CSIS Security Research and Intelligence

Advisory – Microsoft GDI+ Integer division by zero flaw handling .ICO files
VU#290961
CVE-2007-2237

Discovered by Dennis Rand
rand@csis.dk
http://www.csis.dk

Table of contents

Introduction

The installation that this flaw has been tested on is a Windows XP Service Pack 2 with all patches applied.

Current Severity rating: Low risk
CVSS Vector: (AV:L/AC:L/Au:NR/C:N/I:N/A:P/B:N)

Timeline of public disclosure

- 02-04-2007 Vulnerability discovered.
- 17-04-2007 Research ended.
- 18-04-2007 CERT/CC informed
- 18-04-2007 Recieved VU#290961 from CERT/CC
- 25-04-2007 Recieved CVE-2007-2237 from CERT/CC
- 03-05-2007 Reported to Microsoft MSRC (secure@microsoft.com)
- 03-05-2007 Received response from MSRC (Case: 7402)
- 31-05-2007 Received response from MSRC – Flaw will be fixed in next Service Pack
- 31-05-2007 Information released on CSIS Platinum mailing list
- 06-06-2007 Public release

Contact information

The following vulnerability were discovered by Dennis Rand at CSIS Security Group
Questions regarding this issue should be directed to:

Dennis Rand
rand@csis.dk

File description

Program file

File name:	GdiPlus.dll
Company Name	Microsoft Corporation
Program version:	5.1.3102.2180
File version:	5.1.3102.2180 (xpsp_sp2_rtm.040803-2158)
Description:	Microsoft GDI+
MD5 Checksum:	78bdc89c5d9e206209bec5a5a73f91f7
SHA-1 Checksum:	5f6eb616b854cc698451f96bbe9cf5049f25245e

Technical details

Abstract

CSIS Security Group has discovered an "Integer division by zero" flaw in the GDI+ component in Windows XP. This condition are activated when a malformed ICO file are viewed through either Windows Explorer or other components like "Windows Picture and Fax Viewer".

The consequence of this flaw is a Denial of Service condition and doing a restart of the explorer process.

Further exploitation has not been verified.

Description

CSIS Security Group has discovered an "Integer division by zero" flaw in the GDI+ component in Windows XP. This condition are activated when a malformed ICO file are viewed through either Windows Explorer or other components like "Windows Picture and Fax Viewer".

The consequence of this flaw is a Denial of Service condition, to applications using the vulnerable GDI+ component, and doing a restart of the explorer process.

The flaw is in the "**InfoHeader**" → "**Height**" value within the malformed .ICO file, when inserting 0x00000000 at byte location 31 to 34.

Disassembly of the affected area

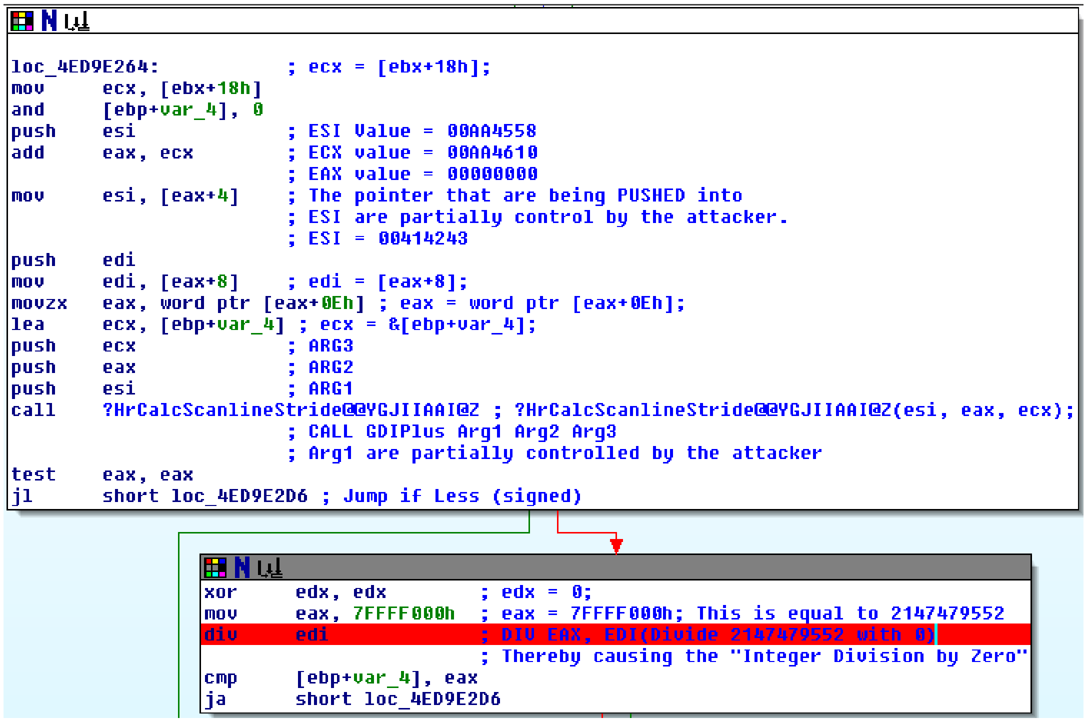
The flaw goes into the following memory area and throws the exception "Integer division by zero" at 4ED9E28F, Causing a restart of the explorer process.

Below is the vulnerable function:

```
.text:4ED9E209 ; private: int __thiscall GpIcoCodec::IsValidDIB(unsigned int)
.text:4ED9E209 ?IsValidDIB@GpIcoCodec@@AAEHI@Z proc near
.text:4ED9E209                                          ; CODE XREF: GpIcoCodec::ReadHeaders(void)+188p
```

"Integer division by Zero"

```
4ED9E28A  mov   eax,7FFFF000h    ; 7FFFF000h = 2147479552
4ED9E28F  div   eax,edi          ; 2147479552 / 0
```



```
loc_4ED9E264:                ; ecx = [ebx+18h];
mov     ecx, [ebx+18h]
and     [ebp+var_4], 0
push    esi                  ; ESI Value = 00AA4558
add     eax, ecx             ; ECX value = 00AA4610
                             ; EAX value = 00000000
mov     esi, [eax+4]         ; The pointer that are being PUSHED into
                             ; ESI are partially control by the attacker.
                             ; ESI = 00414243
push    edi
mov     edi, [eax+8]         ; edi = [eax+8];
movzx   eax, word ptr [eax+0Eh] ; eax = word ptr [eax+0Eh];
lea     ecx, [ebp+var_4] ; ecx = &[ebp+var_4];
push    ecx                  ; ARG3
push    eax                  ; ARG2
push    esi                  ; ARG1
call    ?HrCalcScanlineStride@@YGJIIAAI@Z ; ?HrCalcScanlineStride@@YGJIIAAI@Z(esi, eax, ecx);
                             ; CALL GDIPlus Arg1 Arg2 Arg3
                             ; Arg1 are partially controlled by the attacker
test    eax, eax
jl      short loc_4ED9E2D6 ; Jump if Less (signed)
```

```
xor     edx, edx             ; edx = 0;
mov     eax, 7FFFF000h       ; eax = 7FFFF000h; This is equal to 2147479552
div     edi                  ; DIV EAX, EDI(Divide 2147479552 with 0)
                             ; Thereby causing the "Integer Division by Zero"
cmp     [ebp+var_4], eax
ja      short loc_4ED9E2D6
```

Icon File format

Source: http://www.daubnet.com/formats/ICO.html

Name	Size	Description
Reserved	2 byte	=0
Type	2 byte	=1
Count	2 byte	Number of Icons in this file
Entries	Count * 16	List of icons
Width	1 byte	Cursor Width (16, 32 or 64)
Height	1 byte	Cursor Height (16, 32 or 64 , most commonly = Width)
ColorCount	1 byte	Number of Colors (2,16, 0=256)
Reserved	1 byte	=0
Planes	2 byte	=1
BitCount	2 byte	bits per pixel (1, 4, 8)
SizeInBytes	4 byte	Size of (InfoHeader + ANDbitmap + XORbitmap)
FileOffset	4 byte	FilePos, where InfoHeader starts
repeated Count times		
InfoHeader	40 bytes	Variant of BMP InfoHeader
Size	4 bytes	Size of InfoHeader structure = 40
Width	4 bytes	Icon Width
Height	**4 bytes**	**Icon Height (added height of XOR-Bitmap and AND-Bitmap)**
Planes	2 bytes	number of planes = 1
BitCount	2 bytes	bits per pixel = 1, 4, 8
Compression	4 bytes	Type of Compression = 0
ImageSize	4 bytes	Size of Image in Bytes = 0 (uncompressed)
XpixelsPerM	4 bytes	unused = 0
YpixelsPerM	4 bytes	unused = 0
ColorsUsed	4 bytes	unused = 0
ColorsImportant	4 bytes	unused = 0
Colors	NumberOfColors * 4 bytes	Color Map for XOR-Bitmap
Red	1 byte	red component
Green	1 byte	green component
Blue	1 byte	blue component
reserved	1 byte	=0
repeated NumberOfColors times		
XORBitmap	see below	bitmap
ANDBitmap	see below	monochrome bitmap

Analysis

Exploitation of the flaw will at least result in a Denial of Service condition against the program using the GDI+ component, and doing a restart of the explorer process. Further code execution has not been verified.

Detection

CSIS Security Group has confirmed this vulnerability in Windows XP with latest service pack and patch level.

Windows 2000 does not look to be vulnerable to this flaw.

Microsoft 2003 and Vista not tested.

Recovery

Currently this will kill the current running explorer.exe, however if code execution is possible it will not be possible to see if the flaw are exploited.

Exploit

Exploitation of the flaw can be triggered if a malformed icon is located in a directory that the user browses.

Proof of concept

A Proof of Concept exploit have been made.

Workaround

There are currently no known workaround available.

Fix

The issue has already been resolved in Windows Vista and in the upcoming release of Windows Server 2008, formerly known as Windows Longhorn Server. Microsoft will address the reported issue in the next Service Pack for the affected supported platforms.

What are CVSS

The National Infrastructure Advisory Council (NIAC) has chosen FIRST to be the custodian of the Common Vulnerability Scoring System (CVSS), the emerging standard in vulnerability scoring. This rating system is designed to provide open and universally standard severity ratings of software vulnerabilities. There is a critical need to help organizations appropriately prioritize security vulnerabilities across their constituency. The lack of a common scoring system has security teams worldwide solving the same problems with little or no coordination. FIRST will closely collaborate with CERT/CC and MITRE on this.

http://www.first.org/cvss/

Disclaimer

The information within this document may change without notice.
Use of this information constitutes acceptance for use in an "AS IS" condition.

There are NO warranties with regard to this information.

In no event shall I be liable for any consequences or damages, including direct, indirect, incidental, consequential, loss of business profits or special damages, arising out of or in connection with the use or spread of this information.

Any use of this information lies within the user's responsibility. All registered and unregistered trademarks represented in this document is the sole property of their respective owners.

If you use the following information you have to credit Dennis Rand from CSIS Security Group for the discovery.