batman-adv - Bug #169

General protection error in batadv_tt_global_del_orig()

03/17/2013 02:16 AM - Linus Lüssing

Status:	Closed	**Start date:**	03/17/2013
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2013.3.0		

Description

With two VMs connected to each other, running a 3.8.1 kernel, I'm hitting the following bug when unloading batman-adv (master, 18491357187f9fb8f0e56eb4a9a8af8f793dfd08):

```
[  131.611176] batman_adv: B.A.T.M.A.N. advanced 2013.1.0-24-g1849135-dirty (compatibility version
 14) loaded
[  132.099069] 8139cp 0000:00:03.0 eth1: link up, 100Mbps, full-duplex, lpa 0x05E1
[  132.140967] 8139cp 0000:00:04.0 eth2: link up, 100Mbps, full-duplex, lpa 0x05E1
[  132.186194] 8139cp 0000:00:05.0 eth3: link up, 100Mbps, full-duplex, lpa 0x05E1
[  132.234821] 8139cp 0000:00:06.0 eth4: link up, 100Mbps, full-duplex, lpa 0x05E1
[  132.277455] batman_adv: bat0: Adding interface: eth1
[  132.284719] batman_adv: bat0: Interface activated: eth1
[  132.294374] batman_adv: bat0: Adding interface: eth2
[  132.300770] batman_adv: bat0: Interface activated: eth2
[  132.322999] batman_adv: bat0: Adding interface: eth3
[  132.329421] batman_adv: bat0: Interface activated: eth3
[  132.340253] batman_adv: bat0: Adding interface: eth4
[  132.358525] batman_adv: bat0: Interface activated: eth4
[  137.945588] batman_adv: bat0: Interface deactivated: eth1
[  137.948567] batman_adv: bat0: Removing interface: eth1
[  137.952896] batman_adv: bat0: Interface deactivated: eth2
[  137.965527] batman_adv: bat0: Removing interface: eth2
[  137.968676] batman_adv: bat0: Interface deactivated: eth3
[  137.970727] batman_adv: bat0: Removing interface: eth3
[  137.975008] batman_adv: bat0: Interface deactivated: eth4
[  137.980301] batman_adv: bat0: Removing interface: eth4
[  138.052107] general protection fault: 0000 [#1] SMP
[  138.054058] Modules linked in: batman_adv(O-) crc32c libcrc32c crc16 dm_crypt md_mod snd_pcm sn
d_page_alloc snd_timer processor snd thermal_sys soundcore evdev microcode psmouse pcspkr button s
erio_raw 8139too 8139cp i2c_piix4 mii i2c_core floppy ata_generic ata_piix libata virtio_pci scsi_
mod 9p fscache dm_mirror dm_region_hash dm_log dm_mod 9pnet_virtio virtio_ring virtio 9pnet
[  138.083200] CPU 0
[  138.083912] Pid: 3049, comm: net.agent Tainted: G           O 3.8.1 #1 Bochs Bochs
[  138.086892] RIP: 0010:[<ffffffffa0212b86>]  [<ffffffffa0212b86>] batadv_tt_global_del_orig+0x98
/0xb5 [batman_adv]
[  138.092602] RSP: 0018:ffff880007003de0  EFLAGS: 00010246
[  138.099017] RAX: ffff880003ea4000 RBX: 6b6b6b6b6b6b6b6b RCX: 0000000000000102
[  138.101751] RDX: ffffffffa0218c0e RSI: ffff880007bee000 RDI: ffff880004916a40
[  138.115026] RBP: ffff880007003e20 R08: 0000000000000000 R09: 0000000000002b78
[  138.117664] R10: 0000000000002b78 R11: ffff880006cea048 R12: 0000000000000000
[  138.120347] R13: ffff880007bee000 R14: ffff880007bee000 R15: ffff880007bee188
[  138.123026] FS:  00007f7fb3270700(0000) GS:ffff880007000000(0000) knlGS:0000000000000000
[  138.134173] CS:  0010 DS: 0000 ES: 0000 CR0: 000000008005003b
[  138.136613] CR2: 00007f7fb2d9c890 CR3: 00000000053ef000 CR4: 00000000000006f0
[  138.139578] DR0: 0000000000000000 DR1: 0000000000000000 DR2: 0000000000000000
[  138.142336] DR3: 0000000000000000 DR6: 00000000ffff0ff0 DR7: 0000000000000400
[  138.150435] Process net.agent (pid: 3049, threadinfo ffff880007a80000, task ffff880003ea4000)
[  138.153571] Stack:
[  138.158098]  ffff880007003df0 ffffffff8104c80a ffff880007003e10 ffff880007bee1c0
[  138.161032]  0000000000000000 ffff880007bee000 ffff880007bee0e0 ffff880007bee188
[  138.167069]  ffff880007003e70 ffffffffa020c10a ffff88000700f1f0 ffff880007bee1a8
[  138.171184] Call Trace:
[  138.174598]  <IRQ>
```

```
[  138.175333]  [<ffffffff8104c80a>] ? local_bh_enable_ip+0xe/0x10
[  138.178984]  [<ffffffffa020c10a>] batadv_orig_node_free_rcu+0xdb/0x116 [batman_adv]
[  138.187690]  [<ffffffff810c1a9c>] rcu_process_callbacks+0x264/0x41d
[  138.190109]  [<ffffffff8104cd5a>] __do_softirq+0x122/0x277
[  138.195653]  [<ffffffff8108c11c>] ? clockevents_program_event+0xa1/0xbe
[  138.202145]  [<ffffffff810d7672>] ? time_hardirqs_off+0x15/0x2b
[  138.204419]  [<ffffffff8140527c>] call_softirq+0x1c/0x30
[  138.209996]  [<ffffffff81011dcc>] do_softirq+0x4a/0xa2
[  138.211939]  [<ffffffff8104cfd7>] irq_exit+0x51/0xbc
[  138.214951]  [<ffffffff81405bf1>] smp_apic_timer_interrupt+0x7c/0x8a
[  138.220024]  [<ffffffff81404b32>] apic_timer_interrupt+0x72/0x80
[  138.225962]  <EOI>
[  138.226730]  [<ffffffff813fdc73>] ? retint_restore_args+0x13/0x13
[  138.231374]  [<ffffffff81099157>] ? arch_local_irq_restore+0x6/0xd
[  138.239423]  [<ffffffff813fd6e6>] _raw_spin_unlock_irqrestore+0x4d/0x61
[  138.243742]  [<ffffffffa001f019>] spin_unlock_irqrestore+0x9/0xb [9pnet_virtio]
[  138.249851]  [<ffffffffa001f7d9>] p9_virtio_request+0x187/0x198 [9pnet_virtio]
[  138.252556]  [<ffffffff8112c6ba>] ? kfree_debugcheck+0x13/0x2c
[  138.258815]  [<ffffffffa00017b2>] p9_client_rpc+0xd6/0x3e3 [9pnet]
[  138.261155]  [<ffffffff81090d26>] ? mark_held_locks+0x71/0x99
[  138.270555]  [<ffffffffa00025ab>] ? p9_client_getattr_dotl+0x34/0xdf [9pnet]
[  138.273149]  [<ffffffffa00025ab>] ? p9_client_getattr_dotl+0x34/0xdf [9pnet]
[  138.275720]  [<ffffffffa00025da>] p9_client_getattr_dotl+0x63/0xdf [9pnet]
[  138.285947]  [<ffffffffa00711a2>] v9fs_inode_from_fid_dotl+0x2c/0x13c [9p]
[  138.288509]  [<ffffffffa0070424>] v9fs_get_new_inode_from_fid+0x13/0x1c [9p]
[  138.291823]  [<ffffffffa00704e4>] v9fs_vfs_lookup+0xb7/0x111 [9p]
[  138.294156]  [<ffffffffa0071324>] v9fs_vfs_atomic_open_dotl+0x56/0x2ba [9p]
[  138.302032]  [<ffffffff8114c27f>] ? spin_unlock+0x9/0xb
[  138.303927]  [<ffffffff8114e63b>] ? d_alloc+0x57/0x62
[  138.305862]  [<ffffffff81143efd>] ? lookup_dcache+0x8a/0xa3
[  138.314036]  [<ffffffff81147381>] do_last+0x3b7/0xa25
[  138.326134]  [<ffffffff811453c4>] ? inode_permission+0x45/0x47
[  138.328304]  [<ffffffff81147ab2>] path_openat+0xc3/0x330
[  138.330240]  [<ffffffff8109192a>] ? __lock_is_held+0x32/0x54
[  138.332312]  [<ffffffff81148056>] do_filp_open+0x38/0x86
[  138.334259]  [<ffffffff811533d9>] ? __alloc_fd+0x17d/0x18f
[  138.339404]  [<ffffffff8113b07a>] do_sys_open+0x6c/0xf9
[  138.341329]  [<ffffffff8113b128>] sys_open+0x21/0x23
[  138.347651]  [<ffffffff81403ed9>] system_call_fastpath+0x16/0x1b
[  138.349942] Code: 4c 89 f6 48 89 45 c0 4c 89 ff 48 89 55 c8 e8 e9 e7 ff ff 48 8b 45 c0 48 83 78
 38 00 75 d2 eb c0 4c 89 ef 41 ff c4 e8 2f e4 ff ff <44> 3b 63 10 72 90 41 c6 86 a4 00 00 00 00 48
 83 c4 18 5b 41 5c
[  138.368859] RIP  [<ffffffffa0212b86>] batadv_tt_global_del_orig+0x98/0xb5 [batman_adv]
[  138.379250]  RSP <ffff880007003de0>
[  138.380607] ---[ end trace 6ddb79344d954af5 ]---
[  138.382343] Kernel panic - not syncing: Fatal exception in interrupt
```

```
$ gdb batman-adv.ko
GNU gdb (GDB) 7.4.1-debian
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /var/batman/batman-adv-t_x/batman-adv.ko...done.
(gdb) l *(&batadv_tt_global_del_orig+0x98)
0xebaa is in batadv_tt_global_del_orig (/var/batman/batman-adv-t_x/translation-table.c:1206).
1201            spinlock_t *list_lock; /* protects write access to the hash lists */
1202
1203            if (!hash)
```

```
1204                    return;
1205
1206            for (i = 0; i < hash->size; i++) {
1207                    head = &hash->table[i];
1208                    list_lock = &hash->list_locks[i];
1209
1210                    spin_lock_bh(list_lock);
(gdb) l *(&batadv_orig_node_free_rcu+0xdb)
0x812e is in batadv_orig_node_free_rcu (/var/batman/batman-adv-t_x/originator.c:153).
148
149             batadv_frag_list_free(&orig_node->frag_list);
150             batadv_tt_global_del_orig(orig_node->bat_priv, orig_node,
151                                       "originator timed out");
152
153             kfree(orig_node->tt_buff);
154             kfree(orig_node->bcast_own);
155             kfree(orig_node->bcast_own_sum);
156             kfree(orig_node);
157     }
(gdb) quit
```

History

#1 - 03/17/2013 03:01 AM - Linus Lüssing

batadv_mesh_free()->batadv_originator_free() schedules the batadv_orig_node_free_rcu().

Before batadv_orig_node_free_rcu() is executed (which happens on the rcu_barrier() call in batadv_exit() the latest), batadv_mesh_free()->batadv_tt_free()->batadv_tt_global_table_free()->batadv_hash_destroy(hash)->kfree(hash) is called, freeing the global tt hash.

When batadv_orig_node_free_rcu()->batadv_tt_global_del_orig() now gets executed it tries to access this just freed global tt hash, causing a kernel panic.

#2 - 05/07/2013 08:07 PM - Antonio Quartulli

- % Done changed from 0 to 100

- Status changed from New to Resolved

Fixed by
batman-adv: Fix rcu_barrier() miss due to double call_rcu() in TT code
and
batman-adv: fix global protection fault during soft_iface destruction

#3 - 08/31/2013 08:42 PM - Antonio Quartulli

- Status changed from Resolved to Closed

#4 - 04/16/2016 11:53 PM - Sven Eckelmann

- Description updated

#5 - 02/11/2017 09:15 AM - Sven Eckelmann

- Target version set to 2013.3.0