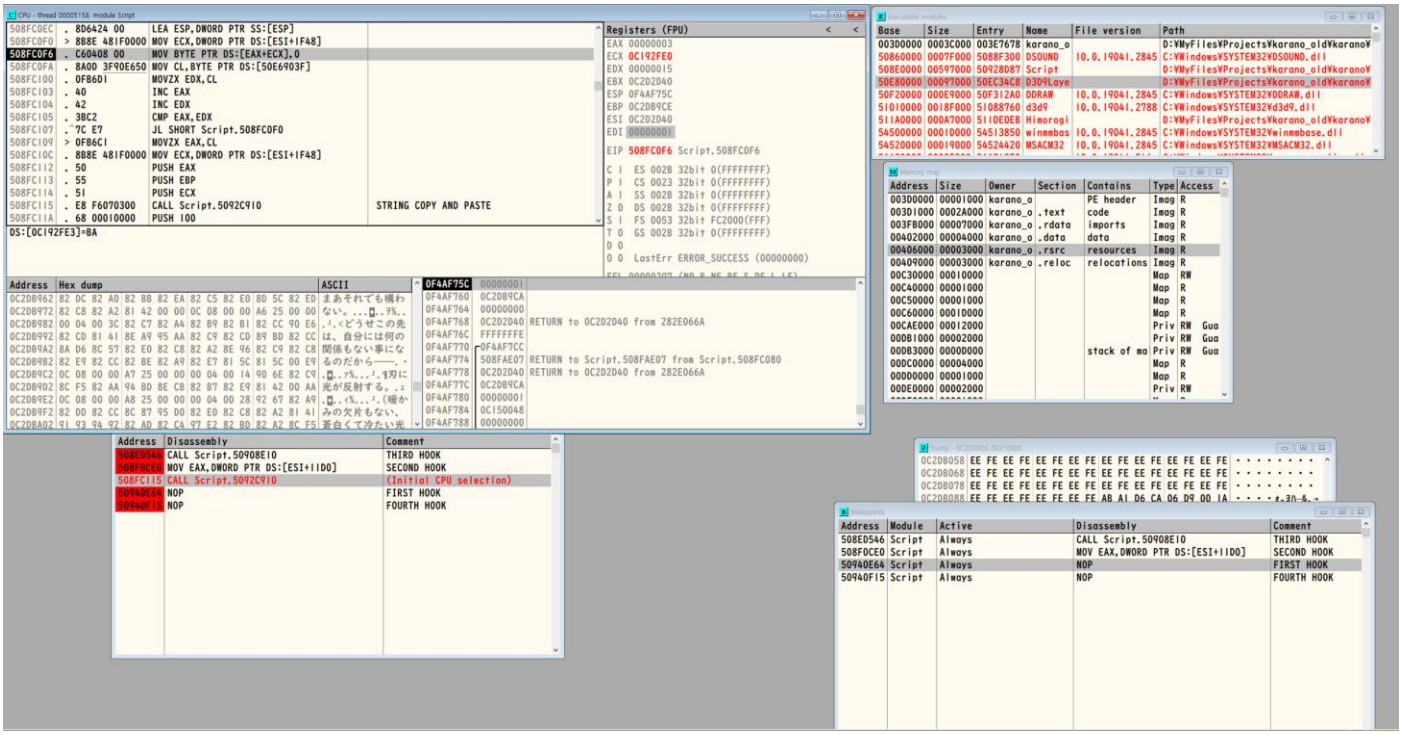
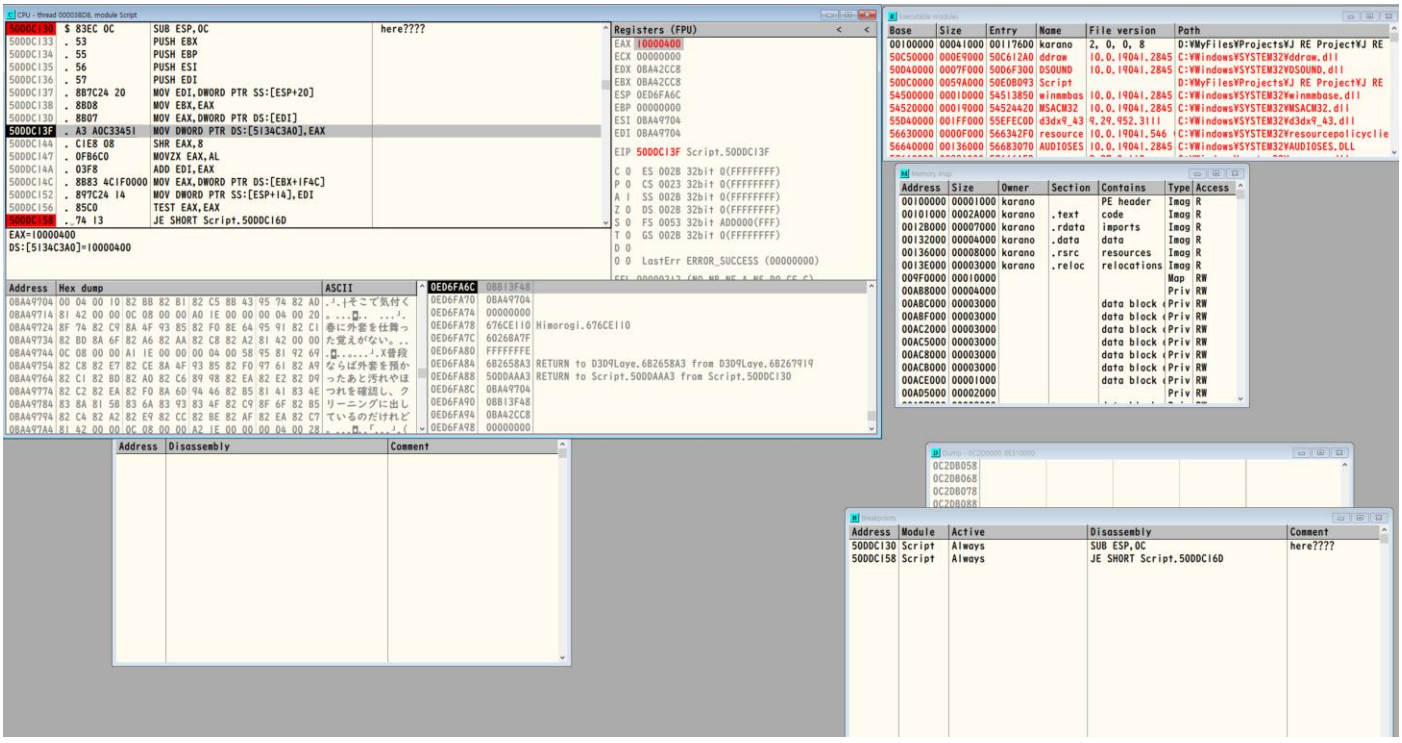
첫페이지는 메모…



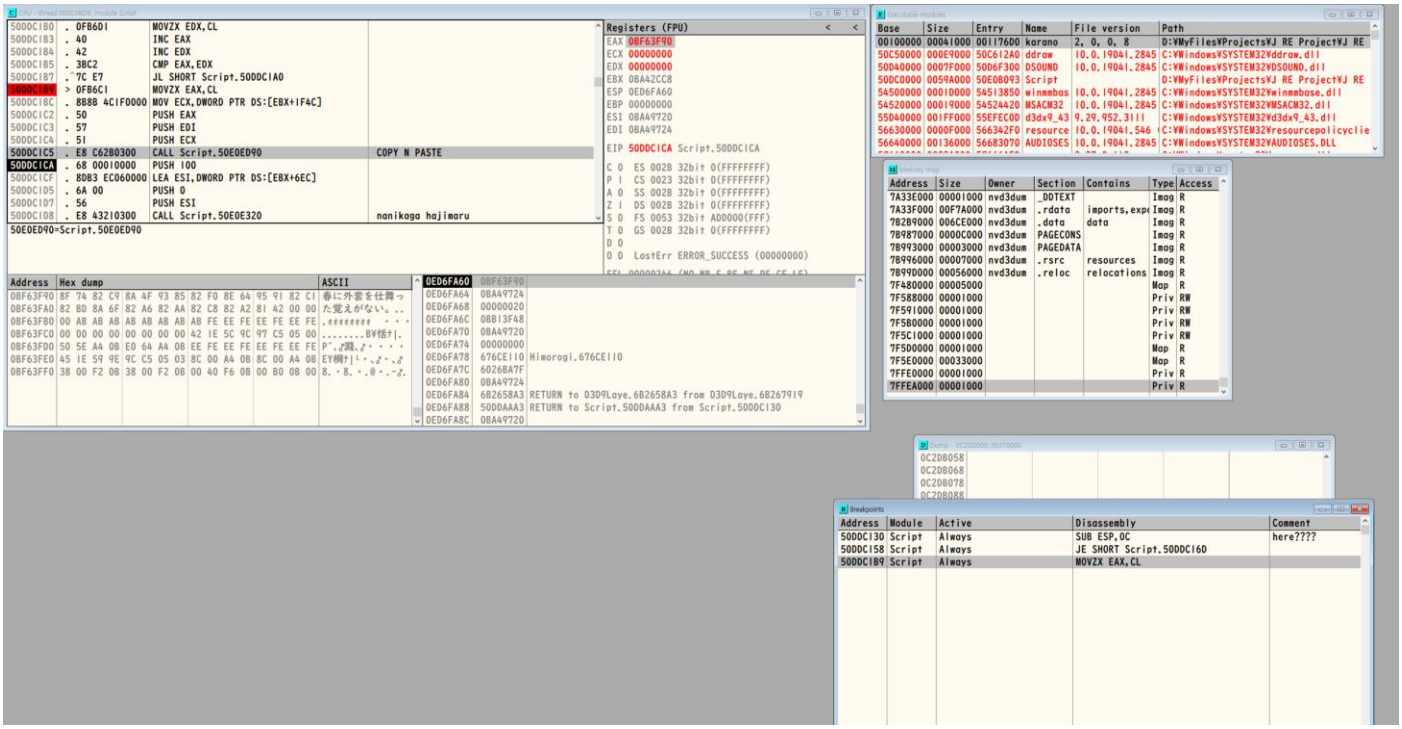
위 사진은 구버전..



EBX 붙여넣기하는곳

EDI 원본주소

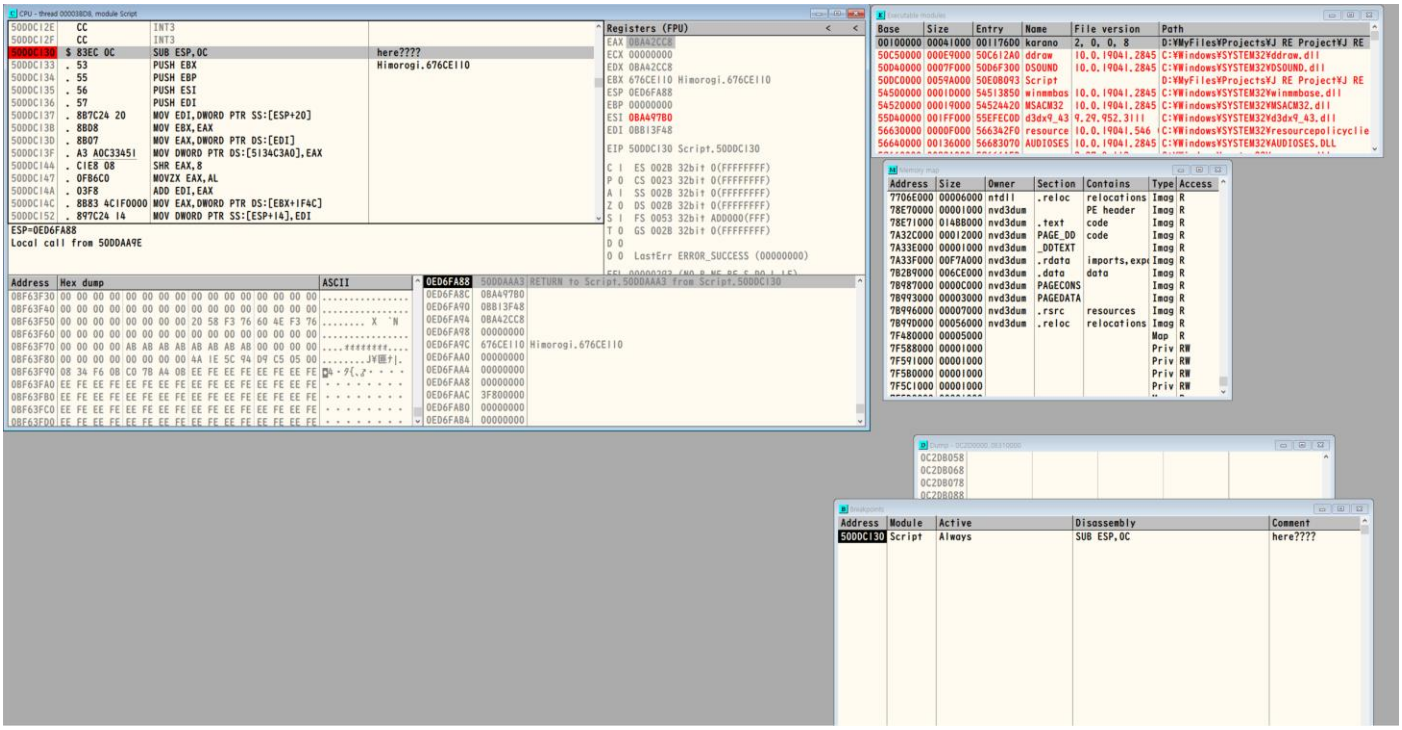
이렇게 테스트해보다가 갑자기 뭔가 떠올라서 기록 남기기 시작



드디어 문자열 copy and paste를 시작하는 최상위 지점을 찾은듯하다..

50DDC1C5 (0018C1C5) = 문자열 붙여넣는 스크립트 이동지점

이 위쪽으로 거슬러 올라가보면…



50DDC130 (0018C130) = 시작지점(?)

이곳이 시작점같다. 그렇다면 이곳 언저리에 후킹하면 되지 않을까 싶은데..

조금 테스트를 해보니 무언가 수상하다

이런식으로 스프라이트가 표시되는 화면에서 브포에 두번 걸린다.

좀더 테스트를 해보며 정확히 어느때 걸리는건지 확인해보자.

테스트 시작은 ??? なにがかしら 가 출력되어있는 상태인데 엄청 많이 봐왔으니 알겠지요?

테스트 시작 화면은 스프라이트 없고, 이름+대사 출력된 상태..

그리고 클릭하여 대사를 넘기면 위 상태가 되며 브포에 걸린다.

다시 F9 를 눌러 진행하면 바로 대사가 출력되며 상태는 Running 이 된다.

➔ 대사만 출력되는 상황에서는 브포 한번걸림.

그렇다면 스프라이트가 나오면 어찌될지 보기위해 계속 대사를 넘겨보자

아래는 위 상태에서 클릭을 한번하여 브포에 걸린상태,

이 아래는 왼쪽에서 F9 한번 누른 것

→

→ 또 F9 를 누르니 이름+대사가 출력됨 (상태는 Running)

➔ 아하! 스프라이트 출력시 한번, 이름+대사 출력시 한번이 확실하다.

→

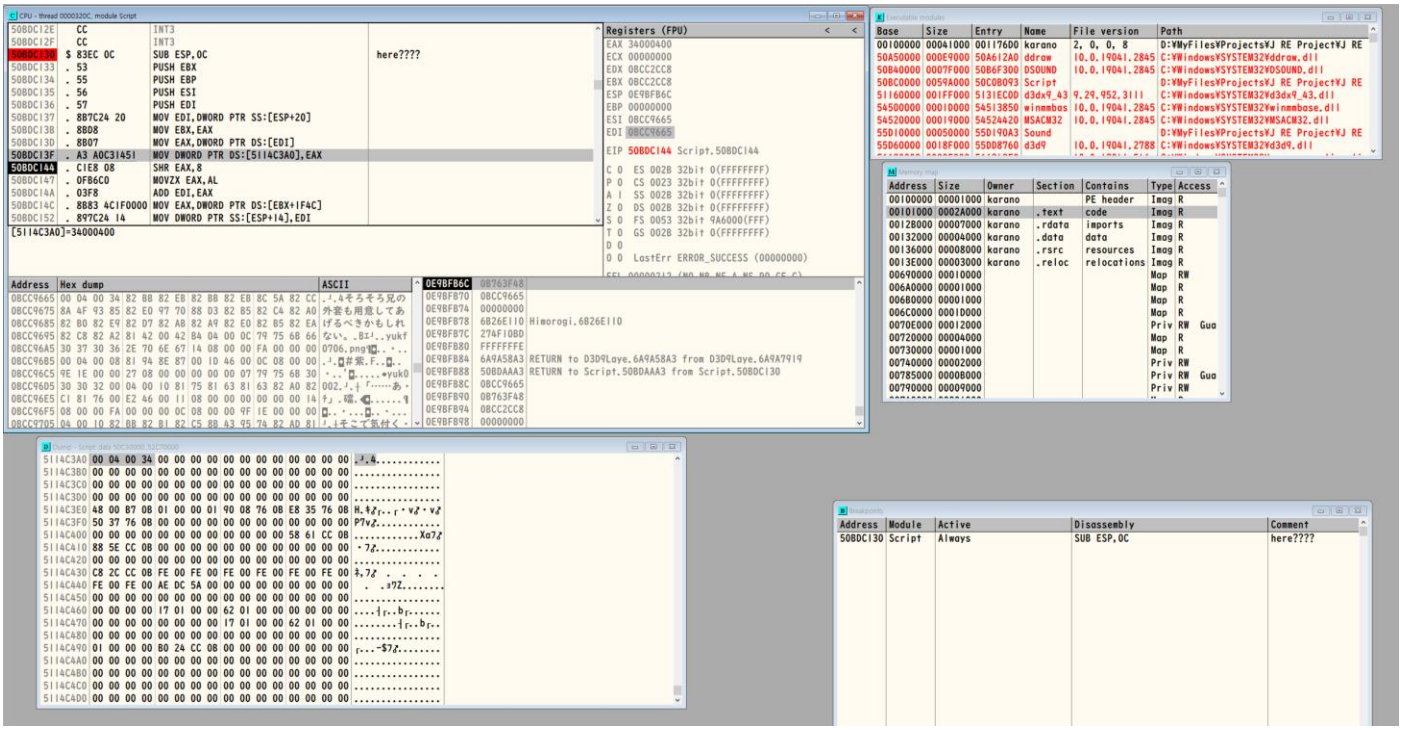

참고로 스프라이트 표시는 이렇게 커다랗게 튀어나올때도 한번 걸린다..(집중해서 테스트해보다가 갑자기 튀어나와서 깜짝놀람;)

올리씨를 잠시 쉬게 돌려보내고 아랄씨를 불러오자.

라고 생각하며 올리디버거를 종료했는데 아이고 어떤 레지스터를 납치해야하는지 안보고 껐네~

다시 키기도 귀찮고 이미 껐으니 무지성으로 이건가 싶은거 가져와봤는데 아무래도 이건 아닌거같다.

〈여담 1〉 - 원본데이터구조



올리씨를 켜서 레지스터를 확인하던더중, 코드에 [5114C3A0]이라는 의문의 절대주소가 눈에 띈다.

EAX 를 그대로 옮겨갖다놓는데.. 주시해보자.

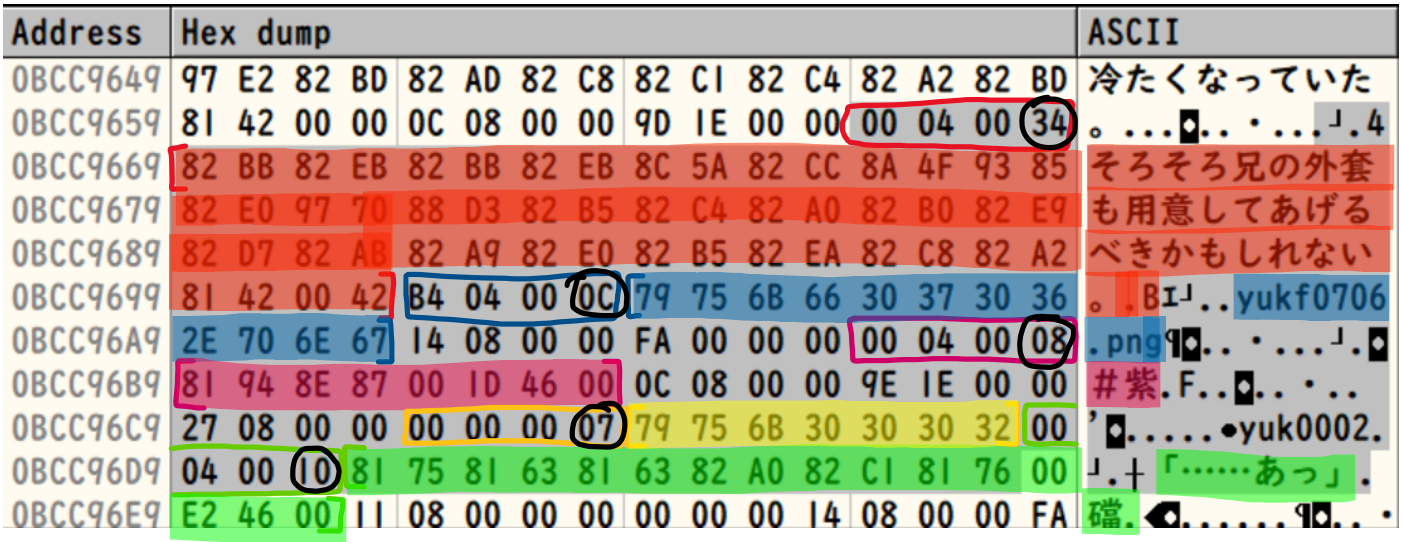
SHR EAX,8 MOVZX EAX,AL

3400 0400 -> 0034 0004 -> 0000 0004

그리고 EDI 에 EAX 를 더한다. 이때 EDI 는 PASTE 하는주소값인데, 그럼 1 WORD 만큼 EDI 가 오른것이다.

다시 정리해보면…

- EAX 는 절대주소[5114C3A0]에의해 3400 0400 으로 고정
- EAX 는 Shift Right 이후 AL 만 추출되어 0000 0004 로 변환
- 붙여넣기할 주소를 가리키는 EDI 에 EAX 가 더해져 다음 EDI 로 1WORD 를 추출해올수 있는 준비 완료

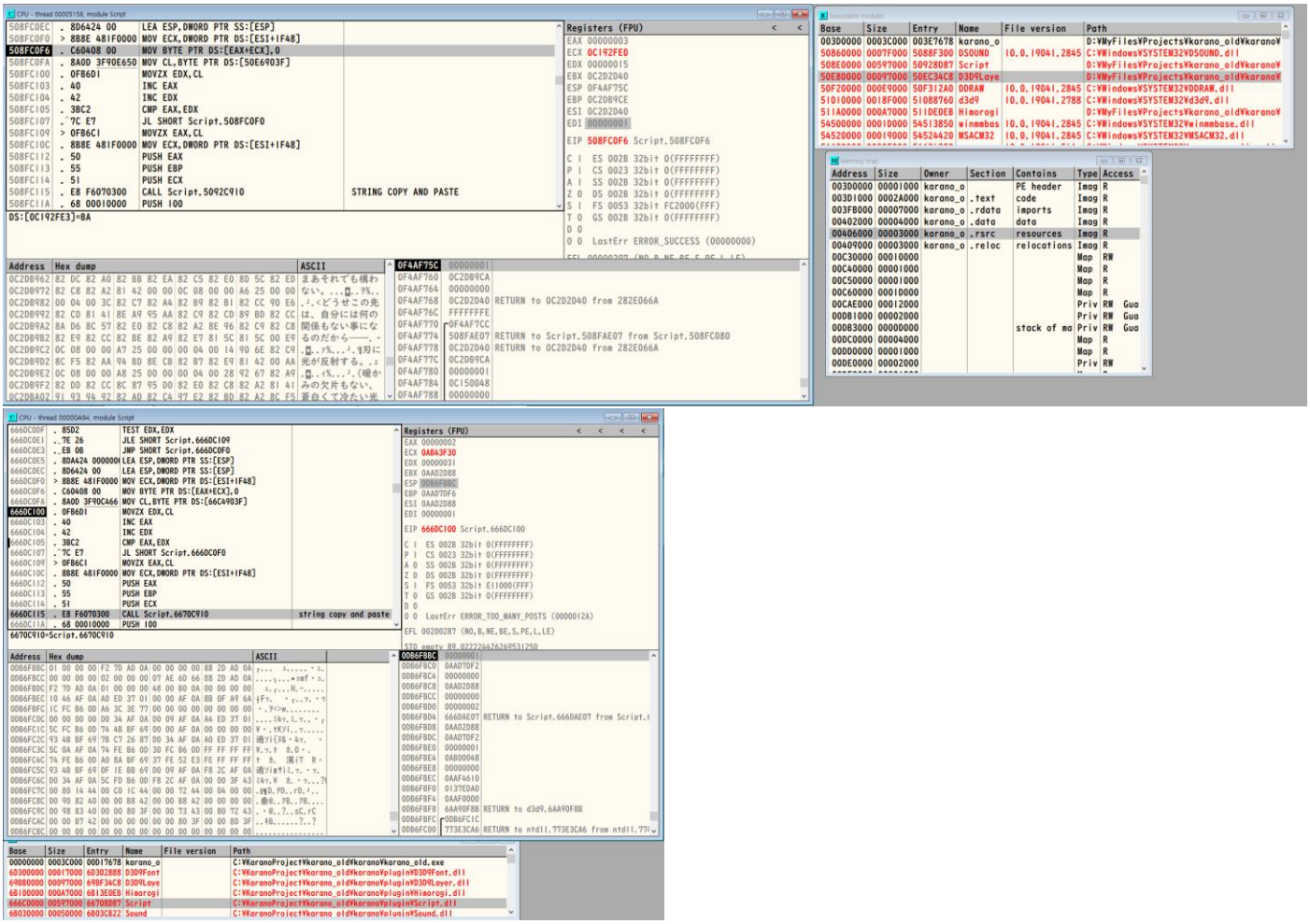


이제 이쪽을 한번 보자..

Hex dump 와 ASCII 를 보며 비교해보는거다.

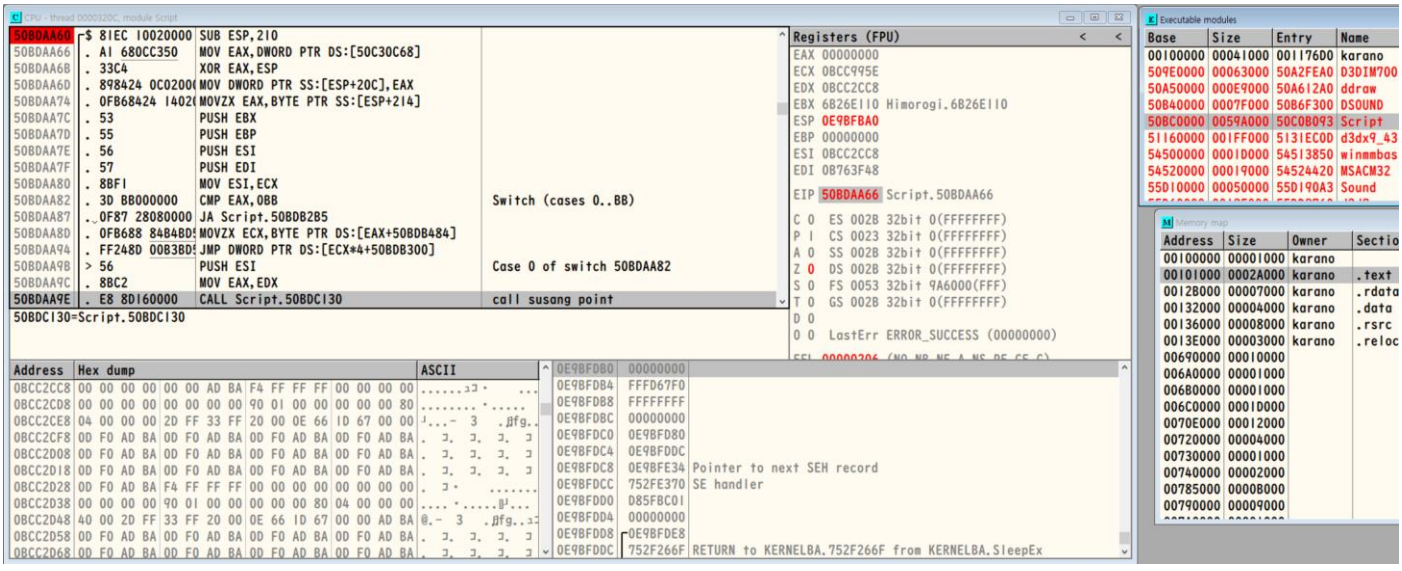
대사, 스프라이트파일명, 인물명 등등 직전에 각 문자열의 길이가 나와있다.

〈여담 2〉 - 구버전과의 비교



String copy and paste 콜 하는부분은 상대주소로 0004 C910 에 존재하나보다.

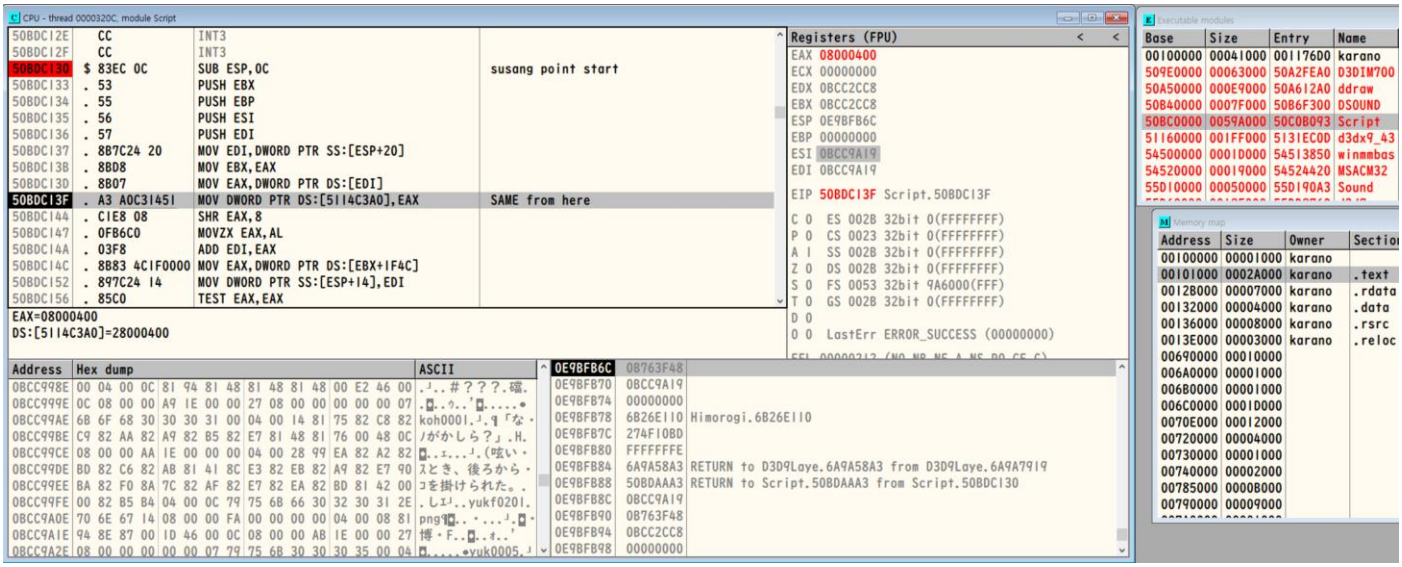
〈여담 3〉 - string copy and paste 이전 상위 코드를 찾아보자.



이곳이 더 상위지점. 이곳을 브포 걸면 현재 화면을 지울 때도 걸리고 다음화면을 그릴때도 걸린다. 다음 코드파인딩기록은 이것을 중점적으로 살펴보는것으로 시작할듯하다. 이쪽어딘가 후킹 지점을 잡으면 되는거 아닐까..?

저기 call susang point(수상한지점) 잡아놓은곳을 타고 가서 50BDC130(0001 C130)에 브포를 걸고 테스트하는건 이번 코드파인딩기록에서 테스트 해본것이다.

그럼 이제 다시한번 정리해보자.

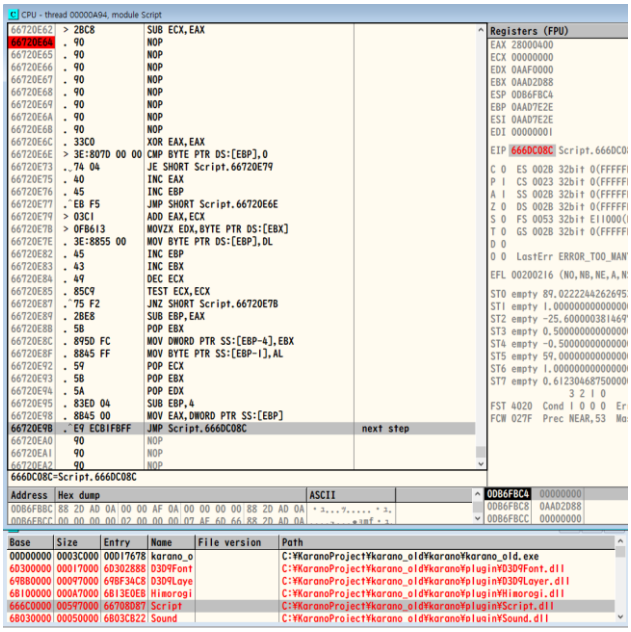


이렇게 타고와서 후킹하면 다음 스프라이트, (이름+)대사 출력할 때 한번씩 걸리는걸 이번 기록에서 확인할수 있었다.

구버전과 비교했을 때 SAME from here 지점부터 String copy and paste(아래쪽이라 사진에서 잘림)까지가 완전히 똑같음.

그렇다면 위 사진기준 50BDC130~50BDC13D 이 다른 부분이다.

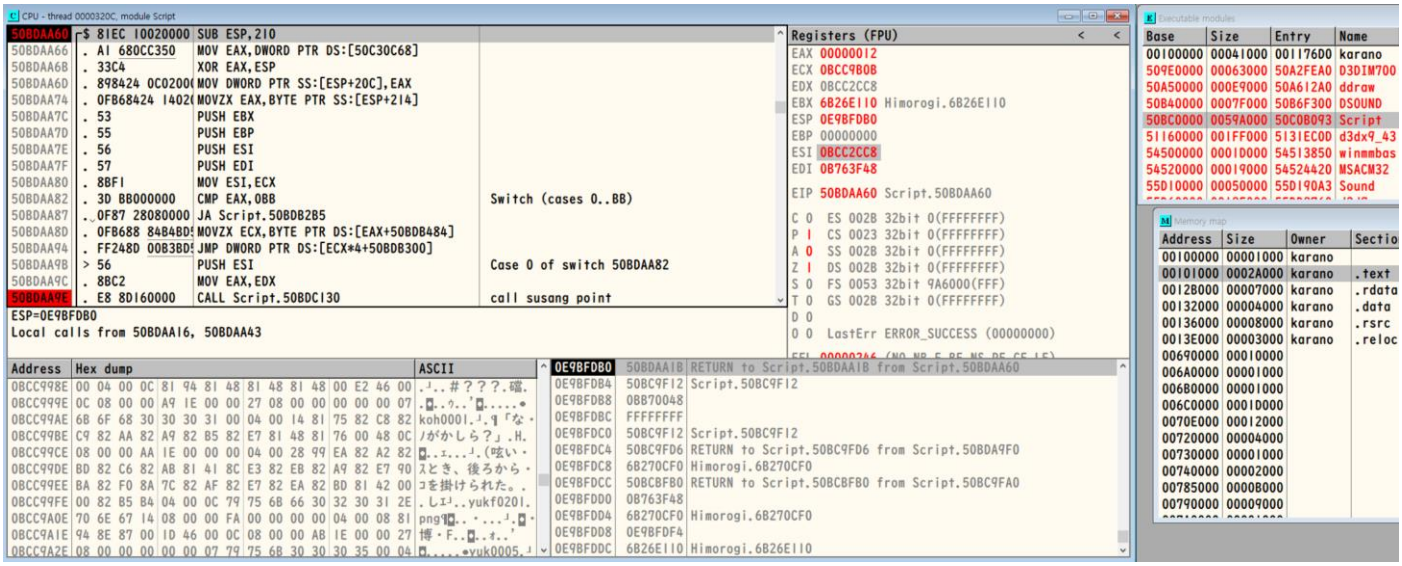
참고로 SAME from here 은 구버전에서 후킹지점으로부터 어떤 작업이 끝나고 점프해오면 도착하는지점이었음.



-> 구버전 (빨강게 브포걸어둔곳이 후킹지점, EBX 를 납치하는곳..)

그렇다면 추가된 50BDC130~50BDC13D 코드를 좀 자세히 봐야겠다..

〈여담 4〉



이렇게 브포를 걸면 다음 대사로 넘어갈 때 스프라이트와 대사를 지우는 동작을 하며 걸린다.

참고로 지울때는 위에서 찾은 수상한 지점이 브레이크에 걸리지 않고 화면의 스프라이트와 대사가 지워지고,

새로운 스프라이트와 대사를 출력할 때 그제야 비로소 수상한 지점의 콜이 호출되어 브레이크가 걸리게 된다.

그렇다면 확실한건 위의 두 브포 사이는 화면 지우기를 담당하고, 화면 출력은 아래쪽의 콜을 타고 간 이후에 작동한다는것이다!

다음 코드 파인딩 기록은 이곳을 중점적으로 바라보게 될 듯하다.