GunP4ng / **README.md**

Last active 3 days ago

Code Revisions 2

CVE-2025-70069

README.md

CVE-2025-70069

> Public reference for CVE-2025-70069 (Issue: uncontrolled memory allocation in Assimp FBX multi-material mesh conversion).

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, `FBXConverter::ConvertMeshMultiMaterial`, `code/AssetLib/FBX/FBXConverter.cpp`

Summary

Assimp's FBX importer converts multi-material meshes by grouping faces per material and allocating output mesh arrays for each group. In the reported vulnerable path, face index counts are summed into `count_vertices` without a practical validation cap before the value is used to size output arrays.

A crafted FBX file can provide oversized face index counts that inflate `count_vertices`, causing Assimp to attempt very large allocations while converting a mesh. This can terminate the importing process through out-of-memory or allocation-failure behavior, 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 force excessive allocation and terminate the importing process.) 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 FBX file whose face index counts are oversized for a multi-material mesh.
3. Reach `FBXConverter::ConvertMeshMultiMaterial`.
4. Observe that `count_vertices` is inflated and later used for mesh-array allocation, leading to excessive memory allocation 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-2025-70069.

Technical Notes / Root Cause (for triage)

Location:

- File: `code/AssetLib/FBX/FBXConverter.cpp`
- Function: `Assimp::FBX::FBXConverter::ConvertMeshMultiMaterial`

Relevant behavior:

```
std::vector<unsigned int>::const_iterator itf = faces.begin();
for (MatIndexArray::const_iterator it = mindices.begin(), end = mindices.end(
        it != end; ++it, ++itf) {
```

```
    if ((*it) != index) {
        continue;
    }
    ++count_faces;
    count_vertices += *itf;
}

out_mesh->mNumVertices = count_vertices;
out_mesh->mVertices = new aiVector3D[count_vertices];
```

The reported issue is that untrusted face index counts can drive `count_vertices` to an excessive value before allocation.

Mitigation / Fix Guidance

- Validate face index counts before summing them into allocation sizes.
- Enforce sane upper bounds for generated vertex and face counts.
- Reject malformed FBX meshes whose face counts exceed implementation limits or file-size-derived expectations.
- Add regression tests for oversized face index counts in multi-material meshes.

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