New issue

Jump to bottom

there are some vulnerabilities in binary mp4tag #770

Open **yuhanghuang** opened this issue on Sep 23 · 0 comments

yuhanghuang commented on Sep 23 • edited

Summary

Hello, I use my fuzzer to fuzz binary mp4tag , the three binary all crashede, and shows that allocator is out of memory trying to allocate 0xxxxxxx bytes. Then I use the crash input to test binary mpesplit and mp42hevc,and all crashed beacuse of same situation. The version of Bento4 is the latest commit5b7cc25 and the operation system is Ubuntu 18.04(docker). The following is the details.And the issue is different from #342. Beacuse I test the poc,and it didn't work.

Bug1

```
root@76fc65f1cc2f:/Bento4/build# ./mp4tag crash_1.mp4
=================================================================
==206601==ERROR: AddressSanitizer: allocator is out of memory trying to allocate 0xfffeffee bytes
    #0 0x4f4778 in operator new[](unsigned long) /llvm-project/compiler-
rt/lib/asan/asan_new_delete.cpp:102
    #1 0x532595 in AP4_DataBuffer::ReallocateBuffer(unsigned int)
/Bento4/Source/C++/Core/Ap4DataBuffer.cpp:210:28
    #2 0x532595 in AP4_DataBuffer::SetDataSize(unsigned int)
/Bento4/Source/C++/Core/Ap4DataBuffer.cpp:151:33

==206601==HINT: if you don't care about these errors you may set allocator_may_return_null=1
SUMMARY: AddressSanitizer: out-of-memory /llvm-project/compiler-
rt/lib/asan/asan_new_delete.cpp:102 in operator new[](unsigned long)
==206601==ABORTING
```

Bug2

```
root@76fc65f1cc2f:/Bento4/build# ./mp4tag crash_2.mp4
=================================================================
==233834==ERROR: AddressSanitizer: allocator is out of memory trying to allocate 0x1fffffff8 bytes
    #0 0x4f4618 in operator new(unsigned long) /llvm-project/compiler-
rt/lib/asan/asan_new_delete.cpp:99
    #1 0x537e3d in AP4_Array<AP4_ElstEntry>::EnsureCapacity(unsigned int)
/Bento4/Source/C++/Core/Ap4Array.h:172:25
    #2 0x537e3d in AP4_ElstAtom::AP4_ElstAtom(unsigned int, unsigned char, unsigned int,
AP4_ByteStream&) /Bento4/Source/C++/Core/Ap4ElstAtom.cpp:87:15
    #3 0x537b15 in AP4_ElstAtom::Create(unsigned int, AP4_ByteStream&)
/Bento4/Source/C++/Core/Ap4ElstAtom.cpp:51:16
    #4 0x50e244 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned int, unsigned
int, unsigned long long, AP4_Atom*&) /Bento4/Source/C++/Core/Ap4AtomFactory.cpp:590:20
    #5 0x50cfd4 in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, unsigned long long&,
AP4_Atom*&) /Bento4/Source/C++/Core/Ap4AtomFactory.cpp:234:14
    #6 0x50c7fe in AP4_AtomFactory::CreateAtomFromStream(AP4_ByteStream&, AP4_Atom*&)
/Bento4/Source/C++/Core/Ap4AtomFactory.cpp:154:12
    #7 0x53a50e in AP4_File::ParseStream(AP4_ByteStream&, AP4_AtomFactory&, bool)
/Bento4/Source/C++/Core/Ap4File.cpp:104:12
    #8 0x53a9ed in AP4_File::AP4_File(AP4_ByteStream&, bool)
/Bento4/Source/C++/Core/Ap4File.cpp:78:5
    #9 0x4f9403 in main /Bento4/Source/C++/Apps/Mp4Tag/Mp4Tag.cpp:821:20
    #10 0x7f0a40dd5c86 in __libc_start_main /build/glibc-CVJwZb/glibc-2.27/csu/../csu/libc-
start.c:310

==233834==HINT: if you don't care about these errors you may set allocator_may_return_null=1
SUMMARY: AddressSanitizer: out-of-memory /llvm-project/compiler-rt/lib/asan/asan_new_delete.cpp:99
in operator new(unsigned long)
==233834==ABORTING
```

Environment

clang 11.0.1
clang++ 11.0.1
version:master branch(commit5b7cc25)

Platform

```
$ uname -a
Linux kali 5.10.0-kali9-amd64 #1 SMP Debian 5.10.46-4kali1 (2021-08-09) x86_64 GNU/Linux
```

How to compile

```
export CC=clang
export CXX=clang++
export CFLAGS="-fsanitize=address -g"
export CXXFLAGS="-fsanitize=address -g"
mkdir cmakebuild
cd cmakebuild
cmake -DCMAKE_BUILD_TYPE=Release ..
make
```

POC

crash.zip

NOTE

I find the two bugs not only exist in latest branch but also exist in latest release version Bento4-1.6.0-639.

Credit

Yuhang Huang (NCNIPC of China)
Han Zheng (NCNIPC of China, Hexhive)
Yin li,Jiayuan Zhang(NCNIPC of China)

Thansk for your time!

No milestone

Development

No branches or pull requests

1 participant