Stack-buffer-overflow was found at ./src/core/ddsi/include/ddsi/q_bitset.h:50:13 in nn_bitset_one. #476

Closed **luckyzfl** opened this issue on Apr 7, 2020 · 6 comments

luckyzfl commented on Apr 7, 2020

I used Peach Fuzzer to fuzz the HelloworldSubscriber at ./build/bin/HelloworldSubscriber.
After a period of time, A stack-buffer-overflow crash was found by AddressSanitizer. Next is the full crash information.

```
`=================================================================
==11082==ERROR: AddressSanitizer: stack-buffer-overflow on address 0x7f1997bf6278 at pc 0x7f199e12cdd0 bp 0x7f1997bf6010 sp 0x7f1997bf6008
READ of size 4 at 0x7f1997bf6278 thread T5 (recv)
#0 0x7f199e12cdcf in nn_bitset_one /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/include/ddsi/q_bitset.h:50:13
#1 0x7f199e12cdcf in nn_defrag_nackmap /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_radmin.c:1467
#2 0x7f199e1484f6 in handle_HeartbeatFrag /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_receive.c:1426:11
#3 0x7f199e1484f6 in handle_submsg_sequence /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_receive.c:2803
#4 0x7f199e13f0f8 in do_packet /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_receive.c:3051:7
#5 0x7f199e13dbbb in recv_thread /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_receive.c:3416:16
#6 0x7f199e162af2 in create_thread_wrapper /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_thread.c:177:9
#7 0x7f199e2559a5 in os_startRoutineWrapper /root/fouzhe/cyclonedds-0.1.0-coverage/src/os/src/posix/../snippets/code/os_posix_thread.c:155:17
#8 0x7f199ddb86b9 in start_thread (/lib/x86_64-linux-gnu/libpthread.so.0+0x76b9)
#9 0x7f199d1c141c in clone /build/glibc-LK5gWL/glibc-2.23/misc/../sysdeps/unix/sysv/linux/x86_64/clone.S:109

Address 0x7f1997bf6278 is located in stack of thread T5 (recv) at offset 344 in frame
#0 0x7f199e13fe8f in handle_submsg_sequence /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_receive.c:2645

=== [Subscriber] Received : Message (1, Hello World) amount: 1
This frame has 37 object(s):
[32, 48) 'sc.i.i' (line 2295)
[64, 68) 'refc_adjust.i.i' (line 2298)
[80, 96) 'pwr_guid.i443' (line 329)
[112, 136) 'src.i444' (line 379)
[176, 192) 'pwr_guid.i' (line 439)
[208, 232) 'src.i366' (line 515)
[272, 288) 'src.i324' (line 1334)
[304, 344) 'nackfrag.i' (line 1422) <== Memory access at offset 344 overflows this variable
[384, 392) 'sm_marker.i.i223' (line 575)
[416, 464) 'sample.i224' (line 1446)
[496, 512) 'src.i225' (line 1447)
[528, 544) 'dst.i226' (line 1447)
[560, 568) 'reply.i' (line 1520)
[592, 616) 'whcst.i227' (line 1553)
[656, 672) 'dst.i186' (line 1577)
[688, 704) 'src.i110' (line 1698)
[720, 736) 'dst.i111' (line 1698)
[752, 756) 'refc_adjust.i112' (line 1748)
[768, 784) 'a.i.i' (line 1106)
[800, 816) 'b.i.i' (line 1106)
[832, 880) 'arg.i' (line 1204)
[912, 928) 'src.i38' (line 1206)
[944, 960) 'dst.i39' (line 1206)
[976, 992) 'sc.i' (line 1238)
[1008, 1012) 'refc_adjust.i' (line 1239)
[1024, 1040) 'sc138.i' (line 1277)
[1056, 1060) 'refc_adjust139.i' (line 1278)
[1072, 1080) 'sm_marker.i.i' (line 575)
[1104, 1120) 'src.i' (line 702)
[1136, 1152) 'dst.i' (line 702)
[1168, 1200) 'gapbits.i' (line 708)
[1232, 1240) 'deferred_free_list.i' (line 717)
[1264, 1288) 'whcst.i' (line 718)
[1328, 1376) 'sample.i' (line 937)
[1408, 2432) 'tmp.i' (line 2521)
[2560, 2616) 'sampleinfo' (line 2809)
[2656, 2712) 'sampleinfo110' (line 2824)
HINT: this may be a false positive if your program uses some custom stack unwind mechanism or swapcontext
(longjmp and C++ exceptions *are* supported)
Thread T5 (recv) created by T0 here:
#0 0x431fdd in pthread_create /root/CL/llvm/projects/compiler-rt/lib/asan/asan_interceptors.cc:317
#1 0x7f199e255444 in os_threadCreate /root/fouzhe/cyclonedds-0.1.0-coverage/src/os/src/posix/../snippets/code/os_posix_thread.c:275:21
#2 0x7f199e1625be in create_thread /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_thread.c:272:7
#3 0x7f199e0d4e78 in setup_and_start_recv_threads /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_init.c:838:34
#4 0x7f199e0d4e78 in rtps_init /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/src/q_init.c:1312
#5 0x7f199e1a0c8e in dds_init /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsc/src/dds_init.c:133:7
#6 0x7f199e1929f2 in dds_create_participant /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsc/src/dds_participant.c:160:11
#7 0x511106 in main /root/fouzhe/cyclonedds-0.1.0-coverage/src/examples/helloworld/subscriber.c:24:19
#8 0x7f199d0da82f in __libc_start_main /build/glibc-LK5gWL/glibc-2.23/csu/../csu/libc-start.c:291
```

```
SUMMARY: AddressSanitizer: stack-buffer-overflow /root/fouzhe/cyclonedds-0.1.0-coverage/src/core/ddsi/include/ddsi/q_bitset.h:50:13 in nn_bitset_one
Shadow bytes around the buggy address:
0x0fe3b2f76bf0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe3b2f76c00: 00 00 00 00 f1 f1 f1 f1 00 f3 f3 f3 00 00 00 00
0x0fe3b2f76c10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0fe3b2f76c20: 00 00 00 00 f1 f1 f1 f1 f8 f8 f2 f2 f8 f2 f8 f8
0x0fe3b2f76c30: f2 f2 f8 f8 f8 f2 f2 f2 f2 f2 f8 f8 f2 f2 f8 f8
=>0x0fe3b2f76c40: f8 f2 f2 f2 f2 f2 00 00 f2 f2 00 00 00 00[f2]
0x0fe3b2f76c50: f2 f2 f2 f2 f8 f2 f2 f2 f8 f8 f8 f8 f8 f8 f2 f2
0x0fe3b2f76c60: f2 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f2 f2 f2 f8 f8
0x0fe3b2f76c70: f8 f2 f2 f2 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f8
0x0fe3b2f76c80: f2 f2 f8 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f8 f8 f8
0x0fe3b2f76c90: f8 f8 f2 f2 f2 f2 f8 f8 f2 f2 f8 f8 f2 f2 f8 f8
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
==11082==ABORTING`
```

I guess it is a potential vulnerability in cyclone project.
Please detect whether it is a problem.
Thanks!

eboasson commented on Apr 7, 2020 — Contributor

Cool!

To state the obvious: it is not the existence of the bug that is cool. What is cool is that you've been fuzzing Cyclone 🙂. And any crash is serious as far as I am concerned, also if it turns out to be non-exploitable or impossible to cause any trouble when run without address sanitizer.

luckyzfl commented on Apr 8, 2020 — Author

And the version of cyclonedds I used is v0.1.0

eboasson commented on Apr 9, 2020 — Contributor

@luckyzfl thanks for letting me know you used v0.1.0: that means I'm pretty certain that this is something I fixed in `23fe452`.

luckyzfl commented on Apr 21, 2020 — Author

Thanks a lot for answering my question.
Now I understand detail of this problem after comparing these two copies of code.

nluedtke commented on Aug 31, 2021

This was assigned CVE-2020-18734.

k0ekk0ek commented on Sep 1, 2021 — Contributor

Thanks **@nluedtke**. Like #501, thanks for creating the CVE (or letting us know). Theoretically this is easily backported to the 0.1.0 branch, but since users are required to update because of the aforementioned issue, I think it's best not to make a 0.1.1 and advice them to upgrade to a newer release.

k0ekk0ek closed this as completed on Sep 1, 2021

Assignees
No one assigned

Labels
None yet

Projects
None yet

Milestone
No milestone

Development

No branches or pull requests

4 participants