
Subject: [PATCH][NEIGH] Fix race between neigh_parms_release and neightbl_fill_parms

Posted by [Pavel Emelianov](#) on Thu, 10 Jan 2008 10:56:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

The neightbl_fill_parms() is called under the write-locked tbl->lock and accesses the parms->dev. The neigh_parm_release() calls the dev_put(parms->dev) without this lock. This creates a tiny race window on which the parms contains potentially stale dev pointer.

To fix this race it's enough to move the dev_put() upper under the tbl->lock, but note, that the parms are held by neighbors and thus can live after the neigh_parms_release() is called, so we still can have a parm with bad dev pointer.

I didn't find where the neigh->parms->dev is accessed, but still think that putting the dev is to be done in a place, where the parms are really freed. Am I right with that?

Signed-off-by: Pavel Emelyanov <xemul@openvz.org>

diff --git a/net/core/neighbour.c b/net/core/neighbour.c
index 29b8ee4..cc8a2f1 100644

--- a/net/core/neighbour.c

+++ b/net/core/neighbour.c

@@ -1316,8 +1316,6 @@ void neigh_parms_release(struct neigh_table *tbl, struct neigh_parms *parms)

```
    *p = parms->next;
    parms->dead = 1;
    write_unlock_bh(&tbl->lock);
-   if (parms->dev)
-       dev_put(parms->dev);
    call_rcu(&parms->rcu_head, neigh_rcu_free_parms);
    return;
}
```

@@ -1328,6 +1326,8 @@ void neigh_parms_release(struct neigh_table *tbl, struct neigh_parms *parms)

```
void neigh_parms_destroy(struct neigh_parms *parms)
{
+   if (parms->dev)
+       dev_put(parms->dev);
    kfree(parms);
}
```
