New issue

Jump to bottom

# Heap-buffer-overflow in OD_ReadUTF8String() odf_code.c #1481

Closed **14isnot40** opened this issue on May 12, 2020 · 1 comment

**14isnot40** commented on May 12, 2020

- [ y] I looked for a similar issue and couldn't find any.
- [ y] I tried with the latest version of GPAC. Installers available at http://gpac.io/downloads/gpac-nightly-builds/
- [ y] 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

**Describe the bug**
A heap-based buffer overflow was discovered in libgpac. The issue is being triggered in the function OD_ReadUTF8String() at odf_code.c

**To Reproduce**
Steps to reproduce the behavior:

1. Compile according to the default configuration

```
$ CC="gcc -fsanitize=address -g" CXX="g++ -fsanitize=address -g" ./configure --static-mp4box
$ make
```

2. execute command

```
MP4Box -hint $poc
```

poc can be found here.

**Expected behavior**
An attacker can exploit this vulnerability by submitting a malicious media file that exploits this issue. This will result in a Denial of Service (DoS) and potentially Information Exposure when the application attempts to process the file.

**Screenshots**
ASAN Reports

```
==42612==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000ef51 at pc 0x7ffff6eda20b bp 0x7fffffff8b60 sp 0x7fffffff8308
READ of size 2 at 0x60200000ef51 thread T0
    #0 0x7ffff6eda20a in __interceptor_strlen (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x7020a)
    #1 0x79532f in OD_SizeUTF8String odf/odf_code.c:49
    #2 0x79532f in gf_odf_size_sup_cid odf/odf_code.c:3208
    #3 0x79739d in gf_odf_desc_size odf/odf_codec.c:364
    #4 0xac28c0 in iods_Size isomedia/box_code_base.c:2818
    #5 0x6aa2f7 in gf_isom_box_size_listing isomedia/box_funcs.c:1588
    #6 0x6aa2f7 in gf_isom_box_size isomedia/box_funcs.c:1601
    #7 0xac6157 in moov_Size isomedia/box_code_base.c:3833
    #8 0x6aa2f7 in gf_isom_box_size_listing isomedia/box_funcs.c:1588
    #9 0x6aa2f7 in gf_isom_box_size isomedia/box_funcs.c:1601
    #10 0x6e1599 in GetMoovAndMetaSize isomedia/isom_store.c:352
    #11 0x6e7a1e in WriteInterleaved isomedia/isom_store.c:1356
    #12 0x6e8be9 in WriteToFile isomedia/isom_store.c:1498
    #13 0x6c9001 in gf_isom_write isomedia/isom_read.c:483
    #14 0x6c9392 in gf_isom_close isomedia/isom_read.c:507
    #15 0x429a8e in mp4boxMain (/usr/local/bin/MP4Box+0x429a8e)
    #16 0x7ffff615e82f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #17 0x41d668 in _start (/usr/local/bin/MP4Box+0x41d668)

0x60200000ef51 is located 0 bytes to the right of 1-byte region [0x60200000ef50,0x60200000ef51)
allocated by thread T0 here:
    #0 0x7ffff6f02602 in malloc (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x98602)
    #1 0x79525c in OD_ReadUTF8String odf/odf_code.c:40
    #2 0x79525c in gf_odf_read_sup_cid odf/odf_code.c:3197

SUMMARY: AddressSanitizer: heap-buffer-overflow ??:0 __interceptor_strlen
Shadow bytes around the buggy address:
  0x0c047fff9d90: fa fa 00 00 fa fa 00 00 fa fa 00 fa fa fa 01 fa
  0x0c047fff9da0: fa fa 00 00 fa fa 01 fa fa fa 00 00 fa fa 00 00
  0x0c047fff9db0: fa fa 00 00 fa fa 00 00 fa fa 00 00 fa fa 00 00
  0x0c047fff9dc0: fa fa 01 fa fa fa 00 00 fa fa 00 00 fa fa 01 fa
  0x0c047fff9dd0: fa fa 00 00 fa fa 01 fa fa fa fd fd fa fa fd fa
=>0x0c047fff9de0: fa fa fd fd fa fa fd fd fa fa[01]fa fa fa 01 fa
  0x0c047fff9df0: fa fa fd fd fa fa 00 00 fa fa 00 04 fa fa 00 00
  0x0c047fff9e00: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff9e10: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff9e20: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff9e30: 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
  Heap right redzone:      fb
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack partial redzone:   f4
  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
==42612==ABORTING
[Inferior 1 (process 42612) exited with code 01]
```

**System (please complete the following information):**

- OS version : Ubuntu 16.04
- GPAC Version : GPAC 0.8.0-e10d39d-master branch

**jeanlf** added a commit that referenced this issue on Jun 11, 2020

`fixed potential crashes on broken fragmented files - cf #1481 and #1480` ✓ `47d8bc5`

**jeanlf** commented on Jun 11, 2020 — Contributor

fixed, thanks for the report

**jeanlf** closed this as completed on Jun 11, 2020

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants