Segmentation Fault caused by MP4Box -lsr in gpac/gpac

1

Valid Reported on Mar 16th 2022

Version:

```
MP4Box -version
MP4Box - GPAC version 2.1-DEV-rev48-gf6d6225a9-master
(c) 2000-2022 Telecom Paris distributed under LGPL v2.1+ - http://gpac.io/
 MINI build (encoders, decoders, audio and video output disabled)

Please cite our work in your research:
 GPAC Filters: https://doi.org/10.1145/3339825.3394929
 GPAC: https://doi.org/10.1145/1291233.1291452

GPAC Configuration: --static-bin
Features: GPAC_CONFIG_LINUX GPAC_64_BITS GPAC_HAS_SOCK_UN GPAC_MINIMAL_ODF
```

Command: `MP4Box -lsr poc1`
Result: `Segmentation fault`
poc1.zip
bt:

```
[iso file] Unknown box type FF00ak in parent moov
[iso file] Unknown box type t0E18 in parent trak
[iso file] extra box maxr found in hinf, deleting
[iso file] Unknown box type 80rak in parent moov
[iso file] Incomplete box mdat - start 11495 size 803729
[iso file] Incomplete file while reading for dump - aborting parsing
[iso file] Unknown box type FF00ak in parent moov
[iso file] Unknown box type t0E18 in parent trak
[iso file] extra box maxr found in hinf, deleting
[iso file] Unknown box type 80rak in parent moov
[iso file] Incomplete box mdat - start 11495 size 803729
```

```
[iso file] Incomplete file while reading for dump - aborting parsing
MPEG-4 BIFS Scene Parsing
[ODF] Reading bifs config: shift in sizes (not supported)
[BIFS] Node Group not allowed as field/child of NDT type 37
Scene loaded - dumping 1 systems streams

Program received signal SIGSEGV, Segmentation fault.
gf_dump_vrml_simple_field (sdump=sdump@entry=0x5555558873e0, parent=<optimi
759         gf_dump_vrml_node(sdump, *(GF_Node **)field.far_ptr, 0, NULL);

   0x00007ffff71d1e1a <+346>:   add    %rbp,%r8
   0x00007ffff71d1e1d <+349>:   movzbl (%r8),%eax
   0x00007ffff71d1e21 <+353>:   add    $0x1,%al
   0x00007ffff71d1e23 <+355>:   jb     0x7ffff71d23b5 <gf_dump_vrml_simple_
   0x00007ffff71d1e29 <+361>:   add    %r14d,%eax
   0x00007ffff71d1e2c <+364>:   xor    %ecx,%ecx
   0x00007ffff71d1e2e <+366>:   xor    %edx,%edx
   0x00007ffff71d1e30 <+368>:   mov    %r12,%rdi
   0x00007ffff71d1e33 <+371>:   mov    %al,(%r8)
   0x00007ffff71d1e36 <+374>:   movl   $0x516d,%fs:(%rbx)
=> 0x00007ffff71d1e3d <+381>:   mov    0x0(%r13),%rsi
   0x00007ffff71d1e41 <+385>:   callq  0x7ffff71cd170 <gf_dump_vrml_node>
   0x00007ffff71d1e46 <+390>:   mov    %fs:(%rbx),%r8d
   0x00007ffff71d1e4a <+394>:   xor    $0x2a76,%r8d
   0x00007ffff71d1e51 <+401>:   add    %rbp,%r8
   0x00007ffff71d1e54 <+404>:   xor    %r12d,%r12d
   0x00007ffff71d1e57 <+407>:   movzbl (%r8),%ebp
   0x00007ffff71d1e5b <+411>:   add    $0x1,%bpl
   0x00007ffff71d1e5f <+415>:   jb     0x7ffff71d237f <gf_dump_vrml_simple_
   0x00007ffff71d1e65 <+421>:   add    %r12d,%ebp
   0x00007ffff71d1e68 <+424>:   mov    %bpl,(%r8)

Stack level 0, frame at 0x7fffffff6f30:
 rip = 0x7ffff71d1e3d in gf_dump_vrml_simple_field (scene_manager/scene_dum
 called by frame at 0x7fffffff6fc0
 source language c.
 Arglist at 0x7fffffff6eb8, args: sdump=sdump@entry=0x5555558873e0, parent=
 Locals at 0x7fffffff6eb8, Previous frame's sp is 0x7fffffff6f30
 Saved registers:
  rbx at 0x7fffffff6ef8, rbp at 0x7fffffff6f00, r12 at 0x7fffffff6f08, r13
```

```
(gdb) where

#0  gf_dump_vrml_simple_field (sdump=sdump@entry=0x5555558873e0, parent=<op
#1  0x00007ffff71d27aa in DumpMultipleIndexedReplace (sdump=sdump@entry=0x5
#2  0x00007ffff71c7ace in gf_sm_dump_command_list
    (sdump=sdump@entry=0x5555558873e0, comList=<optimized out>, indent=inde
#3  0x00007ffff71db286 in gf_sm_dump
    (ctx=ctx@entry=0x55555587e660, rad_name=rad_name@entry=0x555555832de0 <
#4  0x00005555555b9e20 in dump_isom_scene
    (file=<optimized out>, inName=<optimized out>, is_final_name=<optimized
    at filedump.c:217
#5  0x000055555559ae41 in mp4boxMain (argc=<optimized out>, argv=<optimized
#6  0x00007ffff65d60b3 in __libc_start_main (main=
    0x55555556e920 <main>, argc=3, argv=0x7fffffffe068, init=<optimized out
    at ../csu/libc-start.c:308
#7  0x000055555556eb2e in _start () at main.c:6601
```

CVE
CVE-2022-1035
(Published)

Vulnerability Type
CWE-476: NULL Pointer Dereference

Severity
Low (2.9)

Visibility
Public

Status
Fixed

Found by

lyonnhk
@lyonnhk
unranked

This report was seen 731 times.

We are processing your report and will contact the **gpac** team within 24 hours. 8 months ago

We have contacted a member of the **gpac** team and are waiting to hear back 8 months ago

A **gpac/gpac** maintainer 8 months ago — Maintainer

https://github.com/gpac/gpac/issues/2146

A **gpac/gpac** maintainer validated this vulnerability 8 months ago

lyonnhk has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

A **gpac/gpac** maintainer marked this as fixed in **2.1.0-DEV** with commit **3718d5** 8 months ago

The fix bounty has been dropped ✘

This vulnerability will not receive a CVE ✘

lyonnhk 8 months ago — Researcher

@maintainer @admin may i have CVE assigned to this case? Thanks!

A **gpac/gpac** maintainer 8 months ago — Maintainer

We do not interfere with this. Create whatever is necessary on your side.

Jamie Slome 8 months ago — Admin

@maintainer - from our perspective, a report with "none" as severity is not deserving of a CVE, as CVEs are reserved for security issues only.

Do you agree here?

A **gpac/gpac** maintainer 8 months ago — Maintainer

A gpac/gpac maintainer 8 months ago Maintainer

Yes. That's a more general issue: reporter submit crashes but they don't give us clue on how exploitable they are. We are no security experts at this point.

Jamie Slome 8 months ago Admin

In that case, we will `not` assign a CVE here - thanks 👍

lyonnhk 8 months ago Researcher

@admin @maintainer
I have forgot to set the severity under CVSS and after digging deeper, i have found that this is a null-pointer dereference vulnerability.

This kind of vulnerability in C program usually have a medium severity for e.g:
CVE-2022-0712

Attached is the analysis on GDB:

```
751 {
752  u32 i, sf_type;
753  GF_ChildNodeItem *list;
754  void *slot_ptr;
(gdb)
755
756  switch (field.fieldType) {
757  case GF_SG_VRML_SFNODE:
758   assert ( *(GF_Node **)field.far_ptr);
759   gf_dump_vrml_node(sdump, *(GF_Node **)field.far_ptr, 0, NULL);
760   return;
761  case GF_SG_VRML_MFNODE:
762   list = * ((GF_ChildNodeItem **) field.far_ptr);
763   assert( list );
764   sdump->indent++;
(gdb)
765   while (list) {
766    gf_dump_vrml_node(sdump, list->node, 1, NULL);
767    list = list->next;
768   }
769   sdump->indent--;
770   return;
771  case GF_SG_VRML_SFCOMMANDBUFFER:
772   return;
```

```
773  }
774  if (gf_sg_vrml_is_sf_field(field.fieldType)) {
```

```
#1  0x00007ffff71d27aa in DumpMultipleIndexedReplace (sdump=sdump@entry=0x5555558873e0
1731   gf_dump_vrml_simple_field(sdump, field, com->node);
(gdb) l
1726   if (sdump->XMLDump) {
1727    gf_fprintf(sdump->trace, "<repValue position=\"%d\" ", inf->pos);
1728   } else {
1729    gf_fprintf(sdump->trace, "%d BY ", inf->pos);
1730   }
1731   gf_dump_vrml_simple_field(sdump, field, com->node);
1732   if (sdump->XMLDump) {
1733    gf_fprintf(sdump->trace, "/>");
1734   } else {
1735    gf_fprintf(sdump->trace, "\n");
(gdb) p field
$2 = {fieldIndex = 1, fieldType = 10, far_ptr = 0x0, name = 0x7ffff7d93e5c "removeChil
```

In the above segment, the value of far_ptr is 0x0, this occurred after executing the program which can proceed to cause null-pointer dereferencing.

Can you assist to update the report severity as well as assign the CVE accordingly. Sorry for the inconveniences.

lyonnhk 8 months ago Researcher

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

Score: 7.1 (Medium)

Jamie Slome 8 months ago Admin

@maintainer, in light of the new information, do you believe this to be a security issue?

@lyonnhk - what is the impact and implications of this vulnerability? What co
do?

lyonnhk 8 months ago Researcher

Typically, this can allow attackers to read sensitive information from other memory locations or cause a crash. For more information please see cve-125

lyonnhk 8 months ago Researcher

@maintainer @admin any updates?

A gpac/gpac maintainer 8 months ago Maintainer

I confirm this is a null-pointer dereference. Please proceed as per the best practices.

Jamie Slome 8 months ago Admin

@maintainer - sure, we can update the CWE for you. Can you please also confirm what the severity of this is? Would you consider it to be NONE, LOW, MEDIUM, HIGH, CRITICAL?

Also, would you like us to proceed with a CVE for this report?

A gpac/gpac maintainer 8 months ago Maintainer

I would say LOW but I am no security expert. Please go ahead with the CVE if that's what makes sense here.

Jamie Slome 8 months ago Admin

Changes:

None (0) -> Low (2.9)

CWE-125 -> CWE-476 (Null Pointer Dereference)

CVE assigned/published

Sign in to join this conversation

Chat with us