 DC1D4000    PUSH 401DDC
00402A2F   E8 16E7FFFF    CALL 0040114A
00402A34   F7D8           NEG EAX

00402A27
<JMP.&MSVBVM50.__vbaHresultCheckObj>
UNICODE "2G83G35Hs2"
<JMP.&MSVBVM50.__vbaStrCmp>
```

그림 2.3 vbaStrCmp

찾아가 보니 어떤 수상한 문자열과 바로 밑에 vbaStrCmp라고 스트링을 비교하는 함수가 있다. 함수의 이름만 봐도 문자열을 비교한다는 것을 알 수 있다. 또한 이전에 보이는 문자열과 비교한다고 예측할 수 있다. 그래서 저 문자열을 regcode에 입력하여 보자.

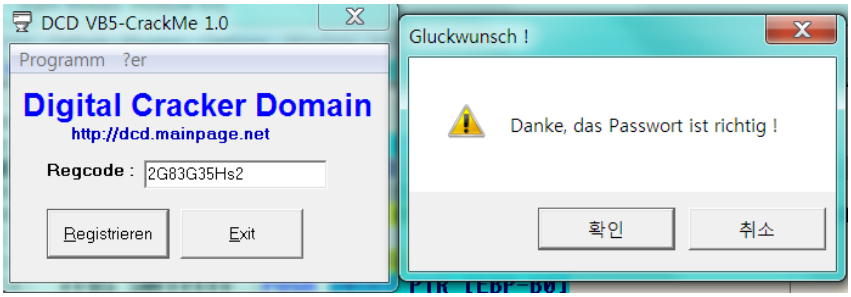


그림 2.4 크랙

올바른 메시지 박스가 나타났다. 즉, '2G83G35Hs2' 문자열이 regcode이고 이 문자열을 비교하는 함수의 이름은 vbaStrCmp라는 것을 알 수 있다.