Heap-based Buffer Overflow in gpac/gpac

0

✓ Valid Reported on Dec 30th 2021

Description

Heap-based Buffer Overflow SFS_AddString () at bifs/script_dec.c:76

Proof of Concept

POC1 is here.

Result

```
MP4Box -disox -ttxt -2 -dump-chap-ogg -dump-cover -drtp -bt -out /dev/null
...

[5]    538135 abort      ./source/gpac/bin/gcc/MP4Box -disox -ttxt -2 -dump
```

Bt

```
Program received signal SIGABRT, Aborted.
0x0000000000d18d6b in raise ()
LEGEND: STACK | HEAP | CODE | DATA | RWX | RODATA
─────────────────────────────────────────────────────────────────────
 RAX  0x0
 RBX  0x10dd8c0 ◂— 0x10dd8c0
 RCX  0xd18d6b (raise+203) ◂— mov    rax, qword ptr [rsp + 0x108]
 RDX  0x0
 RDI  0x2
 RSI  0x7fffffff73b0 ◂— 0x0
 R8   0x0
```

```
R9   0x7fffffff73b0 ◂— 0x0
R10  0x8
R11  0x246
R12  0x7fffffff7620 —▸ 0x1108750 ◂— 0x33333333333333f3
R13  0x10
R14  0x7ffff7ff8000 ◂— 0x6c6c616d00001000
R15  0x1
RBP  0x7fffffff7700 ◂— 0x5dc
RSP  0x7fffffff73b0 ◂— 0x0
RIP  0xd18d6b (raise+203) ◂— mov    rax, qword ptr [rsp + 0x108]
───────────────────────────────────────────────────────────────────────────
 ► 0xd18d6b <raise+203>          mov    rax, qword ptr [rsp + 0x108]
   0xd18d73 <raise+211>          xor    rax, qword ptr fs:[0x28]
   0xd18d7c <raise+220>          jne    raise+260                  <rais
    ↓
   0xd18da4 <raise+260>          call   __stack_chk_fail_local

   0xd18da9                      nop    dword ptr [rax]
   0xd18db0 <sigprocmask>        endbr64
   0xd18db4 <sigprocmask+4>      sub    rsp, 0x98
   0xd18dbb <sigprocmask+11>     xor    r8d, r8d
   0xd18dbe <sigprocmask+14>     mov    rax, qword ptr fs:[0x28]
   0xd18dc7 <sigprocmask+23>     mov    qword ptr [rsp + 0x88], rax
   0xd18dcf <sigprocmask+31>     xor    eax, eax
───────────────────────────────────────────────────────────────────────────
00:0000│ rsi r9 rsp 0x7fffffff73b0 ◂— 0x0
01:0008│            0x7fffffff73b8 —▸ 0xd437e2 (malloc+114) ◂— mov    r8, r
02:0010│            0x7fffffff73c0 ◂— 0x5
03:0018│            0x7fffffff73c8 —▸ 0x10e6370 ◂— 0x0
04:0020│            0x7fffffff73d0 ◂— 0x1
05:0028│            0x7fffffff73d8 —▸ 0xd465cf (strdup+31) ◂— test   rax, r
06:0030│            0x7fffffff73e0 —▸ 0x7fffffff7410 —▸ 0x7fffffff7880 —▸ 0
07:0038│            0x7fffffff73e8 —▸ 0x445bec (gf_bs_read_int+68) ◂— movzx
───────────────────────────────────────────────────────────────────────────
 ► f 0         0xd18d6b raise+203
   f 1         0x4013d8 abort+299
   f 2         0xd37836 __libc_message+662
   f 3         0xd3eabc
   f 4         0xd41e1c _int_malloc+3116
   f 5         0xd437e2 malloc+114
```

```
  f 6         0x450afc gf_malloc+28
  f 7         0x56de8f SFS_AddString+118
```

```
pwndbg> bt
#0  0x0000000000d18d6b in raise ()
#1  0x00000000004013d8 in abort ()
#2  0x0000000000d37836 in __libc_message ()
#3  0x0000000000d3eabc in malloc_printerr ()
#4  0x0000000000d41e1c in _int_malloc ()
#5  0x0000000000d437e2 in malloc ()
#6  0x0000000000450afc in gf_malloc (size=1500) at utils/alloc.c:150
#7  0x000000000056de8f in SFS_AddString (parser=0x7fffffff78d0, str=0xe13f
#8  0x000000000056e7bf in SFS_Arguments (parser=0x7fffffff78d0, is_var=GF_
#9  0x000000000056e540 in SFScript_Parse (codec=0x10f6d90, script_field=0x
#10 0x0000000000564ddb in gf_bifs_dec_sf_field (codec=0x10f6d90, bs=0x10e6
#11 0x0000000000565384 in BD_DecMFFieldVec (codec=0x10f6d90, bs=0x10e6370,
#12 0x000000000056588c in gf_bifs_dec_field (codec=0x10f6d90, bs=0x10e6370,
#13 0x0000000000565b0e in gf_bifs_dec_node_list (codec=0x10f6d90, bs=0x10e
#14 0x0000000000566701 in gf_bifs_dec_node (codec=0x10f6d90, bs=0x10e6370,
#15 0x00000000005653d4 in BD_DecMFFieldVec (codec=0x10f6d90, bs=0x10e6370,
#16 0x000000000056588c in gf_bifs_dec_field (codec=0x10f6d90, bs=0x10e6370,
#17 0x0000000000565b0e in gf_bifs_dec_node_list (codec=0x10f6d90, bs=0x10e
#18 0x0000000000566701 in gf_bifs_dec_node (codec=0x10f6d90, bs=0x10e6370,
#19 0x000000000055d31b in BD_DecSceneReplace (codec=0x10f6d90, bs=0x10e637
#20 0x000000000056c81d in BM_SceneReplace (codec=0x10f6d90, bs=0x10e6370,
#21 0x000000000056ca9e in BM_ParseCommand (codec=0x10f6d90, bs=0x10e6370,
#22 0x000000000056cf48 in gf_bifs_decode_command_list (codec=0x10f6d90, ES
#23 0x00000000006be0e9 in gf_sm_load_run_isom (load=0x7fffffff8850) at sce
#24 0x00000000006a2059 in gf_sm_load_run (load=0x7fffffff8850) at scene_ma
#25 0x000000000041786e in dump_isom_scene (file=0x7fffffffe649 "discxx/__G
#26 0x000000000041521f in mp4boxMain (argc=11, argv=0x7fffffffe2d8) at mai
#27 0x000000000041719b in main (argc=11, argv=0x7fffffffe2d8) at main.c:64
#28 0x0000000000d09840 in __libc_start_main ()
#29 0x000000000040211e in _start ()
```

Chat with us

Vulnerability Type

CWE-122: Heap-based Buffer Overflow

Severity
Medium (6.8)

Visibility
Public

Status
Fixed

Found by

zfeixq
@zfeixq
unranked

This report was seen 552 times.

We are processing your report and will contact the **gpac** team within 24 hours. a year ago

We have contacted a member of the **gpac** team and are waiting to hear back a year ago

We have sent a follow up to the **gpac** team. We will try again in 7 days. a year ago

We have sent a second follow up to the **gpac** team. We will try again in 10 days. 10 months ago

A **gpac/gpac** maintainer validated this vulnerability 10 months ago

zfeixq has been awarded the disclosure bounty ✓

The fix bounty is now up for grabs

A **gpac/gpac** maintainer marked this as fixed in **1.1.0-DEV HEAD** with commit **b5741d**
10 months ago

The fix bounty has been dropped ✗

This vulnerability will not receive a CVE ✗

Sign in to join this conversation

Chat with us