gpac / **gpac** Public

Code · Issues 11 · Pull requests 2 · Wiki · Security · Insights

New issue

Jump to bottom

[BUG] heap-buffer-overflow in gf_base64_encode #2138

Closed · 3 tasks done · **kdsjZh** opened this issue on Mar 10 · 3 comments

kdsjZh commented on Mar 10 • edited

Thanks for reporting your issue. Please make sure these boxes are checked before submitting your issue - thank you!

- ☑ I looked for a similar issue and couldn't find any.
- ☑ I tried with the latest version of GPAC. Installers available at http://gpac.io/downloads/gpac-nightly-builds/
- ☑ I give enough information for contributors to reproduce my issue (meaningful title, github labels, platform and compiler, command-line ...). I can share files anonymously with this dropbox: https://www.mediafire.com/filedrop/filedrop_hosted.php?drop=eec9e058a9486fe4e99c33021481d9e1826ca9dbc242a6cfaab0fe95da5e5d95

Detailed guidelines: http://gpac.io/2013/07/16/how-to-file-a-bug-properly/

Describe the bug
There is a heap-buffer-overflow bug, which can be triggered via MP4Box+ ASan

To Reproduce
Steps to reproduce the behavior:

```
./configure --cc=clang --cxx=clang++ --enable-sanitizer
make -j$(nproc)
./bin/gcc/MP4Box -diso POC
```

Output:

```
[iso file] Box "moof" (start 0) has 3 extra bytes
[iso file] Movie fragment but no moov (yet) - possibly broken parsing!
[iso file] Box "moof" (start 23) has 3 extra bytes
[iso file] Box "moof" (start 34) has 3 extra bytes
[iso file] Box "moof" (start 77) has 3 extra bytes
[iso file] Box "tref" (start 45) has 4 extra bytes
[iso file] Unknown top-level box type 0005hEB
```

```
=================================================================
==1787100==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000001012 at pc
0x0000005b4fdc bp 0x7ffde5e08a70 sp 0x7ffde5e08a68
WRITE of size 1 at 0x602000001012 thread T0
    #0 0x5b4fdb in gf_base64_encode
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/utils/base_encoding.c:48:13
    #1 0x8fdb6b in colr_box_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_dump.c:5493:15
    #2 0x90c095 in gf_isom_box_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_funcs.c:2076:2
    #3 0x8cf29c in gf_isom_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_dump.c:135:3
    #4 0x539be2 in dump_isom_xml
/home/hzheng/workspace/benchmarks/reproduce/gpac/applications/mp4box/filedump.c:1954:6
    #5 0x51939b in mp4boxMain
/home/hzheng/workspace/benchmarks/reproduce/gpac/applications/mp4box/main.c:6155:7
    #6 0x7faccbbfc0b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)
    #7 0x41fdad in _start
(/home/hzheng/workspace/benchmarks/reproduce/gpac/bin/gcc/MP4Box+0x41fdad)

0x602000001012 is located 0 bytes to the right of 2-byte region [0x602000001010,0x602000001012)
allocated by thread T0 here:
    #0 0x4c58ff in malloc /home/hzheng/env/llvm-project/compiler-
rt/lib/asan/asan_malloc_linux.cpp:145:3
    #1 0x8fdb37 in gf_malloc
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/utils/alloc.c:150:9
    #2 0x8fdb37 in colr_box_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_dump.c:5492:20
    #3 0x90c095 in gf_isom_box_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_funcs.c:2076:2
    #4 0x8cf29c in gf_isom_dump
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/isomedia/box_dump.c:135:3
    #5 0x539be2 in dump_isom_xml
/home/hzheng/workspace/benchmarks/reproduce/gpac/applications/mp4box/filedump.c:1954:6
    #6 0x51939b in mp4boxMain
/home/hzheng/workspace/benchmarks/reproduce/gpac/applications/mp4box/main.c:6155:7
    #7 0x7faccbbfc0b2 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x270b2)

SUMMARY: AddressSanitizer: heap-buffer-overflow
/home/hzheng/workspace/benchmarks/reproduce/gpac/src/utils/base_encoding.c:48:13 in
gf_base64_encode
Shadow bytes around the buggy address:
  0x0c047fff81b0: fa fa 07 fa fa fa fd fa fa fa 04 fa fa fa 00 02
  0x0c047fff81c0: fa fa fd fa fa fa 00 07 fa fa 00 00 fa fa 00 00
  0x0c047fff81d0: fa fa 00 fa fa fa fd fa fa fa 00 04 fa fa 00 00
  0x0c047fff81e0: fa fa 00 00 fa fa 01 fa fa fa 00 00 fa fa 00 00
  0x0c047fff81f0: fa fa 04 fa fa fa 00 00 fa fa 04 fa fa fa 01 fa
=>0x0c047fff8200: fa fa[02]fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8210: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8220: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8230: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8240: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8250: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
```

```
    Heap left redzone:       fa
    Freed heap region:       fd
    Stack left redzone:      f1
    Stack mid redzone:       f2
    Stack right redzone:     f3
    Stack after return:      f5
    Stack use after scope:   f8
    Global redzone:          f9
    Global init order:       f6
    Poisoned by user:        f7
    Container overflow:      fc
    Array cookie:            ac
    Intra object redzone:    bb
    ASan internal:           fe
    Left alloca redzone:     ca
    Right alloca redzone:    cb
    Shadow gap:              cc
  ==1787100==ABORTING
```

Environment
gpac commit `54e9ed8`
clang release/12.x
ubuntu 20.04

POC
POC.zip

Credit
Han Zheng
NCNIPC of China
Hexhive

kdsjZh commented on Mar 10 Author

I've just verified that it can be reproduced in the latest commit `6c51dde` .

jeanlf closed this as completed in `ea1eca0` on Mar 10

risicle commented on Jul 31

This looks like it should have a CVE

kdsjZh commented on Aug 1 Author

yes, CVE-2022-26967.

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants