New issue

Jump to bottom

Memory-leak and heap-overflow bugs in Bento4 #776

Open DylanSec opened this issue on Sep 25 · 0 comments

DylanSec commented on Sep 25 • edited

Summary

Hi, developers of Bento4:
I tested the binary mp4edit and mp42hevc with my fuzzer, and three crashes incurred, including two memory-leaks from mp4edit and a heap-overflow from mp42hevc. And I think Bug1 and Bug2 are different. The following is the details.

Bug1

Detected memory leaks in mp4edit.

```
root@25467sd2gsg311:/fuzz-mp4edit/mp4edit# ./mp4edit poc_mp4edit_111062493 /dev/null
WARNING: atom serialized to fewer bytes than declared size
WARNING: atom serialized to fewer bytes than declared size

=================================================================
==1561403==ERROR: LeakSanitizer: detected memory leaks

Direct leak of 88 byte(s) in 1 object(s) allocated from:
    #0 0x8eaf60 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fb56f9ad297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x45f83f in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x45f83f)
    #3 0x55da45 in AP4_Processor::Process(AP4_ByteStream&, AP4_ByteStream&, AP4_ByteStream*,
AP4_Processor::ProgressListener*, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x55da45)
    #4 0x413a42 in main (/fuzz-mp4edit/mp4edit/mp4edit+0x413a42)
    #5 0x7fb56f332c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: 88 byte(s) leaked in 1 allocation(s).
```

Bug2

Another memory-leak-bug in mp4edit.

```
root@25467sd2gsg311:/fuzz-mp4edit/mp4edit# ./mp4edit ../out/crashes/poc_mp4edit_285234531
/dev/null

=================================================================
==2508445==ERROR: LeakSanitizer: detected memory leaks

Indirect leak of 3380 byte(s) in 6 object(s) allocated from:
    #0 0x8eaf60 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fe0fef0e297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x46ae44 in AP4_AvccAtom::Create(unsigned int, AP4_ByteStream&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x46ae44)
    #3 0x45540f in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x45540f)
    #4 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #5 0x48de17 in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&, unsigned
long long) (/fuzz-mp4edit/mp4edit/mp4edit+0x48de17)
    #6 0x5d7069 in AP4_SampleEntry::Read(AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x5d7069)
    #7 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #8 0x62020e in AP4_StsdAtom::AP4_StsdAtom(unsigned int, unsigned char, unsigned int,
AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x62020e)
    #9 0x61f694 in AP4_StsdAtom::Create(unsigned int, AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x61f694)
    #10 0x4546d3 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4546d3)
    #11 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #12 0x48de17 in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&, unsigned
long long) (/fuzz-mp4edit/mp4edit/mp4edit+0x48de17)
    #13 0x48d616 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x48d616)
    #14 0x45cb77 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x45cb77)
    #15 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #16 0x45f83f in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x45f83f)
    #17 0x55da45 in AP4_Processor::Process(AP4_ByteStream&, AP4_ByteStream&, AP4_ByteStream*,
AP4_Processor::ProgressListener*, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x55da45)
    #18 0x413a42 in main (/fuzz-mp4edit/mp4edit/mp4edit+0x413a42)
    #19 0x7fe0fe893c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

…… ……
Indirect leak of 1 byte(s) in 1 object(s) allocated from:
    #0 0x8eaf60 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
```

```
    #1 0x7fe0fef0e297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x5d6bbf in AP4_SampleEntry::Read(AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x5d6bbf)
    #3 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #4 0x62020e in AP4_StsdAtom::AP4_StsdAtom(unsigned int, unsigned char, unsigned int,
AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x62020e)
    #5 0x61f694 in AP4_StsdAtom::Create(unsigned int, AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x61f694)
    #6 0x4546d3 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4546d3)
    #7 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #8 0x48de17 in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&, unsigned
long long) (/fuzz-mp4edit/mp4edit/mp4edit+0x48de17)
    #9 0x48d616 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x48d616)
    #10 0x45cb77 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x45cb77)
    #11 0x4618ff in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) (/fuzz-mp4edit/mp4edit/mp4edit+0x4618ff)
    #12 0x45f83f in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&) (/fuzz-
mp4edit/mp4edit/mp4edit+0x45f83f)
    #13 0x55da45 in AP4_Processor::Process(AP4_ByteStream&, AP4_ByteStream&, AP4_ByteStream*,
AP4_Processor::ProgressListener*, AP4_AtomFactory&) (/fuzz-mp4edit/mp4edit/mp4edit+0x55da45)
    #14 0x413a42 in main (/fuzz-mp4edit/mp4edit/mp4edit+0x413a42)
    #15 0x7fe0fe893c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: 6486 byte(s) leaked in 58 allocation(s).
```

Bug3

Heap-buffer-overflow on address 0x6020000002d4 in mp42hevc.

```
root@2e47aa8b3277:/# ./Bento4/cmakebuild/mp42hevc POC_mp42hevc_8055240 /dev/null
Video Track:
  duration: 200 ms
  sample count: 6
=================================================================
==2354250==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000002d4 at pc
0x0000004fb753 bp 0x7fffdc3cf910 sp 0x7fffdc3cf908
READ of size 1 at 0x6020000002d4 thread T0
    #0 0x4fb752 in WriteSample(AP4_DataBuffer const&, AP4_DataBuffer&, unsigned int,
AP4_ByteStream*) (/Bento4/cmakebuild/mp42hevc+0x4fb752)
    #1 0x4f9a2d in main (/Bento4/cmakebuild/mp42hevc+0x4f9a2d)
    #2 0x7f79552d9c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #3 0x41d999 in _start (/Bento4/cmakebuild/mp42hevc+0x41d999)

0x6020000002d4 is located 0 bytes to the right of 4-byte region [0x6020000002d0,0x6020000002d4)
allocated by thread T0 here:
```

```
    #0 0x4f5c98 in operator new[](unsigned long) /llvm-project/compiler-
rt/lib/asan/asan_new_delete.cpp:102
    #1 0x501bd8 in AP4_DataBuffer::SetDataSize(unsigned int)
(/Bento4/cmakebuild/mp42hevc+0x501bd8)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/Bento4/cmakebuild/mp42hevc+0x4fb752) in
WriteSample(AP4_DataBuffer const&, AP4_DataBuffer&, unsigned int, AP4_ByteStream*)
Shadow bytes around the buggy address:
  0x0c047fff8000: fa fa 00 00 fa fa 00 00 fa fa 00 00 fa fa fd fd
  0x0c047fff8010: fa fa 04 fa fa fa fd fd fa fa 00 05 fa fa 01 fa
  0x0c047fff8020: fa fa fd fa fa fa fd fa fa fa 06 fa fa fa 00 fa
  0x0c047fff8030: fa fa fd fa fa fa 04 fa fa fa fd fd fa fa fd fa
  0x0c047fff8040: fa fa 01 fa fa fa fd fd fa fa fd fa fa fa fd fa
=>0x0c047fff8050: fa fa 06 fa fa fa 01 fa fa fa[04]fa fa fa fd fa
  0x0c047fff8060: fa fa fd fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff80a0: 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
==2354250==ABORTING
```

POC

Bug_1_POC.zip
Bug_2_POC.zip
Bug_3_POC.zip

Environment

Ubuntu 18.04.6 LTS (docker)
clang 12.0.1
clang++ 12.0.1
Bento4 master branch(5b7cc25) && Bento4 latest release version(1.6.0-639)

Credit

Xudong Cao (NCNIPC of China)
Yuhang Huang (NCNIPC of China)
Han Zheng (NCNIPC of China, Hexhive)

Thank you for your time!

DylanSec mentioned this issue on Oct 6

Some Memory leaks exist in mp4xx #792

Open

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant