New issue

Jump to bottom

SEGV exrheader in ImfMultiPartInputFile.cpp:579 #491

Closed **strongcourage** opened this issue on Jul 24, 2019 · 2 comments

Labels Bug

strongcourage commented on Jul 24, 2019

Hi,

I found a crash due to a heap buffer overflow bug on exrheader (the latest commit `9410823` on master).

PoC: https://github.com/strongcourage/PoCs/blob/master/openexr_9410823/PoC_hbo_chunkOffsetReconstruction
Command: exrheader $PoC

ASAN says:

```
==976==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x60200000ec98 at pc 0x7f12af53f92f bp 0x7ffc0b8eb5e0 sp 0x7ffc0b8eb5d0
READ of size 8 at 0x60200000ec98 thread T0
    #0 0x7f12af53f92e in Imf_2_3::MultiPartInputFile::Data::chunkOffsetReconstruction(Imf_2_3::IStream&, std::vector<Imf_2_3::InputPartData*,
std::allocator<Imf_2_3::InputPartData*> > const&) /home/dungnguyen/gueb-testing/openexr/OpenEXR/IlmImf/ImfMultiPartInputFile.cpp:579
    #1 0x7f12af53fee0 in Imf_2_3::MultiPartInputFile::Data::readChunkOffsetTables(bool) /home/dungnguyen/gueb-testing/openexr/OpenEXR/IlmImf/ImfMultiPartInputFile.cpp:759
    #2 0x7f12af54217f in Imf_2_3::MultiPartInputFile::initialize() /home/dungnguyen/gueb-testing/openexr/OpenEXR/IlmImf/ImfMultiPartInputFile.cpp:429
    #3 0x7f12af5446bb in Imf_2_3::MultiPartInputFile::MultiPartInputFile(char const*, int, bool) /home/dungnguyen/gueb-testing/openexr/OpenEXR/IlmImf/ImfMultiPartInputFile.cpp:136
    #4 0x4048d8 in printInfo(char const*) /home/dungnguyen/gueb-testing/openexr/OpenEXR/exrheader/main.cpp:290
    #5 0x4033eb in main /home/dungnguyen/gueb-testing/openexr/OpenEXR/exrheader/main.cpp:562
    #6 0x7f12aea3982f in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x2082f)
    #7 0x403508 in _start (/home/dungnguyen/PoCs/openexr_9410823/exrheader-asan+0x403508)

0x60200000ec98 is located 0 bytes to the right of 8-byte region [0x60200000ec90,0x60200000ec98)
allocated by thread T0 here:
    #0 0x7f12afa61592 in operator new(unsigned long) (/usr/lib/x86_64-linux-gnu/libasan.so.2+0x99592)
    #1 0x7f12af5433de in __gnu_cxx::new_allocator<Imf_2_3::TileOffsets*>::allocate(unsigned long, void const*) /usr/include/c++/5/ext/new_allocator.h:104
    #2 0x7f12af5433de in std::allocator_traits<std::allocator<Imf_2_3::TileOffsets*> >::allocate(std::allocator<Imf_2_3::TileOffsets*>&, unsigned long)
/usr/include/c++/5/bits/alloc_traits.h:491
    #3 0x7f12af5433de in std::_Vector_base<Imf_2_3::TileOffsets*, std::allocator<Imf_2_3::TileOffsets*> >::_M_allocate(unsigned long) /usr/include/c++/5/bits/stl_vector.h:170
    #4 0x7f12af5433de in void std::vector<Imf_2_3::InputPartData*, std::allocator<Imf_2_3::InputPartData*> >::_M_emplace_back_aux<Imf_2_3::InputPartData*>
(Imf_2_3::InputPartData*&&) /usr/include/c++/5/bits/vector.tcc:412
    #5 0x7f12af5433de in void std::vector<Imf_2_3::InputPartData*, std::allocator<Imf_2_3::InputPartData*> >::emplace_back<Imf_2_3::InputPartData*>(Imf_2_3::InputPartData*&&)
/usr/include/c++/5/bits/vector.tcc:101
    #6 0x7f12af5433de in std::vector<Imf_2_3::InputPartData*, std::allocator<Imf_2_3::InputPartData*> >::push_back(Imf_2_3::InputPartData*&&)
/usr/include/c++/5/bits/stl_vector.h:932
    #7 0x7f12af5433de in Imf_2_3::MultiPartInputFile::initialize() /home/dungnguyen/gueb-testing/openexr/OpenEXR/IlmImf/ImfMultiPartInputFile.cpp:427
```

Thanks,
Manh Dung

peterhillman added a commit to peterhillman/openexr that referenced this issue on Jul 25, 2019

`fix off-by-one error in part number validation (Fixes AcademySoftware…` ddc5ae0

kdt3rd added a commit to kdt3rd/openexr that referenced this issue on Jul 25, 2019

`Fix AcademySoftwareFoundation#491, issue with part number range check…` 4c9c7c7

kdt3rd closed this as completed in `8b5370c` on Jul 25, 2019

kdt3rd added the Bug label on Jul 25, 2019

carnil commented on Dec 10, 2020

CVE-2020-16587 seems to have been assigned for this issue.

theta682 commented on Dec 13, 2020 Contributor

Please, communicate with NVD (https://nvd.nist.gov/info) and update the applicable version. As I understand it was fixed in 2.4.0.

DominicJacksonBFX pushed a commit to boris-fx/mocha-openexr that referenced this issue on Jun 22

`Fix AcademySoftwareFoundation#491, issue with part number range check…` 2ccb41f

Assignees

No one assigned

Labels

Bug

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

4 participants