Subject: Re: Re: [PATCH 2/5] net: Make rtnetlink infrastructure network namespace aware
Posted by den on Mon, 01 Oct 2007 08:26:52 GMT
View Forum Message <> Reply to Message

Patrick McHardy wrote:
> Eric W. Biederman wrote:
>> Patrick McHardy <kaber@trash.net> writes:
>>
>>
>>> Maybe I can save you some time: we used to do down_trylock()
>>> for the rtnl mutex, so senders would simply return if someone
>>> else was already processing the queue *or* the rtnl was locked
>>> for some other reason. In the first case the process already
>>> processing the queue would also process the new messages, but
>>> if it the rtnl was locked for some other reason (for example
>>> during module registration) the message would sit in the
>>> queue until the next rtnetlink sendmsg call, which is why
>>> rtnl_unlock does queue processing. Commit 6756ae4b changed
>>> the down_trylock to mutex_lock, so senders will now simply wait
>>> until the mutex is released and then call netlink_run_queue
>>> themselves. This means its not needed anymore.
>>
>> Sounds reasonable.
>>
>> I started looking through the code paths and I currently cannot
>> see anything that would leave a message on a kernel rtnl socket.
>>
>> However I did a quick test adding a WARN_ON if there were any messages
>> found in the queue during rtnl_unlock and I found this code path
>> getting invoked from linkwatch_event. So there is clearly something I
>> don't understand, and it sounds at odds just a bit from your
>> description.
>
>
> That sounds like a bug. Did you place the WARN_ON before or after
> the mutex_unlock()?

The presence of the message in the queue during rtnl_unlock is quite
possible as normal user->kernel message processing path for rtnl is the
following:

```
netlink_sendmsg
  netlink_unicast
    netlink_sendskb
       skb_queue_tail
       netlink_data_ready
          rtnetlink_rcv
		mutex_lock(&rtnl_mutex);
		netlink_run_queue(sk, qlen, &rtnetlink_rcv_msg);
		mutex_unlock(&rtnl_mutex);
```

so, the presence of the packet in the rtnl queue on rtnl_unlock is normal race with a rtnetlink_rcv for me.

Regards,
 Den