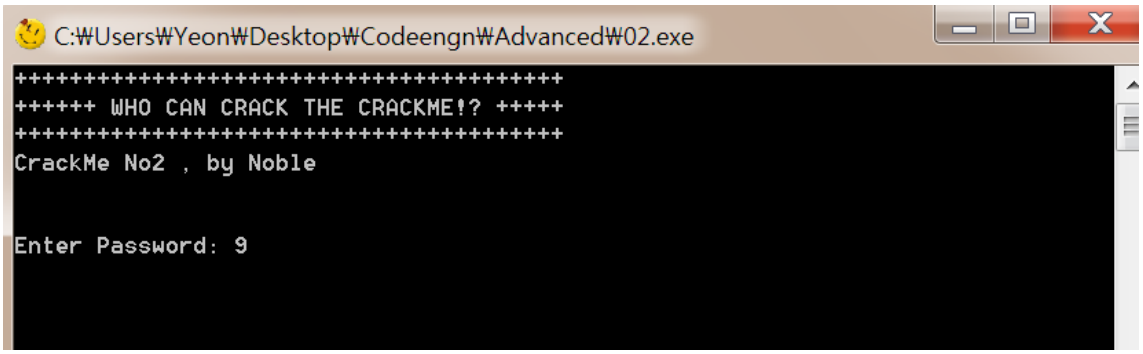
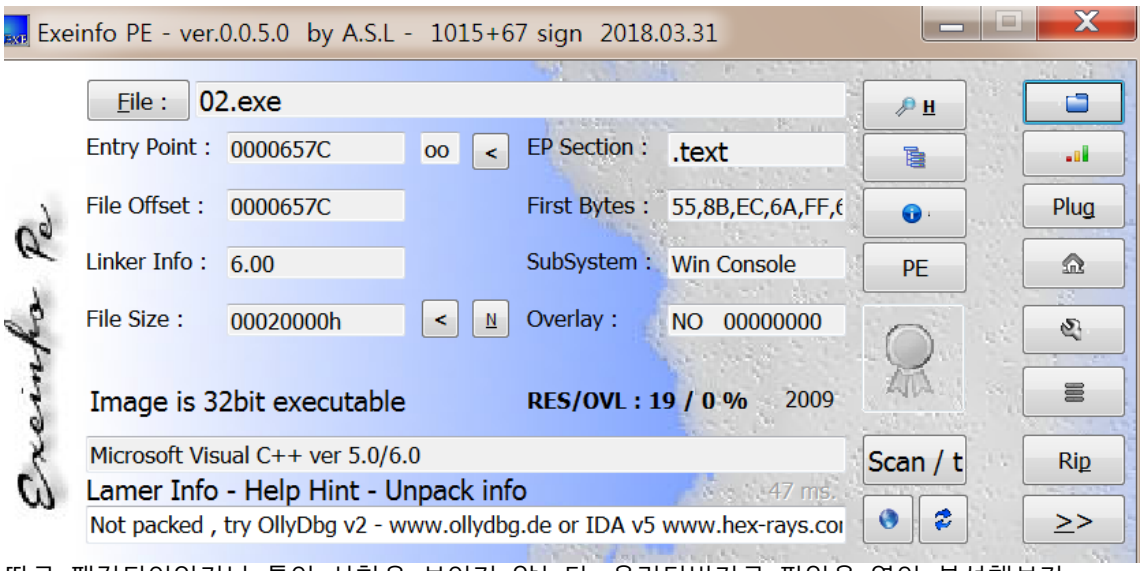
코드 엔진 Challenges: Advance 02

Author: Noble
Korean: 정답은 무엇인가

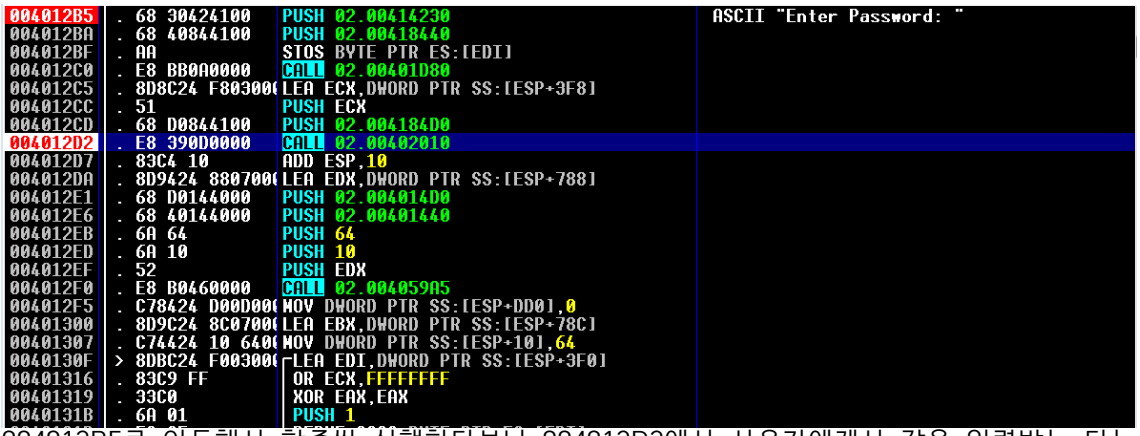
문제에는 따로 힌트가 없다. 파일을 실행 해 보고 어떻게 해볼지 생각해보자.



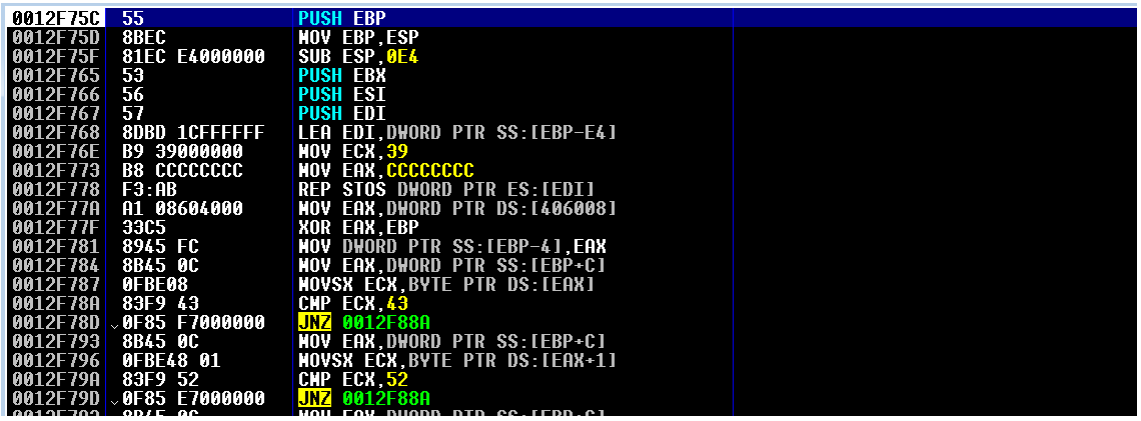
프로그램은 password를 입력하게 되어있고 패스워드에 9를 입력하고 엔터를 누르자 메시지 박스 없이 종료되었다. PEID를 통해 프로그램을 열어 패킹여부를 알아보자.

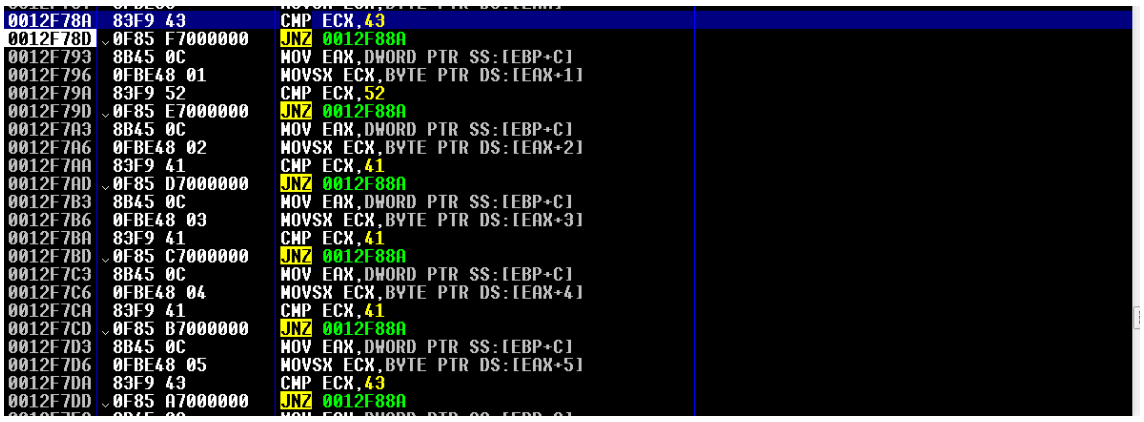


따로 패킹되어있거나 특이 사항은 보이지 않는다. 올리디버거로 파일을 열어 분석해보자.
사용된 함수목록을 봤을때도 특별한 함수는 보이지 않아 문자열 검색을 통해 실행 창에서 본 "Enter Passwod" 문자열을 찾아 이동해보았다.



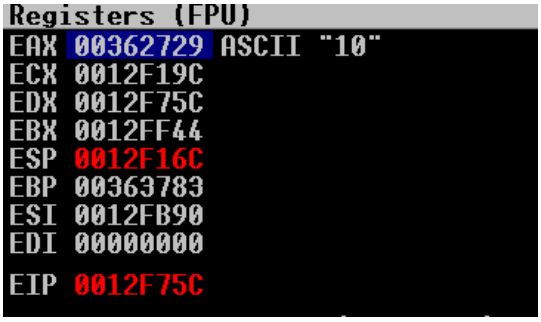
004012B5로 이동해서 한줄씩 실행하다보니 004012D2에서 사용자에게서 값을 입력받는 다는 것을 알 았다. 계속 코드를 실행해보자 . 코드에 뚜라 64번 분기를 한 후 EAX의 값이 0이 되어서 분기문을 빠져나온 후 계속 코드를 실행하다가 004013C5 다음 구문에서 RETN되는 것을 확인했다 . 이 함수의 내부를 확인해보자.

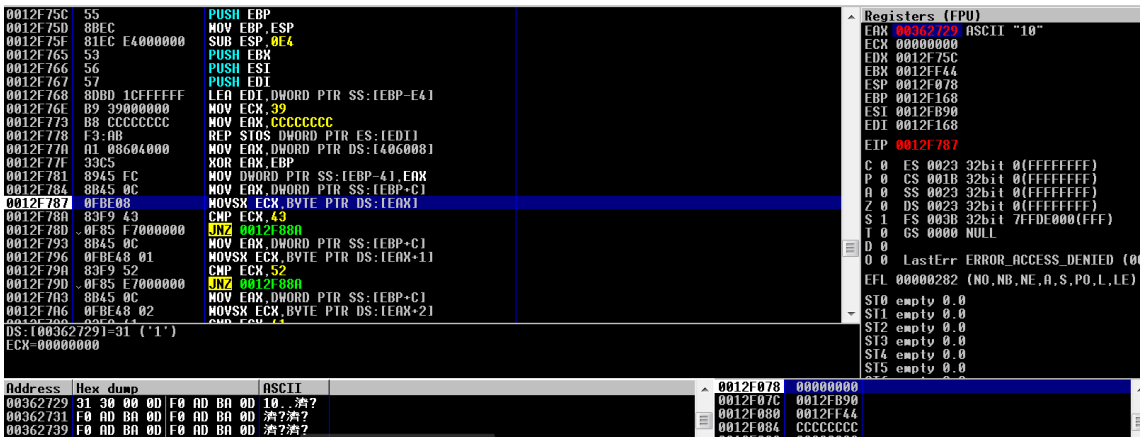




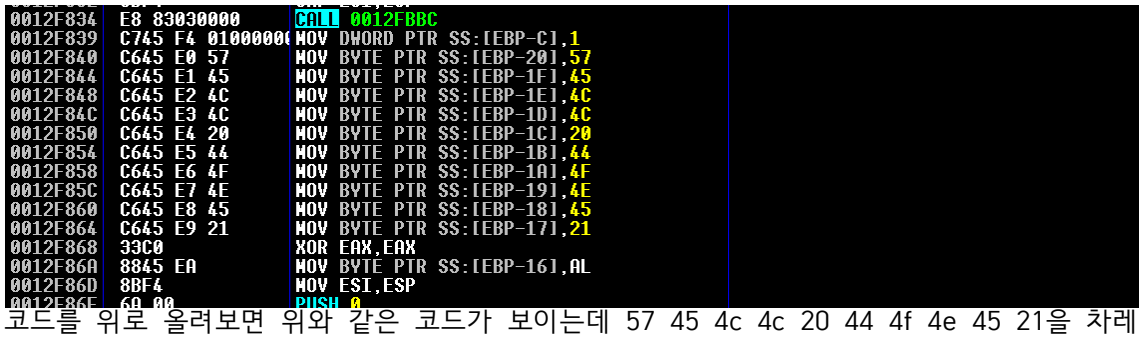
코드를 보면 알 수 있듯이 CMP를 이용해 ECX와 특 정 값을 비교하고 있다 .CMP 부분으로 가고 ECX에 어떤 값이 들어가는지 확인해보자.

※일단 0012F75C의 EAX에 우리가 입력한 값인 10이 아스키 값으로 들어가 있다.





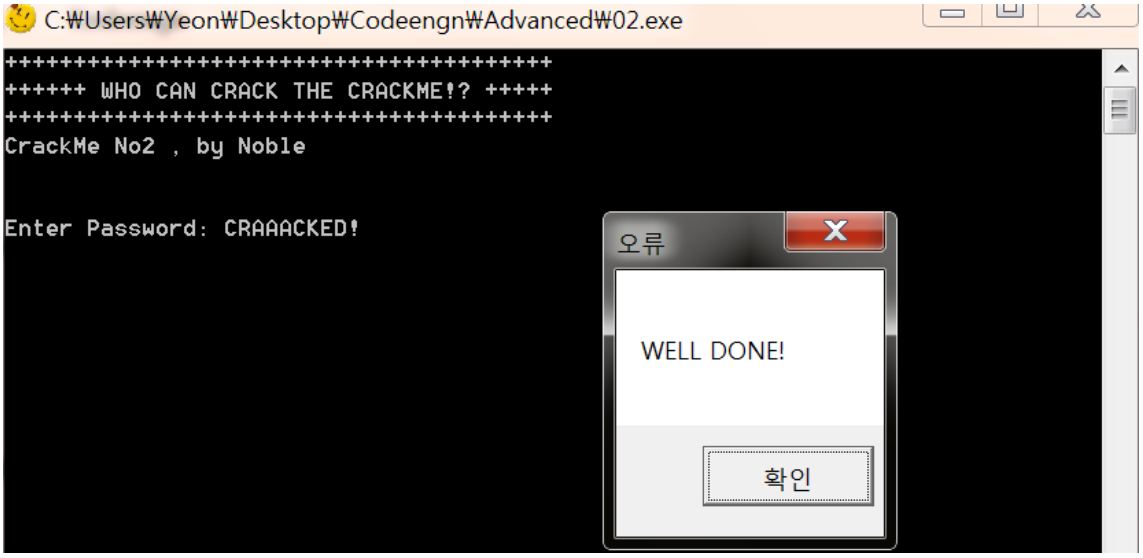
ECX 에 어떤 값이 들어 가는지 확인하기 위해서 CMP위의 부분은 분석해보니 EAX의 1바이트 값에는 31 즉 우리가 입력한 값의 첫째자리인 '1'이 저장되어 있는 것을 알 수 있다.
cmp를 보면 31과 43 이랑 비교해서 ZF가 0이면 JNZ에 의해 18FF8A로 이동하게 된다. ZF 0이라 0012F88A로 이동하게 된다. 18FF8A로 이동하게되면 ECX룰 호출하게 돼서 프로그램이 종료된다.



코드를 위로 올려보면 위와 같은 코드가 보이는데 57 45 4c 4c 20 44 4f 4e 45 21을 차례대로 넣는데 아스키 코드로 확인해보면 WELL DONE! 이라는 것을 알 수 있다. 이 코드를 실행 해 볼 필요가 있다. 이 코드는 0018F839 로 시작하는데 이 주소는

```
0012F7EA  83F9 4B          CMP ECX,4B
0012F7ED  0F85 97000000    JNZ 0012F88A
0012F7F3  8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F7F6  0FBE48 07        MOVSX ECX,BYTE PTR DS:[EAX+7]
0012F7FA  83F9 45          CMP ECX,45
0012F7FD  0F85 87000000    JNZ 0012F88A
0012F803  8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F806  0FBE48 08        MOVSX ECX,BYTE PTR DS:[EAX+8]
0012F80A  83F9 44          CMP ECX,44
0012F80D  75 7B            JNZ SHORT 0012F88A
0012F80F  8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F812  0FBE48 09        MOVSX ECX,BYTE PTR DS:[EAX+9]
0012F816  83F9 21          CMP ECX,21
0012F819  75 6F            JNZ SHORT 0012F88A
0012F81B  8B45 0C          MOV EAX,DWORD PTR SS:[EBP+C]
0012F81E  0FBE48 0A        MOVSX ECX,BYTE PTR DS:[EAX+A]
0012F822  85C9             TEST ECX,ECX
0012F824  74 13            JE SHORT 0012F839
```

0012F824에서 TEST ECX,ECX가 0일 경우 0018F839로 이동하는 것을 알 수 있다. 이렇게 이동하기 위해서는 웨에서 점프하지 않도록 CMP문을 해결해야만한다.
첫 번째 CMP는 ECX와 43을 비교한다. 이때 JNZ에서 분기하지 않기위해서 두 값이 같아야한다. 그럼 ECX의 값을 CMP문마다 다르게 바꿔준다.
즉 43 52 41 41 41 45 4B 45 44 21을 아스키 값으로 바꿔주는 것이다.
아스키 값으로 변경하면 CRAAACKED!가 된다 .이를 입력해보자.



CRACK이 되는 것을 알 수 있다.