```
shellcode1.cpp & X
shellcode1
                                                                    (Global Scope)
      E#include "stdafx.h"
                                                                                                                    byte ptr [ebp-0Dh],65h
byte ptr [ebp-0Ch],70h
byte ptr [ebp-08h],61h
                                                                                             0041179B mov
       #include "windows.h"
                                                                                             0041179F mov
                                                                                             004117A3 mov
                                                                                                                    byte ptr [ebp-0Ah],64h
byte ptr [ebp-9],0
                                                                                             904117A7 mov
      ∃int main(int argo, char* argv[])
                                                                                             004117AB
       {
                                                                                                WinExec(buf, SW_SHOW);
                                                                                           ) 004117AF mov
004117B1 push
                                                                                                                     esi,esp
             char buf[8] = { 'n', 'o', 't', 'e', 'p', 'a', 'd', '\x0' };
            WinExec(buf, SW_SHOW);
                                                                                             004117B3 lea
                                                                                                                     eax,[buf]
                                                                                            00411786 push
00411787 call
                                                                                                                     eax
             ExitProcess(1);
                                                                                                                     dword ptr [__imp__WinExec@8 (041B
                                                                                            0041178D cmp
0041178F call
                                                                                                                    __RTC_CheckEsp (0411235h)
                                                                                            ExitProcess(1);
004117C4 mov
004117C6 nush
                                                                                                                     esi,esp
```

다음과 같이 winexec() 함수를 이용해 메모장을 출력해주는 코드를 짠 후, 비주얼 스튜디오의 디버깅 기능을 이용해 디스어셈블리를 통해 어셈블리어를 추출한다.

그 후 디버깅을 통해 구한 exe 실행 파일을 이뮤니티 디버거를 통해 winexec와 exitptocess() 함수의 실제 주소를 구해 어셈블리어에 넣는다.

```
shellcode2

    (Global Scope)

                                                                                                                           70 mov
                                                                                                             C6 45 FC 70
                                                                                                                                                                  byte ptr [ebp-4],70h
                                                                                                                         mov byte ptr [ebp-3],61h
       ⊡int main(int argc, char* argv[])
                                                                                                                                                                  byte ptr [ebp-3],61h
               asm {

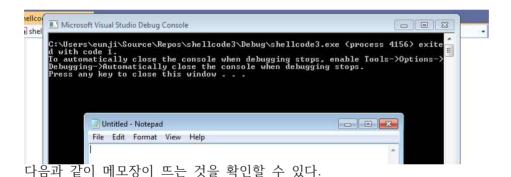
// char buf[8] = { 'n','o','t','e','p','a','d','\x0' };

mov byte ptr [ebp-3],6Fh
mov byte ptr [ebp-6],74h
mov byte ptr [ebp-6],74h
mov byte ptr [ebp-4],76h
mov byte ptr [ebp-4],76h
mov byte ptr [ebp-3],61h
mov byte ptr [ebp-3],64h
mov byte ptr [ebp-1],64
mov byte ptr [ebp-1],6
// xor ebx, ebx
// mov[ebp - 1], ebx
                                                                                                                          mov byte ptr [ebp-2],64h
                                                                                                             C6 45 FE 64
                                                                                                                                                                  byte ptr [ebp-2],64h
                                                                                                                          mov byte ptr [ebp-1],0
                                                                                                             C6 45 FF 00
                                                                                                                                                                 byte ptr [ebp-1],0
                                                                                                                          // xor ebx, ebx
                                                                                                                          //WinExec(buf, SW_SHOW);
                                                                                                                          push 5
                                                                                                         O 6A 05
                                                                                                                                               push
                                                                                                                          lea eax, [ebp - 8]
                   //WinExec(buf, SW_SHOW);
                                                                                                                                                                  eax,[ebp-8]
                                                                                                             8D 45 F8
                  //WinExec(but, SW_SHO
push 5
lea eax, [ebp - 8]
push eax
mov eax, 0x7712F57E
call eax
                                                                                                                         push
     ı
                                                                                                                                               nush
                                                                                                                                                                  eax
                                                                                                                          mov eax, 0x7712F57E
                                                                                                                                                                  eax,7712F57Eh
                                                                                                                                              mov
                   //ExitProcess(1);
                                                                                                                          call eax
                   push 1
mov eax, 0x770FBED2
call eax
                                                                                                                                              call
                                                                                                                                                                  eax
                                                                                                                         //ExitProcess(1);
```

위에서 구한 어셈블리어를 브레이크 포인트를 걸고 디버깅 해서 기계어 코드를 얻는다.

```
Fl#include "stdafx.h"
 #include "windows.h"
  char shellcode[] = "\xC6\x45\xF8\x6E"
"\xC6\x45\xF9\x6F"
  "\xC6\x45\xFA\x74"
  "\xC6\x45\xF8\x65"
  "\xC6\x45\xFC\x70"
  "\xC6\x45\xFD\x61"
  "\xC6\x45\xFE\x64"
 "\xC6\x45\xFF\x00"
⊡//"\x33\xDB"
 //"\x89\x5D\xFF"
  "\x6A\x05"
 "\x8D\x45\xF8"
 "\x50"
 "\x88\x7E\xF5\x12\x77"
 "\xFF\xD0"
 "\x6A\x01"
 "\xB8\xD2\xBE\x0F\x77"
 "\xFF\xD0";
⊟int main(int argc, char* argv[])
 {
      int* shell = (int*)shellcode;
      __asm {
         jmp shell
     };
```

다음과 같이 기계어 코드를 \x로 16진수인 것을 표현하여 코드를 짜준 후 실행하면



shellcode에 주석처리된 것은 널바이트 처리를 한 부분이다. 주석 위 한줄을 지우고 주석들을 해제해도 동일하게 실행되는 것을 확인할 수 있다.

