Basic 05

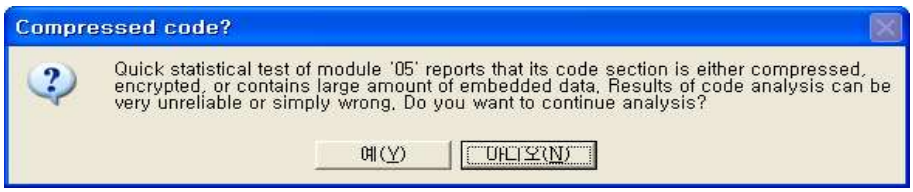
05파일을 실행시키면 다음과 같은 창이 뜬다.

창의 Register now ! 버튼을 누르면

시리얼값이 잘못됐다는 메시지가 뜨게 된다.

이 메시지를 이용하여 시리얼을 찾을 수 있을 것 같다.

OllyDbg를 이용하여 파일을 열면



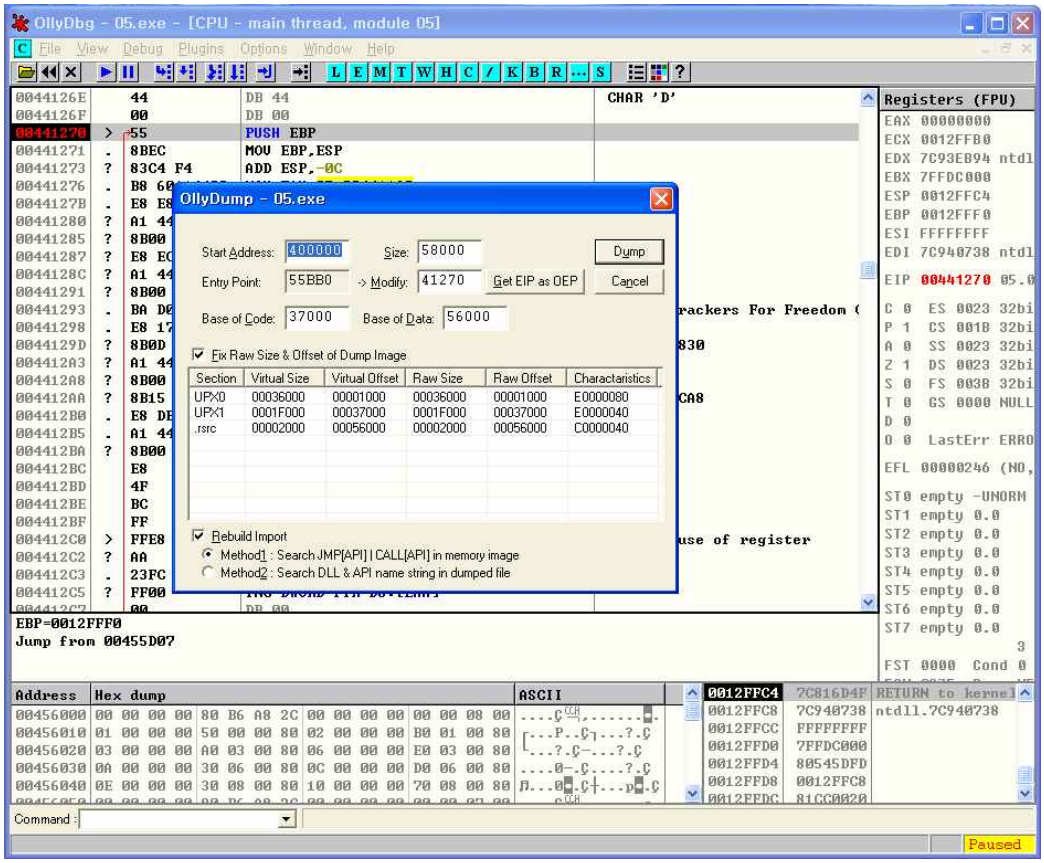
위와 같은 창이 뜨게 되는데 코드 섹션이 압축되었다는 얘기 같다. PEiD 툴을 사용하여 보면

UPX로 패킹되어 있는 것을 볼 수 있다.

패킹 되어 있는 코드는 언패킹하여 보지않으면 제대로 분석 할 수 없으므로 언패킹을 먼저 실

행한다.

OllyDbg로 다시 파일을 열고 실행되는 라인의 제일 밑 까지 내려보면 JMP 05.441270이라는 부분이 있다. 이 점프가 가르키고 있는 곳이 바로 Original Entry Point라고 볼 수 있다. 이부분에 브레이크를 걸고 디버깅을 한 후, OllyDump라는 플러그인을 사용하여 언팩하여 본다.

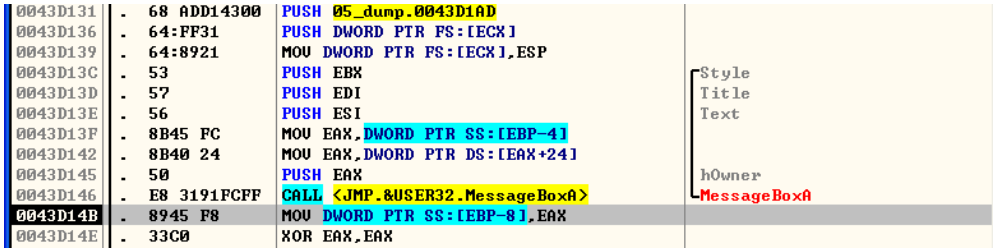


Dump버튼을 눌러 언팩킹된 파일을 생성(05_dump.exe)하였다.

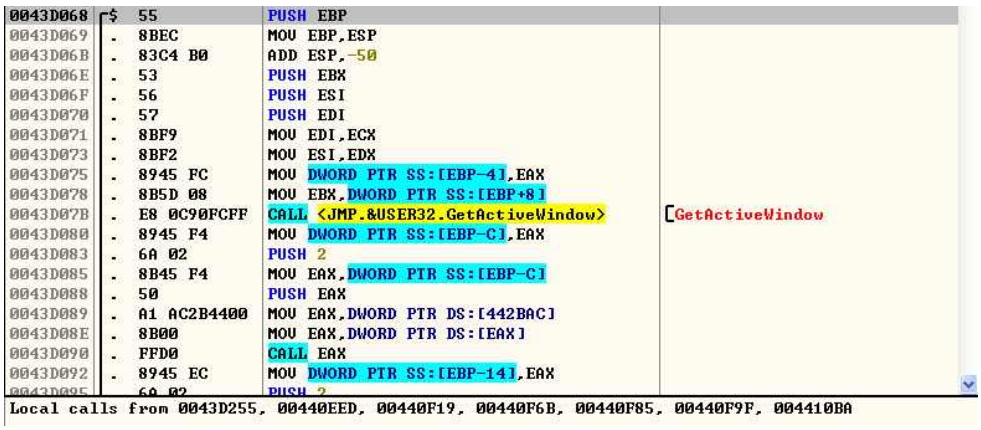
생성된 파일을 PEiD로 확인해 보면

패킹이 해제된 것을 볼 수 있다.

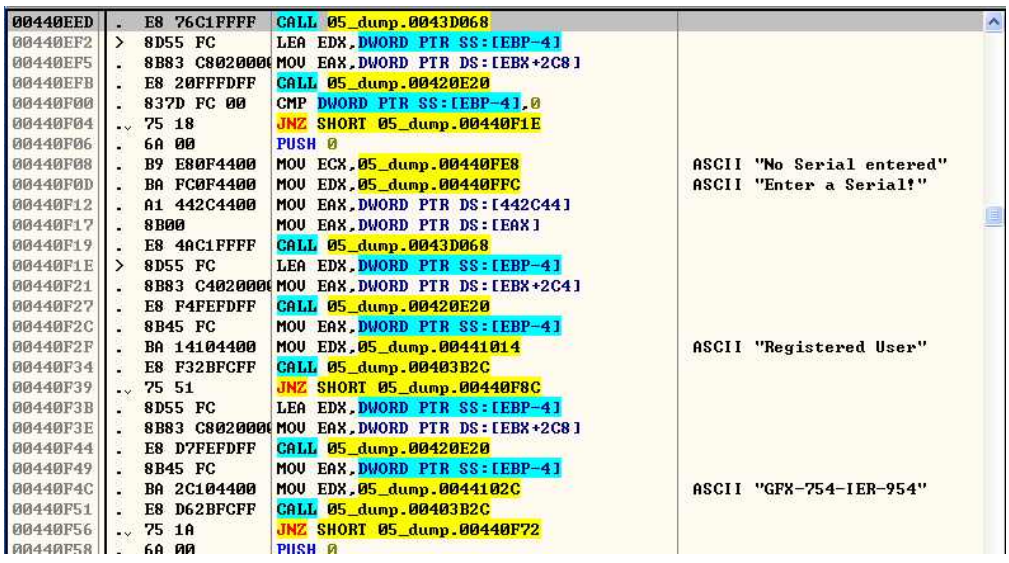
이제 분석을 할 수 있게 되었다.
시리얼이 틀렸을 때 메시지창이 뜨기 때문에 Back to user mode를 이용하여 메시지창이 생성되는 위치를 찾을 수 있다.



위치를 보니 MessageBoxA함수가 호출 되었고, 이 함수를 호출하는 함수의 내부에는 시리얼이 참인지 거짓인지를 판단하는 명령어가 없다. 따라서 이 함수를 호출 하기 전에 text에 출력할 내용을 미리 저장한 것 이라고 볼 수 있다. 명령어의 제일 위쪽을 따라가면 다음과 같이 함수 시작점을 찾을 수 있고, 아래에 이 함수를 호출하는 위치를 찾을 수 있다.



함수를 호출하는 경로들 중 2번째 주소를 따라가보면 다양한 텍스트 값들이 사용되고 있는 것을 알 수 있다.
그 중 시리얼 키처럼 보이는 텍스트도 발견 할 수 있다.



위 그림에서 프로그램이 시작되고 위쪽 칸에는 Registered User가 입력되고 아래쪽 시리얼

칸에는 GFX-754-IER-954가 입력되어야 할 것 같다. 프로그램을 구동 시켜서 그대로 입력하면 성공 메시지가 뜨는 것을 볼 수 있다.

따라서 문제의 시리얼은 GFX-754-IER-954 이다.