

New issue

[Jump to bottom](#)

Possible memory exhaustion in AP4_SgpdAtom::AP4_SgpdAtom(). The process has exhausted 65536MB memory. #712

Open 0xdd96 opened this issue on May 31 · 0 comments

Assignees



Labels

fuzzing

0xdd96 commented on May 31

Vulnerability description

version: Bento4-1.6.0-639

command: ./mp42aac \$POC /dev/null

Download: [poc](#)

Here is the trace reported by ASAN:

```
$ mp42aac poc /dev/null
```

```
AddressSanitizer: Out of memory. The process has exhausted 65536MB for size class 48.
```

```
=====
```

```
==29843==ERROR: AddressSanitizer: allocator is out of memory trying to allocate 0x18 bytes
```

```
#0 0x7ffff769b947 in operator new(unsigned long) (/lib/x86_64-linux-gnu/libasan.so.5+0x10f947)
```

```
#1 0x555555911f52 in AP4_List<AP4_DataBuffer>::Add(AP4_DataBuffer*)
```

```
/path_to_Bento4/Source/C++/Core/Ap4List.h:160
```

```
#2 0x5555559114bd in AP4_SgpdAtom::AP4_SgpdAtom(unsigned int, unsigned char, unsigned int, AP4_ByteStream&) /path_to_Bento4/Source/C++/Core/Ap4SgpdAtom.cpp:111
```

```
#3 0x555555910da4 in AP4_SgpdAtom::Create(unsigned int, AP4_ByteStream&)
```

```
/path_to_Bento4/Source/C++/Core/Ap4SgpdAtom.cpp:54
```

```
#4 0x55555589399c in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned int, unsigned long long, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:729
```

```
#5 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233
```

```

#6 0x5555558b9c5f in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&,
unsigned long long) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:194
#7 0x5555558b96c2 in AP4_ContainerAtom::AP4_ContainerAtom(unsigned int, unsigned long long,
bool, AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:139
#8 0x5555558b9229 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:88
#9 0x555555893d26 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int,
unsigned int, unsigned long long, AP4_Atom*&)
/path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:796
#10 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long
long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233
#11 0x5555558c7b47 in AP4_DrefAtom::AP4_DrefAtom(unsigned int, unsigned char, unsigned int,
AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4DrefAtom.cpp:84
#12 0x5555558c768b in AP4_DrefAtom::Create(unsigned int, AP4_ByteStream&, AP4_AtomFactory&)
/path_to_Bento4/Source/C++/Core/Ap4DrefAtom.cpp:50
#13 0x555555892ccd in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int,
unsigned int, unsigned long long, AP4_Atom*&)
/path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:560
#14 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long
long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233
#15 0x5555558b9c5f in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&,
unsigned long long) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:194
#16 0x5555558b96c2 in AP4_ContainerAtom::AP4_ContainerAtom(unsigned int, unsigned long long,
bool, AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:139
#17 0x5555558b9229 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:88
#18 0x555555893d26 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int,
unsigned int, unsigned long long, AP4_Atom*&)
/path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:796
#19 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long
long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233
#20 0x5555558b9c5f in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&,
unsigned long long) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:194
#21 0x5555558b96c2 in AP4_ContainerAtom::AP4_ContainerAtom(unsigned int, unsigned long long,
bool, AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:139
#22 0x5555558b9229 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:88
#23 0x555555893d26 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int,
unsigned int, unsigned long long, AP4_Atom*&)
/path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:796
#24 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long
long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233
#25 0x5555558b9c5f in AP4_ContainerAtom::ReadChildren(AP4_AtomFactory&, AP4_ByteStream&,
unsigned long long) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:194
#26 0x5555558b96c2 in AP4_ContainerAtom::AP4_ContainerAtom(unsigned int, unsigned long long,
bool, AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:139
#27 0x5555558b9229 in AP4_ContainerAtom::Create(unsigned int, unsigned long long, bool, bool,
AP4_ByteStream&, AP4_AtomFactory&) /path_to_Bento4/Source/C++/Core/Ap4ContainerAtom.cpp:88
#28 0x555555893d26 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int,
unsigned int, unsigned long long, AP4_Atom*&)
/path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:796
#29 0x555555890224 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long
long&, AP4_Atom*&) /path_to_Bento4/Source/C++/Core/Ap4AtomFactory.cpp:233

```

==29843==HINT: if you don't care about these errors you may set allocator_may_return_null=1

SUMMARY: AddressSanitizer: out-of-memory (/lib/x86_64-linux-gnu/libasan.so.5+0x10f947) in operator

```
new(unsigned long)
==29843==ABORTING
```

Vulnerability analysis

[Bento4/Source/C++/Core/Ap4SgpdAtom.cpp](#)

Lines 89 to 114 in 0735fe8

```
89      AP4_UI32 entry_count = 0;
90      AP4_Result result = stream.ReadUI32(entry_count);
91      if (AP4_FAILED(result)) return;
92      bytes_available -= 4;
93
94      // read all entries
95      for (unsigned int i=0; i<entry_count; i++) {
96          AP4_UI32 description_length = m_DefaultLength;
97          if (m_Version == 0) {
98              // entry size unknown, read the whole thing
99              description_length = bytes_available;
100      } else {
```

```
pwndbg> p entry_count
$1 = 4278190081
pwndbg> p m_DefaultLength
$2 = 20
pwndbg> p m_Version
$3 = 1 '\001'
pwndbg> p bytes_available
$4 = 20
```

The possible cause of this issue is that a crafted input can set `entry_count` to a large value (4,278,190,081) in line 90. Such a long loop (line 95-114) will allocate a lot of memory in line 106 and line 111, which eventually exhausts the memory. Since the return value of `stream.Read` is not checked in line 109, the loop will not terminate at the end of the input file.

  **barbibulle** self-assigned this on Jun 4

  **barbibulle** added the **fuzzing** label on Jun 4

Assignees

 **barbibulle**

Labels

fuzzing

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants

