



author D. Wythe <alibuda@linux.alibaba.com> 2023-03-08 16:17:12 +0800
committer Jakub Kicinski <kuba@kernel.org> 2023-03-13 16:03:58 -0700
commit [22a825c541d775c1dbe7b2402786025acad6727b](#) (patch)
tree [de9a699ecc429220f9173bb439d3a13b39f44844](#)
parent [131db499162274858bdbd7b5323a639da4aab86c](#) (diff)
download [linux-22a825c541d775c1dbe7b2402786025acad6727b.tar.gz](#)

diff options

context:

3 ▼

space:

include ▼

mode:

unified ▼

net/smc: fix NULL sndbuf_desc in smc_cdc_tx_handler()

When performing a stress test on SMC-R by `rmmod mlx5_ib` driver during the `wrk/nginx` test, we found that there is a probability of triggering a panic while terminating all link groups.

This issue dues to the race between `smc_smcr_terminate_all()` and `smc_buf_create()`.

```
smc_smcr_terminate_all

smc_buf_create
/* init */
conn->sndbuf_desc = NULL;
...

__smc_lgr_terminate
    smc_conn_kill
        smc_close_abort
            smc_cdc_get_slot_and_msg_send

__softirqentry_text_start
    smc_wr_tx_process_cqe
        smc_cdc_tx_handler
            READ(conn->sndbuf_desc->len);
            /* panic dues to NULL sndbuf_desc */

conn->sndbuf_desc = xxx;
```

This patch tries to fix the issue by always to check the `sndbuf_desc` before send any cdc msg, to make sure that no null pointer is seen during cqe processing.

Fixes: [0b29ec643613](#) ("net/smc: immediate termination for SMCR link groups")

Signed-off-by: D. Wythe <alibuda@linux.alibaba.com>

Reviewed-by: Tony Lu <tonylu@linux.alibaba.com>

Reviewed-by: Wenjia Zhang <wenjia@linux.ibm.com>

Link: <https://lore.kernel.org/r/1678263432-17329-1-git-send-email-alibuda@linux.alibaba.com>

Signed-off-by: Jakub Kicinski <kuba@kernel.org>

Diffstat

```
-rw-r--r-- net/smc/smc_cdc.c 3
```

1 files changed, 3 insertions, 0 deletions

```
diff --git a/net/smc/smc_cdc.c b/net/smc/smc_cdc.c
index 53f63bfbaf5f92..89105e95b4523f 100644
--- a/net/smc/smc_cdc.c
+++ b/net/smc/smc_cdc.c
@@ -114,6 +114,9 @@ int smc_cdc_msg_send(struct smc_connection *conn,
    union smc_host_cursor cfed;
    int rc;

+    if (unlikely(!READ_ONCE(conn->sndbuf_desc)))
+        return -ENOBUFFS;
+
    smc_cdc_add_pending_send(conn, pend);

    conn->tx_cdc_seq++;
```