New issue

Jump to bottom

Heap-buffer-overflow in libjasper/jpc/jpc_enc.c:629 #252

Closed **zodf0055980** opened this issue on Nov 30, 2020 · 8 comments

zodf0055980 commented on Nov 30, 2020 Contributor

I found a heap buffer overflow in the current master (9975856).
I build jasper with ASAN, this is an ASAN report.
POC picture : sample.zip

```
➜  appl git:(master) ✗ ./jasper --input ./sample.pgx --output ./out --output-format jpc -O numrlvls=40
=================================================================
==12383==ERROR: AddressSanitizer: heap-buffer-overflow on address 0x617000000350 at pc 0x7fba3fab9db8 bp 0x7ffc56cf7fa0 sp 0x7ffc56cf7f90
WRITE of size 8 at 0x617000000350 thread T0
    #0 0x7fba3fab9db7 in cp_create /home/yuan/afl-target/jasper/src/libjasper/jpc/jpc_enc.c:629
    #1 0x7fba3fab9db7 in jpc_encode /home/yuan/afl-target/jasper/src/libjasper/jpc/jpc_enc.c:287
    #2 0x5571432fbe8a in main /home/yuan/afl-target/jasper/src/appl/jasper.c:276
    #3 0x7fba3f5f0bf6 in __libc_start_main (/lib/x86_64-linux-gnu/libc.so.6+0x21bf6)
    #4 0x5571432fd339 in _start (/home/yuan/afl-target/jasper/build/src/appl/jasper+0x5339)

0x617000000350 is located 0 bytes to the right of 720-byte region [0x617000000080,0x617000000350)
allocated by thread T0 here:
    #0 0x7fba3fe71b40 in __interceptor_malloc (/usr/lib/x86_64-linux-gnu/libasan.so.4+0xdeb40)
    #1 0x7fba3fa2e5f2 in jas_malloc /home/yuan/afl-target/jasper/src/libjasper/base/jas_malloc.c:238

SUMMARY: AddressSanitizer: heap-buffer-overflow /home/yuan/afl-target/jasper/src/libjasper/jpc/jpc_enc.c:629 in cp_create
Shadow bytes around the buggy address:
  0x0c2e7fff8010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2e7fff8020: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2e7fff8030: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2e7fff8040: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  0x0c2e7fff8050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
=>0x0c2e7fff8060: 00 00 00 00 00 00 00 00 00 00[fa]fa fa fa fa fa
  0x0c2e7fff8070: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2e7fff8080: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2e7fff8090: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2e7fff80a0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
  0x0c2e7fff80b0: fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa fa
Shadow byte legend (one shadow byte represents 8 application bytes):
  Addressable:           00
  Partially addressable: 01 02 03 04 05 06 07
  Heap left redzone:       fa
  Freed heap region:       fd
  Stack left redzone:      f1
  Stack mid redzone:       f2
  Stack right redzone:     f3
  Stack after return:      f5
  Stack use after scope:   f8
  Global redzone:          f9
  Global init order:       f6
  Poisoned by user:        f7
  Container overflow:      fc
  Array cookie:            ac
  Intra object redzone:    bb
  ASan internal:           fe
  Left alloca redzone:     ca
  Right alloca redzone:    cb
==12383==ABORTING
```

I also try to prove it without ASAN.
It malloc 720 bytes in jas_malloc.c:238.
When `-O numrlvls=x > 36`
It tries to write `tccp->prcheightexpns[35]` and causes heap-buffer-overflow-write.

jubalh commented on Nov 30, 2020 Member

I saw that you did some PRs for openjpeg. Do you plan to create one for this issue too?

zodf0055980 commented on Nov 30, 2020 • edited Contributor Author

This is an easy bug. I can try to fix it.

zodf0055980 commented on Nov 30, 2020 • edited Contributor Author

Fixed in #253 . I add the upper bound check.
Could I try to submit this problem to get CVE ID?

jubalh commented on Nov 30, 2020 Member

> Could I try to submit this problem to get CVE ID?

Yes.

jubalh mentioned this issue on Dec 1, 2020

Avoid maxrlvls more than upper bound to cause heap-buffer-overflow #253

Merged

jubalh commented on Dec 1, 2020 — Member

@zodf0055980 thank you for your PR! Next time please use 'Fixes #252' in the commit body.

Closing this since the PR got merged. Please mention the CVE here in case you will get one, ok?

jubalh closed this as completed on Dec 1, 2020

---

zodf0055980 commented on Dec 1, 2020 — Contributor, Author

@jubalh OK. I am sorry I forget it.
If I get a CVE id, I will report here.
Thanks a lot.

👍 1

zodf0055980 commented on Dec 8, 2020 — Contributor, Author

This problem gets CVE-2020-27828.

jubalh commented on Dec 8, 2020 — Member

Thanks for letting us know **@zodf0055980**

shipujin pushed a commit to shipujin/slackware-loongarch64 that referenced this issue on Jul 8

Wed Dec 9 21:10:40 UTC 2020 ··· cf14860

Assignees

No one assigned

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

2 participants