Code Engn Basic

CodeEngn Basic 풀이 보고서 02~20

제출일 2018.12.25

이름 지현근

Basic06

1. 문제	Unpack을 한 후 Serial을 찾으시오. 정답인증은 OEP + Serial Ex) 00400000PASSWORD

2. 본문 제목1

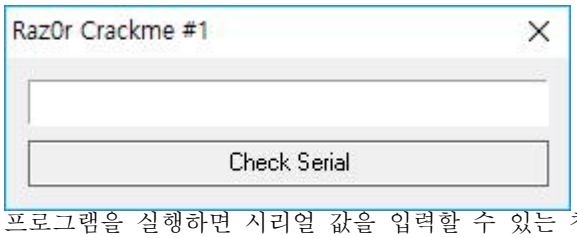
이 단계는 '본문1' 또는 '본문1 에코' 목록으로 작성합니다. 나눔명조 또는 나눔명조 에코 9pt 크기로 작성합니다. 이 단계는 '본문1' 또는 '본문1 에코' 목록으로 작성합니다.

[그림 캡션1]

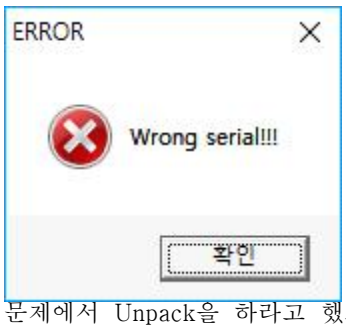
이 단계는 '본문1' 또는 '본문1 에코' 목록으로 작성합니다. 나눔명조 또는 나눔명조 에코 9pt 크기로 작성합니다. 이 단계는 '본문1' 또는 '본문1 에코' 목록으로 작성합니다..

1. 프로그램 실행

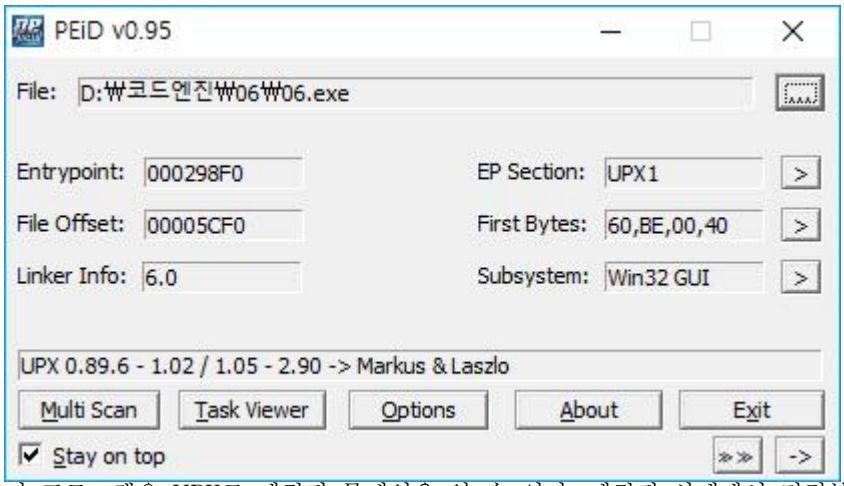
① 분석하기



프로그램을 실행하면 시리얼 값을 입력할 수 있는 창이 뜬다. 임의의 값을 입력하면 에러창을 띄운다.



문제에서 Unpack을 하라고 했으니 뭘로 패킹된 문제인지 확인하기 위해 Peid를 실행해서 확인해준다.



이 프로그램은 UPX로 패킹된 문제임을 알 수 있다. 패킹된 상태에서 정적분석을 할 경우 패킹 전의 프로그램 코드와 다르므로 언패킹을 해준 뒤 정적분석 단계로 넘어가도록 하자.

UPX 언패킹 방식은 굉장히 간단하다. upx를 설치하고 설치한 경로로 가서 upx를 실행해주면 된다.

UPX 실행 모습

필자는 UPX를 D드라이브에 설치하여서 D로 이동 후 실행해주었다. 해당 화면에서

```
[D:\]$ upx
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

Usage: upx [-123456789dlthVL] [-qvfk] [-o file] file..

Commands:
  -1     compress faster                  -9    compress better
  -d     decompress                       -l    list compressed file
  -t     test compressed file             -V    display version number
  -h     give more help                   -L    display software license
Options:
  -q     be quiet                         -v    be verbose
  -oFILE write output to 'FILE'
  -f     force compression of suspicious files
  -k     keep backup files
file..   executables to (de)compress

Type 'upx --help' for more detailed help.

UPX comes with ABSOLUTELY NO WARRANTY; for details visit https://upx.github.io
```

UPX의 기본적인 명령어들을 알려주고 있다. 우리는 언패킹을 해주어야 하므로 -d 명령어를 사용할 것이다.

```
[D:\]$ upx -d D:\코드엔진\06\06.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    159744 <-     26112   16.35%    win32/pe     06.exe

Unpacked 1 file.
```

성공적으로 언패킹한 모습

언패킹은 upx -d [문제파일 경로] 이렇게 입력해주면 가능하다. 언패킹을 할 때 주의해야 할 점은 프로세스를 로드시킨 상태. 즉 실행하고 있거나 디버거로 열어만 놓아도 언패킹이 불가능하다.

```
[D:\]$ upx -d D:\코드엔진\06\06.exe
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2018
UPX 3.95w       Markus Oberhumer, Laszlo Molnar & John Reiser   Aug 26th 2018

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
    159744 <-     26112   16.35%    win32/pe     06.exe
upx: D:\코드엔진\06\06.exe: IOException: D:\코드엔진\06\06.exe: Permission denied

Unpacked 1 file: 0 ok, 1 error.
```

프로세스를 로드시켜놔서 오류가 난 모습

② 결과정리

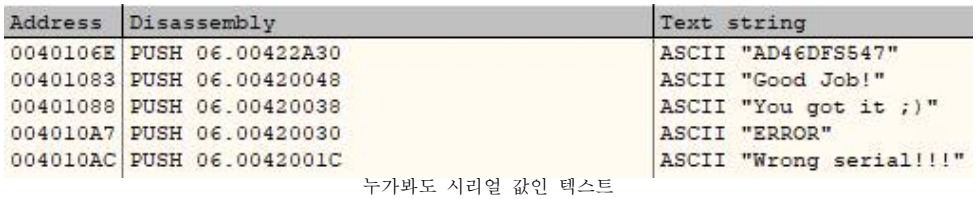
파일이름	06.exe
실행	가능
동적분석을 통해 알아낸 정보	1. 잘못된 시리얼 값을 입력하면 오류창이 뜬다. 2. UPX로 패킹되어 있다.

3. 정적분석

이 문제에서는 OEP와 시리얼값을 찾아야 한다. 그렇기 때문에 디버거로 프로그램을 열어주고 바로 텍스트를 검색하는 것이 아니라 열자마자 현제 있는 주소를 메모장에 적어두자.

OEP 주소

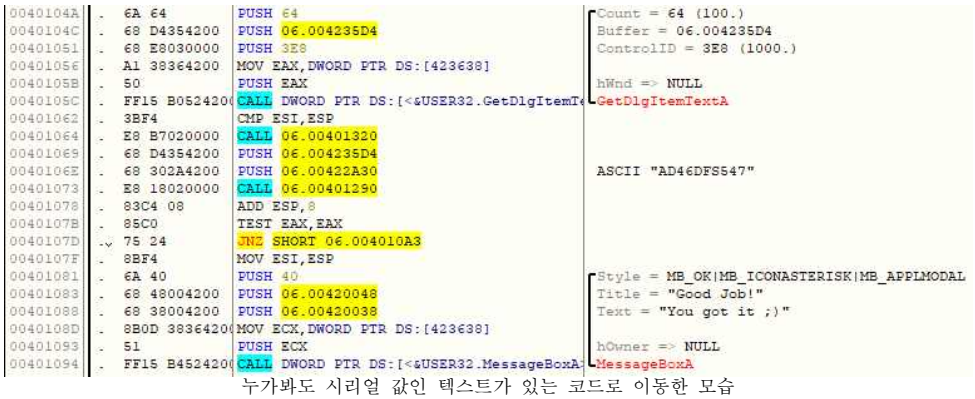
여기서 OEP는 오리지날 엔트리 포인트, 즉 원래 시작 주소이다. OEP주소를 찾았으니 이제 시리얼 값을 찾기 위해 텍스트를 검색해주자.



Address	Disassembly	Text string
0040106E	PUSH 06.00422A30	ASCII "AD46DFS547"
00401083	PUSH 06.00420048	ASCII "Good Job!"
00401088	PUSH 06.00420038	ASCII "You got it ;)"
004010A7	PUSH 06.00420030	ASCII "ERROR"
004010AC	PUSH 06.0042001C	ASCII "Wrong serial!!!"

누가봐도 시리얼 값인 텍스트

검색하자마자 맨 위에 누가봐도 시리얼 값이 문자열이 적혀있다. 누가봐도 시리얼 값인 문자열로 이동해서 분석해보자.



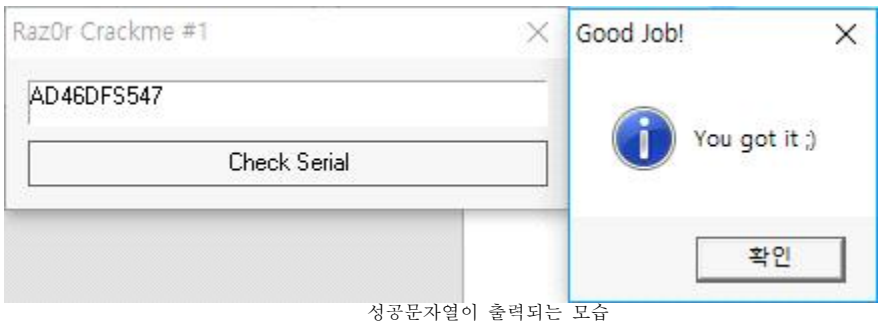
누가봐도 시리얼 값인 텍스트가 있는 코드로 이동한 모습

코드를 보면 [GetDlgItemTextA]함수에서 시리얼값을 입력받고 00401073에서 호출되는 함수로 “AD46DFS547”와 비교하여 일치하면 EAX에 0을 넣어서 JNZ로 통해서 점프되지 않고 성공문자열을 출력하는 모습이다.

---

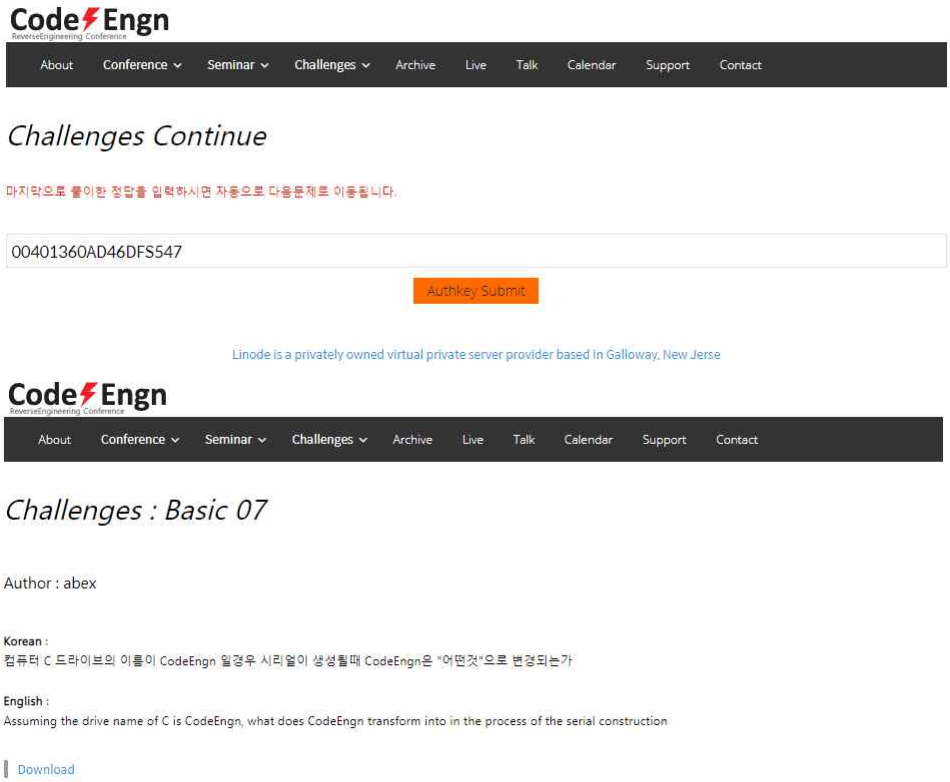
4. 문제해결

이제 알아낸 정보를 바탕으로 문제를 해결해보자. 우리가 알아낸 정보는 OEP주소가 00401360이라는 것과 시리얼 코드로 예상되는 코드는 “AD46DFS547”이라는 것이다. 문제를 해결하기 위해 프로그램을 실행해서 시리얼 코드가 맞는지 확인해보자.



성공문자열이 출력되는 모습

이제 정답을 적으면 된다. 문제에서는 정답인증이 OEP + Serial이라고 했으니 OEP 주소인 00401360과 시리얼 코드인 AD46DFS547을 합쳐서 00401360AD46DFS547이 정답일 것이다. 정답이 맞는지 확인해보자



정답이 맞기 때문에 다음 문제로 넘어가는걸 확인할 수 있다.