New issue

Jump to bottom

Some heap-buffer-overflow bugs in Bento4 #773

Open DylanSec opened this issue on Sep 23 · 0 comments

DylanSec commented on Sep 23 • edited

Summary

Hello, I found three heap buffer overflow bugs in AP4_Atom::TypeFromString(char const*), AP4_BitReader::ReadBit() and AP4_BitReader::ReadBits(unsigned int). They come from mp4tag and mp4mux, respectively.

Bug1

Heap-buffer-overflow on address 0x602000000332 in mp4tag:

```
root@728d9sls452:/fuzz-mp4tag/mp4tag# ./mp4tag --remove 1 ../out/crashes/mp4tag_poc_1 /dev/null
=================================================================
==1647110==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000332 at pc
0x000000468f25 bp 0x7fff3510c600 sp 0x7fff3510c5f8
READ of size 1 at 0x602000000332 thread T0
    #0 0x468f24 in AP4_Atom::TypeFromString(char const*) (/fuzz-mp4tag/mp4tag/mp4tag+0x468f24)
    #1 0x755566 in AP4_MetaData::Entry::FindInIlst(AP4_ContainerAtom*) const (/fuzz-
mp4tag/mp4tag/mp4tag+0x755566)
    #2 0x75a3f2 in AP4_MetaData::Entry::RemoveFromFileIlst(AP4_File&, unsigned int) (/fuzz-
mp4tag/mp4tag/mp4tag+0x75a3f2)
    #3 0x42fc2c in RemoveTag(AP4_File*, AP4_String&, bool) (/fuzz-mp4tag/mp4tag/mp4tag+0x42fc2c)
    #4 0x418531 in main (/fuzz-mp4tag/mp4tag/mp4tag+0x418531)
    #5 0x7fdb89b2ec86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #6 0x407f09 in _start (/fuzz-mp4tag/mp4tag/mp4tag+0x407f09)

0x602000000332 is located 0 bytes to the right of 2-byte region [0x602000000330,0x602000000332)
allocated by thread T0 here:
    #0 0x996920 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fdb8a1a9297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x418531 in main (/fuzz-mp4tag/mp4tag/mp4tag+0x418531)
    #3 0x7fdb89b2ec86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
```

```
SUMMARY: AddressSanitizer: heap-buffer-overflow (/fuzz-mp4tag/mp4tag/mp4tag+0x468f24) in
AP4_Atom::TypeFromString(char const*)
Shadow bytes around the buggy address:
  0x0c047fff8010: fa fa fd fd fa fa 04 fa fa fa fd fd fa fa 00 05
  0x0c047fff8020: fa fa 01 fa fa fa 01 fa fa fa fd fa fa fa 03 fa
  0x0c047fff8030: fa fa fd fa fa fa 06 fa fa fa 00 fa fa fa fd fa
  0x0c047fff8040: fa fa 04 fa fa fa fd fd fa fa fd fa fa fa 01 fa
  0x0c047fff8050: fa fa fd fa fa fa 00 00 fa fa 05 fa fa fa 00 00
=>0x0c047fff8060: fa fa 02 fa fa fa[02]fa fa fa 05 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
  0x0c047fff80b0: 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
==1647110==ABORTING
```

Bug2

Heap-buffer-overflow on address 0x6020000000f8 in mp4mux (AP4_BitReader::ReadBits):

```
root@23iq42wasf35:/fuzz-mp4mux/mp4mux# \./mp4mux --track
h264:../out/crashes/id\:000045\,sig\:06\,src\:000002\,op\:int32\,pos\:33\,val\:\+0\,470985
/dev/null
=================================================================
==2473731==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x6020000000f8 at pc
0x000000649cb8 bp 0x7ffced185f90 sp 0x7ffced185f88
READ of size 1 at 0x6020000000f8 thread T0
    #0 0x649cb7 in AP4_BitReader::ReadBits(unsigned int) (/fuzz-mp4mux/mp4mux/mp4mux+0x649cb7)
    #1 0x4d6040 in ReadGolomb(AP4_BitReader&) (/fuzz-mp4mux/mp4mux/mp4mux+0x4d6040)
    #2 0x4d6ef9 in AP4_AvcFrameParser::ParsePPS(unsigned char const*, unsigned int,
```

```
AP4_AvcPictureParameterSet&) (/fuzz-mp4mux/mp4mux/mp4mux+0x4d6ef9)
    #3 0x4f01dd in AP4_AvcFrameParser::Feed(unsigned char const*, unsigned int,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4f01dd)
    #4 0x4ecbf1 in AP4_AvcFrameParser::Feed(void const*, unsigned int, unsigned int&,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4ecbf1)
    #5 0x4349a5 in main (/fuzz-mp4mux/mp4mux/mp4mux+0x4349a5)
    #6 0x7fb87db03c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #7 0x407df9 in _start (/fuzz-mp4mux/mp4mux/mp4mux+0x407df9)

0x6020000000f8 is located 0 bytes to the right of 8-byte region [0x6020000000f0,0x6020000000f8)
allocated by thread T0 here:
    #0 0xa84ba0 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fb87e17e297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x4f01dd in AP4_AvcFrameParser::Feed(unsigned char const*, unsigned int,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4f01dd)
    #3 0x4349a5 in main (/fuzz-mp4mux/mp4mux/mp4mux+0x4349a5)
    #4 0x7fb87db03c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/fuzz-mp4mux/mp4mux/mp4mux+0x649cb7) in
AP4_BitReader::ReadBits(unsigned int)
Shadow bytes around the buggy address:
  0x0c047fff7fc0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff8000: fa fa 00 00 fa fa 00 00 fa fa 00 00 fa fa fd fd
=>0x0c047fff8010: fa fa 00 03 fa fa 06 fa fa fa 06 fa fa fa 00[fa]
  0x0c047fff8020: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8060: 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
==2473731==ABORTING
```

Bug3

Heap-buffer-overflow on address 0x602000000158 in mp4mux (AP4_BitReader::ReadBit):

```
root@345sadsf12w332:/fuzz-mp4mux/mp4mux# ./mp4mux --track
h264:../out/crashes/id\:000001\,sig\:06\,src\:000002\,op\:flip1\,pos\:8\,10085 /dev/null
=================================================================
==1606856==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x602000000158 at pc
0x00000064a882 bp 0x7ffd08428400 sp 0x7ffd084283f8
READ of size 1 at 0x602000000158 thread T0
    #0 0x64a881 in AP4_BitReader::ReadBit() (/fuzz-mp4mux/mp4mux/mp4mux+0x64a881)
    #1 0x4d6456 in ReadGolomb(AP4_BitReader&) (/fuzz-mp4mux/mp4mux/mp4mux+0x4d6456)
    #2 0x4dcd9e in AP4_AvcFrameParser::ParseSliceHeader(unsigned char const*, unsigned int,
unsigned int, unsigned int, AP4_AvcSliceHeader&) (/fuzz-mp4mux/mp4mux/mp4mux+0x4dcd9e)
    #3 0x4ed906 in AP4_AvcFrameParser::Feed(unsigned char const*, unsigned int,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4ed906)
    #4 0x4ecbf1 in AP4_AvcFrameParser::Feed(void const*, unsigned int, unsigned int&,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4ecbf1)
    #5 0x4349a5 in main (/fuzz-mp4mux/mp4mux/mp4mux+0x4349a5)
    #6 0x7fd9e3df9c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #7 0x407df9 in _start (/fuzz-mp4mux/mp4mux/mp4mux+0x407df9)

0x602000000158 is located 0 bytes to the right of 8-byte region [0x602000000150,0x602000000158)
allocated by thread T0 here:
    #0 0xa84ba0 in malloc /llvm-project/compiler-rt/lib/asan/asan_malloc_linux.cpp:145
    #1 0x7fd9e4474297 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libstdc++.so.6+0x93297)
    #2 0x4ed906 in AP4_AvcFrameParser::Feed(unsigned char const*, unsigned int,
AP4_AvcFrameParser::AccessUnitInfo&, bool) (/fuzz-mp4mux/mp4mux/mp4mux+0x4ed906)
    #3 0x4349a5 in main (/fuzz-mp4mux/mp4mux/mp4mux+0x4349a5)
    #4 0x7fd9e3df9c86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)

SUMMARY: AddressSanitizer: heap-buffer-overflow (/fuzz-mp4mux/mp4mux/mp4mux+0x64a881) in
AP4_BitReader::ReadBit()
Shadow bytes around the buggy address:
  0x0c047fff7fd0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7fe0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff7ff0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c047fff8000: fa fa 00 00 fa fa 00 00 fa fa 00 00 fa fa fd fd
  0x0c047fff8010: fa fa 00 03 fa fa 06 fa fa fa fd fa fa fa fd fa
=>0x0c047fff8020: fa fa 06 fa fa fa 07 fa fa fa 00[fa]fa fa fa fa
  0x0c047fff8030: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8040: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8050: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c047fff8060: fa fa fa 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
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
  ==1606856==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 release version(1.6.0-639)

Credit

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

Thank you for your time!

Assignees

No one assigned

Labels

None vet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant