Talos Vulnerability Report

TALOS-2020-1104

Pixar OpenUSD binary file format offset seek information leak vulnerability

NOVEMBER 12, 2020

CVE NUMBER

CVE-2020-9973

Summary

An exploitable vulnerability exists in the way Pixar OpenUSD 20.05 handles file offsets in binary USD files. A specially crafted malformed file can trigger an arbitrary out-of-bounds memory access that could lead to the disclosure of sensitive information. This vulnerability could be used to bypass mitigations and aid additional exploitation. To trigger this vulnerability, the victim needs to access an attacker-provided file.

Tested Versions

Pixar OpenUSD 20.05
Apple macOS Catalina 10.15.3

Product URLs

https://openusd.org

CVSSv3 Score

4.3 - CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:N/A:N

CWE

CWE-119 - Improper Restriction of Operations within the Bounds of a Memory Buffer

Details

OpenUSD stands for “Open Universal Scene Descriptor” and is a software suite by Pixar that facilitates, among other things, interchange of arbitrary 3-D scenes that may be composed of many elemental assets.

Most notably, USD and its backing file format `usd` are used on Apple iOS and macOS as part of ModelIO framework in support of SceneKit and ARKit for sharing and displaying 3D scenes in, for example, augmented reality applications. On macOS, these files are automatically rendered to generate thumbnails, while on iOS they can be shared via iMessage and opened with user interaction.

USD binary file format consists of a header pointing to a table of contents that in turn points to individual sections that comprise the whole file. Pointer to a table of contents, as well as back to individual sections from the table of contents, is represented as a 64 bit integer specifying a file offset where the table of contents, or specific section, can be found.

For example, table of contents is read using the following code:

```
template <class Reader>
CrateFile::_TableOfContents
CrateFile::_ReadTOC(Reader reader, _BootStrap const &b) const
{
    reader.Seek(b.tocOffset);                                       [1]
    return reader.template Read<_TableOfContents>();                [2]
}
```

At [1] , table of contents offset read from the file header is used to seek into the file and then the parser proceeds to parse the table of contents at [2]. When dealing with regular files, `seek` operations usually just shift the current file pointer to the specified offset, however in most practical uses of OpenUSD (including macOS and iOS utilities and applications) USD file will first be memory mapped. When seek is executed, no check is performed to ensure the offset still falls inside the currently memory mapped file. Since the `b.tocOffset` is a 64 bit integer that is read directly from the mapped file, this can be used to access and read any address in process’ address space.

Similarly to above example, same issue can be triggered when trying to parse individual sections of the file. Following code that parses `FIELDS` section illustrates this:

```
template <class Reader>
void
CrateFile::_ReadFieldSets(Reader reader)
{
    TfAutoMallocTag tag("_ReadFieldSets");
    if (auto fieldSetsSection = _toc.GetSection(_FieldSetsSectionName)) {              [3]
        reader.Seek(fieldSetsSection->start);                                          [4]

        if (Version(_boot) < Version(0,4,0)) {                                         [5]
            _fieldSets = reader.template Read<decltype(_fieldSets)>();
        } else {
            // Compressed fieldSets in 0.4.0.
            auto numFieldSets = reader.template Read<uint64_t>();
            _fieldSets.resize(numFieldSets);
```

When parsing the `FIELDS` section, an offset from table of contents is retrieved at [3] and used in a call to `Seek` at 4. Then at [5] the parsing continues. Again, no check is performed to ensure that the offset falls inside the file, enabling arbitrary memory read. While only `FIELDS` section example is given here, other section offsets are vulnerable, too.

USD files are usually distributed as `usdz` archives which can contains multiple distinct `usd` files each referencing each other, with careful file layout , this sort of vulnerability could be abused to probe the memory layout and influence which files are successfully loaded. This could be abused to achieve information leak and defeat exploitation mitigations such as ASLR.

Crash Information

This vulnerability has been tested on latest version of macOS Catalina 10.15.3.

```
Current executable set to 'qlmanage' (x86_64).
(lldbinit) settings set -- target.run-args  "-t" "poc_toc_offset.usdc" "-o" "./"
(lldbinit) r
Process 59980 launched: '/usr/bin/qlmanage' (x86_64)
Testing Quick Look thumbnails with files:
    poc_toc_offset.usdc
2020-06-23 18:59:28.405728-0500 qlmanage[59980:11301935] [General] Disabling connection to window server
2020-06-23 18:59:28.406581-0500 qlmanage[59980:11301935] [General] Number of mach ports: 42
2020-06-23 18:59:28.408263-0500 qlmanage[59980:11301959] [General] Cache is disabled (real server: NO - cache enabled: NO)
2020-06-23 18:59:28.832447-0500 qlmanage[59980:11301959] MessageTracer: Falling back to default whitelist
-------------------------------------------------------------------------------------------------------------------------[regs]
  RAX: 0xF041414241603141  RBX: 0x00007000060A1350  RBP: 0x00007000060A1250  RSP: 0x00007000060A1210  o d I t s Z a P c
  RDI: 0x00007000060A1290  RSI: 0x00007000060A12B8  RDX: 0x0000000000000000  RCX: 0x0000000000000000  RIP: 0x00007FFF379A6F00
  R8:  0x0000000000000000  R9:  0x0000000000000000  R10: 0x0000000100000000  R11: 0x00006FFF05EB22B8  R12: 0x0000000105822A00
  R13: 0x00007000060A12B8  R14: 0x00007000060A1290  R15: 0x00007FFF8DFD9890
  CS:  002B  FS: 0000  GS: 0000
-------------------------------------------------------------------------------------------------------------------------[code]
@ /System/Library/Frameworks/ModelIO.framework/Versions/A/ModelIO:
->  0x7fff379a6f00 (0x8d5f00): 4c 8b 38        mov    r15, qword ptr [rax]
    0x7fff379a6f03 (0x8d5f03): 48 83 c0 08     add    rax, 0x8
    0x7fff379a6f07 (0x8d5f07): 49 89 45 08     mov    qword ptr [r13 + 0x8], rax
    0x7fff379a6f0b (0x8d5f0b): 4c 89 f7        mov    rdi, r14
    0x7fff379a6f0e (0x8d5f0e): 4c 89 fe        mov    rsi, r15
    0x7fff379a6f11 (0x8d5f11): e8 72 01 00 00  call   0x7fff379a7088 ; ___lldb_unnamed_symbol87486$$ModelIO
    0x7fff379a6f16 (0x8d5f16): 4d 8b 26        mov    r12, qword ptr [r14]
    0x7fff379a6f19 (0x8d5f19): 49 c1 e7 05     shl    r15, 0x5
-------------------------------------------------------------------------------------------------------------------------------

Process 59980 stopped
* thread #7, queue = 'quicklookd.serialgenerator', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
    frame #0: 0x00007fff379a6f00 ModelIO`___lldb_unnamed_symbol87485$$ModelIO + 122
Target 0: (qlmanage) stopped.
(lldbinit) bt
* thread #7, queue = 'quicklookd.serialgenerator', stop reason = EXC_BAD_ACCESS (code=EXC_I386_GPFLT)
  * frame #0: 0x00007fff379a6f00 ModelIO`___lldb_unnamed_symbol87485$$ModelIO + 122
    frame #1: 0x00007fff379944ed ModelIO`___lldb_unnamed_symbol87000$$ModelIO + 273
    frame #2: 0x00007fff379941c3 ModelIO`___lldb_unnamed_symbol86998$$ModelIO + 555
    frame #3: 0x00007fff37993e06 ModelIO`___lldb_unnamed_symbol86997$$ModelIO + 392
    frame #4: 0x00007fff37992ec9 ModelIO`___lldb_unnamed_symbol86984$$ModelIO + 447
    frame #5: 0x00007fff37a8bbf6 ModelIO`___lldb_unnamed_symbol91742$$ModelIO + 242
    frame #6: 0x00007fff37a8bac8 ModelIO`___lldb_unnamed_symbol91741$$ModelIO + 54
    frame #7: 0x00007fff37a39d2f ModelIO`___lldb_unnamed_symbol90729$$ModelIO + 287
```

Timeline

2020-07-01 - Vendor Disclosure (notified Pixar and Apple individually)

2020-07-01 - Vendor (Pixar) acknowledged report

2020-07-02 - Vendor (Apple) acknowledged report

2020-07-14 - Talos tested latest beta of macOS Catalina 10.15.6

2020-08-04 - Talos follow up with Pixar; no response

2020-09-16 - 2nd follow up with Pixar

2020-11-12 - Public Release

CREDIT

Discovered by Aleksandar Nikolic of Cisco Talos.
