batman-adv - Bug #372

Remove hash refcnt only when entry was removed from hash

02/21/2019 08:47 AM - Sven Eckelmann

Status:	Closed	**Start date:**	02/21/2019
Priority:	Normal	**Due date:**	
Assignee:	batman-adv developers	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	2019.1		

Description

I just got following backtrace from Martin Weinelt:

```
[162531.744207] ------------[ cut here ]------------
[162531.745794] refcount_t: underflow; use-after-free.
[162531.747368] WARNING: CPU: 2 PID: 0 at /build/linux-Ut6wTa/linux-4.19.12/lib/refcount.c:187 ref
count_sub_and_test_checked+0x3e/0x50
[162531.750642] Modules linked in: ebt_mark ebt_arp ebtable_filter ebtables iptable_mangle ipt_REJ
ECT nf_reject_ipv4 xt_set iptable_filter xt_nat iptable_nat nf_nat_ipv4 xt_TCPMSS ip6table_mangle
ip6t_REJECT nf_reject_ipv6 xt_tcpudp xt_conntrack ip6table_filter ip6table_nat nf_nat_ipv6 nf_nat
ip6_tables ip_set_hash_net ip_set nfnetlink sb_edac crct10dif_pclmul crc32_pclmul ppdev ghash_clmu
lni_intelbochs_drm ttm drm_kms_helper pcspkr serio_raw evdev drm joydev virtio_balloon sg parport_
pc parport tun button nf_conntrack nf_defrag_ipv6 nf_defrag_ipv4 batman_adv(OE) bridge stp llc cfg
80211 rfkill libcrc32c ip_tables x_tables autofs4 ext4 crc16 mbcache jbd2 crc32c_generic fscrypto
ecb hid_generic usbhid hid crc32c_intel sr_mod cdrom virtio_net net_failover virtio_blk failover a
esni_intel aes_x86_64 psmouse
[162531.769771]  crypto_simd cryptd glue_helper ata_generic uhci_hcd ehci_hcd ata_piix libata usbc
ore scsi_mod virtio_pci virtio_ringvirtio i2c_piix4 usb_common floppy
[162531.774446] CPU: 2 PID: 0 Comm: swapper/2 Tainted: G           OE     4.19.0-0.bpo.1-amd64 #1
Debian 4.19.12-1~bpo9+1
[162531.777179] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1 04/01/2014
[162531.779097] RIP: 0010:refcount_sub_and_test_checked+0x3e/0x50
[162531.780419] Code: 75 0c f0 0f b1 16 75 27 85 d2 0f 94 c0 c3 80 3d db 18 d3 00 00 75 15 48 c7 c
7 e8 91 c7 90 c6 05 cb 18 d3 00 01 e8 d2 52 cb ff <0f> 0b 31 c0 c3 83 f8 ff 75 bf eb f6 66 0f 1f 4
4 00 00 48 89 fe bf
[162531.784594] RSP: 0018:ffff8b129e283a68 EFLAGS: 00010282
[162531.785832] RAX: 0000000000000000 RBX: ffff8b1299578c80 RCX: 0000000000000006
[162531.787442] RDX: 0000000000000007 RSI: 0000000000000082 RDI: ffff8b129e2966a0
[162531.789112] RBP: 0000000000000000 R08: 0000000000000000 R09: 00000000000003e3
[162531.790734] R10: 000000006130427b R11: 0000000000000001 R12: ffff8b129c0c78c0
[162531.792355] R13: ffff8b12756ab600 R14: ffff8b1299578c80 R15: ffff8b129d7c4380
[162531.793996] FS:  0000000000000000(0000) GS:ffff8b129e280000(0000) knlGS:0000000000000000
[162531.797386] CS:  0010 DS: 0000 ES: 0000 CR0: 0000000080050033
[162531.799584] CR2: 000055716132e468 CR3: 000000001220a003 CR4: 0000000000060ee0
[162531.802086] Call Trace:
[162531.804178]  <IRQ>
[162531.805670]  batadv_tt_global_entry_put+0x12/0x40 [batman_adv]
[162531.807971]  _batadv_tt_update_changes+0x452/0x490 [batman_adv]
[162531.810300]  batadv_tt_tvlv_ogm_handler_v1+0x1ed/0x440 [batman_adv]
[162531.812716]  ? __kmalloc_reserve.isra.48+0x2e/0x80
[162531.814778]  batadv_tvlv_containers_process+0xea/0x170 [batman_adv]
[162531.817175]  batadv_tvlv_ogm_receive+0x30/0x40 [batman_adv]
[162531.819430]  batadv_iv_ogm_process_per_outif+0xa31/0xc60 [batman_adv]
[162531.821874]  ? batadv_orig_hash_find+0xed/0x100 [batman_adv]
[162531.824205]  ? batadv_iv_ogm_orig_get+0x1a/0x190 [batman_adv]
[162531.826551]  batadv_iv_ogm_receive+0x34d/0x420 [batman_adv]
[162531.828903]  batadv_batman_skb_recv+0x102/0x160 [batman_adv]
[162531.831326]  __netif_receive_skb_one_core+0x52/0x70
[162531.833573]  netif_receive_skb_internal+0x34/0xe0
[162531.835715]  napi_gro_receive+0xb8/0xe0
[162531.837695]  receive_buf+0x3c1/0x1310 [virtio_net]
[162531.839913]  ? vring_unmap_one+0x16/0x70 [virtio_ring]
[162531.842251]  ? detach_buf+0x68/0x110 [virtio_ring]
```

```
[162531.844731]  virtnet_poll+0xc2/0x303 [virtio_net]
[162531.846913]  net_rx_action+0x297/0x400
[162531.848853]  __do_softirq+0x10d/0x2c3
[162531.850842]  irq_exit+0xc2/0xd0
[162531.852716]  do_IRQ+0x52/0xe0
[162531.854543]  common_interrupt+0xf/0xf
[162531.856471]  </IRQ>
[162531.857995] RIP: 0010:native_safe_halt+0x2/0x10
[162531.860186] Code: 6f ff ff ff 7f c3 65 48 8b 04 25 40 5c 01 00 f0 80 48 02 20 48 8b 00 a8 08 7
4 8c eb c2 f3 c3 90 90 90 90 90 90 90 90 90 fb f4 <c3> 0f 1f 00 66 2e 0f 1f 84 00 00 00 00 00 f4 c
3 90 90 90 90 90 90
[162531.867297] RSP: 0018:ffffb1508038fe90 EFLAGS: 00000246 ORIG_RAX: ffffffffffffffd9
[162531.870219] RAX: ffffffff90518670 RBX: 0000000000000002 RCX: ffff8b129e29a720
[162531.872992] RDX: ffffffff90e4adb8 RSI: ffff8b129e29a720 RDI: 0000000000000082
[162531.875489] RBP: 0000000000000002 R08: 000127aa4740cf62 R09: 0000000000000007
[162531.878113] R10: 0000000000000002 R11: 000000000000013a R12: ffff8b129d4d0ec0
[162531.880643] R13: ffff8b129d4d0ec0 R14: 0000000000000000 R15: 0000000000000000
[162531.883093]  ? __sched_text_end+0x3/0x3
[162531.884841]  default_idle+0x1c/0x140
[162531.886492]  do_idle+0x1c6/0x280
[162531.888048]  cpu_startup_entry+0x6f/0x80
[162531.889749]  start_secondary+0x1a4/0x1f0
[162531.891426]  secondary_startup_64+0xa4/0xb0
[162531.893201] ---[ end trace 639d397811967375 ]---
```

I don't know at the moment what is causing this but my idea was related to the case that batadv_tt_global_free doesn't check whether the entry was actually removed or whether it was already removed before.

Here are some discussion things from IRC:

```
batadv_tt_global_free looks weird. shouldn't it only put the reference for the hash when it was re
ally removed from the hash?
otherwise I don't see why batadv_tt_global_del might not be called twice for the same entry at the
 same time
and have one of them scheduled after batadv_tt_global_hash_find while the other one then starts to
 delete the entry. the second caller will then also try to remove the entry from the hash (doesn't
 do anything) but then still reduce the refcnt
in our case, it is most likely slightly more complicated. but _batadv_tt_update_changes is also ca
lling batadv_tt_global_del.
```

History

#1 - 02/23/2019 12:29 PM - Sven Eckelmann

I found three places which have this (or a very similar) problem with batadv_hash_remove:

```
diff --git a/net/batman-adv/bridge_loop_avoidance.c b/net/batman-adv/bridge_loop_avoidance.c
index ef39aabd..f56464a1 100644
--- a/net/batman-adv/bridge_loop_avoidance.c
+++ b/net/batman-adv/bridge_loop_avoidance.c
@@ -815,6 +815,8 @@ static void batadv_bla_del_claim(struct batadv_priv *bat_priv,

     batadv_hash_remove(bat_priv->bla.claim_hash, batadv_compare_claim,
                batadv_choose_claim, claim);
+
+    // TODO: Fixes: a9ce0dc43e2c ("batman-adv: add basic bridge loop avoidance code")
     batadv_claim_put(claim); /* reference from the hash is gone */

     /* don't need the reference from hash_find() anymore */
diff --git a/net/batman-adv/translation-table.c b/net/batman-adv/translation-table.c
index f73d7913..3e621843 100644
--- a/net/batman-adv/translation-table.c
+++ b/net/batman-adv/translation-table.c
@@ -623,6 +623,7 @@ static void batadv_tt_global_free(struct batadv_priv *bat_priv,

     batadv_hash_remove(bat_priv->tt.global_hash, batadv_compare_tt,
```

```
                batadv_choose_tt, &tt_global->common);
+    // TODO: Fixes: 7bad46397eff ("batman-adv: protect the local and the global trans-tables with rcu")
     batadv_tt_global_entry_put(tt_global);
 }

@@ -1375,6 +1376,8 @@ u16 batadv_tt_local_remove(struct batadv_priv *bat_priv, const u8 *addr,
     if (!tt_entry_exists)
         goto out;

+    // TODO: Fixes: af912d77181f ("batman-adv: protect tt_local_entry from concurrent delete events")
+
     /* extra call to free the local tt entry */
     batadv_tt_local_entry_put(tt_local_entry);
```

#2 - 02/23/2019 03:11 PM - Sven Eckelmann

- % Done changed from 0 to 100

- Target version set to 2019.1

- Status changed from New to Resolved

Submitted patches (unfortunately with a minor problem in my git setup):

- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/20190223140906.28979-1-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/20190223140906.28979-2-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/20190223140906.28979-3-sven@narfation.org/

#3 - 03/03/2019 06:15 PM - Sven Eckelmann

- Status changed from Resolved to Closed

Added for 2019.1