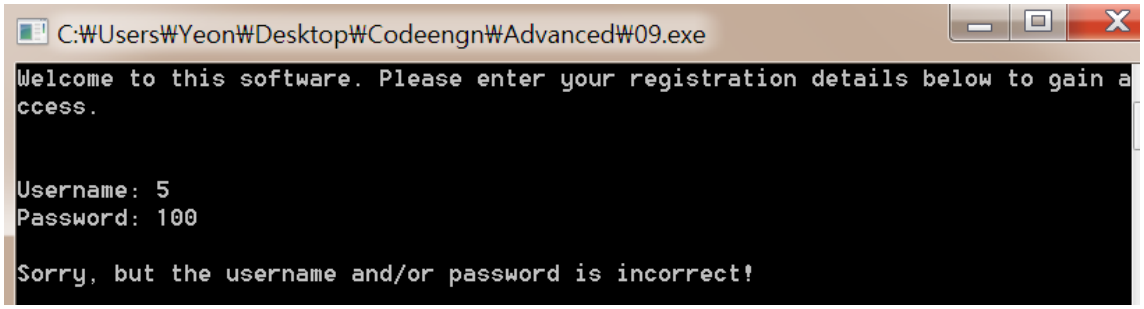
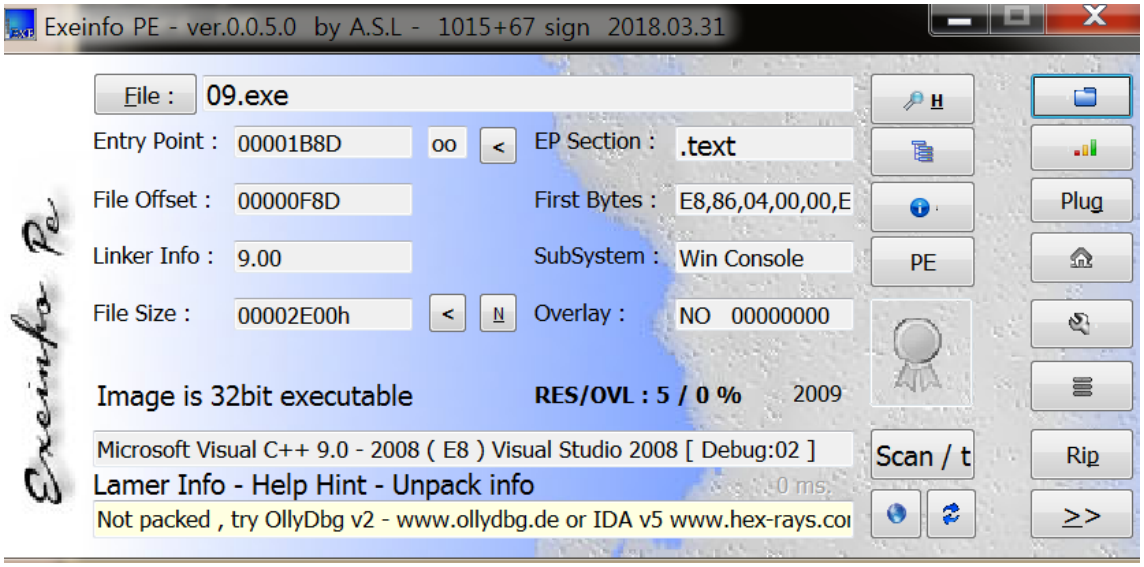
코드 엔진 Challenges: Advance 09

Author: MulleDK13
Korean: password는 무엇인가

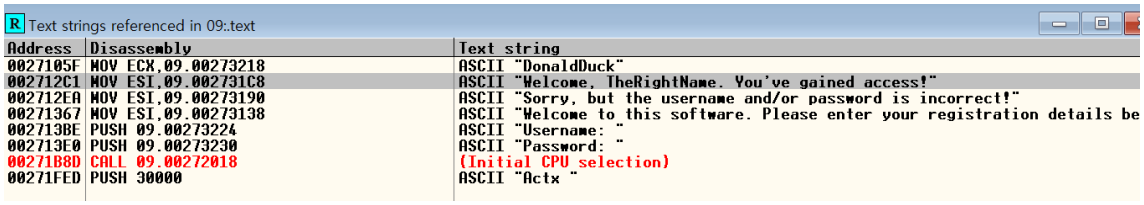
파일을 실행하면 username과 password를 입력하는 화면이 나오는데 문제에서 어떠한 설명도 없으므로 username과 password 두 개 모두에 임의의 값을 넣는다.



PEID로 특이사항과 패킹여부를 확인한후 올리디버거로 분석해보자.

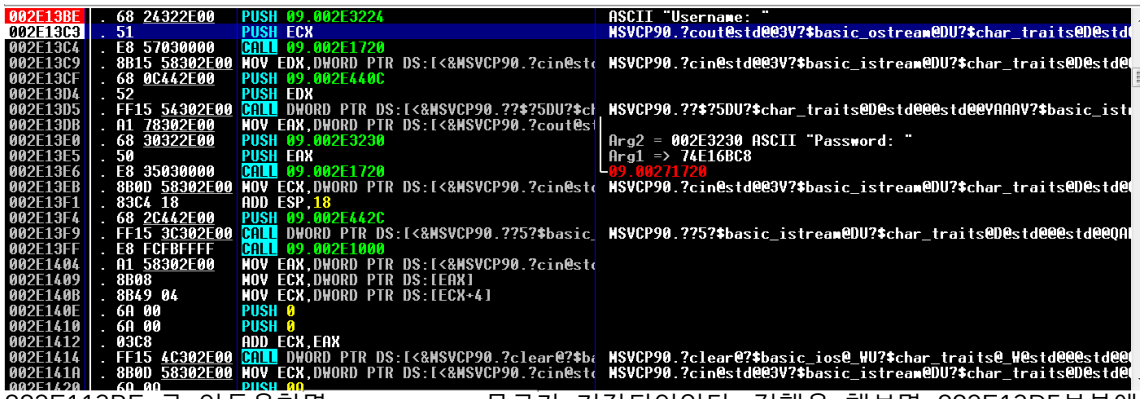


파일은 C++로 작성되어있고 패킹되어있지 않기 때문에 바로 올리디버거로 분석하면 될 것 같다.



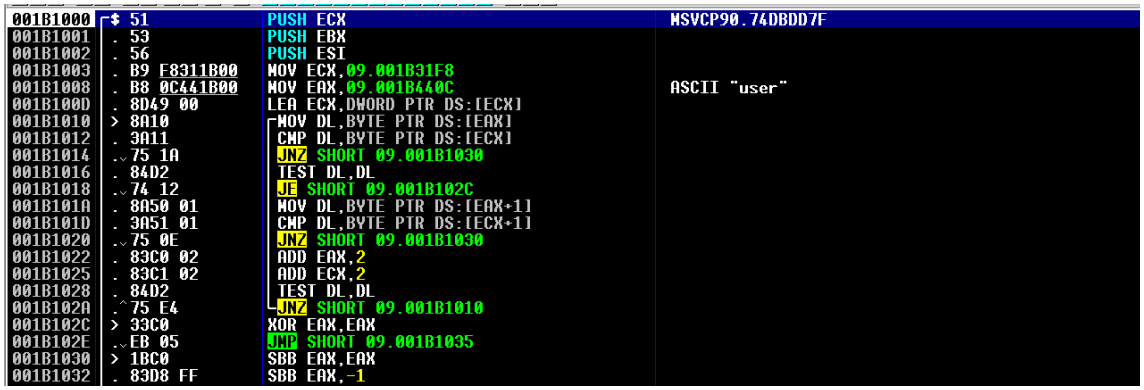
문자열 찾기를 보면 Username과 ,Password가 보이고 성공문자열과 실패문자열이 보인다. 따라 들어가보자.

일단 어떤 아이디와 어떤 패스워드를 입력해야하는지 모르기 때문에 아이디부터 찾아보자.



002E113BE 로 이동을하면 username 문구가 저장되어있다. 진행을 해보면 002E13D5부분에서 username을 입력받는다. 간단하게 user라고 입력해보자. EAX에 74E16BC8이 저장되고 다음 명령어를 실행하면 passwd 출력 문자열이 스택에 첫 번째 인자로 저장 되고 EAX값이 스택에 두 번째 인자로 들어간다 그리고 함수 002E1720이 호출된다. 그리고 002E13F9에서 비빌번호를 100을 입력했다. 그리고 002EC1000이 출력되면서 실패 문자열이 출력되는 것을 확인하였다 .

이 함수를 따라 들어가보자.

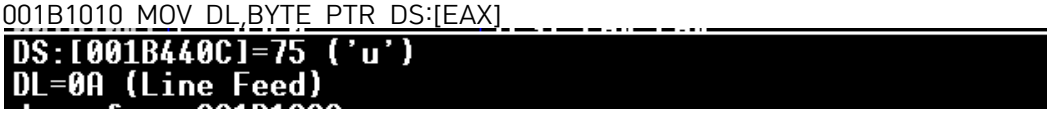


우리가 입력한 user도 보이고 비교문,분기문 등이 보인다. 한줄씩 실행해보자.

우리가 입력한 001b440에 저장된 user가 eax에 저장된다. 그리고 001b31f8에 저장된 001b31f8이 ECX에도 저장되었다.

차례대로 실행해보자.

001B1010 MOV DL,BYTE PTR DS:[EAX]



001B1012 CMP DL,BYTE PTR DS:[ECX]

```
DS:[001B31F8]=00
DL=75 ('u')
```

001B1014 JNZ SHORT 09.001B1030

//비교한 값이 같지않으면 분기하는데 같지않아 분기하게된다.

```
001F1030  > 1BC0          SBB EAX,EAX                              09.001F440C
001F1032  . 83D8 FF       SBB EAX,-1
001F1035  > 8B0D 04441F00 MOV ECX,DWORD PTR DS:[1F4404]
001F103B  . 8B15 50301F00 MOV EDX,DWORD PTR DS:[<&MSVCP90.?endl@s  MSVCP90.?endl@std@@YAAAV?$basic_ostream@DU?$char_t
001F1041  . 85C0          TEST EAX,EAX
001F1043  . A1 2C441F00   MOV EAX,DWORD PTR DS:[1F442C]
001F1048  . 0F94C3        SETE BL
001F104B  . 3B01          CMP EAX,DWORD PTR DS:[ECX]
001F104D  . 8B0D 78301F00 MOV ECX,DWORD PTR DS:[<&MSVCP90.?cout@s  MSVCP90.?cout@std@@3V?$basic_ostream@DU?$char_trai
001F1053  . 52            PUSH EDX
001F1054  . 0F944424 0F   SETE BYTE PTR SS:[ESP+F]
001F1059  . FF15 44301F00 CALL DWORD PTR DS:[<&MSVCP90.??6?$basic_  MSVCP90.??6?$basic_ostream@_WU?$char_traits@_W@std
001F105F  . B9 18321F00   MOV ECX,09.001F3218                      ASCII "DonaldDuck"
001F1064  . B8 0C441F00   MOV EAX,09.001F440C                      ASCII "user"
001F1069  . 8DA424 000000( LEA ESP,DWORD PTR SS:[ESP]
001F1070  > 8A10          MOV DL,BYTE PTR DS:[EAX]
001F1072  . 3A11          CMP DL,BYTE PTR DS:[ECX]
001F1074  .v75 1A         JNZ SHORT 09.001F1090
001F1076  . 84D2          TEST DL,DL
001F1078  .v74 12         JE SHORT 09.001F108C
001F107A  . 8A50 01       MOV DL,BYTE PTR DS:[EAX+1]
001F107D  . 3A51 01       CMP DL,BYTE PTR DS:[ECX+1]
001F1080  .v75 0E         JNZ SHORT 09.001F1090
```

분기하게된 곳에서부터 다시 분석해보도록 하자. 001f1059까지 진행한 결과 EAX에는 74e16bc8 ECX에는 74e16bc8이 EDX에는 74d91850이 저장된다. 그리고 나서 001f1059에서 어떤 함수가 호출되고 001f3218 에 저장된 값 DonaldDuck을 ECX 저장하고 001f440c에 우리가 입력한 user의 값이 저장되고 이 값을 EAX에 저장한다.

001f1070에서 EAX의 값(u)의 1바이트를 DL에 옮긴다. 그리고 ECX의 1바이트 값(D)을을 u가 들어있는 DL가 비교한다. 그리고 같지않으면 분기를하는데 이때 실패메시지가 나온다.

즉 DonaldDuck이 우리가 입력해야하는 name인 것을 알 수 있다.

username을 찾았으니 password를 찾아보도록하자. username에 대한 검사를 마친뒤에 디버깅을 계속 진행해보자.

```
010A1030  > 1BC0          SBB EAX,EAX
010A1032  . 83D8 FF       SBB EAX,-1
010A1035  > 8B0D 04440A01 MOV ECX,DWORD PTR DS:[10A4404]
010A103B  . 8B15 50300A01 MOV EDX,DWORD PTR DS:[<&MSVCP90.?endl@s  MSVCP90.?endl@std@@YAAAV?$basic_ostream@DU?$char_tra
010A1041  . 85C0          TEST EAX,EAX
010A1043  . A1 2C440A01   MOV EAX,DWORD PTR DS:[10A442C]
010A1048  . 0F94C3        SETE BL
010A104B  . 3B01          CMP EAX,DWORD PTR DS:[ECX]
010A104D  . 8B0D 78300A01 MOV ECX,DWORD PTR DS:[<&MSVCP90.?cout@s  MSVCP90.?cout@std@@3V?$basic_ostream@DU?$char_traits
010A1053  . 52            PUSH EDX
010A1054  . 0F944424 0F   SETE BYTE PTR SS:[ESP+F]
```

010A1043에서 우리가 입력한 값이 EAX에 저장된다. 그다음 000702A0에 저장된 0x0088228f와 우리가 입력한 값이 저장된 EAX와 비교된다. 원래 의 패스워드 값이 0088228f인 것을 알 수 있다. 그럼 이 값의 10진수 값인 8921743을 입력하고 파일을 실행해보자.

```
Welcome to this software. Please enter your registration details below to gain a
ccess.


Username: DonaldDuck
Password: 8921743



I can't believe you felt for that! xD


Sorry, but the username and/or password is incorrect!
```

여전히 틀렸다고 나온다.

성공문을 출력하는 부분으로 찾아가보니 맞아도 00a112ad부분에서 실패할 때 나오는 문자열을 출력한다. 패치를 해주도록 하자.

패치후 다시 입력해주니 제대로 돌아가는 것을 확인할 수 있따.