

---

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 Emelianov <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);  
}
```

---

Subject: Re: [PATCH][NEIGH] Fix race between neigh\_parms\_release and neighbl\_fill\_parms

Posted by [davem](#) on Thu, 10 Jan 2008 11:50:32 GMT

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

---

From: Pavel Emelyanov <xemul@openvz.org>

Date: Thu, 10 Jan 2008 13:56:53 +0300

> The neighbl\_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>

It is accessed in lookup\_neigh\_parms(), neighbl\_fill\_parms(), and  
neighbl\_fill\_info() (hmmm, that BUG\_ON(tbl->parms.dev) is cute).

You fix looks correct, patch applied, thanks!

---