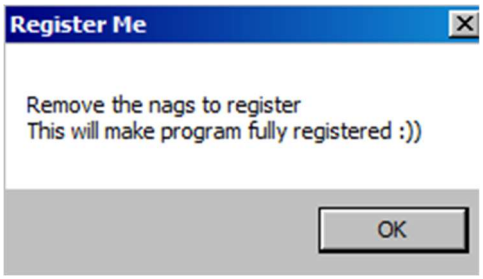
악성코드 분석 보고서

(sand-reversingwithlena-tutorials)

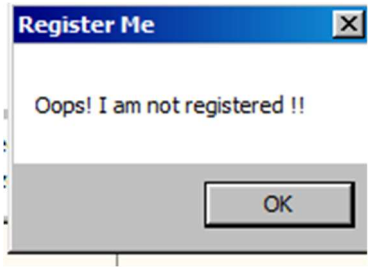
2025.05.27

02



해석: 등록 메시지를 제거하세요
그러면 프로그램이 완전히 등록됩니다 :))

해석: 지금 저를 등록해야 합니다!
등록은 꽤 쉬운 일입니다.
프로그램을 패치해서 nag 메시지를 제거하세요.



해석: 이런! 저는 등록되지 않았어요!!

즉, 프로그램을 패치해서 nag 메시지를 제거해야함.

```
pModule = NULL
GetModuleHandleA


Style = MB_OK|MB_APPLMODAL
Title = "Register Me"
Text = "Remove the nags to registerThis
hOwner = NULL
MessageBoxA
Arg4 = 0000000A
Arg3 = 00000000
Arg2 = 00000000
Arg1 = 00400000
Register.00401052
Style = MB_OK|MB_APPLMODAL
Title = "Register Me"
Text = "Oops! I am not registered !!"
hOwner = NULL
MessageBoxA
ExitCode
ExitProcess
```

GetModuleHandleA: 지정된 모듈에 대한 모듈 핸들을 검색하고 호출 프로세스에서 모듈을 로드하는 함수

MessageBoxA : 시스템 아이콘, 단추 집합 및 상태 또는 오류 정보와 같은 간단한 애플리케이션 관련 메시지가 포함된 모달 대화 상자를 표시하는 함수

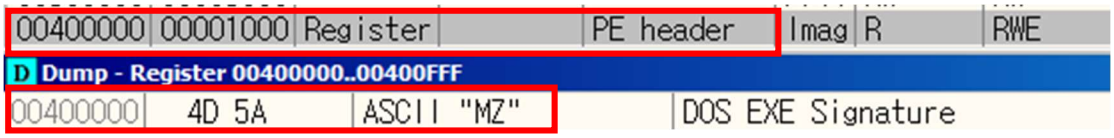
Register.00401052: 실행 파일 안에 있는 **자체 함수의 주소**

우선 첫 번째 등록메시지를 없애기 위해서는 저 함수를 뛰어 넘어야함.

```
00401000  $  6A 00          PUSH 0
00401002  .  E8 0D020000    CALL <JMP.&KERNEL32.GetModuleHandleA>
00401007  .  A3 1C314000    MOV DWORD PTR DS:[40311C],EAX
0040100C  .  83F8 00        CMP EAX,0
0040100F  .  74 13          JE SHORT Register.00401024
00401011  .  6A 00          PUSH 0
00401013  .  68 7D304000    PUSH Register.0040307D
00401018  .  68 34304000    PUSH Register.00403034
0040101D  .  6A 00          PUSH 0
0040101F  .  E8 C6010000    CALL <JMP.&USER32.MessageBoxA>
00401024  >  6A 0A          PUSH 0A
00401026  .  FF35 2031400(  PUSH DWORD PTR DS:[403120]
```

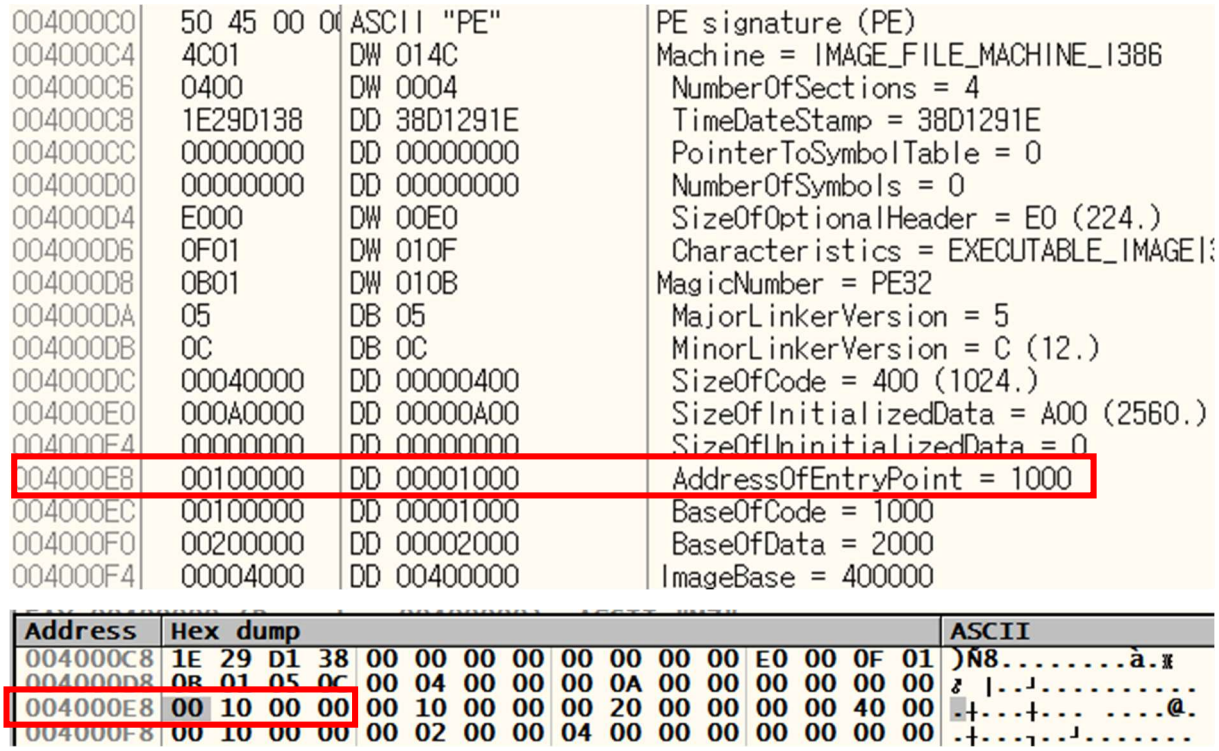
0x0040100F를 뛰어 넘기 위해서는 EAX=0이 되어야함. 하지만 EAX를 보면 0x00400000의 값이 있고 "MZ"가 저장되어 있는 걸 볼 수 있음. 즉, EAX는 절대 0이 될 수 없음.

```
Registers (FPU)
EAX 00400000 ASCII "MZ"
```



"M"(Memory Map)을 누르고 저 "PE header"을 누르면 "Dump"로 들어감.

그럼 EAX에 저장되는 주소가 0x00400000인 걸 알 수 있음.



저장되어 있는 걸 볼 수 있음.

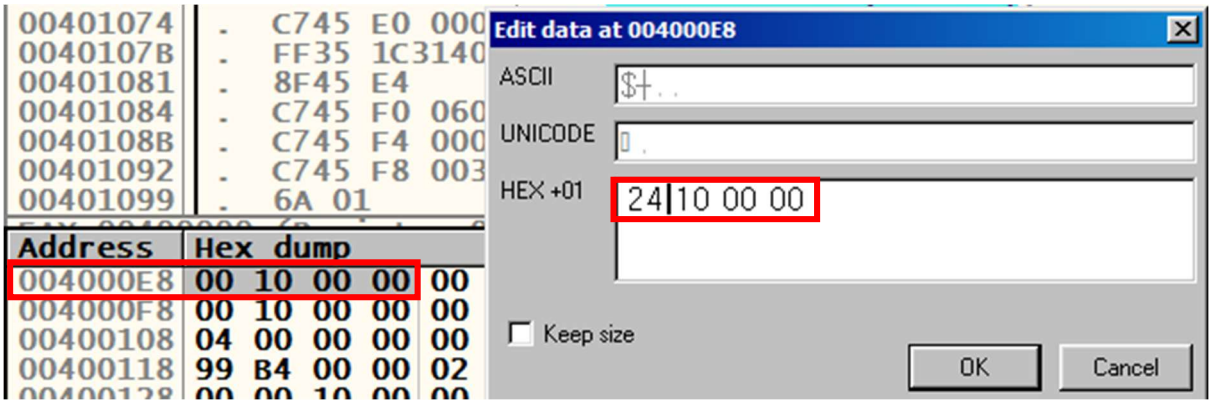
Image_NT_Header > Image_Optional_Header > AddressOfEntryPoint

PE가 로드된 후 실행을 시작할 주소에 대한 RVA를 가지고 있는 중요한 필드로 코드섹션 내에서 특정번지가 됨. EXE PE의 경우 해당 PE가 로드된 후 **이 프로세스의 메인 쓰레드가 실행하는 최초의 코드임**. 일반적으로 이 번지가 가리키는 값은 다음 런타임 시작 루틴의 RVA가 설정됨.

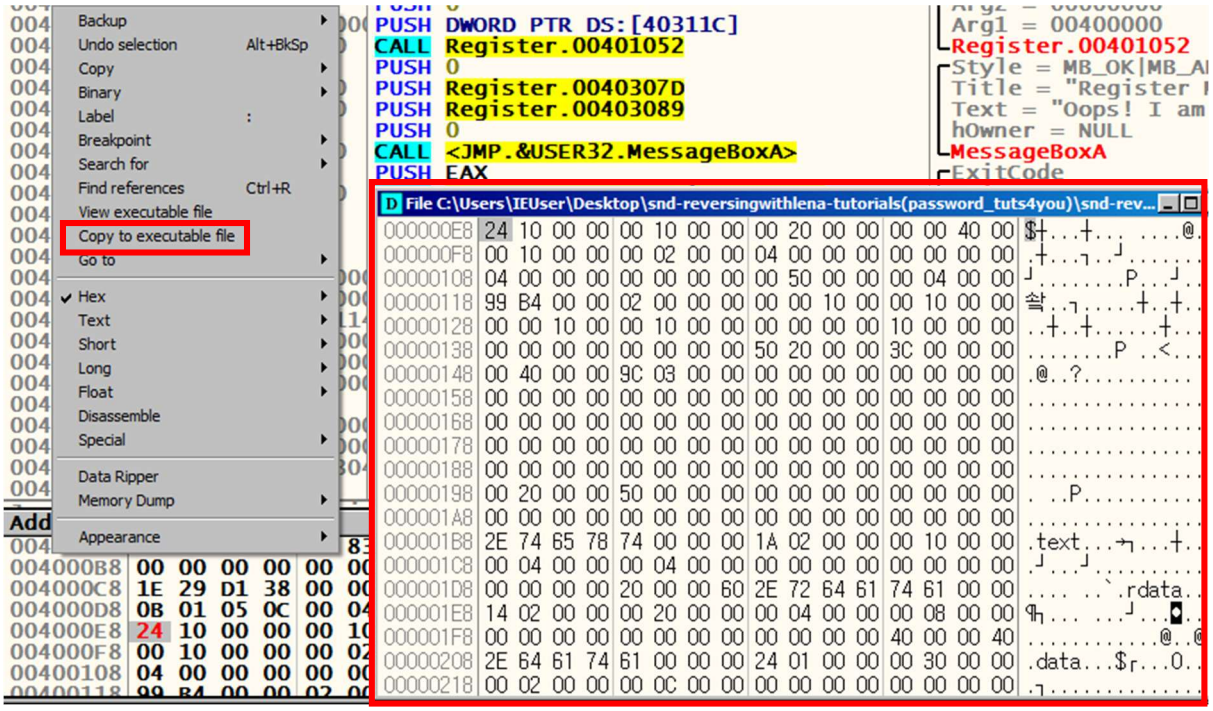
※ RVA (Relative Virtual Address): 특정 기준점(ImageBase)로 부터의 계산된 상대 주소를 뜻함.

VA == RVA + ImageBase

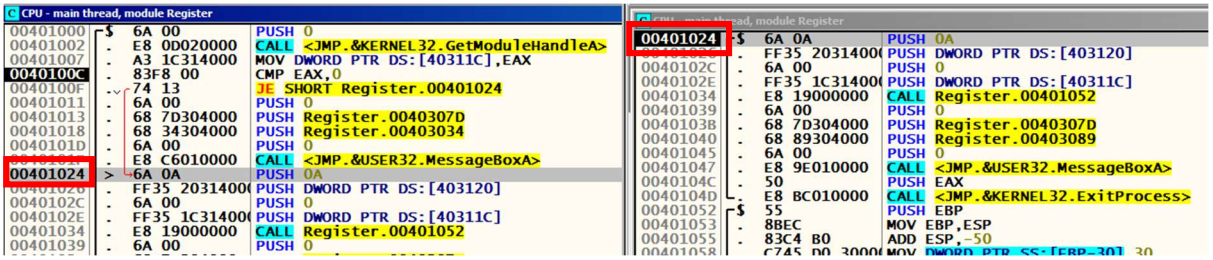
※ VA (Virtual Address): 변하지 않는 프로세스의 가상 메모리의 절대 주소를 뜻함.



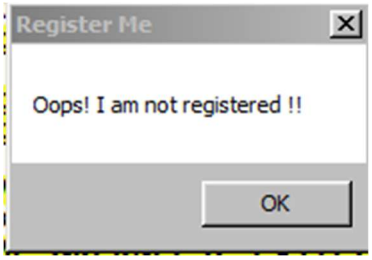
0x004000E8 이 부분을 '24 10 00 00'으로 바꿔줌. 왜냐하면 ImageBase 400000+Entry Point 1000의 값이 시작 점이기 때문에 AddressOfEntryPoint를 바꿔줌. 점프해야 할 주소가 0x00401024이기 때문임.



오른쪽 마우스를 누르고 'Copy to executable file'을 누르면 숫자가 바뀐 덤프 창 하나가 생김. 그 창을 다시 저장해주고 OllyDbug로 켜주면

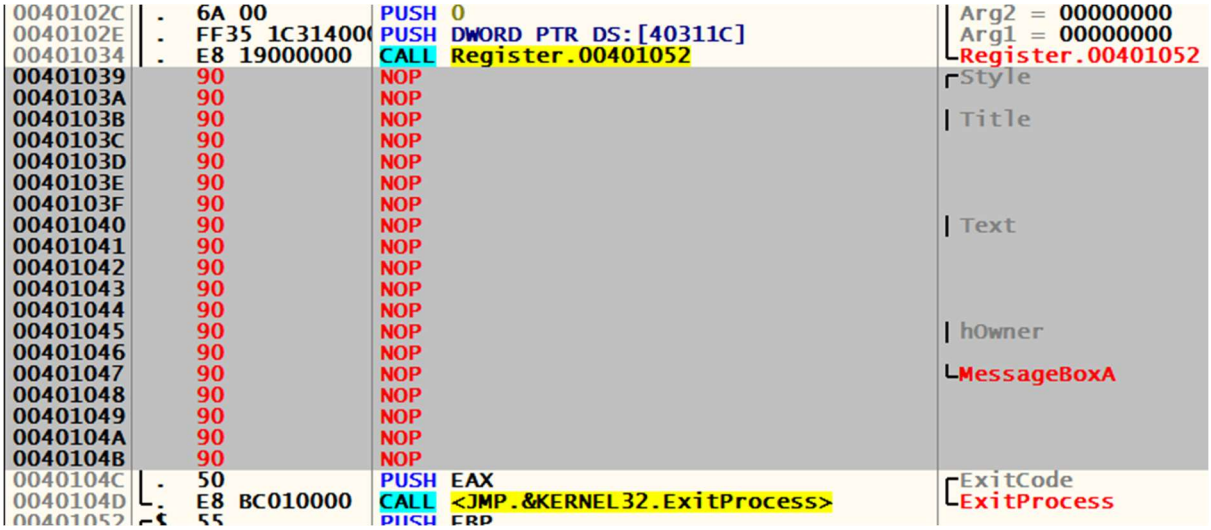


0x00401024부터 시작하는 파일 하나가 생성됨.



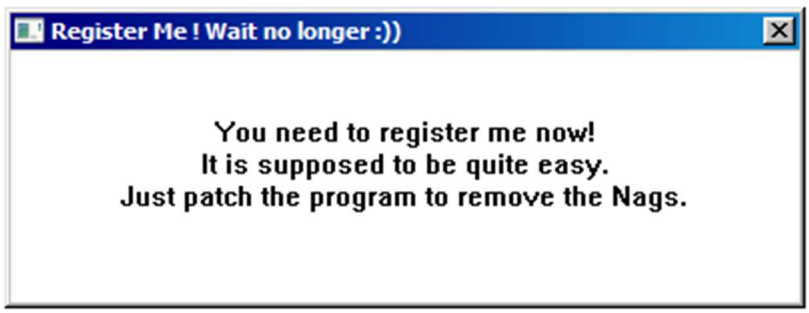
실행해보면 이 창이 또 발생하는 걸 볼 수 있음. 이걸 없애기 위해서는

0x00401039 – 0x00401047까지 Nop으로 바꿔줌.



이런 식으로 변경된 것을 확인할 수 있음.

다시 저장 후 클릭해보면 2번째 메시지만 발생하는 걸 볼 수 있음.



해결 완료!