Bug: SEGV on unknown address still exists in Assimp::XFileImporter::CreateMeshes #4662

Open 0xdd96 opened this issue on Jul 26 · 1 comment

Labels Bug

0xdd96 commented on Jul 26 • edited

Describe the bug

SEGV on unknown address still exists in Assimp::XFileImporter::CreateMeshes.

This is similar to issue #1728. Note that #1728 reported wrong type of the vulnerability, as it is not a NULL pointer dereference. Patch `39ce3e1` *was misguided by #1728, leaving this vulnerability unfixed.*

To Reproduce

Steps to reproduce the behavior:
version: latest commit `3c253ca`
poc:null_CreateMeshes.zip

```
git clone https://github.com/assimp/assimp.git
cd assimp
mkdir build
cd build
CFLAGS="-g -O0" CXXFLAGS="-g -O0" cmake -G "Unix Makefiles" -DBUILD_SHARED_LIBS=OFF  -
DASSIMP_BUILD_ASSIMP_TOOLS=ON  ..
./assimp info $POC
```

Expected behavior

```
user@c3ae4d510abb:$ ./bin/assimp info poc
Launching asset import ...           OK
Validating postprocessing flags ...  OK
0 %
Segmentation fault (core dumped)
```

```
user@c3ae4d510abb:$ ./bin/assimp info poc
Launching asset import ...           OK
Validating postprocessing flags ...  OK
0 %
AddressSanitizer:DEADLYSIGNAL
=================================================================
==20088==ERROR: AddressSanitizer: SEGV on unknown address 0x6120000301c0 (pc 0x555556872ed9 bp
0x7fffffffb4d0 sp 0x7fffffffb100 T0)
==20088==The signal is caused by a READ memory access.
    #0 0x555556872ed8  (bin/assimp+0x131eed8)
    #1 0x55555687151a  (bin/assimp+0x131d51a)
    #2 0x5555568716a0  (bin/assimp+0x131d6a0)
    #3 0x555556870ba0  (bin/assimp+0x131cba0)
    #4 0x555556870829  (bin/assimp+0x131c829)
    #5 0x555555c56ab5  (bin/assimp+0x702ab5)
    #6 0x55555580ecf2  (bin/assimp+0x2bacf2)
    #7 0x5555557f89af  (bin/assimp+0x2a49af)
    #8 0x5555557f5f42  (bin/assimp+0x2a1f42)
    #9 0x555555801399  (bin/assimp+0x2ad399)
    #10 0x5555557f59c8  (bin/assimp+0x2a19c8)
    #11 0x7ffff7070082  (/lib/x86_64-linux-gnu/libc.so.6+0x24082)
    #12 0x5555557cda7d  (bin/assimp+0x279a7d)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (bin/assimp+0x131eed8)
==20088==ABORTING
Aborted
```

Vulnerability analysis

Using gdb to trace this PoC, the vulnerability occurs in line 340 of XFileImporter.cpp, due to `idx=16256` is larger than the capacity of `sourceMesh->mNormals` (24).

assimp/code/AssetLib/X/XFileImporter.cpp
Lines 337 to 342 in `3c253ca`

```
337     if ( mesh->HasNormals() ) {
338         if ( sourceMesh->mNormFaces[ f ].mIndices.size() > d ) {
339             const size_t idx( sourceMesh->mNormFaces[ f ].mIndices[ d ] );
340             mesh->mNormals[ newIndex ] = sourceMesh->mNormals[ idx ];
341         }
342     }
```

After tracing it, I found that `pMesh->mNormals` assigned `numNormals` elements in line 514-519 of `XFileParser.cpp`, then line 535-536 saved the result of `ReadInt` to `pMesh->mNormFaces[a].mIndices` without checking if it is in the correct boundary (`<numNormals`). This eventually leads to the bug above.

assimp/code/AssetLib/X/XFileParser.cpp
Lines 513 to 541 in 3c253ca

```
513     unsigned int numNormals = ReadInt();
514     pMesh->mNormals.resize(numNormals);
515 
516     // read normal vectors
517     for (unsigned int a = 0; a < numNormals; ++a) {
518         pMesh->mNormals[a] = ReadVector3();
519     }
520 
521     // read normal indices
522     unsigned int numFaces = ReadInt();
523     if (numFaces != pMesh->mPosFaces.size()) {
524         ThrowException("Normal face count does not match vertex face count.");
```

Suggested fix

Add a boundary check after `ReadInt` following the convention in line 410 below. Line 410 ensures the number read by `ReadInt` does not exceed the size of the vector.

assimp/code/AssetLib/X/XFileParser.cpp
Lines 394 to 415 in 3c253ca

```
394     unsigned int numVertices = ReadInt();
395     pMesh->mPositions.resize(numVertices);
396 
397     // read vertices
398     for (unsigned int a = 0; a < numVertices; a++)
399         pMesh->mPositions[a] = ReadVector3();
400 
401     // read position faces
402     unsigned int numPosFaces = ReadInt();
403     pMesh->mPosFaces.resize(numPosFaces);
404     for (unsigned int a = 0; a < numPosFaces; ++a) {
405         // read indices
```

0xdd96 added the **Bug** label on Jul 26

krop commented on Sep 7

CVE-2022-38528 was published yesterday and references this bug report.

Assignees

No one assigned

Labels

Bug

Projects

@kimkulling's backlog

Status: NEW New +2 more

1 closed project ▾

Milestone

No milestone

Development

No branches or pull requests

2 participants