Talos Vulnerability Report

TALOS-2020-0999

Accusoft ImageGear ICO icoread code execution vulnerability

MAY 5, 2020

CVE NUMBER

CVE-2020-6076

Summary

An exploitable out-of-bounds write vulnerability exists in the igcore19d.dll ICO icoread parser of the Accusoft ImageGear 19.5.0 library. A specially crafted ICO file can cause an out-of-bounds write, resulting in a remote code execution. An attacker needs to provide a malformed file to the victim to trigger the vulnerability.

Tested Versions

Accusoft ImageGear 19.5.0

Product URLs

https://www.accusoft.com/products/imagegear/overview/

CVSSv3 Score

9.8 - CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

CWE

CWE-787: Out-of-bounds Write

Details

The ImageGear library is a document imaging developer toolkit providing all kinds of functionality related to image conversion, creation, editing, annotation, etc. It supports more than 100 formats, including many image formats, DICOM, PDF, Microsoft Office and others.

There is a vulnerability in the ICO raster image parser. A specially crafted ICO file can lead to an out-of-bounds write, resulting in remote code execution.

If we try to load a malformed ICO file via the `IG_load_file` function we end up in the following situation:

```
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=fffc0060 ebx=0852ffe0 ecx=3fff0018 edx=00000000 esi=267c005c edi=26721000
eip=5b56df2c esp=00afefec ebp=00aff004 iopl=0         nv dn ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010602
MSVCR110!memcpy+0x3a4:
5b56df2c f3a5            rep movs dword ptr es:[edi],dword ptr [esi]
```

Checking attributes related with the destination buffer we can see:

```
0:000> !heap -p -a edi
        address 26720ffc found in
        _DPH_HEAP_ROOT @ bf1000
        in busy allocation (  DPH_HEAP_BLOCK:         UserAddr         UserSize -         VirtAddr         VirtSize)
                                                      b781af8:         13121000         13600000 -         13120000
13602000
        5bbfab70 verifier!AVrfDebugPageHeapAllocate+0x00000240
        77378fcb ntdll!RtlDebugAllocateHeap+0x00000039
        772cbb0d ntdll!RtlpAllocateHeap+0x000000ed
        772cb02f ntdll!RtlpAllocateHeapInternal+0x0000022f
        772cadee ntdll!RtlAllocateHeap+0x0000003e
        5b56daff MSVCR110!malloc+0x00000049
        5b8a289d igCore19d+0x0000289d
        5b8d7736 igCore19d!IG_comm_is_comp_exist+0x000062c6
        5b90f787 igCore19d!IG_mpi_page_set+0x000043f7
        5b914b3e igCore19d!IG_mpi_page_set+0x000097ae
        5b8bc8d2 igCore19d!GPb_image_associate+0x00000092
        5b91c3e0 igCore19d!IG_mpi_page_set+0x00011050
        5b8f84de igCore19d!IG_cpm_profiles_reset+0x0000dfae
        5b9b4b37 igCore19d!IG_mpi_page_set+0x000a97a7
        5b9b441a igCore19d!IG_mpi_page_set+0x000a908a
        5b8e07c9 igCore19d!IG_image_savelist_get+0x00000b29
        5b91fb97 igCore19d!IG_mpi_page_set+0x00014807
        5b91f4f9 igCore19d!IG_mpi_page_set+0x00014169
        5b8b6007 igCore19d!IG_load_file+0x00000047
        00ef59ac simple_exe_141+0x000159ac
        00ef61a7 simple_exe_141+0x000161a7
        00ef6cbe simple_exe_141+0x00016cbe
        00ef6b27 simple_exe_141+0x00016b27
        00ef69bd simple_exe_141+0x000169bd
        00ef6d38 simple_exe_141+0x00016d38
        74f56359 KERNEL32!BaseThreadInitThunk+0x00000019
        772f7b74 ntdll!__RtlUserThreadStart+0x0000002f
        772f7b44 ntdll!_RtlUserThreadStart+0x0000001b
```

We see that the `size` parameter of the memcpy function is huge: `ecx=3fff0018`, which could lead to overflow.

Further analysis revealed that the memcpy size parameter depends on:

```
        DWORD offset : 0x4
        value 0xffff0018

        WORD offset : 0xE
        value : 0020
        based on that value we chose a multiplication 2 or 4
        if x > 8
                multiplier = 4
        else
                multiplier = 2

        so memcpy size  : 0xfffc0060 = 0xffff0018 * 4

        destination buffer is allocated based on :
        DWORD offset : 8
        value 00 00 12 0B

        constant operation 0xb120000 >> 1 = 05890000

        allocation based on : 0xfffc0060 * 05890000     = 13600000
```

As we can see, an attacker controls all presented variables just by proper file content manipulation.

An attacker can cause an out-of-bounds write leading to memory corruption, which can result in remote code execution.

Crash Information

```
First chance exceptions are reported before any exception handling.
This exception may be expected and handled.
eax=fffc0060 ebx=0852ffe0 ecx=3fff0018 edx=00000000 esi=267c005c edi=26721000
eip=5b56df2c esp=00afefec ebp=00aff004 iopl=0         nv dn ei pl nz na po nc
cs=0023  ss=002b  ds=002b  es=002b  fs=0053  gs=002b             efl=00010602
MSVCR110!memcpy+0x3a4:
5b56df2c f3a5           rep movs dword ptr es:[edi],dword ptr [esi]

0:000> kb
 # ChildEBP RetAddr  Args to Child
00 00afeff0 5b8afa86 26760fa0 00000000 fffc0060 MSVCR110!memcpy+0x3a4
WARNING: Stack unwind information not available. Following frames may be wrong.
01 00aff004 5b91c9ca 26760fa0 00000000 fffc0060 igCore19d+0xfa86
02 00aff024 5b8f94fe 00aff77c 00000000 0588ffff igCore19d!IG_mpi_page_set+0x1163a
03 00aff044 5b9b4e59 00aff608 00000000 0588ffff igCore19d!IG_cpm_profiles_reset+0xefce
04 00aff0b8 5b9b441a 00aff608 1000001b 0ac9eff8 igCore19d!IG_mpi_page_set+0xa9ac9
05 00aff580 5b8e07c9 00aff608 0ac9eff8 00000001 igCore19d!IG_mpi_page_set+0xa908a
06 00aff5b8 5b91fb97 00000000 0ac9eff8 00aff608 igCore19d!IG_image_savelist_get+0xb29
07 00aff834 5b91f4f9 00000000 09ff3f88 00000001 igCore19d!IG_mpi_page_set+0x14807
08 00aff854 5b8b6007 00000000 09ff3f88 00000001 igCore19d!IG_mpi_page_set+0x14169
09 00aff874 00ef59ac 09ff3f88 00aff960 00aff984 igCore19d!IG_load_file+0x47
0a 00aff974 00ef61a7 09ff3f88 00affaa8 00000021 simple_exe_141+0x159ac
0b 00affb40 00ef6cbe 00000004 09fa0f68 09e7bf20 simple_exe_141+0x161a7
0c 00affb54 00ef6b27 2f555bad 00ef15e1 00ef15e1 simple_exe_141+0x16cbe
0d 00affbb0 00ef69bd 00affbc0 00ef6d38 00affbd0 simple_exe_141+0x16b27
0e 00affbb8 00ef6d38 00affbd0 74f56359 00930000 simple_exe_141+0x169bd
0f 00affbc0 74f56359 00930000 74f56340 00affc2c simple_exe_141+0x16d38
10 00affbd0 772f7b74 00930000 25fbdd4a 00000000 KERNEL32!BaseThreadInitThunk+0x19
11 00affc2c 772f7b44 ffffffff 77318ef2 00000000 ntdll!__RtlUserThreadStart+0x2f
12 00affc3c 00000000 00ef15e1 00930000 00000000 ntdll!_RtlUserThreadStart+0x1b

0:000> lmva eip
Browse full module list
start    end        module name
5b8a0000 5bbe9000   igCore19d   (export symbols)       d:\projects\ImageGear\current\Build\Bin\x86\igCore19d.dll
        Loaded symbol image file: d:\projects\ImageGear\current\Build\Bin\x86\igCore19d.dll
        Image path: d:\projects\ImageGear\current\Build\Bin\x86\igCore19d.dll
        Image name: igCore19d.dll
        Browse all global symbols  functions  data
        Timestamp:        Fri Nov 22 15:45:29 2019 (5DD7F489)
        CheckSum:         00356062
        ImageSize:        00349000
        File version:     19.5.0.0
        Product version:  19.5.0.0
        File flags:       0 (Mask 3F)
        File OS:          4 Unknown Win32
        File type:        2.0 Dll
        File date:        00000000.00000000
        Translations:     0409.04b0
        Information from resource tables:
                CompanyName:      Accusoft Corporation
                ProductName:      Accusoft ImageGear
                InternalName:     igcore19d.dll
                OriginalFilename: igcore19d.dll
                ProductVersion:   19.5.0.0
                FileVersion:      19.5.0.0
                FileDescription:  Accusoft ImageGear CORE DLL
                LegalCopyright:   Copyright 1996-2019 Accusoft Corporation. All rights reserved.
                LegalTrademarks:  ImageGearÆ and AccusoftÆ are registered trademarks of Accusoft Corporation
```

Timeline

2020-01-30 - Vendor Disclosure

2020-04-30 - Vendor Patched

2020-05-05 - Public Release

CREDIT

Discovered by a member of Cisco Talos.