author	Dean Luick <dean.luick@cornelisnetworks.com>	2022-10-18 10:27:50 -0400
committer	Greg Kroah-Hartman <gregkh@linuxfoundation.org>	2022-11-10 17:57:49 +0100
commit	25760a41e3802f54aadcc31385543665ab349b8e (patch)	
tree	20e88e9e5f379ac936c2e47960db70b3ce0ee335	
parent	6b5c87f9b3f87d20935004d528e157dda30e4ea6 (diff)	
download	linux-25760a41e3802f54aadcc31385543665ab349b8e.tar.gz	

diff options

context: 3

space: include

mode: unified

IB/hfi1: Correctly move list in sc_disable()

```
[ Upstream commit 1afac08b39d85437187bb2a92d89a741b1078f55 ]

Commit 13bac861952a ("IB/hfi1: Fix abba locking issue with sc_disable()")
incorrectly tries to move a list from one list head to another.  The
result is a kernel crash.

The crash is triggered when a link goes down and there are waiters for a
send to complete.  The following signature is seen:

  BUG: kernel NULL pointer dereference, address: 0000000000000030
  [...]
  Call Trace:
   sc_disable+0x1ba/0x240 [hfi1]
   pio_freeze+0x3d/0x60 [hfi1]
   handle_freeze+0x27/0x1b0 [hfi1]
   process_one_work+0x1b0/0x380
   ? process_one_work+0x380/0x380
   worker_thread+0x30/0x360
   ? process_one_work+0x380/0x380
   kthread+0xd7/0x100
   ? kthread_complete_and_exit+0x20/0x20
   ret_from_fork+0x1f/0x30

The fix is to use the correct call to move the list.

Fixes: 13bac861952a ("IB/hfi1: Fix abba locking issue with sc_disable()")
Signed-off-by: Dean Luick <dean.luick@cornelisnetworks.com>
Signed-off-by: Dennis Dalessandro <dennis.dalessandro@cornelisnetworks.com>
Link: https://lore.kernel.org/r/166610327042.674422.6146908799669288976.stgit@awfm-02.cornelisnetworks.com
Signed-off-by: Leon Romanovsky <leon@kernel.org>
Signed-off-by: Sasha Levin <sashal@kernel.org>
```

Diffstat

-rw-r--r-- drivers/infiniband/hw/hfi1/pio.c 3

1 files changed, 1 insertions, 2 deletions

```
diff --git a/drivers/infiniband/hw/hfi1/pio.c b/drivers/infiniband/hw/hfi1/pio.c
index 1a82ea73a0fc26..fa5de362010f2f 100644
--- a/drivers/infiniband/hw/hfi1/pio.c
+++ b/drivers/infiniband/hw/hfi1/pio.c
@@ -955,8 +955,7 @@ void sc_disable(struct send_context *sc)
        spin_unlock(&sc->release_lock);

        write_seqlock(&sc->waitlock);
-       if (!list_empty(&sc->piowait))
-               list_move(&sc->piowait, &wake_list);
```

```
+       list_splice_init(&sc->piowait, &wake_list);
        write_sequnlock(&sc->waitlock);
        while (!list_empty(&wake_list)) {
                struct iowait *wait;
```