syzbot found a crash in batadv_iv_send_outstanding_bat_ogm_packet when interfaces are removed from the bat0

Hello,

syzbot found the following crash on:

HEAD commit: f7571657 Merge tag 'fuse-fixes-5.6-rc1' of git://git.kerne.. 
git tree: upstream
console output: https://syzkaller.appspot.com/x/log.txt?x=12dddbbee00000
kernel config: https://syzkaller.appspot.com/x/.config?x=7f1d914a74bd6ddc
dashboard link: https://syzkaller.appspot.com/bug?extid=ac36b6a33c28a491e929
compiler: clang version 10.0.0 (https://github.com/llvm/llvm-project/ c2443155a0fb245c8f17f2 c1c72b6ea391e86e81)

Unfortunately, I don't have any reproducer for this crash yet.

IMPORTANT: if you fix the bug, please add the following tag to the commit:
Reported-by: syzbot+ac36b6a33c28a491e929@syzkaller.appspotmail.com

general protection fault, probably for non-canonical address 0xdffffc0000000002: 0000 [#1] PREEMPT
SMP KASAN
KASAN: null-ptr-deref in range [0x0000000000000010-0x0000000000000017]
CPU: 0 PID: 465 Comm: kworker/u4:5 Not tainted 5.5.0-syzkaller #0
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS Google 01/01/2011
Workqueue: bat_events batadv_iv_send_outstanding_bat_ogm_packet
RIP: 0010:batadv_iv_send_outstanding_bat_ogm_packet+0x664/0x770 (net/batman-adv/bat_iv_ogm.c:1718)
RIP: 0010:process_one_work+0x7f5/0x10f0 (kernel/workqueue.c:2264)
RIP: 0010:worker_thread+0xbcc/0x1630 (kernel/workqueue.c:2410)
RIP: 0010:kthread+0x332/0x350 (kernel/kthread.c:255)
RIP: 0010:ret_from_fork+0x24/0x28 (arch/x86/entry/entry_64.S:352)

Call Trace: 
batadv_iv_send_outstanding_bat_ogm_packet+0x664/0x770 net/batman-adv/bat_iv_ogm.c:1718
process_one_work+0x7f5/0x10f0 kernel/workqueue.c:2264
worker_thread+0xbcc/0x1630 kernel/workqueue.c:2410
kthread+0x332/0x350 kernel/kthread.c:255
ret_from_fork+0x24/0x28 arch/x86/entry/entry_64.S:352

Modules linked in:

---[ end trace eddf69e5e4c9f596 ]---
This bug is generated by a bot. It may contain errors.
See https://goo.gl/tpsmEJ for more information about syzbot.
syzbot engineers can be reached at syzkaller@googlegroups.com.
syzbot will keep track of this bug report. See:
https://goo.gl/tpsmEJ#status for how to communicate with syzbot.

The relevant code section is:

```c
if (hard_iface == primary_if) {
    /* tt changes have to be committed before the tvlv data is
    appended as it may alter the tt tvlv container
    */
    batadv_tt_local_commit_changes(bat_priv);
    tvlv_len = batadv_tvlv_container_ogm_append(bat_priv, ogm_buff,
                                               ogm_buff_len,
                                               BATADV_OGM_HLEN);
}

batadv_ogm_packet = (struct batadv_ogm_packet *)(*ogm_buff);
batadv_ogm_packet->tvlv_len = htons(tvlv_len);

/* change sequence number to network order */
seqno = (u32)atomic_read(&hard_iface->bat_iv.ogm_seqno);
batadv_ogm_packet->seqno = htonl(seqno);
atomic_inc(&hard_iface->bat_iv.ogm_seqno);
batadv_iv_ogm_slide_own_bcast_window(hard_iface)
```

Could this be some kind of use after free?

History

#1 - 02/16/2020 01:22 PM - Sven Eckelmann
See also
- https://lists.open-mesh.org/pipermail/b.a.t.m.a.n/2020-February/019127.html
- https://lists.open-mesh.org/pipermail/b.a.t.m.a.n/2020-February/019129.html

#2 - 02/17/2020 05:46 PM - Sven Eckelmann
- % Done changed from 0 to 100
- Target version set to 2020.0
- Status changed from New to Resolved

The untested [1] patch was submitted as https://patchwork.open-mesh.org/patch/18075/

[1] there is currently no good reproducer for this bug