



author Maciej Fijalkowski <maciej.fijalkowski@intel.com> 2023-03-14 10:45:43 -0700
committer Greg Kroah-Hartman <gregkh@linuxfoundation.org> 2023-03-22 13:31:27 +0100
commit [b89a453c6918e0f346fb0562e8c7812b94d28c73](#) (patch)
tree [bdae67c1211f625fd9d345bf9f17a22c45d4ca](#)
parent [d1bfd4cf6bb8b1c0da00ab25ad6cdb85121cbbbf](#) (diff)
download [linux-b89a453c6918e0f346fb0562e8c7812b94d28c73.tar.gz](#)

diff options

context:
space:
mode:

ice: xsk: disable txq irq before flushing hw

[Upstream commit [b830c9642386867863ac64295185f896ff2928ac](#)]

ice_qp_dis() intends to stop a given queue pair that is a target of xsk pool attach/detach. One of the steps is to disable interrupts on these queues. It currently is broken in a way that txq irq is turned off *after* HW flush which in turn takes no effect.

```
ice_qp_dis():
-> ice_qvec_dis_irq()
--> disable rxq irq
--> flush hw
-> ice_vsi_stop_tx_ring()
--> disable txq irq
```

Below splat can be triggered by following steps:

- start xdpsock WITHOUT loading xdp prog
- run xdp_rxq_info with XDP_TX action on this interface
- start traffic
- terminate xdpsock

```
[ 256.312485] BUG: kernel NULL pointer dereference, address: 0000000000000018
[ 256.319560] #PF: supervisor read access in kernel mode
[ 256.324775] #PF: error_code(0x0000) - not-present page
[ 256.329994] PGD 0 P4D 0
[ 256.332574] Oops: 0000 [#1] PREEMPT SMP NOPTI
[ 256.337006] CPU: 3 PID: 32 Comm: ksoftirqd/3 Tainted: G          OE        6.2.0-rc5+ #51
[ 256.345218] Hardware name: Intel Corporation S2600WFT/S2600WFT, BIOS SE5C620.86B.02.01.0008.031920191559 03/19/2019
[ 256.355807] RIP: 0010:ice_clean_rx_irq_zc+0x9c/0x7d0 [ice]
[ 256.361423] Code: b7 8f 8a 00 00 00 66 39 ca 0f 84 f1 04 00 00 49 8b 47 40 4c 8b 24 d0 41 0f b7 45 04 66 25 ff 3f 66 89 04 24 0f 84 85 02 00 00 <
[ 256.380463] RSP: 0018:ffffc900088bfd20 EFLAGS: 00010206
[ 256.385765] RAX: 000000000000003c RBX: 0000000000000035 RCX: 000000000000067f
[ 256.393012] RDX: 0000000000000775 RSI: 0000000000000000 RDI: ffff8881deb3ac80
[ 256.400256] RBP: 000000000000003c R08: ffff888847982710 R09: 0000000000010000
[ 256.407500] R10: ffffffff82c060c0 R11: 0000000000000004 R12: 0000000000000000
[ 256.414746] R13: ffff88811165eea0 R14: ffff888111650000 R15: ffff888119b37600
[ 256.421990] FS:  0000000000000000(0000) GS:ffff8897e0cc0000(0000) knlGS:0000000000000000
[ 256.430207] CS:  0010 DS:  0000 ES:  0000 CR0: 0000000080050033
[ 256.436036] CR2: 0000000000000018 CR3: 000000005c0a006 CR4: 0000000007706e0
[ 256.443283] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[ 256.450527] DR3: 0000000000000000 DR6: 00000000fffe0ff0 DR7: 0000000000000400
[ 256.457770] PKRU: 55555554
[ 256.460529] Call Trace:
[ 256.463015] <TASK>
[ 256.465157] ? ice_xmit_zc+0x6e/0x150 [ice]
[ 256.469437] ice_napi_poll+0x46d/0x680 [ice]
[ 256.473815] ? __raw_spin_unlock_irqrestore+0x1b/0x40
[ 256.478863] __napi_poll+0x29/0x160
[ 256.482409] net_rx_action+0x136/0x260
[ 256.486222] __do_softirq+0xe8/0x2e5
[ 256.489853] ? smpboot_thread_fn+0x2c/0x270
[ 256.494108] run_ksoftirqd+0x2a/0x50
[ 256.497747] smpboot_thread_fn+0x1c1/0x270
[ 256.501907] ? __pfx_smpboot_thread_fn+0x10/0x10
[ 256.506594] kthread+0xea/0x120
[ 256.509785] ? __pfx_kthread+0x10/0x10
[ 256.513597] ret_from_fork+0x29/0x50
[ 256.517238] </TASK>
```

In fact, irqs were not disabled and napi managed to be scheduled and run while xsk_pool pointer was still valid, but SW ring of xdp_buff pointers was already freed.

To fix this, call ice_qvec_dis_irq() after ice_vsi_stop_tx_ring(). Also while at it, remove redundant ice_clean_rx_ring() call - this is handled in ice_qp_clean_rings().

Fixes: [2d4238f55697](#) ("ice: Add support for AF_XDP")
Signed-off-by: Maciej Fijalkowski <maciej.fijalkowski@intel.com>
Reviewed-by: Larysa Zaremba <larysa.zaremba@intel.com>
Tested-by: Chandan Kumar Rout <chandanx.rout@intel.com> (A Contingent Worker at Intel)
Acked-by: John Fastabend <john.fastabend@gmail.com>
Signed-off-by: Tony Nguyen <anthony.l.nguyen@intel.com>
Reviewed-by: Leon Romanovsky <leonro@nvidia.com>

Signed-off-by: David S. Miller <davem@davemloft.net>
Signed-off-by: Sasha Levin <sashal@kernel.org>

Diffstat

```
-rw-r--r-- drivers/net/ethernet/intel/ice/ice_xsk.c 4
```

1 files changed, 2 insertions, 2 deletions

```
diff --git a/drivers/net/ethernet/intel/ice/ice_xsk.c b/drivers/net/ethernet/intel/ice/ice_xsk.c
index 60d8ef0c885950..070be30cbaa917 100644
--- a/drivers/net/ethernet/intel/ice/ice_xsk.c
+++ b/drivers/net/ethernet/intel/ice/ice_xsk.c
@@ -166,8 +166,6 @@ static int ice_qp_dis(struct ice_vsi *vsi, u16 q_idx)
     }
     netif_tx_stop_queue(netdev_get_tx_queue(vsi->netdev, q_idx));

-    ice_qvec_dis_irq(vsi, rx_ring, q_vector);
-
     ice_fill_txq_meta(vsi, tx_ring, &txq_meta);
     err = ice_vsi_stop_tx_ring(vsi, ICE_NO_RESET, 0, tx_ring, &txq_meta);
     if (err)
@@ -182,6 +180,8 @@ static int ice_qp_dis(struct ice_vsi *vsi, u16 q_idx)
     if (err)
         return err;
     }
+    ice_qvec_dis_irq(vsi, rx_ring, q_vector);
+
     err = ice_vsi_ctrl_one_rx_ring(vsi, false, q_idx, true);
     if (err)
         return err;
```