

CVE-2020-10809: Heap overflow in decompress.c – HDF5 – 1.13.0

Heap overflow in decompress.c – HDF5 – 1.13.0

Loginsoft-2020-1002

11 March, 2020

CVE Number

CVE-2020-10809

CWE

CWE – 122 : Heap-based Buffer Overflow

Product Details

HDF5 is a data model, library, and file format for storing and managing data. It supports an unlimited variety of data types and is designed for flexible and efficient I/O and for high volume and complex data. HDF5 is portable and is extensible, allowing applications to evolve in their use of HDF5. The HDF5 Technology suite includes tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format.

URL: <https://www.hdfgroup.org/downloads>

Vulnerable Versions

1.13.0

Vulnerability Details

During our research we observed Heap overflow in the function `Decompress()` located in `decompress.c`. The same be triggered by sending a crafted file to the `gif2h5` binary. It allows an attacker to cause Denial of Service.

SYNOPSIS

We observed that in function `Gif2Mem()` in the line `gifImageDesc[imageCount-1]->image = Decompress(gifImageDesc[imageCount-1], gifHead)`; from this it calls another function `Decompress()` located in `decompress.c` to convert gif to hdf image, here in line `OutCode[OutCount++] = Suffix[CurCode]`; at the time of assignment operation left side interger pointer `OutCode` size is much small then right side interger pointer `Suffix`.

vulnerable Source code

```
276         while (CurCode > DataMask) {
277             if (OutCount >= 1024) {
278                 /*return error message*/
279             }
280
281             OutCode[OutCount++] = Suffix[CurCode];
282             CurCode = Prefix[CurCode];
283         }
284
285         /* The last code in the chain is treated as raw data. */
286         FinChar = CurCode & DataMask;
```

Analysis

DEBUG:

GDB:

Unknown Block Separator Character: 0xfe
Unknown Block Separator Character: 0xcb
Unknown Block Separator Character: 0xfd
Unknown Block Separator Character: 0x4
Unknown Block Separator Character: 0xac
Unknown Block Separator Character: 0xfa
Unknown Block Separator Character: 0x5e
Unknown Block Separator Character: 0xa2
Unknown Block Separator Character: 0xbc
Unknown Block Separator Character: 0xe4
Unknown Block Separator Character: 0xf
Unknown Block Separator Character: 0x26
Unknown Block Separator Character: 0x68
Unknown Block Separator Character: 0xc2
Unknown Block Separator Character: 0x7e
Unknown Block Separator Character: 0xf2
Unknown Block Separator Character: 0xc1
Unknown Block Separator Character: 0x1a
Unknown Block Separator Character: 0xf4
Unknown Block Separator Character: 0x27
Unknown Block Separator Character: 0x1b
Unknown Block Separator Character: 0xbc
Unknown Extension Label: 0x1a
Unknown Block Separator Character: 0xff
Unknown Block Separator Character: 0x3c
Unknown Block Separator Character: 0xc
Unknown Block Separator Character: 0x39
Unknown Block Separator Character: 0x1f
Unknown Block Separator Character: 0x88
Unknown Block Separator Character: 0x28
Unknown Block Separator Character: 0xe6
Unknown Block Separator Character: 0x46
Unknown Block Separator Character: 0xf
Unknown Block Separator Character: 0x2
Unknown Block Separator Character: 0x2
Unknown Block Separator Character: 0x28
Unknown Block Separator Character: 0x41
Unknown Block Separator Character: 0xf
Unknown Block Separator Character: 0x4
Unknown Block Separator Character: 0x77
Unknown Block Separator Character: 0x28
Unknown Block Separator Character: 0x8
Unknown Block Separator Character: 0x57
Unknown Block Separator Character: 0x12
Unknown Block Separator Character: 0x14
Unknown Block Separator Character: 0x50
Unknown Block Separator Character: 0x19
Unknown Block Separator Character: 0xb0
Unknown Block Separator Character: 0xd3
Unknown Block Separator Character: 0xc4
Unknown Block Separator Character: 0x4c
Unknown Block Separator Character: 0xdc
Unknown Block Separator Character: 0x16
Unknown Block Separator Character: 0x76
Unknown Block Separator Character: 0x20
Unknown Block Separator Character: 0xc5
Unknown Block Separator Character: 0x6a
Unknown Block Separator Character: 0xed
Unknown Block Separator Character: 0xd6
Unknown Block Separator Character: 0x9a
Unknown Block Separator Character: 0xa4
Unknown Block Separator Character: 0x49
Unknown Block Separator Character: 0xde
Unknown Block Separator Character: 0x42
Unknown Block Separator Character: 0x2e
Unknown Block Separator Character: 0xf0
Unknown Block Separator Character: 0xc4
Unknown Block Separator Character: 0x58
Unknown Block Separator Character: 0x6b
Unknown Block Separator Character: 0x8a
Unknown Block Separator Character: 0x4a
Unknown Block Separator Character: 0xb9
Unknown Block Separator Character: 0x2d
Unknown Block Separator Character: 0x4
Unknown Block Separator Character: 0xbf
Unknown Block Separator Character: 0x49
Unknown Block Separator Character: 0xda
Unknown Block Separator Character: 0xa4
Unknown Block Separator Character: 0x35
Unknown Block Separator Character: 0x5a
Unknown Block Separator Character: 0xc3
Unknown Block Separator Character: 0x14
Unknown Block Separator Character: 0x5
Unknown Block Separator Character: 0x75

Program received signal SIGSEGV, Segmentation fault.
[Legend: Modified register | Code | Heap | Stack | String]

registers ———
\$rax : 0x15f
\$rbx : 0x2312
\$rcx : 0x5b2d
\$rdx : 0xfb
\$rsp : 0x00007fffffffdd080 → 0x000000000000ae35
\$rbp : 0x0000555563db34c → 0x0000000000000000
\$r1 : 0x281
\$r1 : 0x0000555563d7340 → 0x0000000000000000
\$rip : 0x00005555656246 → mov DWORD PTR [rbp+rcx*4+0x0], edx
\$r8 : 0x0
\$r9 : 0x1137
\$r10 : 0x0000555563d3330 → 0x0000000000000000
\$r11 : 0x5b2d
\$r12 : 0x00007fffd3b4010 → 0x36c3646096d06a3
\$r13 : 0x226
\$r14 : 0xff
\$r15 : 0xae35
\$eflags: [zero CARRY PARTY adjust SIGN trap INTERRUPT direction overFlow RESUME virtualx86 identification]
\$cs: 0x0033 \$ss: 0x002b \$ds: 0x0000 \$es: 0x0000 \$fs: 0x0000 \$gs: 0x0000

stack ———
0x00007fffffffdd080+0x0000: 0x000000000000ae35 → \$rsp
0x00007fffffffdd080+0x0008: 0x0000000000000000
0x00007fffffffdd080+0x0010: 0x0000000000000027
0x00007fffffffdd080+0x0018: 0x0000555563db350 → 0x0000000f00000000
0x00007fffffffdda0+0x0020: 0x000001e900000171
0x00007fffffffdda8+0x0028: 0x00000000000000ff
0x00007fffffffddb0+0x0030: 0x0000020000000041 ("A")
0x00007fffffffddb8+0x0038: 0x000001b000000040 ("B")

code:x86:64 ———
0x555555565236 nop WORD PTR cs:[rax+rax*1+0x0]
0x555555565240 mov edx, DWORD PTR [rdi+rax*4]
0x555555565243 mov r11d, ecx
→ 0x555555565246 mov DWORD PTR [rbp+rcx*4+0x0], edx
0x55555556524a movsx rax, DWORD PTR [r10+rax*4]
0x55555556524e add rcx, 0x1
0x555555565252 cmp r14d, rax
0x555555565255 jl 0x555555565240
0x555555565257 nop

source:/home/aceteam/h[...].c+281 ———
276 while (CurCode > DataMask) {
277 if (OutCount >= 1024) {
278 /*return error message*/
279 }
280
→ 281 OutCode[OutCount++] = Suffix[CurCode];
282 CurCode = Prefix[CurCode];
283 }
284

```
threads -----
[00] Id 1, Name: "glf2h5", stopped, reason: SIGSEGV

trace -----
[00] 0x5555556246 -> Decompress (GifImageDesc=0x5555563d2fe0, GifHead=0x5555563d2510)
[01] 0x55555567d3a -> Gif2Mem (MemGif=, GifMemoryStruct=0x7fffffffdea0)
[02] 0x555555635fb -> main (argv=, argc=)

0x00005555556246 in Decompress (GifImageDesc=GifImageDesc@entry=0x5555563d2fe0,
GifHead=GifHead@entry=0x5555563d2510) at /hdfs/hl/tools/glf2h5/decompress.c:281
281      OutCode[OutCount++] = Suffix[CurCode];

gef> bt
#0 0x00005555556246 in Decompress (GifImageDesc=GifImageDesc@entry=0x5555563d2fe0,
GifHead=GifHead@entry=0x5555563d2510) at /hdfs/hl/tools/glf2h5/decompress.c:281
#1 0x000055555567d3a in Gif2Mem (MemGif=, GifMemoryStruct=0x7fffffffdea0) at
/hdfs/hl/tools/glf2h5/glf2mem.c:184
#2 0x0000555555635fb in main (argv=, argc=) at /hdfs/hl/tools/glf2h5/glf2hdf.c:100
gef> i r
rax      0x15f      0x15f
rbx      0x2312     0x2312
rcx      0x5b2d     0x5b2d
rdx      0xf0      0xf0
rsi      0x2e1      0x2e1
rdi      0x5555563d7340 0x5555563d7340
rbp      0x5555563db34c 0x5555563db34c
rsp      0x7fffffffdd80 0x7fffffffdd80
r0       0x0       0x0
r9       0x1137     0x1137
r10      0x5555563d3330 0x5555563d3330
r11      0x5b2d     0x5b2d
r12      0x7ffdf3b4010 0x7ffdf3b4010
r13      0x226      0x226
r14      0xff      0xff
r15      0xae35     0xae35
rip      0x5555556246 0x5555556246
eflags   0x102bf      [ CF PF SF IF RF ]
cs       0x33      0x33
ss       0x2b      0x2b
ds       0x0       0x0
es       0x0       0x0
fs       0x0       0x0
gs       0x0       0x0
gef> x/d OutCode
0x5555563db350:      0
```

ASAN Output:



Proof of Concept
 ./gif2h5 \$POC /dev/null
 Vendor Disclosure: 2020-3-10
 Credit
 Discovered by ACE Team – LoginSoft

Let us know how we can help you

CONTACT

US Office

4437 Brookfield Corporate Drive, Suite 101
Charlottesville, VA USA 20151.
+1 703 956 7410

Canada Office

7-7003 Steeles Ave W, Toronto,
ON M9W 0A2, Canada.

India Office

1-63-S-8B, Kavuri Hills, Jubilee Hills,
Hyderabad-500033.