New issue

Jump to bottom

heap-use-after-free bug in mp42ts #793

Open **burymyname** opened this issue on Oct 10 · 0 comments

burymyname commented on Oct 10

Hello, developers of Bento4. I found a heap use after free bug in AP4_LinearReader::Advance(bool) with ASAN.
The following is the details.

Details

```
=================================================================
==32056==ERROR: AddressSanitizer: heap-use-after-free on address 0x604000001f98 at pc
0x56093865ee11 bp 0x7ffea5a93280 sp 0x7ffea5a93270
READ of size 8 at 0x604000001f98 thread T0
    #0 0x56093865ee10 in AP4_LinearReader::Advance(bool)
Bento4/Source/C++/Core/Ap4LinearReader.cpp:434
    #1 0x560938666716 in AP4_LinearReader::ReadNextSample(unsigned int, AP4_Sample&,
AP4_DataBuffer&) Bento4/Source/C++/Core/Ap4LinearReader.cpp:530
    #2 0x5609386402ea in ReadSample Bento4/Source/C++/Apps/Mp42Ts/Mp42Ts.cpp:181
    #3 0x56093863a518 in WriteSamples Bento4/Source/C++/Apps/Mp42Ts/Mp42Ts.cpp:306
    #4 0x56093863a518 in main Bento4/Source/C++/Apps/Mp42Ts/Mp42Ts.cpp:638
    #5 0x7f8ea7badc86 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21c86)
    #6 0x56093863f9d9 in _start (Bento4/mp42ts+0x3a9d9)

0x604000001f98 is located 8 bytes inside of 48-byte region [0x604000001f90,0x604000001fc0)
freed by thread T0 here:
    #0 0x7f8ea899d9c8 in operator delete(void*, unsigned long) (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0xe19c8)
    #1 0x56093865e49f in AP4_LinearReader::SampleBuffer::~SampleBuffer()
Bento4/Source/C++/Core/Ap4LinearReader.h:104
    #2 0x56093865e49f in AP4_LinearReader::Advance(bool)
Bento4/Source/C++/Core/Ap4LinearReader.cpp:462

previously allocated by thread T0 here:
    #0 0x7f8ea899c448 in operator new(unsigned long) (/usr/lib/x86_64-linux-
gnu/libasan.so.4+0xe0448)
    #1 0x56093865ddb9 in AP4_LinearReader::Advance(bool)
Bento4/Source/C++/Core/Ap4LinearReader.cpp:422

SUMMARY: AddressSanitizer: heap-use-after-free Bento4/Source/C++/Core/Ap4LinearReader.cpp:434 in
```

```
AP4_LinearReader::Advance(bool)
Shadow bytes around the buggy address:
  0x0c087fff83a0: fa fa fd fd fd fd fd fd fa fa fd fd fd fd fd fa
  0x0c087fff83b0: fa fa fd fd fd fd fd fd fa fa fd fd fd fd fd fa
  0x0c087fff83c0: fa fa fd fd fd fd fd fd fa fa fd fd fd fd fd fa
  0x0c087fff83d0: fa fa fd fd fd fd fd fd fa fa fd fd fd fd fd fa
  0x0c087fff83e0: fa fa fd fd fd fd fd fd fa fa fd fd fd fd fd fa
=>0x0c087fff83f0: fa fa fd[fd]fd fd fd fd fa fa fd fd fd fd fd fa
  0x0c087fff8400: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8410: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8420: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8430: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c087fff8440: 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
==32056==ABORTING
```

PoC

mp42ts_poc.zip

Verification Steps

```
git clone https://github.com/axiomatic-systems/Bento4
cd Bento4
mkdir check_build && cd check_build
cmake ../ -DCMAKE_C_COMPILER=clang -DCMAKE_CXX_COMPILER=clang++ -DCMAKE_C_FLAGS="-
fsanitize=address" -DCMAKE_CXX_FLAGS="-fsanitize=address" -DCMAKE_BUILD_TYPE=Release
make -j
./mp42ts poc /dev/null
```

Enviroment

- Ubuntu 18.04
- clang 10.01
- Bento4 master branch 4df7274e commit and version 1.6.0-639

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

1 participant