

Forum:[Hack Forums](#)

Thread: [Access violation - code c0000005 00000000?When run simple inline ASM in windows](#)

mr.cracker - October 15, 2014, 7:22 pm

i was running a simple program inside windows 8.1

Code //cl /EHsc ConnectBack.cpp

```
#include <windows.h>
```

```
#include <stdio.h>
```

```
DWORD __cdecl connect_back()  
{
```

```
    __asm{  
        pushad
```

```
        connectback:  
            jmp startup_bnc
```

```
find:  
    cld  
    mov ebx, 0x00000002  
    mov eax, ebx  
    ret
```

```
find_function:  
    pushad
```

```
find_function_loop:
```

```
compute_hash:  
    xor edi, edi  
    xor eax, eax  
    cld
```

```
compute_hash_again:
```

```
mov ebp, 0x08  
compute_hash_finished:  
find_function_compare:  
    mov eax, 0x00000001  
    mov [esp + 0x1c], eax
```

```
find_function_finished:  
    popad  
    ret
```

```

startup_bnc:
    call find
    mov edx, eax
    mov eax, 0xec0e4e8e
    push eax
    push edx
    call find_function
    add esp, 0x08

```

```

finished:
    mov [esp + 0x1c], eax
    popad
    ret

```

```

}
}

```

```

int main()
{
    DWORD addr = connect_back();
    printf("EAX Now: 0x%08x\n ", addr);

```

```

    return 0;
}error from windbg

```

Code (e18.11e8): Access violation - code c0000005 (!!! second chance !!!)

00000000 ?? ???

0:000:x86> k

ChildEBP RetAddr

WARNING: Frame IP not in any known module. Following frames may be wrong.

0101feac 00000000 0x0how can i return without any error ? i used 'pushad' in the beginning because i change 'ebp, ebx' which should be preserved in C function. Am i right ? But i don't know how to return 'eax' perfectly? How does my EIP changes ?

mr.cracker - October 16, 2014, 8:39 pm

i got answer..when i call "connect_back" function my system does the following,

Code push ebp

mov ebp, esp

push ebx

push esi

push ediso when i simply call "ret" , the return address is actually after 16 bytes long from the current value. i.e before i call "ret" my "esp" points to in the following order ->

ebp ebx esi edi "return address"

so we don't wanna use "pushad" which is in top of the code(no need with that). and use "return ;", so that the loader itself pop out the 4 registers implicitly.

my changed code is, below

Code #include <windows.h>

#include <stdio.h>

DWORD __cdecl connect_back()

```

{

__asm{

    connectback:
        jmp startup_bnc
find:
    cld
    mov ebx, 0x00000002
    mov eax, ebx
    ret

find_function:
    pushad

find_function_loop:

compute_hash:
    xor edi, edi
    ;xor eax, eax
    cld
compute_hash_again:

    mov ebp, 0x08
compute_hash_finished:
find_function_compare:
    ;mov eax, 0x00000001
    mov [esp + 0x1c], eax

find_function_finished:
    popad
    ret

startup_bnc:
    call find
    mov edx, eax
    mov eax, 0xec0e4e8e
    push eax
    push edx
    call find_function
    add esp, 0x08

finished:

}
return ;
}

int main()
{
    DWORD addr = connect_back();
    printf("hellooo\n");
    printf("EAX Now: 0x%08x\n ", addr);
}

```

```
return 0;  
}
```