openscad / **openscad** Public

Code Issues 686 Pull requests 105 Actions Projects 4 Wiki •••

New issue Jump to bottom

Out-of-bounds memory access in comment parser (file without trailing newline) #4043

Closed **eldstal** opened this issue on Jan 9 · 2 comments

eldstal commented on Jan 9 Contributor

Summary

A .scad file with no trailing newline may cause an out-of-bounds read during parsing of annotations.

Vulnerable versions

- OpenSCAD (commit `374fa58`)

Steps to reproduce

1. Unzip the provided proof-of-concept file
2. `openscad --export-format stl -o /dev/null oobr_comment.scad`
3. Observe the segmentation fault

```
albin@KNYTT:openscad$ ./openscad.bin --export-format stl -o /dev/null
pocs_manual/oobr_comment.scad
Could not initialize localization.
UndefinedBehaviorSanitizer:DEADLYSIGNAL
==12619==ERROR: UndefinedBehaviorSanitizer: SEGV on unknown address 0x00000d384000 (pc
0x00000064254d bp 0x7ffe70e82d60 sp 0x7ffe70e82bf0 T12619)
==12619==The signal is caused by a READ memory access.
    #0 0x64254c in getComment(std::__cxx11::basic_string<char, std::char_traits<char>,
std::allocator<char> > const&, int) /home/albin/fuzz/openscad/openscad/src/comment.cc:95:9
    #1 0x6401b9 in CommentParser::collectParameters(std::__cxx11::basic_string<char,
std::char_traits<char>, std::allocator<char> > const&, SourceFile*)
/home/albin/fuzz/openscad/openscad/src/comment.cc:284:25
    #2 0x464526 in cmdline(CommandLine const&)
/home/albin/fuzz/openscad/openscad/src/openscad.cc:411:2
    #3 0x479f04 in main /home/albin/fuzz/openscad/openscad/src/openscad.cc:1230:12
    #4 0x7f39dec7e0b2 in __libc_start_main /build/glibc-eX1tMB/glibc-2.31/csu/../csu/libc-
start.c:308:16
    #5 0x4390cd in _start (/home/albin/fuzz/openscad/openscad/build/openscad+0x4390cd)
```

```
UndefinedBehaviorSanitizer can not provide additional info.
==12619==ABORTING
```

Cause

The seek through the `fulltext` string at comment.cc:95 only searches for `\n`, but ignores the potential end of the text. As a result, a file which ends without a `\n` character will cause this loop to read out of bounds.

Impact

It appears that the out-of-bounds data can only be read, and not written. An out-of-bounds read does not expose a security vulnerability on its own, but can be used to bypass automatic security features such as stack canaries and pointer encryption.

Proposed mitigation

Also check against `fulltext.size()` in the loop, to ensure that `end` does not exceed the length of the file.

eldstal mentioned this issue on Jan 9

Add file bounds check to comment parser, issue #4043 #4044

Merged

thehans commented on Jan 19 Member

Fixed by #4044

thehans closed this as completed on Jan 19

eldstal commented on Feb 4 Contributor Author

Thanks for handling this issue so quickly!

This vulnerability has been assigned CVE-2022-0497 by the Red Hat CNA.

t-paul added a commit that referenced this issue on Feb 5

`CVE-2022-0497 Out-of-bounds memory access in comment parser.` … ✗ 84addf3

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants