Buffer Over Read in gf_utf8_wcslen in gpac/gpac

0

Valid Reported on Sep 7th 2022

Description

Buffer Over Read in function gf_utf8_wcslen at gpac/src/utils/utf.c:442 .

gpac version

```
git log
commit fc4749f9ce8d6ddf50d1f1104366cdacede14d33 (grafted, HEAD -> master, c
Author: Aurelien David <aurelien.david@telecom-paristech.fr>
Date:   Mon Aug 1 06:44:34 2022 -0700

fix quickjs build on osx < 10.12 (#2229)

./MP4Box -version
MP4Box - GPAC version 2.1-DEV-revUNKNOWN-master
(c) 2000-2022 Telecom Paris distributed under LGPL v2.1+ - http://gpac.io
```

Proof of Concept

poc download url:
https://github.com/Janette88/test_pocs/blob/main/poc1_hbo.dat

with asan log:

```
./MP4Box -diso ../../../test/poc1_hbo.dat

[isom] invalid tag size in Xtra !
[isom] not enough bytes in box Xtra: 7 left, reading 8 (file isomedia/box_c
```

```
[iso file] Box "Xtra" (start 24) has 7 extra bytes
[iso file] Read Box type 00000001 (0x00000001) at position 92 has size 0 bu
[iso file] Box "moof" (start 84) has 8 extra bytes

[iso file] Movie fragment but no moov (yet) - possibly broken parsing!
[iso file] Box "vwid" (start 204) has 5 extra bytes
[iso file] Unknown top-level box type 00000B01
[iso file] Incomplete box 00000B01 - start 264 size 34164724
[iso file] Incomplete file while reading for dump - aborting parsing
=================================================================
==95685==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6030000
READ of size 2 at 0x603000000f20 thread T0
    #0 0x7f9c7e90cab7 in gf_utf8_wcslen utils/utf.c:442
    #1 0x7f9c7e90cab7 in gf_utf8_wcslen utils/utf.c:438
    #2 0x7f9c7ede8243 in xtra_box_dump isomedia/box_dump.c:6471
    #3 0x7f9c7edef7ed in gf_isom_box_dump isomedia/box_funcs.c:2108
    #4 0x7f9c7edb5fa9 in gf_isom_dump isomedia/box_dump.c:138
    #5 0x55de6ec03d86 in dump_isom_xml /home/fuzz/gpac2/gpac/applications/m
    #6 0x55de6ebe6c49 in mp4box_main /home/fuzz/gpac2/gpac/applications/mp4
    #7 0x7f9c7c3bf082 in __libc_start_main ../csu/libc-start.c:308
    #8 0x55de6ebc0afd in _start (/home/fuzz/gpac2/gpac/bin/gcc/MP4Box+0xa2c

0x603000000f21 is located 0 bytes to the right of 17-byte region [0x6030000
allocated by thread T0 here:
    #0 0x7f9c82138808 in __interceptor_malloc ../../../../src/libsanitizer/
    #1 0x7f9c7ed9a26b in xtra_box_read isomedia/box_code_base.c:12890
    #2 0x7f9c7edeb593 in gf_isom_box_read isomedia/box_funcs.c:1860
    #3 0x7f9c7edeb593 in gf_isom_box_parse_ex isomedia/box_funcs.c:271
    #4 0x7f9c7edec9e5 in gf_isom_parse_root_box isomedia/box_funcs.c:38
    #5 0x7f9c7ee15a6c in gf_isom_parse_movie_boxes_internal isomedia/isom_i
    #6 0x7f9c7ee1bbdf in gf_isom_parse_movie_boxes isomedia/isom_intern.c:8
    #7 0x7f9c7ee1bbdf in gf_isom_open_file isomedia/isom_intern.c:980
    #8 0x55de6ebe5539 in mp4box_main /home/fuzz/gpac2/gpac/applications/mp4
    #9 0x7f9c7c3bf082 in __libc_start_main ../csu/libc-start.c:308

SUMMARY: AddressSanitizer: heap-buffer-overflow utils/utf.c:442 in gf_utf8_
Shadow bytes around the buggy address:
  0x0c067fff8190: 00 fa fa fa 00 00 00 fa fa fa 00 00 00 fa fa fa
  0x0c067fff81a0: 00 00 00 fa fa fa 00 00 00 fa fa fa 00 00 00 fa
  0x0c067fff81b0: fa fa 00 00 00 fa fa fa 00 00 00 fa fa fa 00 00
  0x0c067fff81c0: 00 fa fa fa 00 00 00 06 fa fa 00 00 00 fa fa fa
```

```
  0x0c067fff81d0: 00 00 00 02 fa fa 00 00 00 03 fa fa fd fd fd fd
=>0x0c067fff81e0: fa fa 00 00[01]fa fa fa 00 00 00 00 fa fa 00 00
  0x0c067fff81f0: 00 fa fa fa fd fd fd fd fa fa fa fa fa fa fa fa

  0x0c067fff8200: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff8210: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff8220: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c067fff8230: 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
==95685==ABORTING
```

valgrind (without asan) log:

```
valgrind ./MP4Box -diso ../../../test/poc1_hbo.dat
==99671== Memcheck, a memory error detector
==99671== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==99671== Using Valgrind-3.19.0 and LibVEX; rerun with -h f
==99671== Command: ./MP4Box -diso ../../../test/poc1_hbo.da
==99671==
```

```
[isom] invalid tag size in Xtra !
[isom] not enough bytes in box Xtra: 7 left, reading 8 (file isomedia/box_c
[iso file] Box "Xtra" (start 24) has 7 extra bytes

[iso file] Read Box type 00000001 (0x00000001) at position 92 has size 0 bu
[iso file] Box "moof" (start 84) has 8 extra bytes
[iso file] Movie fragment but no moov (yet) - possibly broken parsing!
[iso file] Box "vwid" (start 204) has 5 extra bytes
[iso file] Unknown top-level box type 00000B01
[iso file] Incomplete box 00000B01 - start 264 size 34164724
[iso file] Incomplete file while reading for dump - aborting parsing
==99671== Invalid read of size 2
==99671==    at 0x4933D0C: gf_utf8_wcslen (in /home/fuzz/gpac2/gpac/bin/gcc
==99671==    by 0x4A7944D: xtra_box_dump (in /home/fuzz/gpac2/gpac/bin/gcc/
==99671==    by 0x4A7BDF1: gf_isom_box_dump (in /home/fuzz/gpac2/gpac/bin/g
==99671==    by 0x4A684E5: gf_isom_dump (in /home/fuzz/gpac2/gpac/bin/gcc/I
==99671==    by 0x13F532: dump_isom_xml (in /home/fuzz/gpac2/gpac/bin/gcc/M
==99671==    by 0x131C9E: mp4box_main (in /home/fuzz/gpac2/gpac/bin/gcc/MP4
==99671==    by 0x5144082: (below main) (libc-start.c:308)
==99671==  Address 0x5543320 is 16 bytes inside a block of size 17 alloc'd
==99671==    at 0x483C855: malloc (vg_replace_malloc.c:381)
==99671==    by 0x4A5F326: xtra_box_read (in /home/fuzz/gpac2/gpac/bin/gcc/
==99671==    by 0x4A7A7D8: gf_isom_box_parse_ex (in /home/fuzz/gpac2/gpac/b
==99671==    by 0x4A7B022: gf_isom_parse_root_box (in /home/fuzz/gpac2/gpac
==99671==    by 0x4A839D5: gf_isom_parse_movie_boxes_internal (in /home/fuz
==99671==    by 0x4A85196: gf_isom_open_file (in /home/fuzz/gpac2/gpac/bin/
==99671==    by 0x132F03: mp4box_main (in /home/fuzz/gpac2/gpac/bin/gcc/MP4
==99671==    by 0x5144082: (below main) (libc-start.c:308)
==99671==
==99671== Invalid read of size 2
==99671==    at 0x4933D0C: gf_utf8_wcslen (in /home/fuzz/gpac2/gpac/bin/gcc
==99671==    by 0x4933D6B: gf_utf8_wcstombs (in /home/fuzz/gpac2/gpac/bin/g
==99671==    by 0x4A7946E: xtra_box_dump (in /home/fuzz/gpac2/gpac/bin/gcc/
==99671==    by 0x4A7BDF1: gf_isom_box_dump (in /home/fuzz/gpac2/gpac/bin/g
==99671==    by 0x4A684E5: gf_isom_dump (in /home/fuzz/gpac2/gpac/bin/gcc/I
==99671==    by 0x13F532: dump_isom_xml (in /home/fuzz/gpac2/gpac/bin/gcc/M
==99671==    by 0x131C9E: mp4box_main (in /home/fuzz/gpac2/gpac/bin/gcc/MP4
==99671==    by 0x5144082: (below main) (libc-start.c:308)
==99671==  Address 0x5543320 is 16 bytes inside a block of si
==99671==    at 0x483C855: malloc (vg_replace_malloc.c:381)
==99671==    by 0x4A5F326: xtra_box_read (in /home/fuzz/gpac2/gpac/bin/gcc/
```

```
==99671==    by 0x4A7A7D8: gf_isom_box_parse_ex (in /home/fuzz/gpac2/gpac/b
==99671==    by 0x4A7B022: gf_isom_parse_root_box (in /home/fuzz/gpac2/gpac
==99671==    by 0x4A839D5: gf_isom_parse_movie_boxes_internal (in /home/fuz

==99671==    by 0x4A85196: gf_isom_open_file (in /home/fuzz/gpac2/gpac/bin/
==99671==    by 0x132F03: mp4box_main (in /home/fuzz/gpac2/gpac/bin/gcc/MP4
==99671==    by 0x5144082: (below main) (libc-start.c:308)
==99671==
==99671==
==99671== HEAP SUMMARY:
==99671==     in use at exit: 0 bytes in 0 blocks
==99671==   total heap usage: 309 allocs, 309 frees, 739,238 bytes allocate
==99671==
==99671== All heap blocks were freed -- no leaks are possible
==99671==
==99671== For lists of detected and suppressed errors, rerun with: -s
==99671== ERROR SUMMARY: 2 errors from 2 contexts (suppressed: 0 from 0)
```

ps: The vulnerability exits in the newest version. I checked it ever happened in older version ,but the patch didn't work in the newest version.The bug is still there. I tried to fix the bug based on the commit 915e2cb.
(https://github.com/gpac/gpac/commit/915e2cba715f36b7cc29e28888117831ca143d78)
in /gpac/src/isomedia/box_code_base.c#L12890

```
if (prop_size>4) {
            tag_size-=2;
            prop_type = gf_bs_read_u16(bs);
            prop_size -= 6;
            ISOM_DECREASE_SIZE_NO_ERR(ptr, prop_size)
            //add 2 extra bytes for UTF16 case string dump
            data2 = gf_malloc(sizeof(char) * (prop_size+3));
            gf_bs_read_data(bs, data2, prop_size);
            data2[prop_size] = 0;
            data2[prop_size+1] = 0;
            data2[prop_size+2] = 0;                                    //2) add t
            tag_size-=prop_size;
        } else {
            prop_size = 0;
```

```
}
```

Then saved the file and recompiled the gpac, here is my testing log :

```
valgrind ./MP4Box -diso ../../../test/poc1_hbo.dat
==103699== Memcheck, a memory error detector
==103699== Copyright (C) 2002-2022, and GNU GPL'd, by Julian Seward et al.
==103699== Using Valgrind-3.19.0 and LibVEX; rerun with -h for copyright ir
==103699== Command: ./MP4Box -diso ../../../test/poc1_hbo.dat
==103699==
[isom] invalid tag size in Xtra !
[isom] not enough bytes in box Xtra: 7 left, reading 8 (file isomedia/box_c
[iso file] Box "Xtra" (start 24) has 7 extra bytes
[iso file] Read Box type 00000001 (0x00000001) at position 92 has size 0 bu
[iso file] Box "moof" (start 84) has 8 extra bytes
[iso file] Movie fragment but no moov (yet) - possibly broken parsing!
[iso file] Box "vwid" (start 204) has 5 extra bytes
[iso file] Unknown top-level box type 00000B01
[iso file] Incomplete box 00000B01 - start 264 size 34164724
[iso file] Incomplete file while reading for dump - aborting parsing
==103699==
==103699== HEAP SUMMARY:
==103699==     in use at exit: 0 bytes in 0 blocks
==103699==   total heap usage: 309 allocs, 309 frees, 739,239 bytes allocat
==103699==
==103699== All heap blocks were freed -- no leaks are possible
==103699==
==103699== For lists of detected and suppressed errors, rerun with: -s
==103699== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

Hope it's helpful!

Impact

This vulnerabilities are capable of crashing software, Modify Memory, and possible remote execution.

CVE
CVE-2022-3178
(Published)

Vulnerability Type
CWE-126: Buffer Over-read

Severity
High (7.8)

Registry
Other

Affected Version
2.1-DEV-revUNKNOWN-master

Visibility
Public

Status
Fixed

Found by

janette88
@janette88
master

This report was seen 642 times.

We are processing your report and will contact the **gpac** team within 24 hours. 3 months ago

janette88 modified the report 3 months ago

We have contacted a member of the **gpac** team and are waiting to hear back 3 months ago

A **gpac/gpac** maintainer 3 months ago Maintainer

https://github.com/gpac/gpac/issues/2255

We have sent a follow up to the **gpac** team. We will try again in 7 days. 2 mo

A **gpac/gpac** maintainer validated this vulnerability 2 months ago

janette88 has been awarded the disclosure bounty ✔

The fix bounty is now up for grabs

The researcher's credibility has increased: +7

A gpac/gpac maintainer marked this as fixed in 2.1.0-DEV with commit 775107 2 months ago

The fix bounty has been dropped ✖

This vulnerability will not receive a CVE ✖

janette88 2 months ago Researcher

@admin can we get a CVE for this ?

Jamie Slome 2 months ago Admin

Happy to assign and publish once we get the go-ahead from the maintainer 👍

A gpac/gpac maintainer 2 months ago Maintainer

Yes. @Jamie please go ahead each time this is common action to take. Thanks

Jamie Slome 2 months ago Admin

Sorted :)

Sign in to join this conversation