batman-adv - Bug #371

batadv_interface_tx causes invalid (eth_hdr) memory access

12/31/2018 07:56 PM - Sven Eckelmann

Status:	Closed	**Start date:**	12/31/2018
Priority:	Immediate	**Due date:**	
Assignee:	batman-adv developers	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	2019.0		

Description

https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/7K3FEZDV3VUSITUASHQE2NPQPE3UVMFQ/
https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/4DBIPZJUB56K2A6AXUX6URRUHMOOWG4V/

syzbot found the following crash on:

HEAD commit: 903b77c63167 Merge tag 'linux-kselftest-4.21-rc1' of git:/..
git tree: upstream
console output: https://syzkaller.appspot.com/x/log.txt?x=168acbdd400000
kernel config: https://syzkaller.appspot.com/x/.config?x=53a2f2aa0b1f7606
dashboard link: https://syzkaller.appspot.com/bug?extid=7d20bc3f1ddddc0f9079
compiler: gcc (GCC) 8.0.1 20180413 (experimental)
syz repro: https://syzkaller.appspot.com/x/repro.syz?x=11aa3b57400000
C reproducer: https://syzkaller.appspot.com/x/repro.c?x=12c907ed400000

IMPORTANT: if you fix the bug, please add the following tag to the commit:
Reported-by: syzbot+7d20bc3f1ddddc0f9079@syzkaller.appspotmail.com

```
IPv6: ADDRCONF(NETDEV_UP): vxcan1: link is not ready
8021q: adding VLAN 0 to HW filter on device batadv0
==================================================================
BUG: KASAN: use-after-free in batadv_interface_tx+0x160a/0x18b0
net/batman-adv/soft-interface.c:226
Read of size 2 at addr ffff88809f96aa4b by task syz-executor871/8379

CPU: 1 PID: 8379 Comm: syz-executor871 Not tainted 4.20.0+ #395
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS
Google 01/01/2011
Call Trace:
  __dump_stack lib/dump_stack.c:77 [inline]
  dump_stack+0x1d3/0x2c6 lib/dump_stack.c:113
  print_address_description.cold.5+0x9/0x1ff mm/kasan/report.c:187
  kasan_report.cold.6+0x1b/0x39 mm/kasan/report.c:317
  __asan_report_load_n_noabort+0xf/0x20 mm/kasan/generic_report.c:145
  batadv_interface_tx+0x160a/0x18b0 net/batman-adv/soft-interface.c:226
  __netdev_start_xmit include/linux/netdevice.h:4382 [inline]
  netdev_start_xmit include/linux/netdevice.h:4391 [inline]
  dev_direct_xmit+0x36c/0x6a0 net/core/dev.c:3930
  packet_direct_xmit+0xfb/0x170 net/packet/af_packet.c:246
  packet_snd net/packet/af_packet.c:2932 [inline]
  packet_sendmsg+0x298a/0x6ad0 net/packet/af_packet.c:2957
  sock_sendmsg_nosec net/socket.c:621 [inline]
  sock_sendmsg+0xd5/0x120 net/socket.c:631
  __sys_sendto+0x3d7/0x670 net/socket.c:1788
  __do_sys_sendto net/socket.c:1800 [inline]
  __se_sys_sendto net/socket.c:1796 [inline]
  __x64_sys_sendto+0xe1/0x1a0 net/socket.c:1796
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe
RIP: 0033:0x441eb9
Code: e8 4c ad 02 00 48 83 c4 18 c3 0f 1f 80 00 00 00 00 48 89 f8 48 89 f7
```

```
48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff
ff 0f 83 fb 04 fc ff c3 66 2e 0f 1f 84 00 00 00 00
RSP: 002b:00007ffee312ce58 EFLAGS: 00000212 ORIG_RAX: 000000000000002c
RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 0000000000441eb9
RDX: 000000000000000e RSI: 0000000020000180 RDI: 0000000000000003
RBP: 0000000000000001 R08: 0000000000000000 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000212 R12: 0000000000000000
R13: 00000000004028d0 R14: 0000000000000000 R15: 0000000000000000

Allocated by task 8368:
  save_stack+0x43/0xd0 mm/kasan/common.c:73
  set_track mm/kasan/common.c:85 [inline]
  kasan_kmalloc+0xcb/0xd0 mm/kasan/common.c:482
  kasan_slab_alloc+0x12/0x20 mm/kasan/common.c:397
  kmem_cache_alloc+0x130/0x730 mm/slab.c:3541
  getname_flags+0xd0/0x590 fs/namei.c:140
  user_path_at_empty+0x2d/0x50 fs/namei.c:2608
  user_path_at include/linux/namei.h:57 [inline]
  do_faccessat+0x254/0x800 fs/open.c:378
  __do_sys_access fs/open.c:430 [inline]
  __se_sys_access fs/open.c:428 [inline]
  __x64_sys_access+0x59/0x80 fs/open.c:428
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe

Freed by task 8368:
  save_stack+0x43/0xd0 mm/kasan/common.c:73
  set_track mm/kasan/common.c:85 [inline]
  __kasan_slab_free+0x102/0x150 mm/kasan/common.c:444
  kasan_slab_free+0xe/0x10 mm/kasan/common.c:452
  __cache_free mm/slab.c:3485 [inline]
  kmem_cache_free+0x83/0x290 mm/slab.c:3747
  putname+0xf2/0x130 fs/namei.c:261
  filename_lookup+0x39a/0x520 fs/namei.c:2357
  user_path_at_empty+0x40/0x50 fs/namei.c:2608
  user_path_at include/linux/namei.h:57 [inline]
  do_faccessat+0x254/0x800 fs/open.c:378
  __do_sys_access fs/open.c:430 [inline]
  __se_sys_access fs/open.c:428 [inline]
  __x64_sys_access+0x59/0x80 fs/open.c:428
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe

The buggy address belongs to the object at ffff88809f96a2c0
  which belongs to the cache names_cache of size 4096
The buggy address is located 1931 bytes inside of
  4096-byte region [ffff88809f96a2c0, ffff88809f96b2c0)
The buggy address belongs to the page:
page:ffffea00027e5a80 count:1 mapcount:0 mapping:ffff88821bc49080 index:0x0
compound_mapcount: 0
flags: 0x1fffc0000010200(slab|head)
raw: 01fffc0000010200 ffffea0002810f08 ffffea00027d0388 ffff88821bc49080
raw: 0000000000000000 ffff88809f96a2c0 0000000100000001 0000000000000000
page dumped because: kasan: bad access detected

Memory state around the buggy address:
  ffff88809f96a900: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
  ffff88809f96a980: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
> ffff88809f96aa00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
                                          ^
  ffff88809f96aa80: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
  ffff88809f96ab00: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
==================================================================
```

syzbot found the following crash on:

HEAD commit: f346b0becb1b Merge branch 'akpm' (patches from Andrew)
git tree: upstream
console output: https://syzkaller.appspot.com/x/log.txt?x=10f0bbdd400000
kernel config: https://syzkaller.appspot.com/x/.config?x=c255c77ba370fe7c
dashboard link: https://syzkaller.appspot.com/bug?extid=9d7405c7faa390e60b4e
compiler: gcc (GCC) 8.0.1 20180413 (experimental)
syz repro: https://syzkaller.appspot.com/x/repro.syz?x=14c3eabf400000
C reproducer: https://syzkaller.appspot.com/x/repro.c?x=164bfbfb400000

IMPORTANT: if you fix the bug, please add the following tag to the commit:
Reported-by: syzbot+9d7405c7faa390e60b4e@syzkaller.appspotmail.com

```
IPv6: ADDRCONF(NETDEV_UP): vxcan1: link is not ready
8021q: adding VLAN 0 to HW filter on device batadv0
==================================================================
BUG: KASAN: slab-out-of-bounds in batadv_interface_tx+0x160a/0x18b0
net/batman-adv/soft-interface.c:226
Read of size 2 at addr ffff8880a662f5cb by task syz-executor922/8142

CPU: 0 PID: 8142 Comm: syz-executor922 Not tainted 4.20.0+ #173
Hardware name: Google Google Compute Engine/Google Compute Engine, BIOS
Google 01/01/2011
Call Trace:
  __dump_stack lib/dump_stack.c:77 [inline]
  dump_stack+0x1d3/0x2c6 lib/dump_stack.c:113
  print_address_description.cold.5+0x9/0x1ff mm/kasan/report.c:187
  kasan_report.cold.6+0x1b/0x39 mm/kasan/report.c:317
  __asan_report_load_n_noabort+0xf/0x20 mm/kasan/generic_report.c:145
  batadv_interface_tx+0x160a/0x18b0 net/batman-adv/soft-interface.c:226
  __netdev_start_xmit include/linux/netdevice.h:4382 [inline]
  netdev_start_xmit include/linux/netdevice.h:4391 [inline]
  dev_direct_xmit+0x36c/0x6a0 net/core/dev.c:3930
  packet_direct_xmit+0xfb/0x170 net/packet/af_packet.c:246
  packet_snd net/packet/af_packet.c:2932 [inline]
  packet_sendmsg+0x298a/0x6ad0 net/packet/af_packet.c:2957
  sock_sendmsg_nosec net/socket.c:621 [inline]
  sock_sendmsg+0xd5/0x120 net/socket.c:631
  __sys_sendto+0x3d7/0x670 net/socket.c:1788
  __do_sys_sendto net/socket.c:1800 [inline]
  __se_sys_sendto net/socket.c:1796 [inline]
  __x64_sys_sendto+0xe1/0x1a0 net/socket.c:1796
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe
RIP: 0033:0x441619
Code: 18 89 d0 c3 66 2e 0f 1f 84 00 00 00 00 00 0f 1f 00 48 89 f8 48 89 f7
48 89 d6 48 89 ca 4d 89 c2 4d 89 c8 4c 8b 4c 24 08 0f 05 <48> 3d 01 f0 ff
ff 0f 83 3b 0b fc ff c3 66 2e 0f 1f 84 00 00 00 00
RSP: 002b:00007ffd5c9334f8 EFLAGS: 00000212 ORIG_RAX: 000000000000002c
RAX: ffffffffffffffda RBX: 0000000000000000 RCX: 0000000000441619
RDX: 000000000000000e RSI: 0000000020000180 RDI: 0000000000000003
RBP: 0000000000000001 R08: 0000000000000000 R09: 0000000000000000
R10: 0000000000000000 R11: 0000000000000212 R12: 00000000004025e0
R13: 0000000000402670 R14: 0000000000000000 R15: 0000000000000000

Allocated by task 8184:
  save_stack+0x43/0xd0 mm/kasan/common.c:73
  set_track mm/kasan/common.c:85 [inline]
  kasan_kmalloc+0xcb/0xd0 mm/kasan/common.c:482
  kasan_slab_alloc+0x12/0x20 mm/kasan/common.c:397
  kmem_cache_alloc+0x130/0x730 mm/slab.c:3541
  getname_flags+0xd0/0x590 fs/namei.c:140
  getname+0x19/0x20 fs/namei.c:211
  do_sys_open+0x383/0x780 fs/open.c:1057
  __do_sys_open fs/open.c:1081 [inline]
  __se_sys_open fs/open.c:1076 [inline]
```

```
  __x64_sys_open+0x7e/0xc0 fs/open.c:1076
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe

Freed by task 8184:
  save_stack+0x43/0xd0 mm/kasan/common.c:73
  set_track mm/kasan/common.c:85 [inline]
  __kasan_slab_free+0x102/0x150 mm/kasan/common.c:444
  kasan_slab_free+0xe/0x10 mm/kasan/common.c:452
  __cache_free mm/slab.c:3485 [inline]
  kmem_cache_free+0x83/0x290 mm/slab.c:3747
  putname+0xf2/0x130 fs/namei.c:261
  do_sys_open+0x54d/0x780 fs/open.c:1072
  __do_sys_open fs/open.c:1081 [inline]
  __se_sys_open fs/open.c:1076 [inline]
  __x64_sys_open+0x7e/0xc0 fs/open.c:1076
  do_syscall_64+0x1b9/0x820 arch/x86/entry/common.c:290
  entry_SYSCALL_64_after_hwframe+0x49/0xbe

The buggy address belongs to the object at ffff8880a662e5c0
  which belongs to the cache names_cache of size 4096
The buggy address is located 11 bytes to the right of
  4096-byte region [ffff8880a662e5c0, ffff8880a662f5c0)
The buggy address belongs to the page:
page:ffffea0002998b80 count:1 mapcount:0 mapping:ffff88821bc48200 index:0x0
compound_mapcount: 0
flags: 0x1fffc0000010200(slab|head)
raw: 01fffc0000010200 ffffea00029fa208 ffffea00022fd188 ffff88821bc48200
raw: 0000000000000000 ffff8880a662e5c0 0000000100000001 0000000000000000
page dumped because: kasan: bad access detected

Memory state around the buggy address:
  ffff8880a662f480: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
  ffff8880a662f500: fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb fb
> ffff8880a662f580: fb fb fb fb fb fb fb fb fc fc fc fc fc fc fc fc
                                                 ^
  ffff8880a662f600: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
  ffff8880a662f680: fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc fc
==================================================================
```

History

#1 - 12/31/2018 08:55 PM - Sven Eckelmann

- Status changed from New to In Progress

The reproducer1 & reproducer2 can be used to reproduce it with Emulation Environment

The interesting line seems to be

```
    switch (ntohs(ethhdr->h_proto)) {
```

Or with more context:

```
    /* reset control block to avoid left overs from previous users */
    memset(skb->cb, 0, sizeof(struct batadv_skb_cb));

    netif_trans_update(soft_iface);
    vid = batadv_get_vid(skb, 0);
    ethhdr = eth_hdr(skb);

    switch (ntohs(ethhdr->h_proto)) {
    case ETH_P_8021Q:
```

This seems to suggest that the retriever ethernet header isn't backed with data in the skb.

I've added some printk to show the ethhdr + skb->data for the crash:

```
batadv_interface_tx:225 0xffff888013a7e9ff
batadv_interface_tx:226 0xffff888013a6ea02
```

You can see that the eth_hdr (first line) is for some reason 65533 bytes far away. This doesn't make a lot of sense because batman-adv expects that the ethernet header is just in front of the current skb->data

#2 - 12/31/2018 10:14 PM - Sven Eckelmann

Looks like the important piece here is the

```
int opt = 4;
setsockopt(sock, SOL_PACKET, PACKET_QDISC_BYPASS, &opt, 4);
```

right before the bind. It doesn't happen without this. I can also reproduce this by using the lot simpler rawsend.c
https://lists.open-mesh.org/mailman3/hyperkitty/list/b.a.t.m.a.n@lists.open-mesh.org/message/ZS6SNV2S7JYWZWCHQUNKPP3267MTXGZ3/attachment/3/rawsend.c

The batman-adv device was created using:

```
#! /bin/sh
insmod /host/batman-adv/net/batman-adv/batman-adv.ko
ip link add dev batadv0 type batadv
ip link set up dev batadv0
```

#3 - 12/31/2018 10:23 PM - Sven Eckelmann

- File reproducer1_simplified.c added

Here is also the simplified reproducer (with 99% less hardcoded hex values and pid/netns)

#4 - 12/31/2018 10:44 PM - Sven Eckelmann

- Description updated

#5 - 12/31/2018 10:47 PM - Sven Eckelmann

- Target version set to 2019.0

Patch can be found at https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/20181231214609.22378-1-sven@narfation.org/

#6 - 01/04/2019 11:13 AM - Sven Eckelmann

- Status changed from In Progress to Resolved

#7 - 02/01/2019 11:43 PM - Sven Eckelmann

- Status changed from Resolved to Closed

#8 - 05/27/2020 09:58 PM - Sven Eckelmann

- Description updated

Files

reproducer1.c	11.4 KB	12/31/2018	Sven Eckelmann
reproducer2.c	9.46 KB	12/31/2018	Sven Eckelmann
reproducer1_simplified.c	1.42 KB	12/31/2018	Sven Eckelmann