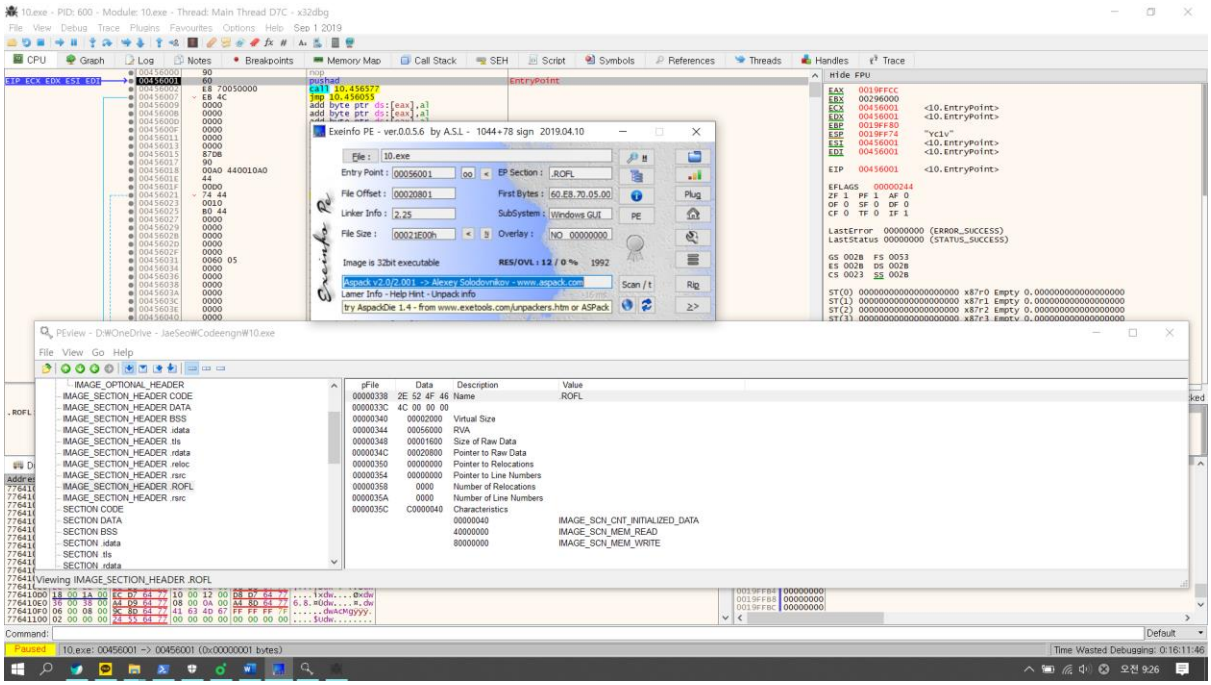
10.exe - OEP를 구한 후 '등록성공' 으로 가는 분기점의 OPCODE를 구하시오.

정답인증은 OEP + OPCODE (EX) 00400000EB03)

디버깅을 하기전에 Packing이 되어 있는지 PE 분석을 해본다.



분석 결과 "Aspack"이라는 Packing 프로그램으로 패킹이 되어 있는 것을 확인할 수 있다.

X32dbg로 분석을 통해 Unpacking을 한다.

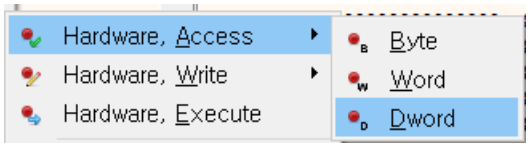
Pushad를 통해 레지스터를 stak에 올리고 있는 모습을 볼 수 있다.

```
EAX   0019FFCC
EBX   00268000
ECX   00456001     <10.EntryPoint>
EDX   00456001     <10.EntryPoint>
EBP   0019FF80
ESP   0019FF54     <&EntryPoint>
ESI   00456001     <10.EntryPoint>
EDI   00456001     <10.EntryPoint>
```

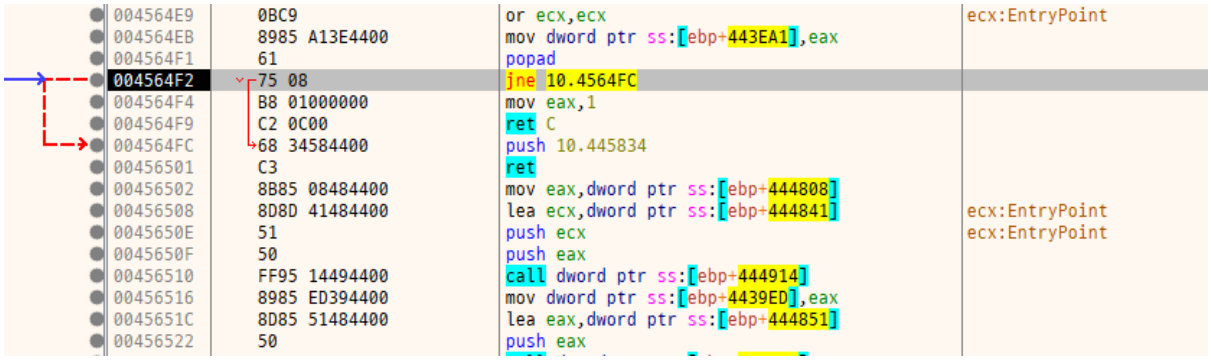
F8를 눌러 ESP의 변경이 되는 모습을 관찰한다.

이제 복호화 작업을 끝내고 popad를 통해 레지스터를 복구할 것이다.

이때 ESP주소를 기점으로 덤프를 뜨고 memory에서 access 될 때 Break Point를 걸어 관찰을 한다.

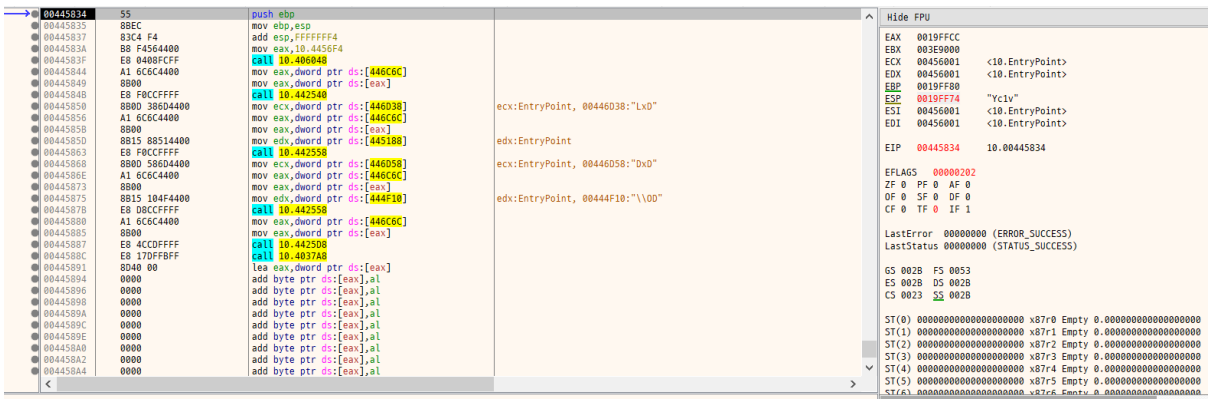


이제 F9를 눌러 Break Point에 걸리도록 한다.



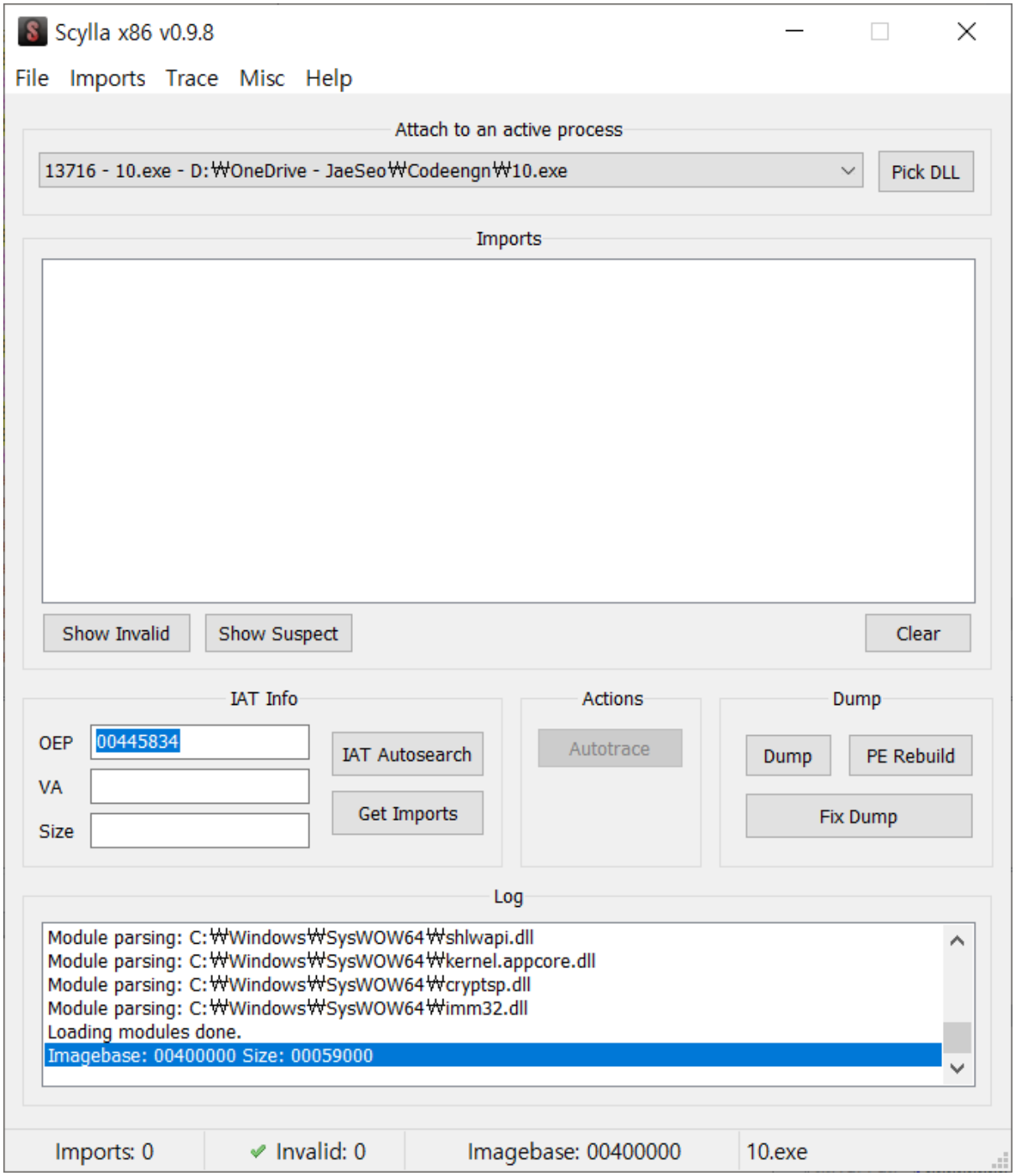
그 결과 위에 Popad로 복구하는 모습을 볼 수가 있다.

이때 return되는 곳을 확인해본다.



초기에 저장된 레지스터가 전부 복구된 것을 볼 수 있다. 이것을 통해 OEP라는 것을 알 수 있다.

이때 x32dbg의 플러그인 Scylla를 실행해서 덤프를 뜨도록 한다.



현재 찾았던 OEP를 기입을 해준다.

그리고 IAT Auosearch를 통해 자동으로 입력 값을 받는다. 그리고 Get Imports를 통해 Import 대상을 불러오고 DUMP를 시킨다.

그리고 FIX DUMP를 통해 정상적인 IAT를 주입시켜 준다.

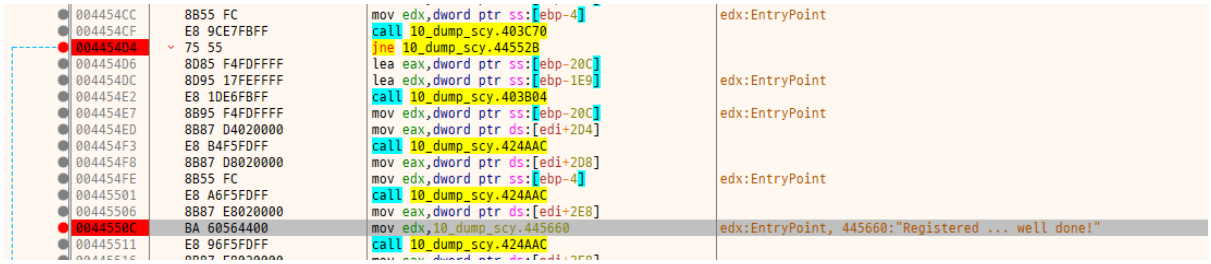
그 결과 이러한 파일이 생성이 된다.

이제 정상적으로 Unpacking 된 10_dump_SCY.exe 를 가지고 디버깅을 진행해본다.

일단 이상태에서는 파악하기 힘들기 때문에 String 검색을 통해 찾기로 한다.

Address	Disassembly	String
00441CEA	push 10_dump_scy.441FA0	"RegisterAutomation"
0044550C	mov edx,10_dump_scy.445660	"Registered ... well done!"

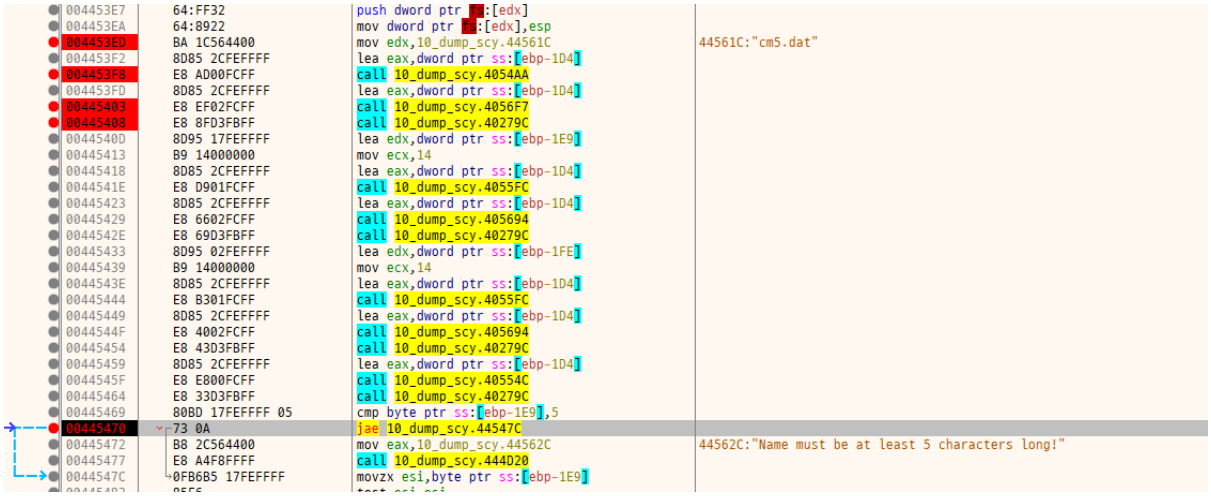
"Register"라는 키워드를 통해 검색을 하니 성공했을 때의 메시지를 발견할 수 있었다.



이때 위에 JNE을 통해 분기점을 가지고 있는데 이것을 통해 체크를 하는 것을 알 수가 있다.

문제로 돌아와 성공으로 가는 분기점의 OPCODE는 004454D4라는 것을 알 수가 있다.

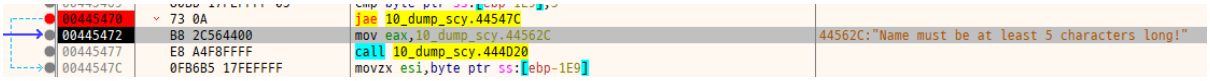
정답은 구하였지만 계속 분석을 해본다. 위로 올려서 String 문자열을 보다 보면



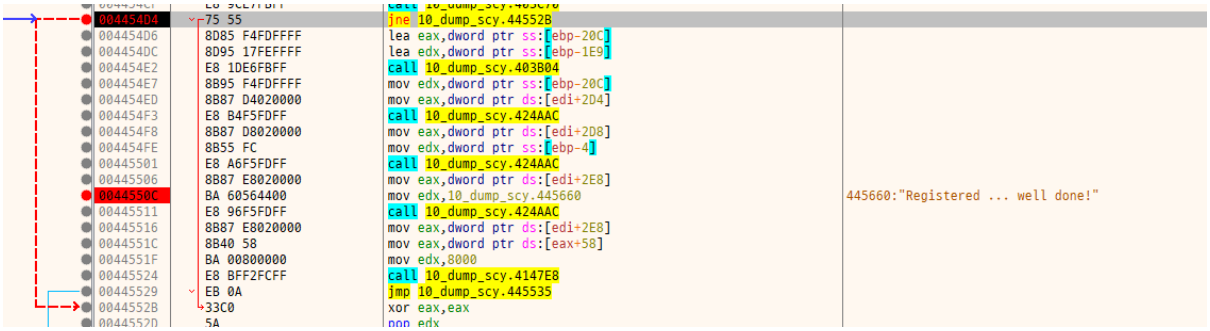
이러한 글자 수에 대한 안내 메시지와 cm5.dat이라는 문자열을 볼 수 있다. 이때 CALL 하는 부분에 Break Point를 걸어 F7으로 진행을 해본다.

중간에 CreatFileA라는 함수를 호출하여 cm5.dat를 확인하고 있다는 것을 알 수 있다.

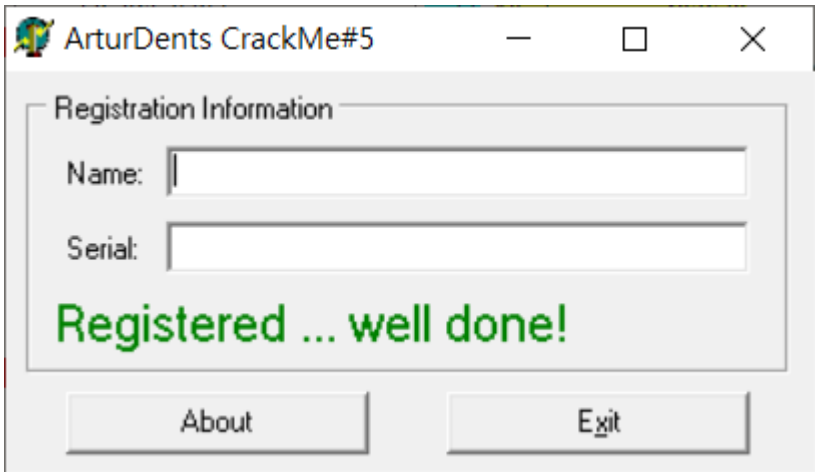
이때 cm5.dat 파일을 생성하여 어떻게 작동하는지 분석한다.



아무 것도 입력을 하지 않아 글자수 경고 메시지를 출력준비 하는 것을 볼 수가 있다.



그 다음 분기점에서 ZF 플래그를 1로 바꿔 성공 쪽으로 이동할 수 있게 한다.



결과 성공 메시지를 볼 수 있다.

정답: 004458347555