GunP4ng / **README.md**

Created 3 days ago

Code Revisions 1

CVE-2025-70071

README.md

CVE-2025-70071

Public reference for CVE-2025-70071 (Issue: uncontrolled memory allocation in Assimp FBX binary vector array 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, `ParseVectorDataArray` for `aiVector2D`, `code/AssetLib/FBX/FBXParser.cpp`

Summary

Assimp's FBX parser reads binary array metadata from FBX elements and uses the array count to size parsing buffers and output vectors. In the reported vulnerable path for vector data, a crafted FBX file can provide a very large binary array count. The parser trusts that count and attempts to allocate memory proportional to it.

This can cause an allocation-size-too-big abort or process termination during parsing, resulting in denial of service.

Vulnerability Type / CWE

Type: Uncontrolled Memory Allocation / Uncontrolled Resource Consumption CWE: CWE-400 (Uncontrolled Resource Consumption), CWE-770 (Allocation of Resources Without Limits or Throttling)

Impact

Denial of Service: Yes (crafted FBX input can cause excessive allocation or allocation-size-too-big termination.) Attack requirements: The target application must import an attacker-supplied FBX file using Assimp.

High-level Verification / Reproduction (no weaponized details)

High-level reproduction conditions:

1. Build Assimp 6.0.2 with the FBX importer enabled.
2. Import a crafted binary FBX containing a vector data array header with a very large count.
3. Reach `ParseVectorDataArray()` for `aiVector2D` data.
4. Observe the parser attempting to allocate memory based on the untrusted count, leading to allocation-size-too-big or process termination.

The current local `crashes/final` corpus in this workspace was rematched and does not contain the reproducer for this CVE; those local files reproduce CVE-2025-70070 instead. They should not be cited as evidence for CVE-2025-70071.

Technical Notes / Root Cause (for triage)

Location:

- File: `code/AssetLib/FBX/FBXParser.cpp`
- Function: `Assimp::FBX::ParseVectorDataArray()` for `aiVector2D`

Relevant behavior:

```
ReadBinaryDataArrayHead(data, end, type, count, el);
std::vector<char> buff;
ReadBinaryDataArray(type, count, data, end, buff, el);
```

```
const uint32_t count2 = count / 2;
out.reserve(count2);
```

The reported issue is that the binary array `count` is accepted from untrusted input and used to size allocations without a sufficient upper bound or consistency check.

Mitigation / Fix Guidance

- Bound binary array counts before allocation.
- Validate array counts against remaining input size and expected element width.
- Reject impossible or excessive vector data arrays early.
- Add regression coverage for huge binary vector array counts.

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