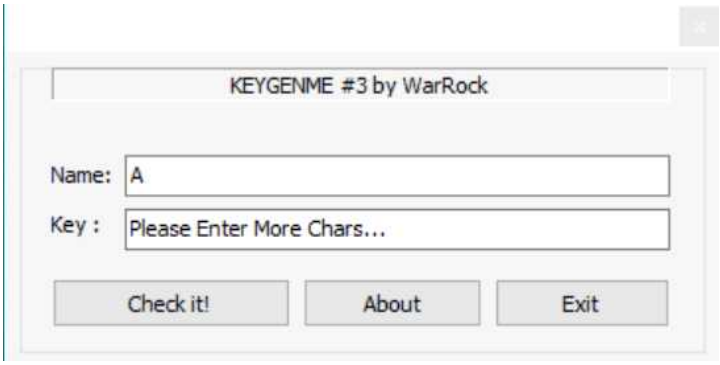
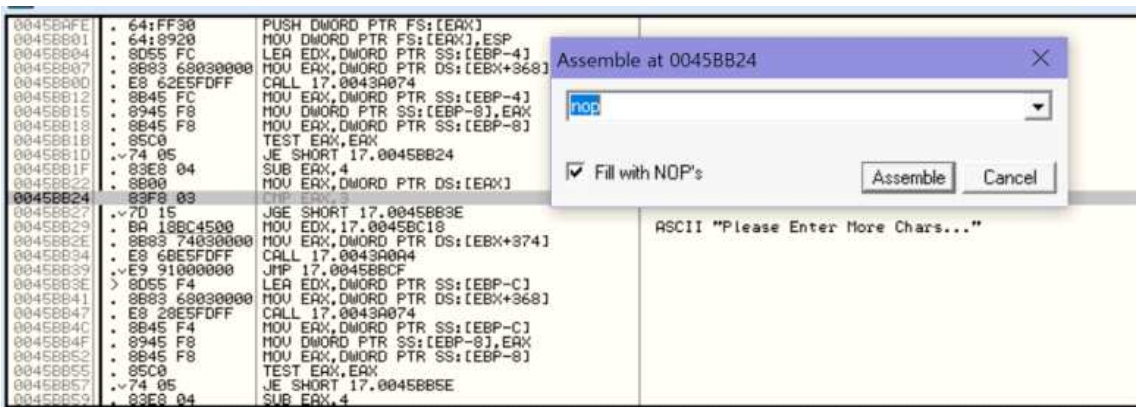
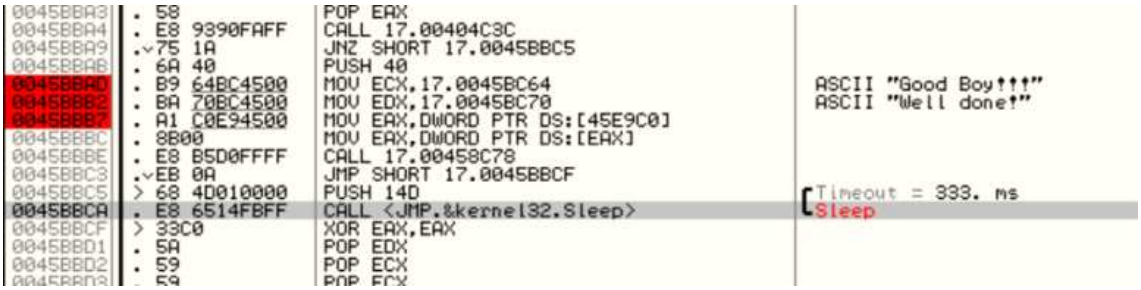
CodeEngn Basic RCE L17
: Key 값이 BEDA-2F56-BC4F4368-8A71-870B 일때 Name은 무엇인가
힌트 : Name은 한자리인데.. 알파벳일수도 있고 숫자일수도 있고..
정답인증은 Name의 MD5 해쉬값(대문자)



name은 한자리라고 해서 아무거나 입력하고 눌렀는데 더 입력하라고 나온다
이 부분은 올리디버거를 열어 수정해주도록 해야겠다 !

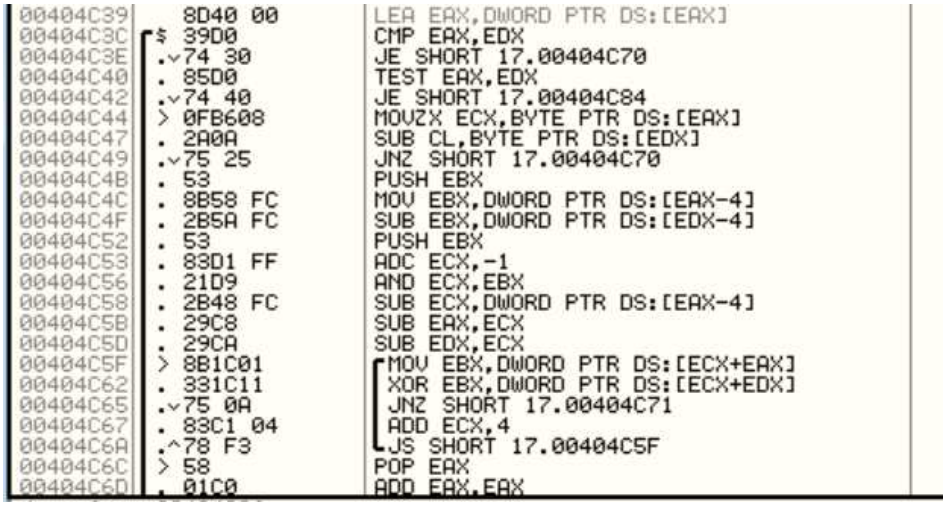


text string 찾기로 해당 메세지를 찾아서 이동했고, cmp eax,3으로 보아 3글자가 아니면 더 입력하라고 하는 것 같다. 이 부분을 nop으로 바꾸어준다.



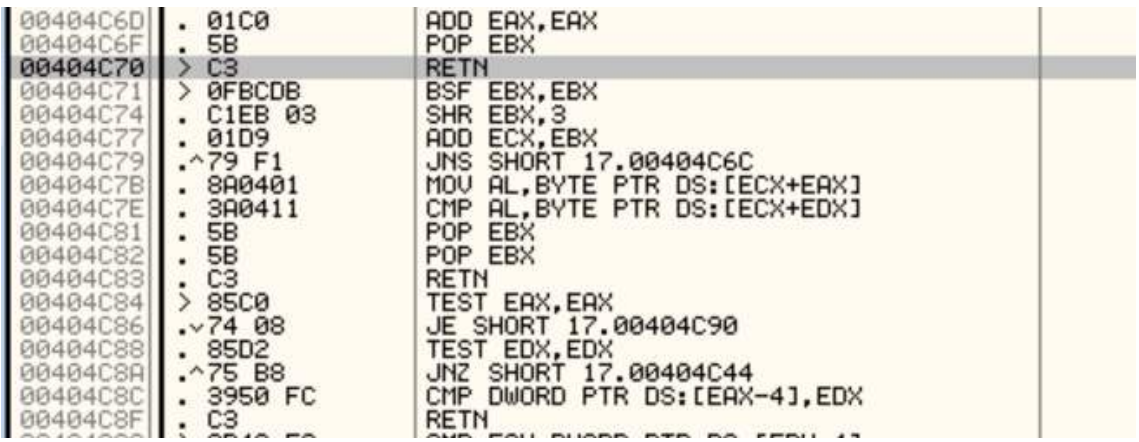
조금만 밑으로 내려가다 보면 성공메세지가 출력되는 부분을 볼 수 있다. 코드를 살펴보면 위에 JNZ 0045BBC5가 있는데 이 주소는 Timeout으로 프로그램이 종료되도록 하는 것 같다.

즉 JNZ을 그냥 통과해야 성공메세지가 출력이 되는 분기점이기 때문에 CALL 00404C3C 부분으로 이동해서 살펴보아야 겠다.

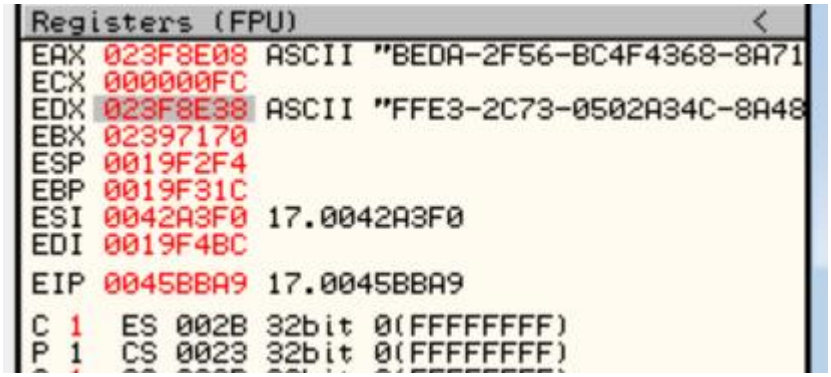


CMP EAX, EDX
JE 00404C70
= EAX - EDX를 한 값이 0이면, 00404C70으로 JUMP하라

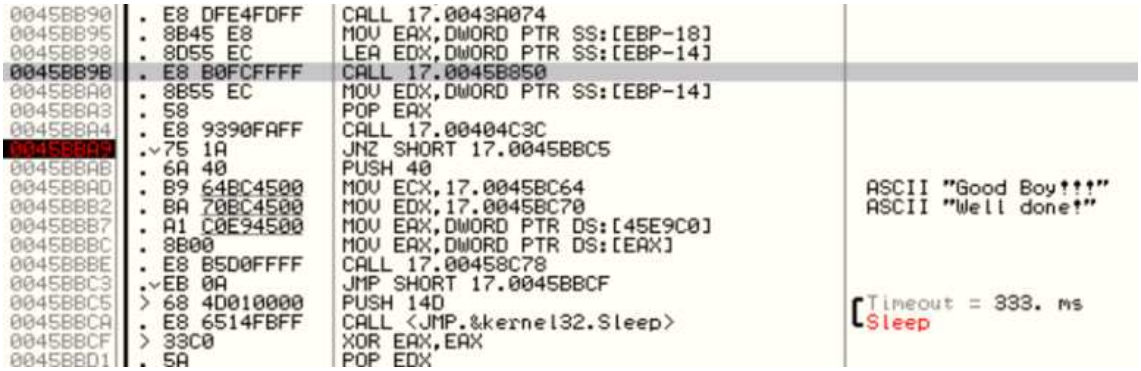
그럼 EAX와 EDX의 값이 같다고 가정하고 00404C70으로 가보겠다.



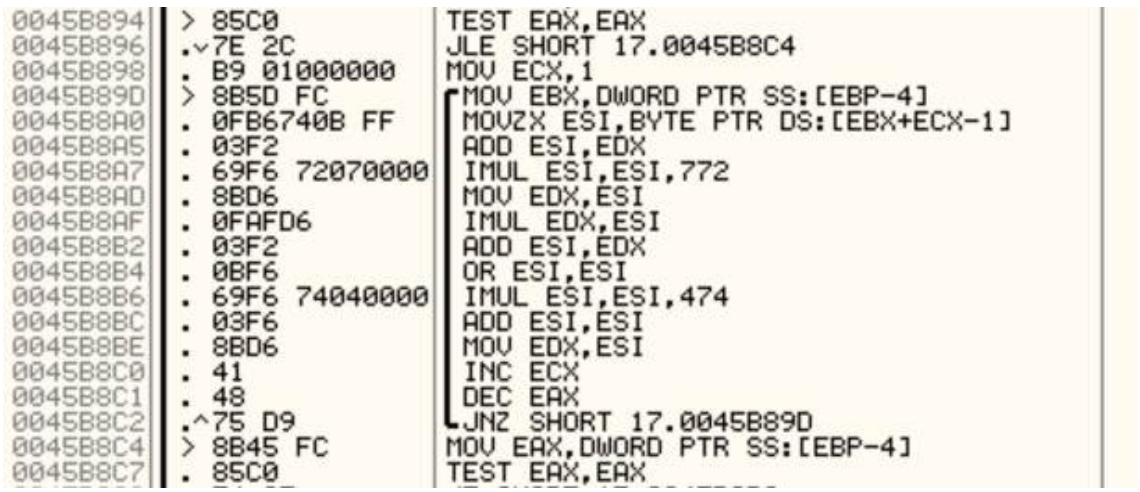
바로 RETN으로 0을 리턴한다. 즉 CALL 00404C3C는 EAX와 EDX를 비교해 같으면 0을 리턴하는 것으로 볼 수 있다.



JNZ에 BP를 걸어주고 실행시켜 NAME에는 'A', KEY에 BEDA-2F56-BC4F4368-8A71-870B를 눌러 CHECK IT을 누르면 EAX에 내가 입력한 시리얼 넘버가 들어간다. 아마도 이름이 A일때의 KEY는 FFE3-2C73-052A34C-8A48인 것을 추측할 수 있다.



그럼 CALL 00404C3C는 사용자의 입력값과 시리얼값을 비교하는 함수를 호출하고, 바로 그 위에 호출되는 함수가 NAME에 대한 시리얼 값을 생성하는 함수라고 추측할 수 있으므로 0045B850으로 이동해서 살펴보겠다.



0045B89D ~ 0045B8C2를 살펴보면

```
MOV EBX, DWORD PTR SS:[EBP-4]
MOVZX ESI, BYTE PTR DS:[EBX+ECX-1]
ADD ESI, EDX // ESI = ESI + EDX
IMUL ESI, ESI, 772 // ESI = ESI * 772
MOV EDX, ESI // EDX = ESI
IMUL EDX, ESI // EDX = EDX * ESI
ADD ESI, EDX // ESI = ESI + EDX
OR ESI, ESI // ESI = ESI
IMUL ESI, ESI, 474 // ESI = ESI * 474
ADD ESI, ESI // ESI = ESI + ESI
MOV EDX, ESI // EDX = ESI
INC ECX // ECI + 1
DEC EAX //  EAX - 1
JNZ 0045B89D
```

JNZ로 다시 맨 위로 올라가 루프를 돌며 생성하는 것 같았다 !
c++로 작성해 보고, 돌려보았다.

```
#include <stdio.h>

int main()
{
	int ESI, EDX = 0;

	for (int i = 0x30; i <= 0x7A; i++) { // 0에서 z 까지 ( ASCII 코드로 0x30 = 0, 0x7A = z )
		ESI += EDX; // ADD ESI, EDX
		ESI = i * 0x772; //IMUL ESI, ESI, 0x772
		EDX = ESI; //MOV EDX, ESI
		EDX *= ESI; //IMUL EDX, ESI
		ESI += EDX; //ADD ESI, EDX
		ESI = ESI; // OR ESI, ESI
		ESI *= 0x474; //IMUL ESI, ESI, 0x474
		ESI += ESI; //ADD ESI, ESI
		EDX = ESI; // MOV EDX, ESI

		printf("NAME = %c , PW = %X\n", i, EDX);
	}
	return 0;
}
```

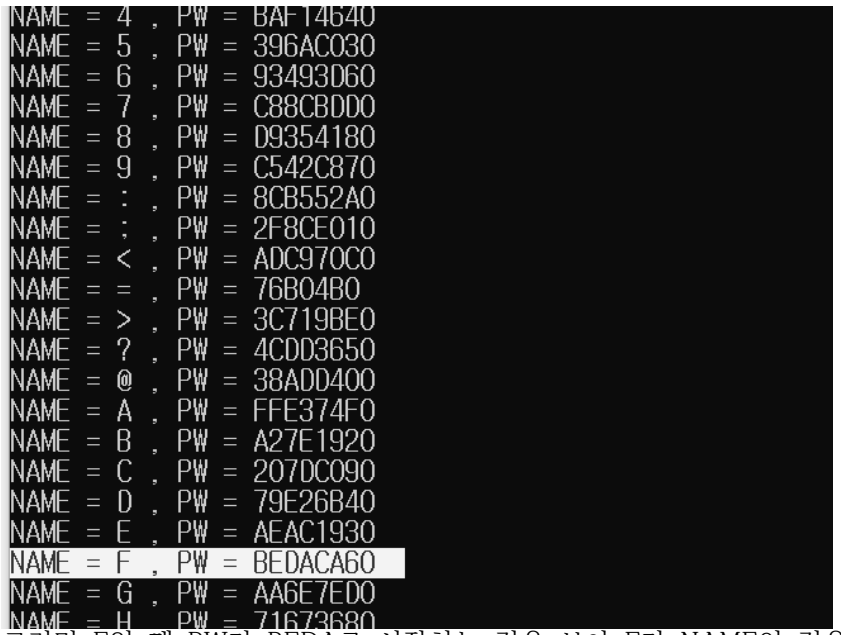
+

IMUL : 인자가 2개일 경우, 첫번째 인자와 두번째 인자를 곱한 결과를 첫번째 인자에 저장한다. 인자가 3개일 경우, 두번째 인자와 세번째 인자를 곱한 결과를 첫번째 인자에 저장한다.

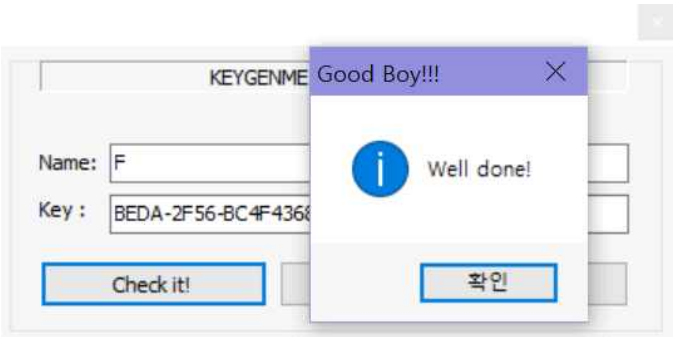
MOV DEST, SRC : SRC에 있는 값을 DEST에 저장

INC : 피연산자 + 1

DEC : 피연산자 - 1



그러면 F일 때 PW가 BEDA로 시작하는 것을 보아 F가 NAME인 것을 알 수 있다.



성공메세지가 출력된다