#8284 closed defect (fixed)

Opened 3 years ago
Closed 3 years ago

memory leaks in fifo_alloc_common()

Reported by:	Suhwan	Owned by:	
Priority:	important	Component:	avformat
Version:	git-master	Keywords:	
Cc:		Blocked By:	
Blocking:		Reproduced by developer:	no
Analyzed by developer:	no		

Description

Summary of the bug:
There are memory leaks in fifo_alloc_common()
How to reproduce:

```
%ffmpeg_g -stream_loop 23 -y -i $PoC1 -i $PoC2 -target svcd -loglevel 0 -psnr -c c

ffmpeg version N-95389-gdd01947397 Copyright (c) 2000-2019 the FFmpeg developers
built with clang version 6.0.0-1ubuntu2 (tags/RELEASE_600/final)
configuration: --cc=clang --cxx=clang++ --ld=clang --enable-debug
```

Here's Valgrind log

```
==48887== HEAP SUMMARY:
==48887==     in use at exit: 88 bytes in 3 blocks
==48887==   total heap usage: 667 allocs, 664 frees, 680,197 bytes allocated
==48887==
==48887== 56 (40 direct, 16 indirect) bytes in 1 blocks are definitely lost in los
==48887==    at 0x9D3CE76: memalign (in /usr/lib/valgrind/vgpreload_memcheck-amd64
==48887==    by 0x9D3CF91: posix_memalign (in /usr/lib/valgrind/vgpreload_memcheck
==48887==    by 0x590EC79: av_malloc (mem.c:87)
==48887==    by 0x590EC79: av_mallocz (mem.c:238)
==48887==    by 0x58DA3AC: fifo_alloc_common (fifo.c:32)
==48887==    by 0x175B433: mpeg_mux_init (mpegenc.c:461)
==48887==    by 0x17BEA4D: avformat_write_header (mux.c:521)
==48887==    by 0x4AB6CF: check_init_output_file (ffmpeg.c:2973)
==48887==    by 0x4A830E: init_output_stream (ffmpeg.c:3631)
==48887==    by 0x492E66: transcode_init (ffmpeg.c:3700)
==48887==    by 0x48A1A5: transcode (ffmpeg.c:4653)
==48887==    by 0x487D53: main (ffmpeg.c:4884)
==48887==
==48887== LEAK SUMMARY:
==48887==    definitely lost: 40 bytes in 1 blocks
==48887==    indirectly lost: 16 bytes in 1 blocks
==48887==      possibly lost: 0 bytes in 0 blocks
==48887==    still reachable: 32 bytes in 1 blocks
==48887==         suppressed: 0 bytes in 0 blocks
==48887== Reachable blocks (those to which a pointer was found) are not shown.
==48887== To see them, rerun with: --leak-check=full --show-leak-kinds=all
==48887==
==48887== For counts of detected and suppressed errors, rerun with: -v
==48887== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 0 from 0)
```

ASAN log

```
=================================================================
==34325==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 40 byte(s) in 1 object(s) allocated from:
    #0 0x4de9e8 in posix_memalign (ffmpeg_asan+0x4de9e8)
    #1 0x8594168 in av_malloc ffmpeg/libavutil/mem.c:87:9
    #2 0x8594168 in av_mallocz ffmpeg/libavutil/mem.c:238
    #3 0x852e42d in fifo_alloc_common ffmpeg/libavutil/fifo.c:32:9

Indirect leak of 16 byte(s) in 1 object(s) allocated from:
    #0 0x4de9e8 in posix_memalign (ffmpeg_asan+0x4de9e8)
    #1 0x8593011 in av_malloc ffmpeg/libavutil/mem.c:87:9
    #2 0x852e3f6 in av_fifo_alloc ffmpeg/libavutil/fifo.c:45:20

SUMMARY: AddressSanitizer: 56 byte(s) leaked in 2 allocation(s).
```

Please confirm.
Thanks

Attachments (2)

- PoC_1.tga(3.5 KB) - added by Suhwan 3 years ago.
  poc1
- PoC_2.flac(440.4 KB) - added by Suhwan 3 years ago.
  poc2

Change History (3)

by Suhwan, 3 years ago

Attachment: *PoC_1.tga*added

poc1

by Suhwan, 3 years ago

Attachment: *PoC_2.flac*added

poc2

comment:1 by mkver, 3 years ago

Component: undetermined → avformat
Resolution: → fixed
Status: new → closed

Fixed in b288a7eb3d963a175e177b6219c8271076ee8590.

Note: See TracTickets for help on using tickets.