New issue

Jump to bottom

buffer overflow in AP4_NullTerminatedStringAtom #418

Open 5hadowblad3 opened this issue on Aug 9, 2019 · 0 comments

Assignees

Labels fuzzing

5hadowblad3 commented on Aug 9, 2019 • edited

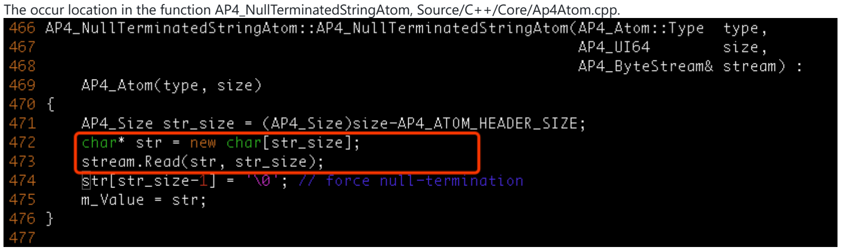
There is a buffer overflow in Ap4ElstAtom.cpp related to AP4_ElstAtom.

Distributor ID: Ubuntu
Description: Ubuntu 16.04.6 LTS
Release: 16.04
Codename: xenial
gcc: 5.4.0

To reproduce the bug,
compile the project with flag
DCMAKE_C_FLAGS=-g -m32 -fsanitize=address,undefined

then run:
./mp42aac input /dev/null

The occur location in the function AP4_NullTerminatedStringAtom, Source/C++/Core/Ap4Atom.cpp.



```
466 AP4_NullTerminatedStringAtom::AP4_NullTerminatedStringAtom(AP4_Atom::Type  type,
467                                                            AP4_UI64        size,
468                                                            AP4_ByteStream& stream) :
469     AP4_Atom(type, size)
470 {
471     AP4_Size str_size = (AP4_Size)size-AP4_ATOM_HEADER_SIZE;
472     char* str = new char[str_size];
473     stream.Read(str, str_size);
474     str[str_size-1] = '\0'; // force null-termination
475     m_Value = str;
476 }
477
```

Here is the trace reported by ASAN:

```
==10577==ERROR: AddressSanitizer: heap-buffer-overflow on address 0xf54006cf at pc 0x085d6d35 bp 0xffe49ac8 sp 0xffe49ab8
WRITE of size 1 at 0xf54006cf thread T0
#0 0x85d6d34 in AP4_NullTerminatedStringAtom::AP4_NullTerminatedStringAtom(unsigned int, unsigned long long, AP4_ByteStream&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4Atom.cpp:474
#1 0x82ccfbb in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned int, unsigned long long, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:529
#2 0x82fa1f7 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:225
#3 0x82fa1f7 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:151
#4 0x809a044 in AP4_File::ParseStream(AP4_ByteStream&, AP4_AtomFactory&, bool) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4File.cpp:104
#5 0x809a044 in AP4_File::AP4_File(AP4_ByteStream&, bool) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4File.cpp:78
#6 0x8082ce7 in main /mnt/data/playground/mp42-a/Source/C++/Apps/Mp42Aac/Mp42Aac.cpp:250
#7 0xf6a25636 in __libc_start_main (/lib/i386-linux-gnu/libc.so.6+0x18636)
#8 0x808df1b (/mnt/data/playground/mp42-patch/Build/mp42aac+0x808df1b)

0xf54006cf is located 1 bytes to the left of 1-byte region [0xf54006d0,0xf54006d1)
allocated by thread T0 here:
#0 0xf729ce46 in operator new[](unsigned int) (/usr/lib32/libasan.so.2+0x97e46)
#1 0x85d6657 in AP4_NullTerminatedStringAtom::AP4_NullTerminatedStringAtom(unsigned int, unsigned long long, AP4_ByteStream&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4Atom.cpp:472
#2 0x82ccfbb in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned int, unsigned long long, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:529
#3 0x82fa1f7 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:225
#4 0x82fa1f7 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4AtomFactory.cpp:151
#5 0x809a044 in AP4_File::ParseStream(AP4_ByteStream&, AP4_AtomFactory&, bool) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4File.cpp:104
#6 0x809a044 in AP4_File::AP4_File(AP4_ByteStream&, bool) /mnt/data/playground/mp42-a/Source/C++/Core/Ap4File.cpp:78
#7 0x8082ce7 in main /mnt/data/playground/mp42-a/Source/C++/Apps/Mp42Aac/Mp42Aac.cpp:250
#8 0xf6a25636 in __libc_start_main (/lib/i386-linux-gnu/libc.so.6+0x18636)
```

```
SUMMARY: AddressSanitizer: heap-buffer-overflow /mnt/data/playground/mp42-a/Source/C++/Core/Ap4Atom.cpp:474 AP4_NullTerminatedStringAtom::AP4_NullTerminatedStringAtom(unsigned int, unsigned long long, AP4_ByteStream&)
Shadow bytes around the buggy address:
0x3ea80080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea80090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea800a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea800b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea800c0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
=>0x3ea800d0: fa fa fa fa fa fa fa fa fa fa[fa]01 fa fa fa 00 04
0x3ea800e0: fa fa 00 04 fa fa 00 fa fa fa 00 04 fa fa 00 fa
0x3ea800f0: fa fa 00 04 fa fa 00 fa fa fa 00 04 fa fa 00 fa
0x3ea80100: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea80110: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
0x3ea80120: 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
Heap right redzone: fb
Freed heap region: fd
Stack left redzone: f1
Stack mid redzone: f2
Stack right redzone: f3
Stack partial redzone: f4
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
==10577==ABORTING
```

This is the POC input:
poc_inputs.zip

barbibulle self-assigned this on Aug 25, 2019

barbibulle added the fuzzing label on Aug 25, 2019

Assignees

barbibulle

Labels

fuzzing

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants