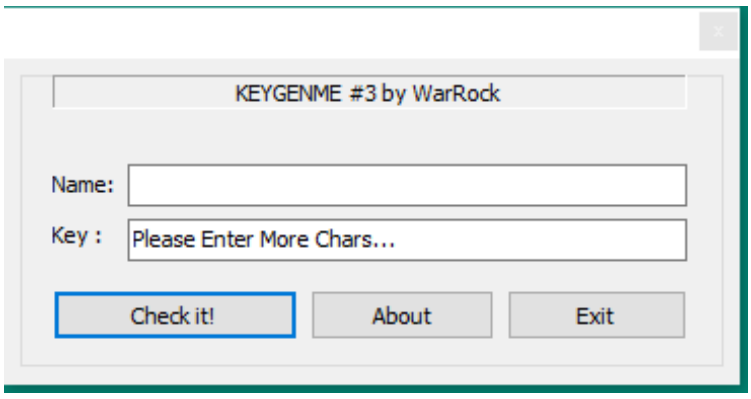
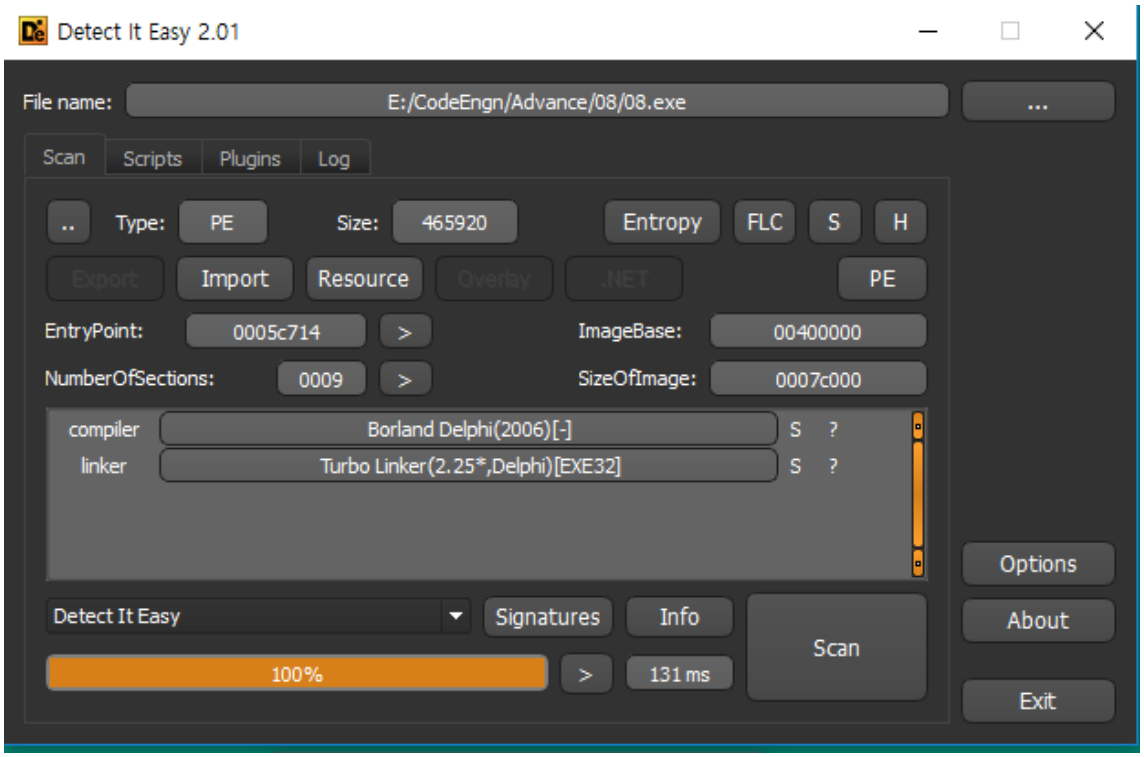
CodeEngn Advance 8 Tree T0 Tree

Name을 알아보자



그냥 Check it을 누르니
더많은 문자열을 입력하라고 나온다.

패킹이따로 되어있지 않다.



Check it 했을 때 나오는 문자열 Pleas Enter More Chars..로 본문을 찾아냄

```
0045BB27   7D 15          jge 08.45BB3E
0045BB29   BA 18BC4500    mov edx,08.45BC18                    45BC18:"Please Enter More Chars..."
0045BB2E   8B83 74030000  mov eax,dword ptr ds:[ebx+374]
0045BB34   E8 68E5FDFF    call 08.43A0A4
0045BB39   E9 91000000    jmp 08.45BBCF
0045BB3E   8D55 F4        lea edx,dword ptr ss:[ebp-C]
0045BB41   8B83 68030000  mov eax,dword ptr ds:[ebx+368]
0045BB47   E8 28E5FDFF    call 08.43A074
0045BB4C   8B45 F4        mov eax,dword ptr ss:[ebp-C]
0045BB4F   8945 F8        mov dword ptr ss:[ebp-8],eax
0045BB52   8B45 F8        mov eax,dword ptr ss:[ebp-8]
0045BB55   85C0           test eax,eax
0045BB57   74 05          je 08.45BB5E
0045BB59   83E8 04        sub eax,4
0045BB5C   8B00           mov eax,dword ptr ds:[eax]
0045BB5E   83F8 1E        cmp eax,1E
0045BB61   7E 12          jle 08.45BB75
0045BB63   BA 3CBC4500    mov edx,08.45BC3C                    45BC3C:"Please Enter Not More Then 30 Chars..."
0045BB68   8B83 74030000  mov eax,dword ptr ds:[ebx+374]
0045BB6E   E8 31E5FDFF    call 08.43A0A4
0045BB73   EB 5A          jmp 08.45BBCF
0045BB75   8D55 F0        lea edx,dword ptr ss:[ebp-10]
0045BB78   8B83 74030000  mov eax,dword ptr ds:[ebx+374]
0045BB7E   E8 F1E4FDFF    call 08.43A074
0045BB83   8B45 F0        mov eax,dword ptr ss:[ebp-10]
0045BB86   50             push eax
0045BB87   8D55 E8        lea edx,dword ptr ss:[ebp-18]
0045BB8A   8B83 68030000  mov eax,dword ptr ds:[ebx+368]
0045BB90   E8 DFE4FDFF    call 08.43A074
0045BB95   8B45 E8        mov eax,dword ptr ss:[ebp-18]
0045BB98   8D55 EC        lea edx,dword ptr ss:[ebp-14]
0045BB9B   E8 B0FCFFFF    call 08.45B850
0045BBA0   8B55 EC        mov edx,dword ptr ss:[ebp-14]
0045BBA3   58             pop eax
0045BBA4   E8 9390FAFF    call 08.404C3C
0045BBA9   75 1A          jne 08.45BBC5
0045BBAB   6A 40          push 40
0045BBAD   B9 64BC4500    mov ecx,08.45BC64                    45BC64:"Good Boy!!!"
0045BBB2   BA 70BC4500    mov edx,08.45BC70                    45BC70:"Well done!"
0045BBB7   A1 C0E94500    mov eax,dword ptr ds:[45E9C0]
0045BBBC   8B00           mov eax,dword ptr ds:[eax]
0045BBBE   E8 85D0FFFF    call 08.458C78
0045BBC3   EB 0A          jmp 08.45BBCF
```

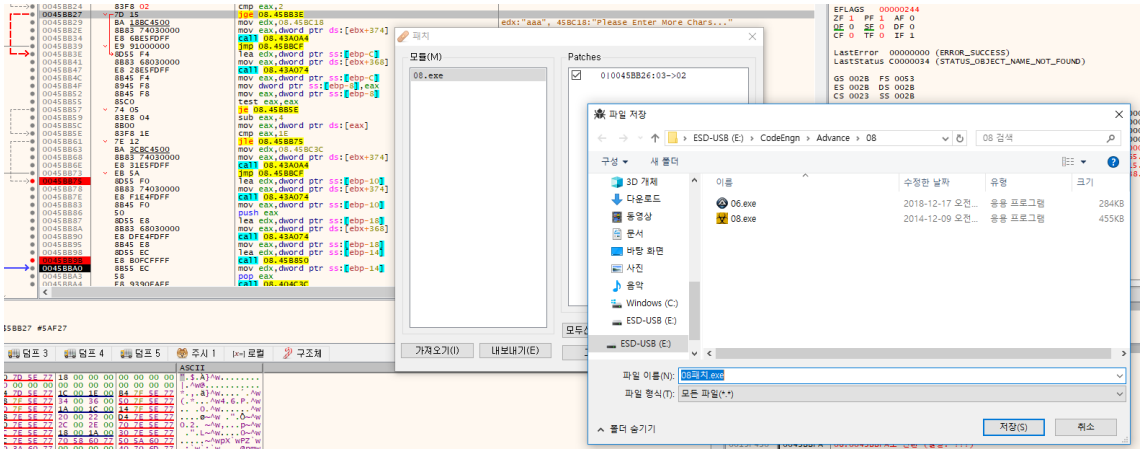
name에 aaa를 준뒤 트레이싱 해보니
비트로 엄청 장난친다.

```
mov ebx,dword ptr ss:[ebp-4]                    [ebp-4]:"aaa"
movzx esi,byte ptr ds:[ebx+ecx-1]
add esi,edx
imul esi,esi,772
mov edx,esi
imul edx,esi
add esi,edx
or esi,esi
imul esi,esi,474
add esi,esi
mov edx,esi
inc ecx
dec eax
jne 08.45B89D
```

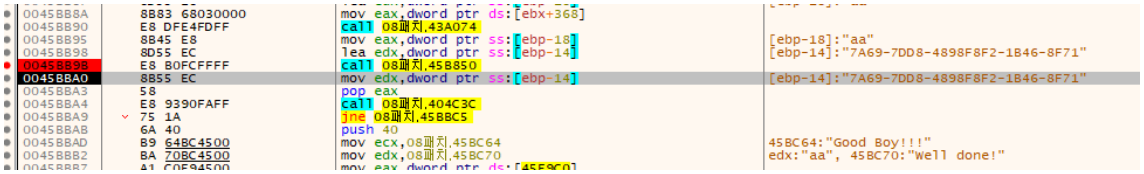
우선 위에 어셈블리어를 분석하여 알고리즘으로 만들었다.

1. 첫글짜와 772 곱
2. 곱한거 ^2
3. 1번 + 2번
4. 3번 * 474
5. 4번 *2

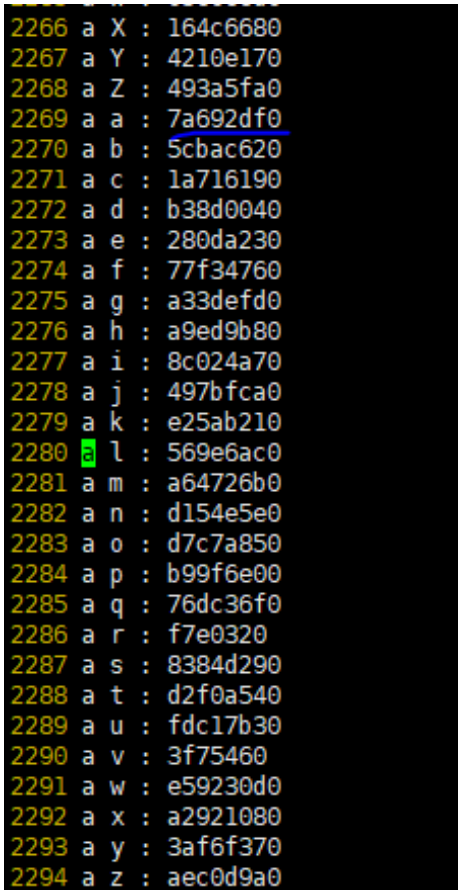
Name에 문자의길이는 최소 3인데 2글자도 받을 수 있도록 패치



aa를 넣어서 시리얼 값을 보고



아까만든 프로그램에서 aa를 넣었을 때 나오는 값을보니
0x7a692df0 으로
상위 16비트가 key 값으로 오는 것을 알 수 있었다.



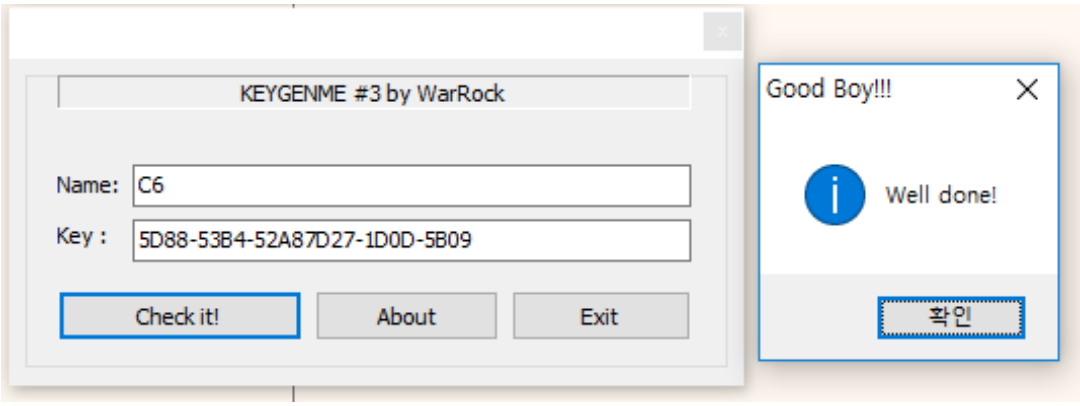
결국 key값 앞부분 5d88을 이용하여 이름을 찾으면 됨 아까 만든 코드를 수정하여
5d88 저격

```c
#include<stdio.h>

int main(){
    unsigned int c[62],edx,esi,temp=0;
    int i,j;
    edx=esi=0;
    for(i=0;i<10;i++) c[i]=i+48;
    for(i=0;i<26;i++) {
        c[i+10]=i+65;
        c[i+36]=i+97;
    }
    for(i=0;i<62;i++){
        edx=0;
        esi=c[i];
        esi+=edx;
        esi*=0x772;
        edx=esi;
        edx*=esi;
        esi+=edx;
        esi*=0x474;
        esi+=esi;
        edx=esi;
        temp=edx;
        for(j=0;j<62;j++){
            edx=temp;
            esi=c[j];
            esi+=edx;
            esi*=0x772;
            edx=esi;
            edx*=esi;
            esi+=edx;
            esi*=0x474;
            esi+=esi;
            edx=esi;
            if((esi>>16)==0x5d88)printf("%c %c : %.x\n",c[i], c[j],esi>>16);
        }
"advance8.c" 38L, 556C
```

```
[kwl3434@linux ~]$ gcc advance8.c
[kwl3434@linux ~]$ ./a.out
C 6 : 5d88
```

C6이라고 나옴



해시화만 하면
http://www.convertstring.com/ko/Hash/MD5

7E8B9F5CAB4A8FE24FAD9FE4B7452702
Clear