악성코드 분석 보고서

(sand-reversingwithlena-tutorials)

2025.06.07

05

1. 문제

체험판이라고 나오는 이 창을 없애야한다.

창을 다 닫았을 때 생기는 결제 유도 창을 없애야한다.

2. 해결 방법

"**결과를 실행하는 함수**"일 땐 Called from을 보고, 그걸 실행하게 만든 "**판단 로직이 있을 함수**"일 땐 Procedure를 눌러라!"

'F8' : 한 줄씩 내려가기

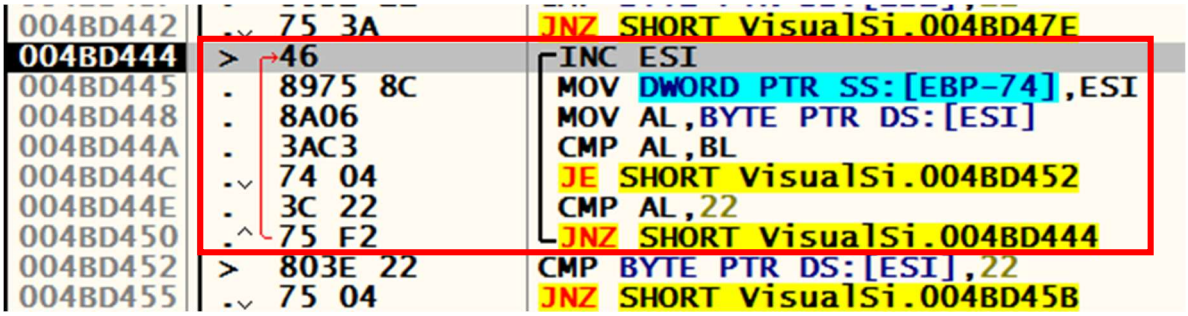
'F8+ctrl' : 한 줄씩 자동으로 내려가기

'F9' : 실행

'F2' : breakpoint 걸기

'F7' : 함수 내부 들어가서 보기

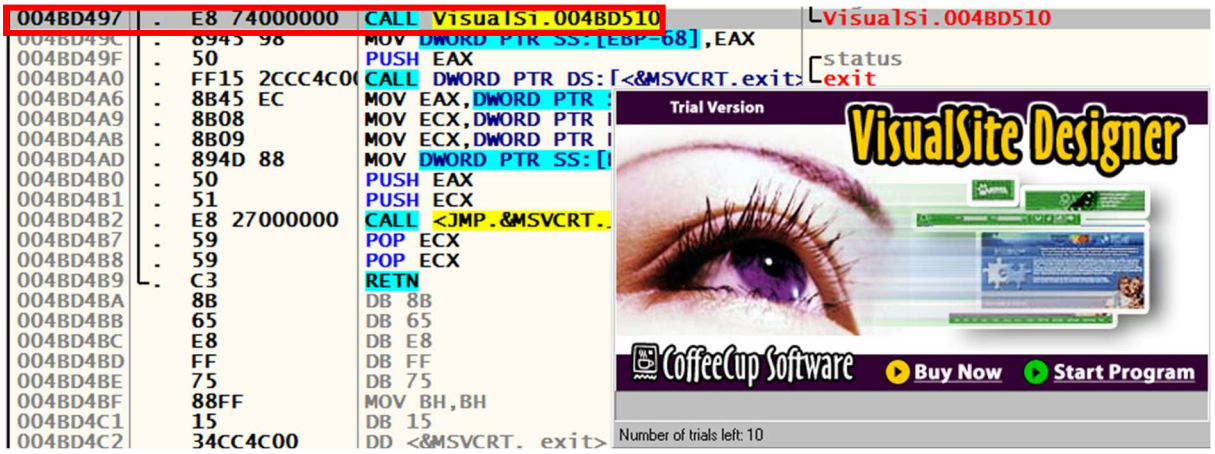
1) 체험판 창



'F8+ctrl'를 눌러서 실행하는 것을 보면 이렇게 반복되는 구간을 볼 수 있다. 여기서 탈출하기 위해서는 그 다음 구문인 0x004BD452에 breakpoint를 걸어준다.

걸고 'F9'를 누르면 빠르게 나올 수 있다.

'F8' 눌러서 실행하다보면 프로그램이 실행되는 구간이 있다.

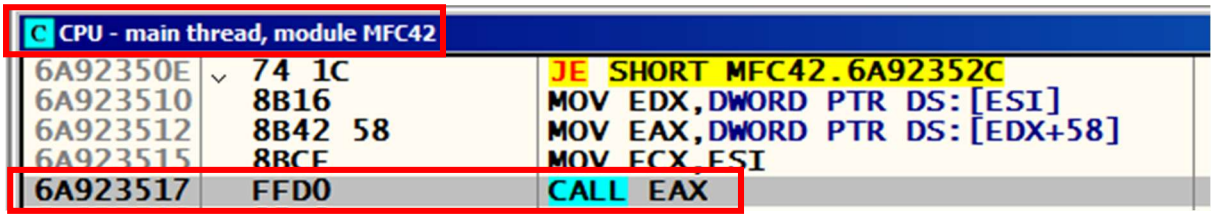


0x004BD497에서 실행되는 걸 볼 수 있다. 그럼 이 부분은 breakpoint를 걸어서 멈춰준다.

'F2+ctrl'을 누르고 재시작 후 'F9'을 눌러서 breakpoint가 걸린 0x004BD497로 와준다. 거기서 'F7'을 눌러 함수 내부를 본다. 또 'F8' 눌러서 실행하다보면 프로그램이 실행되는 구간이 또 있다.

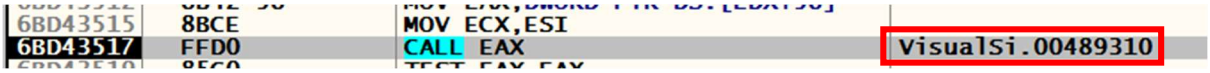
0x004BD520에서 breakpoint를 걸어서 멈춰준다.

'F2+ctrl'을 누르고 재시작 후 'F9'을 눌러서 breakpoint가 걸린 0x004BD520로 와준다. 거기서 'F7'을 눌러 함수 내부를 본다. 또 'F8' 눌러서 실행하다보면 프로그램이 실행되는 구간이 또 있다.

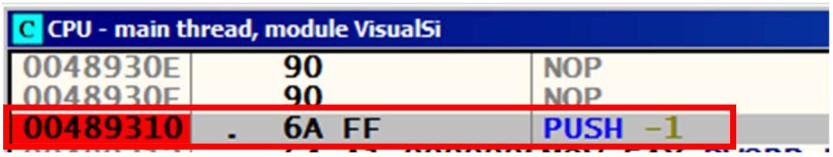


아까 실행하고 있던 주소랑 완전 다른 걸 볼 수 있다. 아까는 00으로 시작했고 지금은 6A로 시작하는 걸 볼 수 있다. 왜냐하면 MFC42.dll이라는 외부 함수로 들어왔기 때문이다. 그래서 주소가 완전 달라지고 여기서는 Breakpoint 사용이 안된다. (Hardware

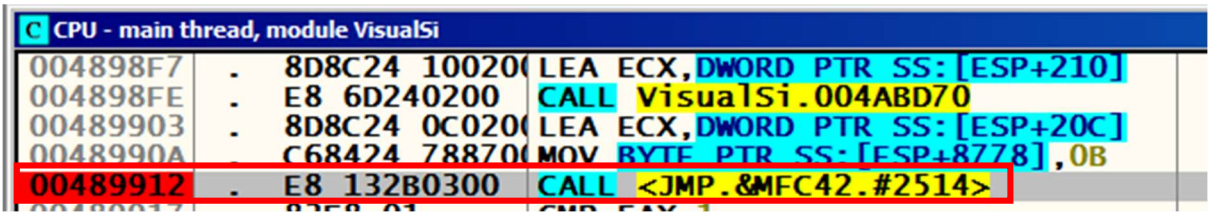
breakpoint가 있긴한데 오류가 발생함.)



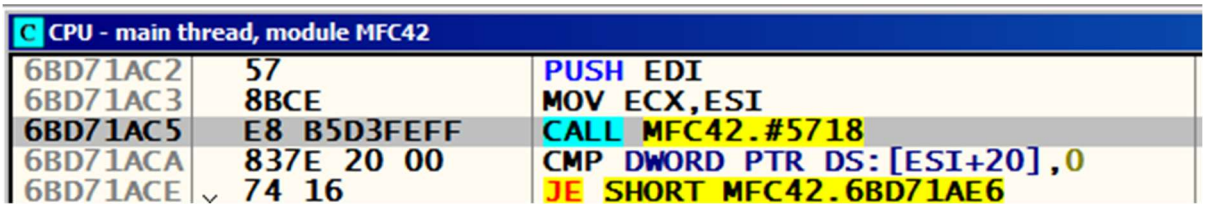
그래서 breakpoint를 걸기 위해서는 0x00489310으로 들어가준다.



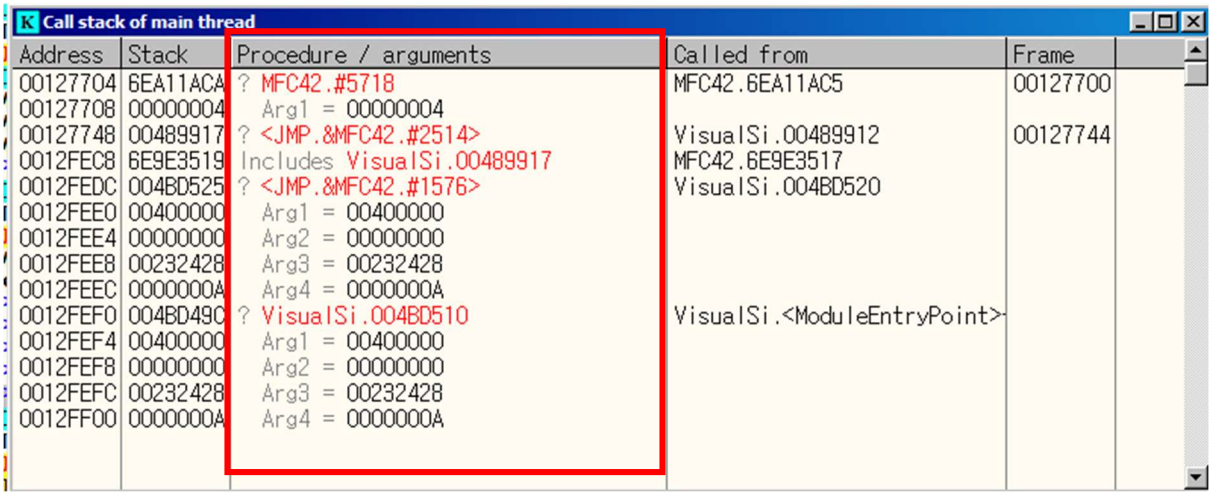
0x00489310에서 breakpoint를 걸어주고 'F8'을 눌러 실행하다보면 프로그램이 실행되는 구간이 또 있다. (0x00489310에서는 굳이 breakpoint를 안걸어주고 실행해줘도 된다.)



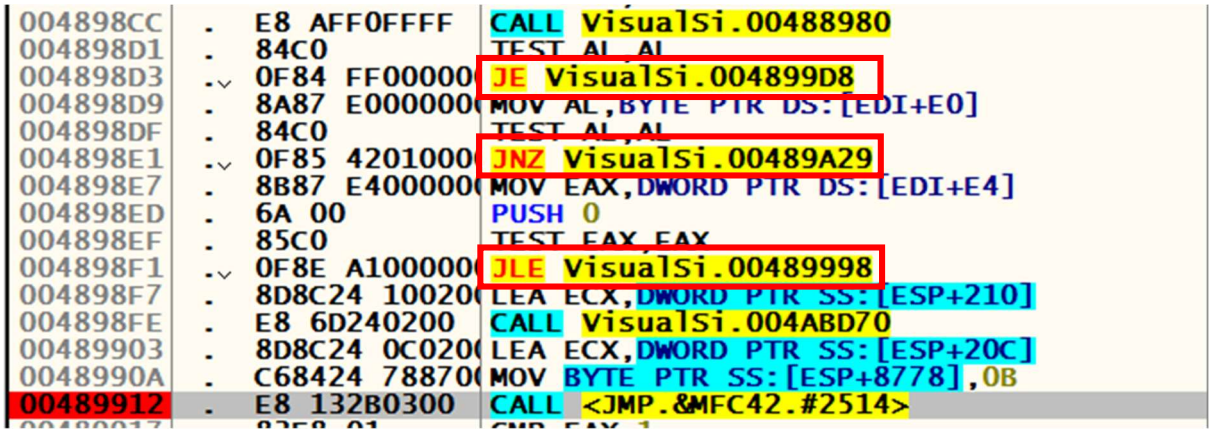
0x00489912에서 breakpoint를 걸어주고 'F2+ctrl'을 누르고 재시작 후 'F9'을 눌러서 breakpoint가 걸린 0x00489912로 와준다. 거기서 'F7'을 눌러 함수 내부를 본다. 또 'F8' 눌러서 실행하다보면 프로그램이 실행되는 구간이 또 있다.



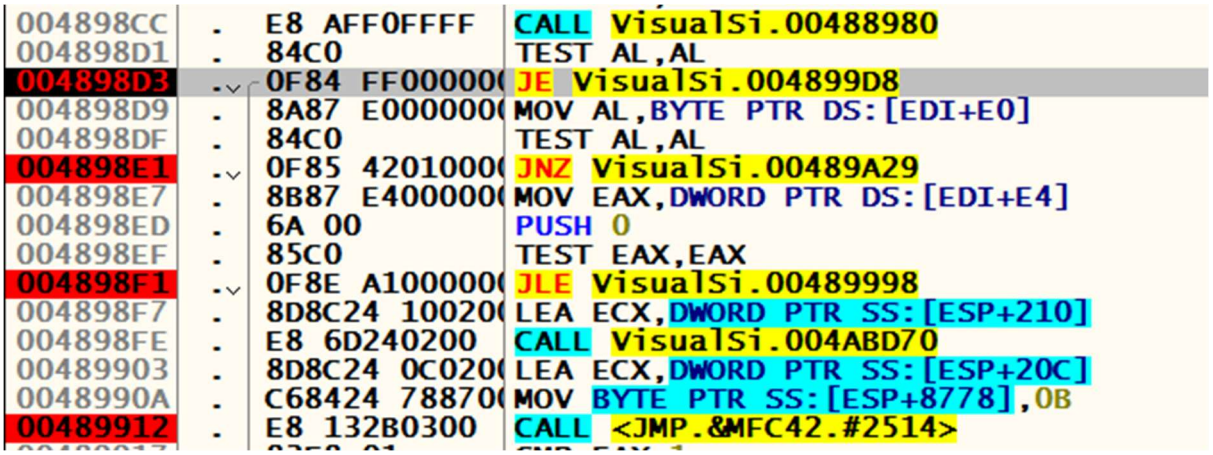
이 부분은 breakpoint를 걸지 못하니까 다시 재시작을 해준 후 바로 'F7을 눌러서 봐준다. 그리고 'F8'를 누르면 계속 반복하는 걸 볼 수 있다. K를 누르면 call stack을 볼 수 있다.



근데 우리는 체험판 창을 완전 없애는게 아니라 판단 로직을 이용해 수정하고 싶으니까 Procedure 쪽을 봐서 내부 함수가 있나 본다. 그럼 VisualSi.00489917 내부 함수 1개가 보인다. (지금 call stack 창은 위->아래로 쌓이기 때문에 위에서부터가 내가 최근에 실행한 함수이다. 만약 내부 함수가 여러 개 있으면 차례로 함수 코드 주변에 조건 분기분이 있나 봐준다.)



클릭해서 코드를 보면 내부 함수 위에 이렇게 3개가 있는 걸 볼 수 있다.



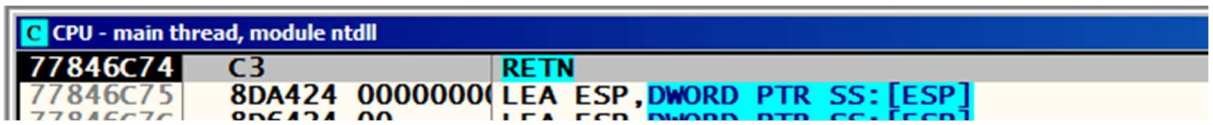
이렇게 3개 다 breakpoint를 걸어 주고 다시 실행해주고 비교해보면

JE (jump equal) : 비교 결과가 같을 때 점프

JNZ (jump not zero) : 결과가 0이 아닐 때 점프

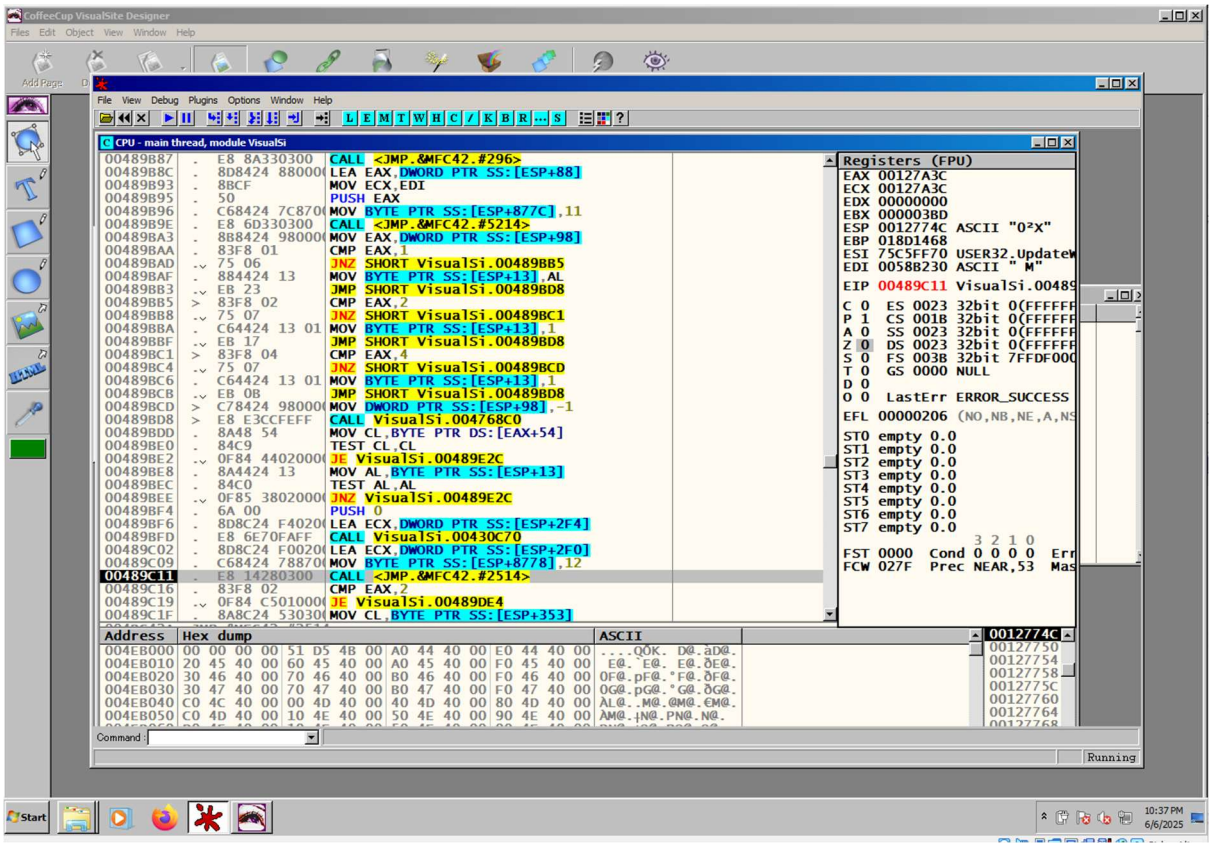
JLE (jump less or equal) : 작거나 같을 때 점프

첫 번째 JE를 보면 같으면 ZF=1이 되어야 점프한다. 그러면 ZF=1로 바꿔주고 'F8'로 실행해보면



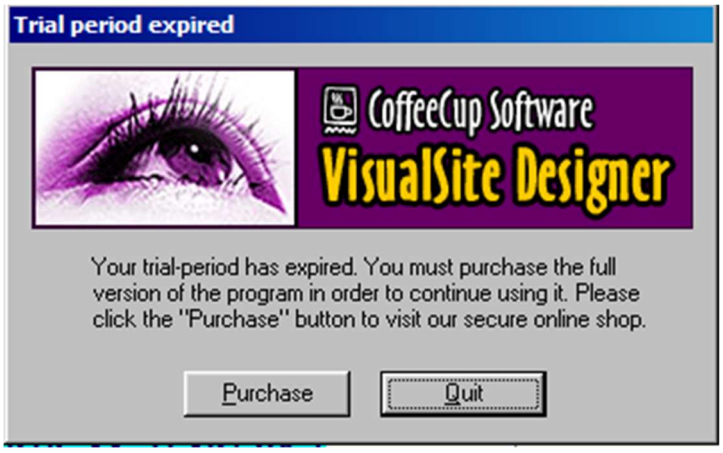
프로그램이 끝나는 걸 볼 수 있다. 그럼 이 부분은 바꾸면 안된다.

두 번째 JNZ를 보면 ZF=0이 되어야 점프를 한다. 그러면 ZF=0으로 바꾸고 'F8'로 실행해보면



뒤에 창이 실행되고 멈추는 걸 볼 수 있다. 그럼 이 코드를 수정해줘야한다.

세 번째 JLE를 보면 같으면 ZF=1이 되어야 점프한다. 그러면 ZF=1로 바꿔주고 'F8'로 실행해보면

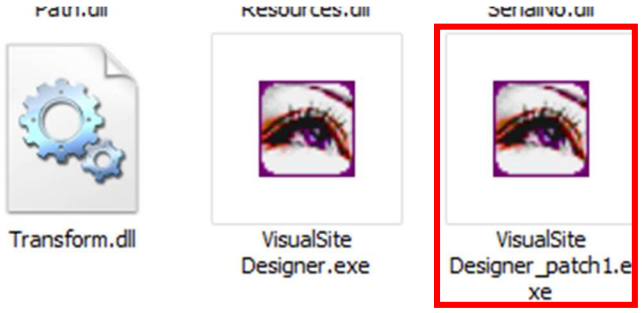


해당 창이 나오고 실행이 안되는 걸 볼 수 있다. 이 코드는 수정하면 안된다.

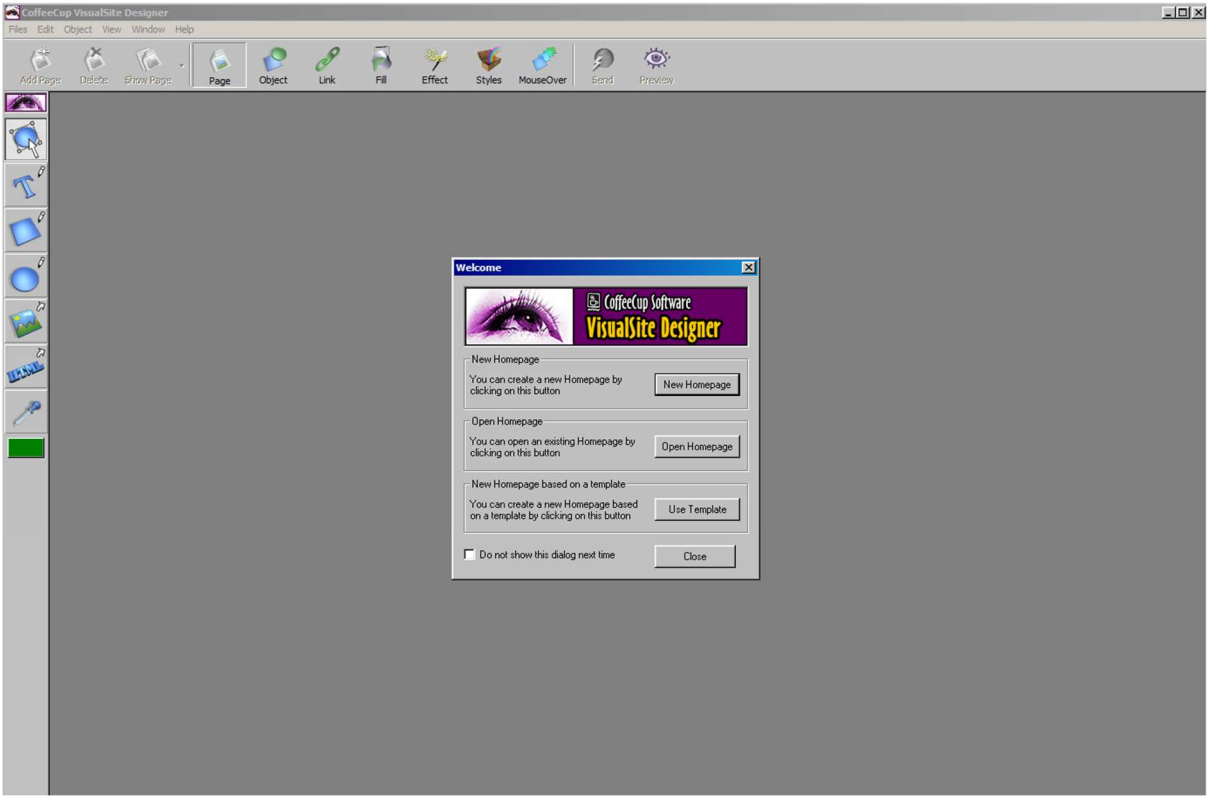
```
004898D1   .   84C0           TEST AL,AL
004898D3   .v  0F84 FF00000(  JE VisualSi.004899D8
004898D9   .   8A87 E000000(  MOV AL,BYTE PTR DS:[EDI+E0]
004898DF   .   84C0           TEST AL,AL
004898E1   v   E9 43010000    JMP VisualSi.00489A29
004898E6       008B 87E4000(  ADD BYTE PTR DS:[EBX+E487],CL
004898EC   ?   006A 00        ADD BYTE PTR DS:[EDX],CH
004898EF   .   85C0           TEST EAX,EAX
004898F1   .v  0F8E A100000(  JLE VisualSi.00489998
004898F7   .   8D8C24 10020(  LEA ECX,DWORD PTR SS:[ESP+210]
004898FE   .   E8 6D240200    CALL VisualSi.004ABD70
00489903   .   8D8C24 0C020(  LEA ECX,DWORD PTR SS:[ESP+20C]
0048990A   .   C68424 78870(  MOV BYTE PTR SS:[ESP+8778],0B
00489912   .   E8 132B0300    CALL <JMP.&MFC42.#2514>
00489917   .   83F8 01        CMP EAX,1
```

JNZ VisualSi.00489A29 -> JMP VisualSi.00489A29로 바꾸고 새로 저장해준다.

Copy to executable > Selection -> 덤프 창에서 Save File



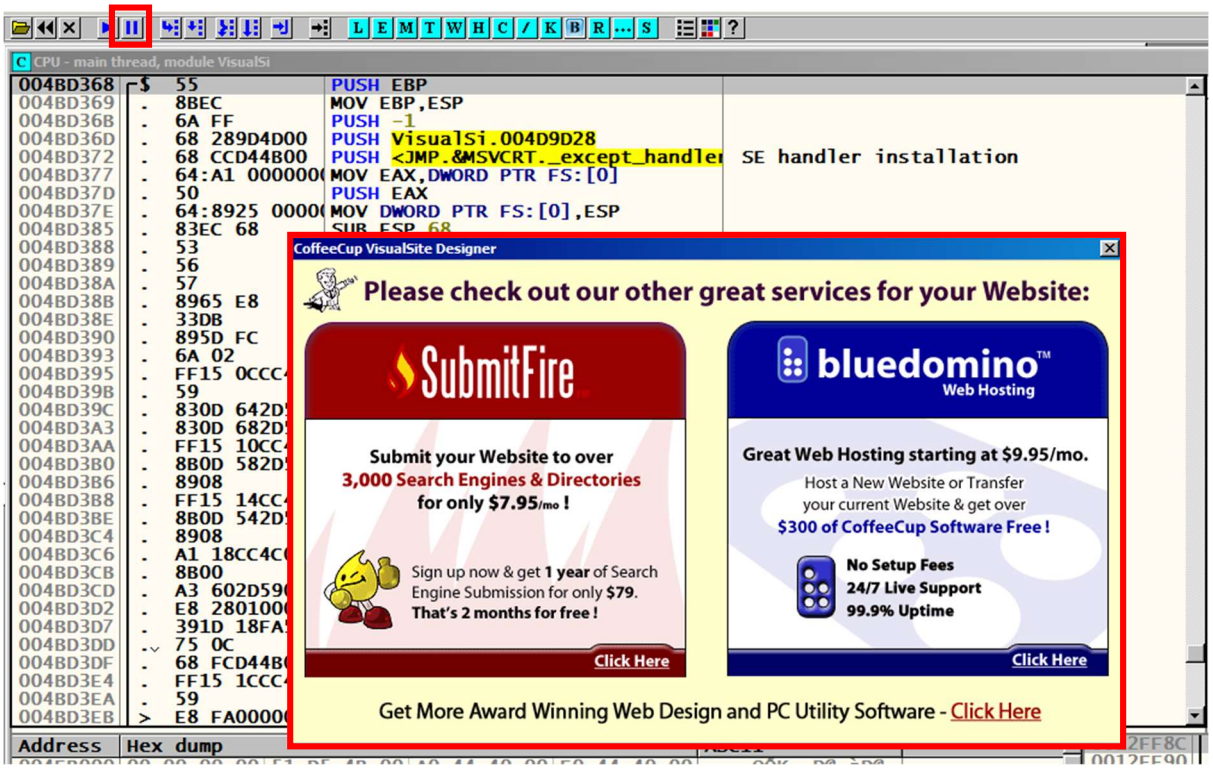
클릭해서 실행해보면


CoffeeCup VisualSite Designer
Files Edit Object View Window Help
Add Page Delete Show Page Page Object Link Fill Effect Styles MouseOver Send Preview
Welcome
CoffeeCup Software
VisualSite Designer
New Homepage
You can create a new Homepage by clicking on this button
New Homepage
Open Homepage
You can open an existing Homepage by clicking on this button
Open Homepage
New Homepage based on a template
You can create a new Homepage based on a template by clicking on this button
Use Template
Do not show this dialog next time
Close

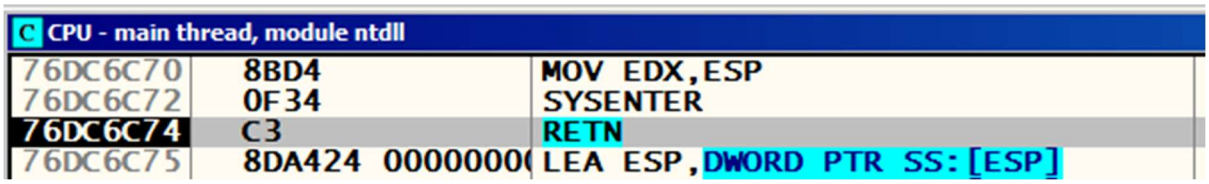
실행하는 걸 볼 수 있다.

2) 결제 유도 창

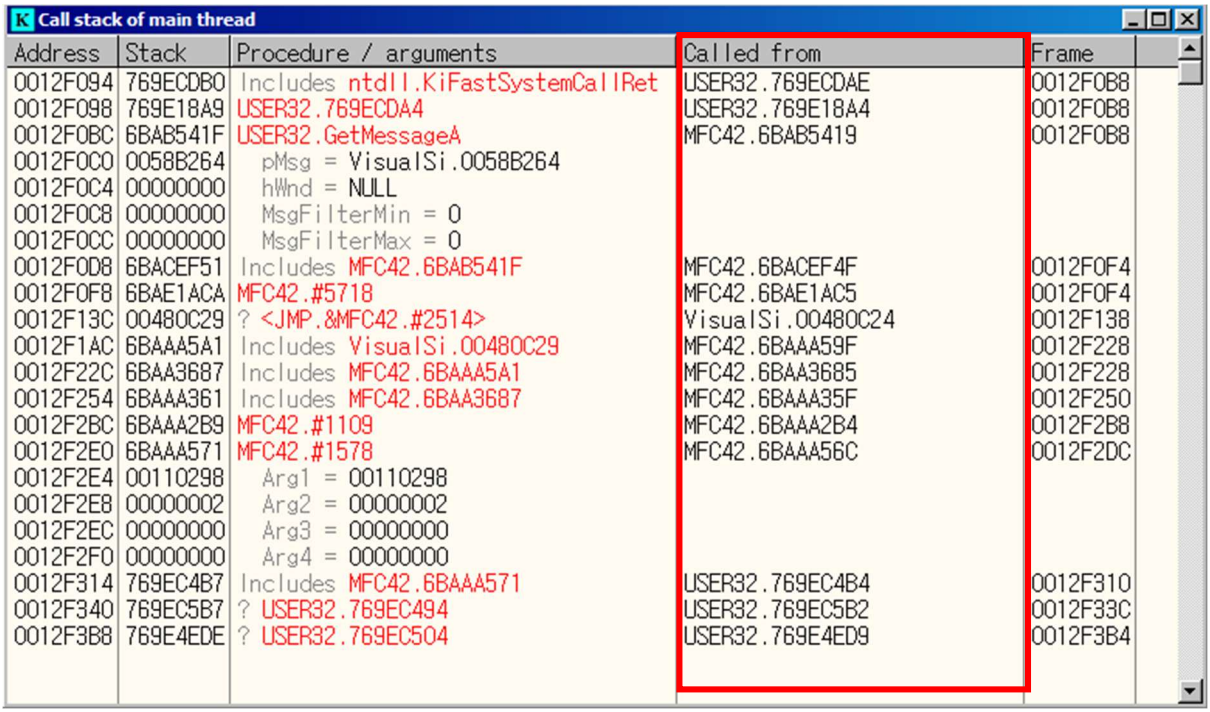
'F9'을 눌러서 프로그램을 실행시켜준다.



모든 창을 끄고 결제 유도창만 남은 채로 빨간 박스인 '멈춤' 버튼을 누르면

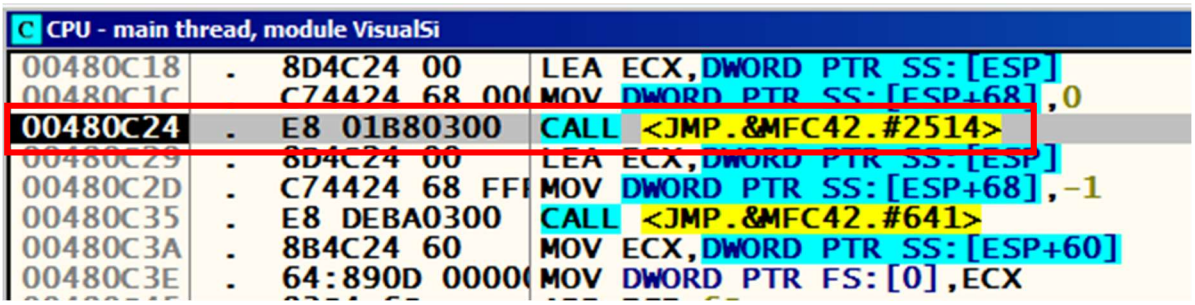


끝난 걸 볼 수 있다. 그 후 call stack창을 열어서 보면 실행된 내부 함수를 볼 수 있다.



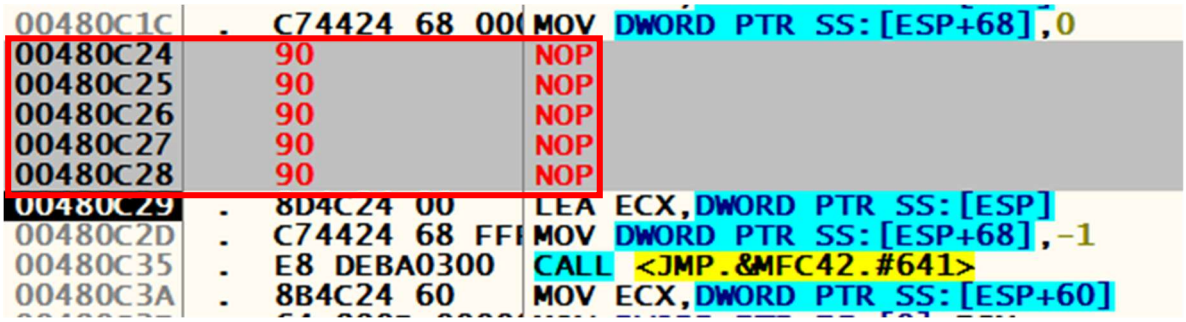
이번에는 Called from 쪽을 봐줘야한다. (왜냐하면 이거는 판단 로직이 아니라 그냥 실행을 안하게 만들거라서 지금 실행되고 있다는 뜻은 'Procedure' 보다는 결과를 실행하기 전의 함수를 보는거라서 'Called from'을 봐줘야한다.)

그럼 최근에 실행된 내부 함수인 VisualSi.00480C24가 보인다. 그걸 클릭해서 들어가면



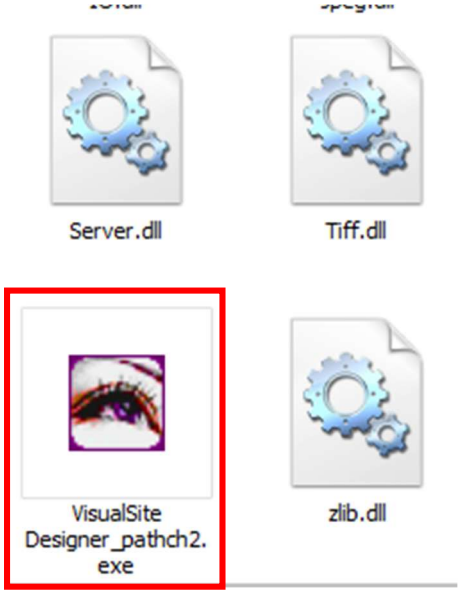
0x00480C24를 breakpoint를 걸어주고 다시 실행해보면 마지막 결제 유도 창이 발생하지 않는걸 볼 수 있다.

그럼 이 부분을 Binary > Fill with Nops로 바꿔준다.



이렇게 바뀌는 걸 볼 수 있다.

Copy to executable > Selection -> 덤프 창에서 Save File



실행해보면 마지막 결제 유도 창도 발생하지 않는 걸 볼 수 있다.

문제 해결!