tjko / **jpegoptim** Public

<> Code   ⊙ Issues 7   Pull requests   Discussions   Actions   Projects   •••

New issue

Jump to bottom

SEGV caused by a READ memory access #107

⊙ Open **Cvjark** opened this issue on Jun 1 · 5 comments

Cvjark commented on Jun 1 • edited ▾

hi, with the help of fuzzing ,I found some crash sample in this repo, here is the sample, are they new bugs?

crash position jpegoptim.c:631:3
crash sample: crash1_SEGV_caused_by_READ_memory_access_at_jpegoptim.c:631:3
sample here:
crash1_SEGV_caused_by_READ_memory_access_at_jpegoptim.zip

command: ./jpegoptim -f --all-progressive crash_sample

```
AddressSanitizer:DEADLYSIGNAL

=================================================================
==48067==ERROR: AddressSanitizer: SEGV on unknown address (pc 0x7f70c95ca086 bp 0x61c000000270 sp
0x7ffe18c37400 T0)
==48067==The signal is caused by a READ memory access.
==48067==Hint: this fault was caused by a dereference of a high value address (see register values
below).  Disassemble the provided pc to learn which register was used.
    #0 0x7f70c95ca086  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1f086)
    #1 0x7f70c95cad87  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1fd87)
    #2 0x7f70c95c8e08  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1de08)
    #3 0x7f70c95c14c6 in jpeg_consume_input (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x164c6)
    #4 0x7f70c95c176f in jpeg_read_header (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1676f)
    #5 0x4f7f0d in main /home/bupt/Desktop/jpegoptim/jpegoptim.c:631:3
    #6 0x7f70c8998c86 in __libc_start_main /build/glibc-CVJwZb/glibc-2.27/csu/../csu/libc-
start.c:310
    #7 0x41cf09 in _start (/home/bupt/Desktop/jpegoptim/jpegoptim+0x41cf09)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1f086)
==48067==ABORTING
```

crash position: jpegoptim.c:710:18
crash sample: crash2_SEGV_caused_by_READ_memory_access_at_jpegoptim.c:710:18
sample here:
crash_SEGV_caused_by_READ_memory_access_at_jpegoptim.zip
command: ./jpegoptim -f --all-progressive crash_sample

```
==48074==ERROR: AddressSanitizer: SEGV on unknown address (pc 0x7f7896911086 bp 0x61c000000270 sp
0x7fffe7677e00 T0)
==48074==The signal is caused by a READ memory access.
==48074==Hint: this fault was caused by a dereference of a high value address (see register values
below).  Disassemble the provided pc to learn which register was used.
    #0 0x7f7896911086  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1f086)
    #1 0x7f7896911d87  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1fd87)
    #2 0x7f789690fe08  (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1de08)
    #3 0x7f78969186ed in jpeg_read_coefficients (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x266ed)
    #4 0x4f8c9a in main /home/bupt/Desktop/jpegoptim/jpegoptim.c:710:18
    #5 0x7f7895cdfc86 in __libc_start_main /build/glibc-CVJwZb/glibc-2.27/csu/../csu/libc-
start.c:310
    #6 0x41cf09 in _start (/home/bupt/Desktop/jpegoptim/jpegoptim+0x41cf09)

AddressSanitizer can not provide additional info.
SUMMARY: AddressSanitizer: SEGV (/usr/lib/x86_64-linux-gnu/libjpeg.so.8+0x1f086)
==48074==ABORTING
```

tjko commented on Jun 23 Owner

Both examples seem to point issue in libjpeg.so.8 (what was the exact version of libjpeg that jpegoptim was linked against?)

Cvjark commented on Jun 23 • edited ▾ Author

I use `ldconfig -v | grep libjpeg` to checkout the version of libjpeg i use ,and the result : libjpeg.so.8 -> libjpeg.so.8.1.2

dfateyev commented on Sep 28

Was registered as CVE-2022-32325

tjko commented on Sep 28 Owner

How exactly is this an issue in jpegoptim?

Stack traces clearly show issue in libjpeg.so.8, and not in jpegoptim.... or am I missing something?

dfateyev commented on Oct 8

I would suggest to check and reproduce it with a newer `"jpegoptim"` over updated `"libjpeg"` .
Probably it's not an issue anymore. Also not sure if it's applicable to `"libjpeg-turbo"` .

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

3 participants