GunP4ng / **README.md**

Created 3 days ago

Code Revisions 1

CVE-2025-70070

README.md

CVE-2025-70070

> Public reference for CVE-2025-70070 (Issue: NULL pointer / invalid dereference in Assimp FBX mesh Layer parsing).

Product / Affected Versions

Vendor: Assimp project Product: Assimp Open Asset Import Library Affected version(s): Assimp 6.0.2 as tested/reported Fixed version: TBD/Unknown Component(s): FBX importer, `MeshGeometry::MeshGeometry`, `code/AssetLib/FBX/FBXMeshGeometry.cpp`

Summary

Assimp's FBX mesh geometry parser processes `Layer` elements while constructing `MeshGeometry`. In the reported vulnerable path, the parser obtains the token list for a `Layer` element and dereferences `tokens[0]` without first checking that the token list is non-empty.

A crafted FBX file can include a `Layer` element with no tokens. When Assimp imports the file, the unchecked access can trigger a NULL pointer / invalid dereference and crash the importing process, resulting in denial of service.

Vulnerability Type / CWE

Type: NULL Pointer Dereference / Invalid Dereference CWE: CWE-476 (NULL Pointer Dereference)

Impact

Denial of Service: Yes (crafted FBX input can deterministically crash the importing process.) Attack requirements: The target application must import an attacker-supplied FBX file using Assimp.

High-level Verification / Reproduction (no weaponized details)

This issue is matched by the current local ASAN/gdb crash corpus in this workspace.

Verification setup:

- Binary: `~/working/sf/fuzzing/assimp/build-afl/fbx_fuzzer`
- Corpus directory: `~/working/sf/fuzzing/assimp/crashes/final`
- Representative minimized input: one of the files in `crashes/final`; all currently present files rematch to the same issue.

Observed gdb top frame for the current corpus:

```
Assimp::FBX::MeshGeometry::MeshGeometry()
code/AssetLib/FBX/FBXMeshGeometry.cpp:171
```

The local crash filenames and folders are misleading for some samples (`heap_bof`, `out-of-memory`, and similar names), but rematching showed that all seven files under the current `crashes/final` directory reproduce this `MeshGeometry::MeshGeometry` Layer-token dereference issue.

Technical Notes / Root Cause (for triage)

Location:

- File: `code/AssetLib/FBX/FBXMeshGeometry.cpp`
- Function: `Assimp::FBX::MeshGeometry::MeshGeometry`

Relevant vulnerable pattern:

```
for (ElementMap::const_iterator it = Layer.first; it != Layer.second; ++it) {
    const TokenList& tokens = (*it).second->Tokens();

    const char* err;
    const int index = ParseTokenAsInt(*tokens[0], err);
```

The code assumes that each `Layer` element has at least one token. If a crafted FBX supplies an empty token list, `tokens[0]` is accessed without validation and the process can crash.

Mitigation / Fix Guidance

- Check that the `Layer` token list is non-empty before accessing `tokens[0]`.
- Reject malformed `Layer` elements or use a safe default handling path.
- Add regression coverage for FBX files with empty `Layer` token lists.

Upstream Fix / References for Triage

Upstream fix commit: TBD/Unknown Upstream pull request: TBD/Unknown Release note status: TBD/Unknown

Credits

Discovered by: TaeYong LEE (GunP4ng)

References

- Assimp project: https://github.com/assimp/assimp