batman-adv - Bug #244

kref_get_unless_zero without visible rcu_read_lock section

03/05/2016 11:13 AM - Sven Eckelmann

Status:	Closed	**Start date:**	03/05/2016
Priority:	Normal	**Due date:**	
Assignee:	Marek Lindner	**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	2016.2		

Description

Andrew detected one instance of a kref_get_unless_zero which didn't seem to belong to a rcu_read_lock protected section. I went through the rest of the code and annotated things which look similar.

```
diff --git a/net/batman-adv/bat_iv_ogm.c b/net/batman-adv/bat_iv_ogm.c
index 2c65668..a27fda0 100644
--- a/net/batman-adv/bat_iv_ogm.c
+++ b/net/batman-adv/bat_iv_ogm.c
@@ -679,9 +679,11 @@ static void batadv_iv_ogm_aggregate_new(const unsigned char *packet_buff,
     unsigned char *skb_buff;
     unsigned int skb_size;

+    /* TODO */
     if (!kref_get_unless_zero(&if_incoming->refcount))
         return;

+    /* TODO */
     if (!kref_get_unless_zero(&if_outgoing->refcount))
         goto out_free_incoming;

diff --git a/net/batman-adv/gateway_client.c b/net/batman-adv/gateway_client.c
index c59aff5..fafc053 100644
--- a/net/batman-adv/gateway_client.c
+++ b/net/batman-adv/gateway_client.c
@@ -135,6 +135,7 @@ static void batadv_gw_select(struct batadv_priv *bat_priv,

     spin_lock_bh(&bat_priv->gw.list_lock);

+    /* TODO */
     if (new_gw_node && !kref_get_unless_zero(&new_gw_node->refcount))
         new_gw_node = NULL;

@@ -440,6 +441,7 @@ static void batadv_gw_node_add(struct batadv_priv *bat_priv,
     if (gateway->bandwidth_down == 0)
         return;

+    /* TODO */
     if (!kref_get_unless_zero(&orig_node->refcount))
         return;

diff --git a/net/batman-adv/hard-interface.c b/net/batman-adv/hard-interface.c
index 3240a67..a4c488d 100644
--- a/net/batman-adv/hard-interface.c
+++ b/net/batman-adv/hard-interface.c
@@ -236,6 +236,7 @@ static void batadv_primary_if_select(struct batadv_priv *bat_priv,

     ASSERT_RTNL();

+    /* TODO */
     if (new_hard_iface && !kref_get_unless_zero(&new_hard_iface->refcount))
         new_hard_iface = NULL;

@@ -464,6 +465,7 @@ int batadv_hardif_enable_interface(struct batadv_hard_iface *hard_iface,
```

```
     if (hard_iface->if_status != BATADV_IF_NOT_IN_USE)
         goto out;

+    /* TODO */
     if (!kref_get_unless_zero(&hard_iface->refcount))
         goto out;

diff --git a/net/batman-adv/main.c b/net/batman-adv/main.c
index d64ddb9..89bc5b1 100644
--- a/net/batman-adv/main.c
+++ b/net/batman-adv/main.c
@@ -736,6 +736,7 @@ static struct batadv_tvlv_container
         if (tvlv_tmp->tvlv_hdr.version != version)
             continue;

+        /* TODO */
         if (!kref_get_unless_zero(&tvlv_tmp->refcount))
             continue;

diff --git a/net/batman-adv/network-coding.c b/net/batman-adv/network-coding.c
index 32f9fa1..e6dc208 100644
--- a/net/batman-adv/network-coding.c
+++ b/net/batman-adv/network-coding.c
@@ -856,6 +856,7 @@ static struct batadv_nc_node
     if (!nc_node)
         return NULL;

+    /* TODO */
     if (!kref_get_unless_zero(&orig_neigh_node->refcount))
         goto free;

diff --git a/net/batman-adv/originator.c b/net/batman-adv/originator.c
index e63d6a5..92e6a41 100644
--- a/net/batman-adv/originator.c
+++ b/net/batman-adv/originator.c
@@ -381,6 +381,7 @@ batadv_orig_ifinfo_new(struct batadv_orig_node *orig_node,
     if (!orig_ifinfo)
         goto out;

+    /* TODO */
     if (if_outgoing != BATADV_IF_DEFAULT &&
         !kref_get_unless_zero(&if_outgoing->refcount)) {
         kfree(orig_ifinfo);
@@ -462,6 +463,7 @@ batadv_neigh_ifinfo_new(struct batadv_neigh_node *neigh,
     if (!neigh_ifinfo)
         goto out;

+    /* TODO */
     if (if_outgoing && !kref_get_unless_zero(&if_outgoing->refcount)) {
         kfree(neigh_ifinfo);
         neigh_ifinfo = NULL;
@@ -539,6 +541,7 @@ batadv_hardif_neigh_create(struct batadv_hard_iface *hard_iface,
     if (hardif_neigh)
         goto out;

+    /* TODO */
     if (!kref_get_unless_zero(&hard_iface->refcount))
         goto out;

@@ -650,6 +653,7 @@ batadv_neigh_node_new(struct batadv_orig_node *orig_node,
     if (!neigh_node)
         goto out;

+    /* TODO */
     if (!kref_get_unless_zero(&hard_iface->refcount)) {
         kfree(neigh_node);
         neigh_node = NULL;
```

```
diff --git a/net/batman-adv/routing.c b/net/batman-adv/routing.c
index 45093c6..39a2e04 100644
--- a/net/batman-adv/routing.c
+++ b/net/batman-adv/routing.c
@@ -101,6 +101,7 @@ static void _batadv_update_route(struct batadv_priv *bat_priv,
         batadv_neigh_node_put(curr_router);

     /* increase refcount of new best neighbor */
+    /* TODO */
     if (neigh_node && !kref_get_unless_zero(&neigh_node->refcount))
         neigh_node = NULL;
```

My proposal was to use kref_get when we already have a valid reference and to move the kref_get_unless_zero up to the rcu_read_lock section code (when there is one). This has the benefit that kref_get will warn us when the reference counting broke end we try to get a reference when the reference counter already reached zero (which should not happen in this situation).

History

#1 - 03/05/2016 04:11 PM - Sven Eckelmann

- Assignee changed from Sven Eckelmann to Marek Lindner

- Status changed from New to In Progress

The changes are now in ecsv/kref_get_unless_zero. The patches were sent to the mailing list as

- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-1-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-2-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-3-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-4-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-5-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-6-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-7-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-8-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457190564-11419-8-git-send-email-sven@narfation.org/
- https://patchwork.open-mesh.org/project/b.a.t.m.a.n./patch/1457201124-8495-1-git-send-email-sven@narfation.org/

#2 - 05/02/2016 09:06 PM - Sven Eckelmann

- Status changed from In Progress to Closed

Patches were accepted

#3 - 02/11/2017 08:37 AM - Sven Eckelmann

- Target version set to 2016.2

#4 - 05/27/2020 10:23 PM - Sven Eckelmann

- Description updated