#8845 closed defect (fixed)

Opened 2 years ago
Closed 2 years ago
Last modified 17 months ago

A stack-buffer-overflow in FFmpeg JIT code

Reported by:	Zhou Anshunkang	Owned by:	
Priority:	important	Component:	avcodec
Version:	git-master	Keywords:	aac
Cc:		Blocked By:	
Blocking:		Reproduced by developer:	no
Analyzed by developer:	no		

Description

System info

Ubuntu x86_64, clang 6.0, ffmpeg (git-master
https://github.com/FFmpeg/FFmpeg/commit/1ead176d874acb489827ace3935fc71e1eea7e0e)

Configure

```
./configure --cc=clang --cxx=clang++ --ld=clang --enable-debug --toolchain=clang-a
```

Command line

```
./ffmpeg -i @@ -y -f mov /dev/null
```

Address Sanitizer

```
  built with clang version 6.0.0-1ubuntu2 (tags/RELEASE_600/final)
  configuration: --cc=clang --cxx=clang++ --ld=clang --enable-debug --toolchain=cl
  libavutil      56. 58.100 / 56. 58.100
  libavcodec     58.100.100 / 58.100.100
  libavformat    58. 50.100 / 58. 50.100
  libavdevice    58. 11.101 / 58. 11.101
  libavfilter     7. 87.100 /  7. 87.100
  libswscale      5.  8.100 /  5.  8.100
  libswresample   3.  8.100 /  3.  8.100
[aac @ 0x61b000000080] Format aac detected only with low score of 1, misdetection
[aac @ 0x619000000580] More than one AAC RDB per ADTS frame is not implemented. Up
[aac @ 0x61b000000080] Packet corrupt (stream = 0, dts = NOPTS).
[aac @ 0x619000000580] Sample rate index in program config element does not match
[aac @ 0x619000000580] Inconsistent channel configuration.
[aac @ 0x619000000580] get_buffer() failed
[aac @ 0x61b000000080] decoding for stream 0 failed
[aac @ 0x61b000000080] Estimating duration from bitrate, this may be inaccurate
[aac @ 0x61b000000080] Could not find codec parameters for stream 0 (Audio: aac (M
Consider increasing the value for the 'analyzeduration' (0) and 'probesize' (50000
Input #0, aac, from '/home/seviezhou//experiment-3/AlphaFuzz-ffmpeg/alpha_out/cras
  Duration: 00:00:00.18, bitrate: 5 kb/s
    Stream #0:0: Audio: aac (Main), 13 channels (FL+FR+FLC+FRC+TC+TFL+TFC+TFR+TBL+
[aac @ 0x619000002380] Channel layout '16 channels (FL+FR+FLC+FRC+TC+TFL+TFC+TFR+T
Stream mapping:
  Stream #0:0 -> #0:0 (aac (native) -> aac (native))
Press [q] to stop, [?] for help
[aac @ 0x619000002380] More than one AAC RDB per ADTS frame is not implemented. Up
[aac @ 0x619000002380] channel element 3.7 is not allocated
Error while decoding stream #0:0: Invalid data found when processing input
[aac @ 0x619000002380] Sample rate index in program config element does not match
=================================================================
==48056==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffffe2fba5
READ of size 1 at 0x7ffffe2fba5 thread T0
    #0 0x29497f0  (/home/seviezhou/ffmpeg/ffmpeg+0x29497f0)
    #1 0x2959670  (/home/seviezhou/ffmpeg/ffmpeg+0x2959670)
    #2 0x2941d83  (/home/seviezhou/ffmpeg/ffmpeg+0x2941d83)
    #3 0x126e91c  (/home/seviezhou/ffmpeg/ffmpeg+0x126e91c)
    #4 0x126e22d  (/home/seviezhou/ffmpeg/ffmpeg+0x126e22d)
    #5 0x567dae  (/home/seviezhou/ffmpeg/ffmpeg+0x567dae)
    #6 0x560488  (/home/seviezhou/ffmpeg/ffmpeg+0x560488)
    #7 0x555ea5  (/home/seviezhou/ffmpeg/ffmpeg+0x555ea5)
    #8 0x7f1571274b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../c
    #9 0x41dea9 in _init (/home/seviezhou/ffmpeg/ffmpeg+0x41dea9)

Address 0x7ffffe2fba5 is located in stack of thread T0 at offset 165 in frame
    #0 0x294179f  (/home/seviezhou/ffmpeg/ffmpeg+0x294179f)

  This frame has 4 object(s):
    [32, 64) 'gb.i' (line 1113)
    [96, 128) 'gb' (line 3413)
    [160, 164) 'new_extradata_size' (line 3417) <== Memory access at offset 165 ov
    [176, 180) 'jp_dualmono_size' (line 3421)
HINT: this may be a false positive if your program uses some custom stack unwind m
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/home/seviezhou/ffmpeg/ffmpeg+0x
Shadow bytes around the buggy address:
  0x10007e3bdf20: f2 f2 f2 f2 f2 f2 f2 f2 00 00 00 00 00 00 00 00
  0x10007e3bdf30: f2 f2 f2 f2 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e3bdf40: 00 00 00 00 00 00 00 00 00 00 00 00 f3 f3 f3 f3
  0x10007e3bdf50: f3 f3 f3 f3 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e3bdf60: f1 f1 f1 f1 f8 f8 f8 f8 f2 f2 f2 f2 00 00 00 00
=>0x10007e3bdf70: f2 f2 f2 f2[04]f2 04 f3 00 00 00 00 00 00 00 00
  0x10007e3bdf80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e3bdf90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e3bdfa0: 00 00 00 00 f1 f1 f1 f1 04 f2 f8 f3 00 00 00 00
  0x10007e3bdfb0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x10007e3bdfc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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
==48056==ABORTING
```

It seems that if I ran this command multiple times, it can crash different objects, here is another ASAN output, using the same input:

```
  built with clang version 6.0.0-1ubuntu2 (tags/RELEASE_600/final)
  configuration: --cc=clang --cxx=clang++ --ld=clang --enable-debug --toolchain=cl
  libavutil      56. 58.100 / 56. 58.100
  libavcodec     58.100.100 / 58.100.100
  libavformat    58. 50.100 / 58. 50.100
  libavdevice    58. 11.101 / 58. 11.101
  libavfilter     7. 87.100 /  7. 87.100
  libswscale      5.  8.100 /  5.  8.100
  libswresample   3.  8.100 /  3.  8.100
[aac @ 0x61b000000080] Format aac detected only with low score of 1, misdetection
[aac @ 0x619000000580] More than one AAC RDB per ADTS frame is not implemented. Up
[aac @ 0x61b000000080] Packet corrupt (stream = 0, dts = NOPTS).
[aac @ 0x619000000580] Sample rate index in program config element does not match
=================================================================
==73492==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7ffe338744d7
READ of size 1 at 0x7ffe338744d7 thread T0
    #0 0x29497f0  (/home/seviezhou/ffmpeg/ffmpeg+0x29497f0)
    #1 0x2959670  (/home/seviezhou/ffmpeg/ffmpeg+0x2959670)
    #2 0x2941d83  (/home/seviezhou/ffmpeg/ffmpeg+0x2941d83)
    #3 0x126e91c  (/home/seviezhou/ffmpeg/ffmpeg+0x126e91c)
    #4 0x126e22d  (/home/seviezhou/ffmpeg/ffmpeg+0x126e22d)
    #5 0xfbf83f  (/home/seviezhou/ffmpeg/ffmpeg+0xfbf83f)
    #6 0xfb6db4  (/home/seviezhou/ffmpeg/ffmpeg+0xfb6db4)
    #7 0x51815a  (/home/seviezhou/ffmpeg/ffmpeg+0x51815a)
    #8 0x516cca  (/home/seviezhou/ffmpeg/ffmpeg+0x516cca)
    #9 0x5166f5  (/home/seviezhou/ffmpeg/ffmpeg+0x5166f5)
    #10 0x555cef  (/home/seviezhou/ffmpeg/ffmpeg+0x555cef)
    #11 0x7f0651f26b96 in __libc_start_main /build/glibc-OTsEL5/glibc-2.27/csu/../
    #12 0x41dea9 in _init (/home/seviezhou/ffmpeg/ffmpeg+0x41dea9)

Address 0x7ffe338744d7 is located in stack of thread T0 at offset 2039 in frame
    #0 0x294596f  (/home/seviezhou/ffmpeg/ffmpeg+0x294596f)

  This frame has 26 object(s):
    [32, 37) '.compoundliteral.sroa.5.i.i' (line 199)
    [64, 69) '.compoundliteral11.sroa.5.i.i' (line 199)
    [96, 101) '.compoundliteral22.sroa.5.i.i' (line 199)
    [128, 1152) 'e2c_vec.i' (line 263)
    [1280, 1285) '.compoundliteral.sroa.5.i' (line 260)
    [1312, 1317) '.compoundliteral60.sroa.5.i' (line 260)
    [1344, 1349) '.compoundliteral80.sroa.5.i' (line 260)
    [1376, 1381) '.compoundliteral99.sroa.5.i' (line 260)
    [1408, 1413) '.compoundliteral118.sroa.5.i' (line 260)
    [1440, 1445) '.compoundliteral133.sroa.5.i' (line 260)
    [1472, 1477) '.compoundliteral156.sroa.5.i' (line 260)
    [1504, 1509) '.compoundliteral176.sroa.5.i' (line 260)
    [1536, 1541) '.compoundliteral193.sroa.5.i' (line 260)
    [1568, 1584) 'SWAP_tmp.i' (line 438)
    [1600, 1616) 'SWAP_tmp219.i' (line 439)
    [1632, 1648) 'SWAP_tmp224.i' (line 440)
    [1664, 1680) 'SWAP_tmp229.i' (line 441)
    [1696, 1712) 'SWAP_tmp234.i' (line 442)
    [1728, 1744) 'SWAP_tmp239.i' (line 443)
    [1760, 1776) 'SWAP_tmp244.i' (line 444)
    [1792, 1808) 'SWAP_tmp249.i' (line 445)
    [1824, 1840) 'SWAP_tmp254.i' (line 446)
    [1856, 1872) 'SWAP_tmp270.i' (line 454)
    [1888, 1892) 'channels' (line 514)
    [1904, 2016) 'id_map' (line 516)
    [2048, 2055) 'type_counts' (line 517) <== Memory access at offset 2039 underfl
HINT: this may be a false positive if your program uses some custom stack unwind m
      (longjmp and C++ exceptions *are* supported)
SUMMARY: AddressSanitizer: stack-buffer-overflow (/home/seviezhou/ffmpeg/ffmpeg+0x
Shadow bytes around the buggy address:
  0x100046706840: f8 f2 f2 f2 f8 f2 f2 f2 f8 f2 f2 f2 f8 f2 f2 f2
  0x100046706850: f8 f2 f2 f2 f8 f2 f2 f2 f8 f2 f2 f2 f8 f2 f2 f2
  0x100046706860: f8 f8 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2
  0x100046706870: f8 f8 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2
  0x100046706880: f8 f8 f2 f2 f8 f8 f2 f2 04 f2 00 00 00 00 00 00
=>0x100046706890: 00 00 00 00 00 00 00 f2 f2[f2]f2 07 f3 f3 f3
  0x1000467068a0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000467068b0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000467068c0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000467068d0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x1000467068e0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
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
==73492==ABORTING
```

After it crashes, it also seems to mess up the terminal.

Attachments (1)

- stack-overflow-ffmpeg-JIT(116 bytes) - added by Carl Eugen Hoyos 2 years ago.

Change History (6)

by Carl Eugen Hoyos, 2 years ago

Attachment: *stack-overflow-ffmpeg-JIT*added

comment:1 by Carl Eugen Hoyos, 2 years ago

Component: ffmpeg → avcodec
Keywords: aac added; asan removed
Priority: normal → important

valgrind shows a possible issue:

```
==17778== Invalid read of size 8
==17778==    at 0x4767E9: che_configure (aacdec_template.c:133)
==17778==    by 0x4767E9: output_configure.cold (aacdec_template.c:543)
==17778==    by 0xE889EC: aac_decode_frame_int.isra.0 (aacdec_template.c:3312)
==17778==    by 0xE8935C: aac_decode_frame (aacdec_template.c:3457)
==17778==    by 0x8B119F: decode_simple_internal (decode.c:342)
==17778==    by 0x8B119F: decode_simple_receive_frame (decode.c:538)
==17778==    by 0x8B119F: decode_receive_frame_internal (decode.c:556)
```

```
==17778==    by 0x8B1E4F: avcodec_send_packet (decode.c:614)
==17778==    by 0x4B85AC: decode (ffmpeg.c:2217)
==17778==    by 0x4B85AC: decode_audio (ffmpeg.c:2274)
==17778==    by 0x4B85AC: process_input_packet (ffmpeg.c:2596)
==17778==    by 0x4BB31A: process_input (ffmpeg.c:4493)
==17778==    by 0x4BB31A: transcode_step (ffmpeg.c:4613)
==17778==    by 0x4BB31A: transcode (ffmpeg.c:4667)
==17778==    by 0x49838D: main (ffmpeg.c:4872)
```

comment:2 by Zhou Anshunkang, 2 years ago

I am not surely if these three bug reports point to the same location. I think it is a little bit strange.

comment:3 by jeeb, 2 years ago

For the record I posted a patch set that would improve the sanity checks for 22.2 so that it is not as easy to get handled as such on the 18th, but so far have received no reviews:

https://patchwork.ffmpeg.org/project/ffmpeg/list/?series=2055

This causes both of the fuzzing samples I have received to no longer be an issue (with both valgrind and clang 10 ASAN), while it still enables valid 22.2 content to decode properly.

comment:4 by Carl Eugen Hoyos, 2 years ago

Resolution: → fixed
Status: new → closed

Fixed by Jan Ekström in d6f293353c94c7ce200f6e0975ae3de49787f91f

comment:5 by Carl Eugen Hoyos, 17 months ago

Since the information in the CVE report is wrong:
This issue is neither reproducible with FFmpeg 4.3 nor earlier versions.

Note: See TracTickets for help on using tickets.