CodeEngn Challenges Basic RCE Level5 풀이

Reverse L05 Start

Author : Acid Bytes [CFF]

Korea :

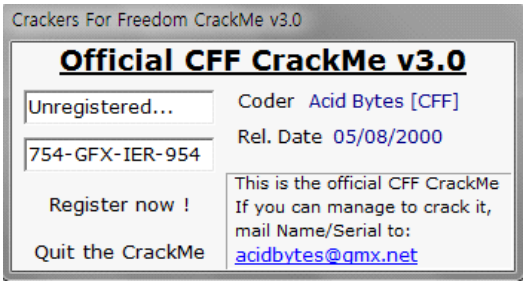
이 프로그램의 등록키는 무엇인가

English :

The registration key of this program is?

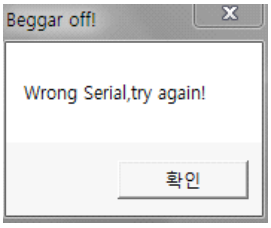
Down

우선 파일을 실행해보았다.



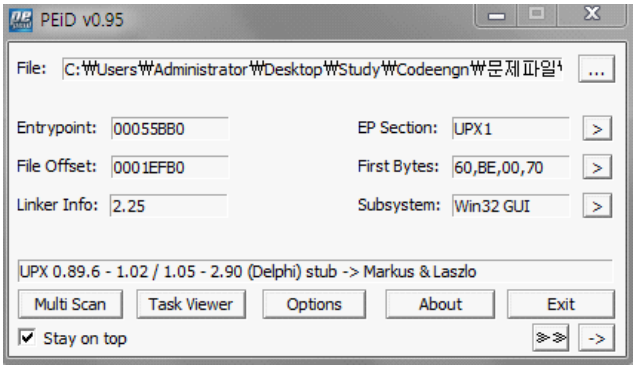
저렇게 시리얼을 입력하는곳이 두군데 있었고,

미리 입력이 되있길래 Register now!를 클릭했는데,



다음과 같이 잘못된 시리얼을 입력했다고 나온다.

일단 Olly로 열기전에 PEID로 파일의 정보를 얻어보았다.



파일이 UPX로 패킹되어있었다.

그러면 olly로 열어도 프로그램의 제대로 된 어셈블리코드를 볼수가 없는데

제대로 어셈을 보기위해서는 파일은 unpacking해주어야한다.

Upx unpacking은 간단하니 툴을 사용하지않고 손수해보겠다.

먼저 olly로 attach하고난 후 스크롤을 밑으로 내리면 수많은 DB 00을 볼수가있을것이다.

다시 스크롤을 올리다보면

DB 00과 어셈코드간의 경계점에 jmp문이있을것이다.

```
00455CEF  .  FF96 0061050( CALL DWORD PTR DS:[ESI+56100]
00455CF5  .  09C0          OR EAX,EAX
00455CF7  .v 74 07         JE SHORT Reverse_.00455D00
00455CF9  .  8903          MOV DWORD PTR DS:[EBX],EAX
00455CFB  .  83C3 04       ADD EBX,4
00455CFE  .^ EB E1         JMP SHORT Reverse_.00455CE1
00455D00  >  FF96 0461050( CALL DWORD PTR DS:[ESI+56104]
00455D06  >  61            POPAD
00455D07  .^ E9 64B5FEFF   JMP Reverse_.00441270
00455D0C     245D4500      DD Reverse_.00455D24
00455D10     345D4500      DD Reverse_.00455D34
00455D14     D0344400      DD Reverse_.004434D0
00455D18     00            DB 00
00455D19     00            DB 00
00455D1A     00            DB 00
00455D1B     00            DB 00
00455D1C     00            DB 00
00455D1D     00            DB 00
```

저 jmp문제 bp를 건후 실행을 시키고

점프를 진행시키면

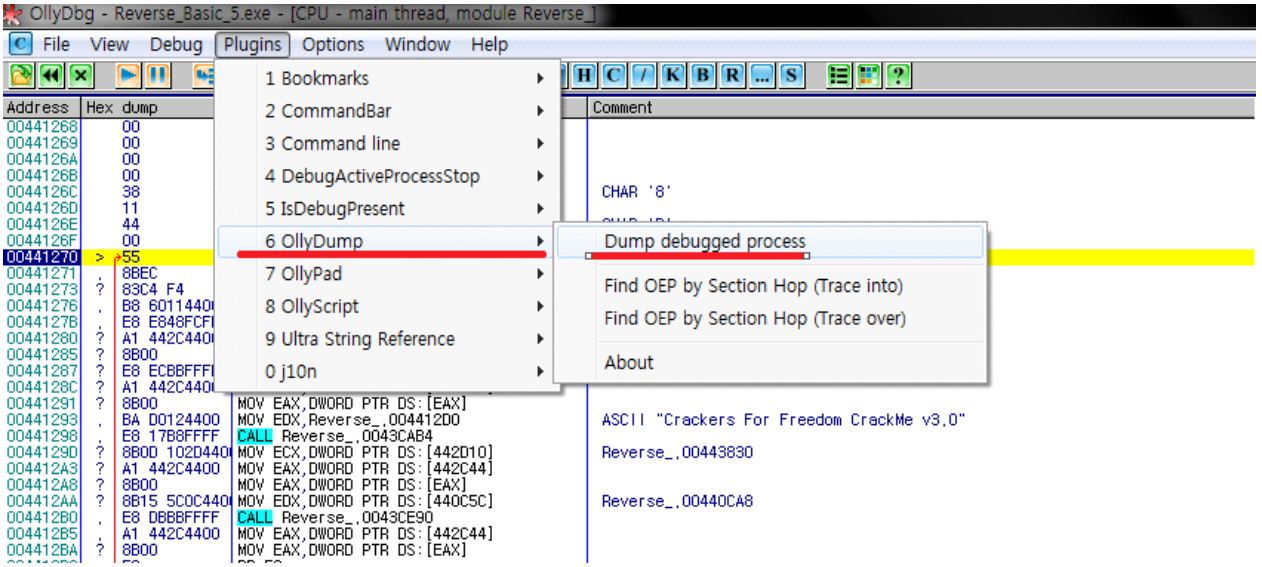
OEP가 있는 곳으로 점프한다.

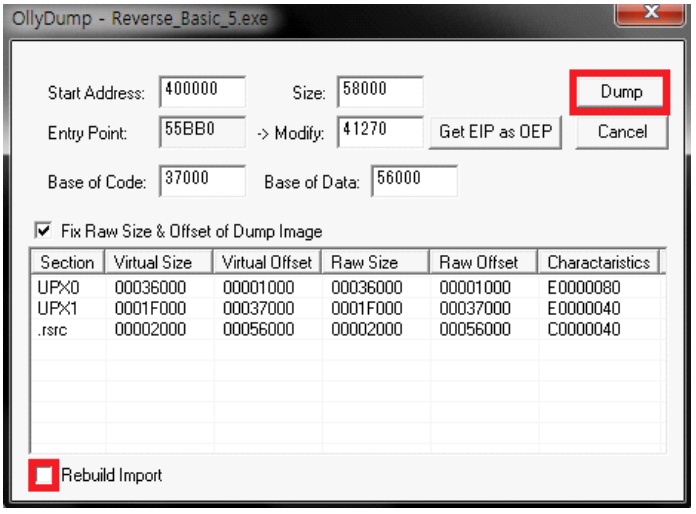
```
0044126B     00            DB 00
0044126C     38            DB 38                    CHAR '8'
0044126D     11            DB 11
0044126E     44            DB 44                    CHAR 'D'
0044126F     00            DB 00
00441270  >  55            PUSH EBP
00441271  .  8BEC          MOV EBP,ESP
00441273  ?  83C4 F4       ADD ESP,-0C
00441276  .  B8 60114400   MOV EAX,Reverse_.00441160
0044127B  .  E8 E848FCFF   CALL Reverse_.00405B68
00441280  ?  A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00441285  ?  8B00          MOV EAX,DWORD PTR DS:[EAX]
00441287  ?  E8 ECBBFFFF   CALL Reverse_.0043CE78
0044128C  ?  A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
00441291  ?  8B00          MOV EAX,DWORD PTR DS:[EAX]
00441293  .  BA D0124400   MOV EDX,Reverse_.004412D0  ASCII "Crackers For Freedom CrackMe v3.0"
00441298  .  E8 17B8FFFF   CALL Reverse_.0043CAB4
0044129D  ?  8B0D 102D440( MOV ECX,DWORD PTR DS:[442D10] Reverse_.00443830
004412A3  ?  A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
004412A8  ?  8B00          MOV EAX,DWORD PTR DS:[EAX]
004412AA  ?  8B15 5C0C440( MOV EDX,DWORD PTR DS:[440C5C] Reverse_.00440CA8
004412B0  .  E8 DBBBFFFF   CALL Reverse_.0043CE90
004412B5  .  A1 442C4400   MOV EAX,DWORD PTR DS:[442C44]
004412BA  ?  8B00          MOV EAX,DWORD PTR DS:[EAX]
004412BC     E8            DB E8
004412BD     4F            DB 4F                    CHAR 'O'
004412BE     BC            DB BC
```

저기서 부터가 진짜 이 프로그램의 어셈블리 코드이다.

저 어셈소스들을 따로 저장을시켜줘야하는데

Plugin의 Olly dump를 이용했다.





해주면

파일을 저장할 이름을 설정하는데

_원본파일이름.exe

이 보기좀 편하다.

프로그램의 경로로 따라가 프로그램을 실행하면 정상적으로 실행이 안되는데

왜냐면 아까 PE를 rebuild 안해줬기에 프로그램이 손상된채로 저장되었기 때문이다.

그래서 lordPE라는 프로그램을 이용해

파일의 PE를 rebuild 해줄것이다.

LordPE라는 프로그램을 다운받아 실행시켜

언패킹된 프로그램의 아이콘을 끌어다놔주면 자동적으로 PE가 rebuild된다.

그래서, 최종적인 파일을 실행시켜보면,

잘 실행되는 것을 볼 수가 있다.

이제 그 프로그램을 분석해볼일만남았다.

다시 언팩된 프로그램을 올리로 attach후

Search For -> All Referenced Text Strings 로 프로그램의 문자열들을 살펴보면,

```
00440E81 DD _Reverse,0043EB24                ASCII "p?"
00440E89 DD _Reverse,0043E170                ASCII "찰C"
00440E96 ASCII "TForm1"
00440E9C DD _Reverse,00440CA8                ASCII "4wB"
00440EA0 DD _Reverse,00433584                ASCII "?C"
00440EA7 ASCII "Unit1"
00440EDC MOV ECX,_Reverse,00440FC8           ASCII "No Name entered"
00440EE1 MOV EDX,_Reverse,00440FD8           ASCII "Enter a Name!"
00440F08 MOV ECX,_Reverse,00440FE8           ASCII "No Serial entered"
00440F0D MOV EDX,_Reverse,00440FFC           ASCII "Enter a Serial!"
00440F2F MOV EDX,_Reverse,00441014           ASCII "Registered User"
00440F4C MOV EDX,_Reverse,0044102C           ASCII "GFX-754-IER-954"
00440F5A MOV ECX,_Reverse,0044103C           ASCII "CrackMe cracked successfully"
00440F5F MOV EDX,_Reverse,0044105C           ASCII "Congrats! You cracked this CrackMe!"
00440F74 MOV ECX,_Reverse,00441080           ASCII "Beggar off!"
00440F79 MOV EDX,_Reverse,0044108C           ASCII "Wrong Serial,try again!"
00440F8E MOV ECX,_Reverse,00441080           ASCII "Beggar off!"
00440F93 MOV EDX,_Reverse,0044108C           ASCII "Wrong Serial,try again!"
00440F9F CALL _Reverse,0043D068              (Initial CPU selection)
```

다음과 같이 정답, 오답을 알려주는 듯한 문자열들을 볼수가 있을것이다.

저 문자열들이 있는곳으로 가보면,

```
00440F27  .  E8 F4FEFDFF  CALL _Reverse,00420E20
00440F2C  .  8B45 FC      MOV EAX,DWORD PTR SS:[EBP-4]
00440F2F  .  BA 14104400  MOV EDX,_Reverse,00441014          ASCII "Registered User"
00440F34  .  E8 F32BFCFF  CALL _Reverse,00403B2C
00440F39  .v 75 51        JNZ SHORT _Reverse,00440F8C
00440F3B  .  8D55 FC      LEA EDX,DWORD PTR SS:[EBP-4]
00440F3E  .  8B83 C802000 MOV EAX,DWORD PTR DS:[EBX+2C8]
00440F44  .  E8 D7FEFDFF  CALL _Reverse,00420E20
00440F49  .  8B45 FC      MOV EAX,DWORD PTR SS:[EBP-4]
00440F4C  .  BA 2C104400  MOV EDX,_Reverse,0044102C          ASCII "GFX-754-IER-954"
00440F51  .  E8 D62BFCFF  CALL _Reverse,00403B2C
00440F56  .v 75 1A        JNZ SHORT _Reverse,00440F72
00440F58  .  6A 00        PUSH 0
00440F5A  .  B9 3C104400  MOV ECX,_Reverse,0044103C          ASCII "CrackMe cracked successfully"
00440F5F  .  BA 5C104400  MOV EDX,_Reverse,0044105C          ASCII "Congrats! You cracked this CrackMe!"
00440F64  .  A1 442C4400  MOV EAX,DWORD PTR DS:[442C44]
00440F69  .  8B00         MOV EAX,DWORD PTR DS:[EAX]
00440F6B  .  E8 F8C0FFFF  CALL _Reverse,0043D068
00440F70  .v EB 32        JMP SHORT _Reverse,00440FA4
00440F72  >  6A 00        PUSH 0
00440F74  .  B9 80104400  MOV ECX,_Reverse,00441080          ASCII "Beggar off!"
00440F79  .  BA 8C104400  MOV EDX,_Reverse,0044108C          ASCII "Wrong Serial,try again!"
00440F7E  .  A1 442C4400  MOV EAX,DWORD PTR DS:[442C44]
```

다음과 같은 어셈소스들을 볼수가있는데.

각각 함수 호출전에 자신이 입력한 값과 뭔가 수상한 문자열을 EDX에 넣어주고 함수를 call하는 것을 볼수가 있다.

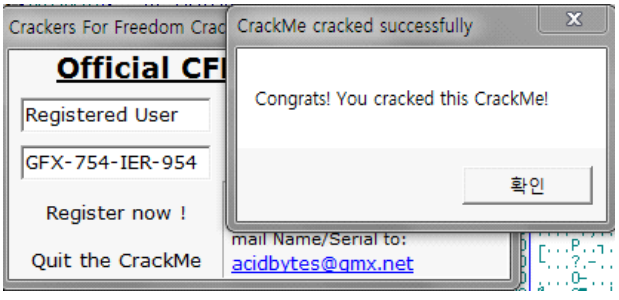
함수를 파고들어가면 내가 입력한값과 저 문자열과 비교를 한다.

즉 저기 저 push되는 문자들이 key값이고,

Registered User는 name값이니

레지스터값은 GFX-754-IER-954가 되는것이다.

실제로 저 두값을 두개의 입력박스에 입력하고 등록키를 누르면



다음과 같이 축하한다는 말을 해준다