Subject: Re: Re: [PATCH 2/5] net: Make rtnetlink infrastructure network namespace aware
Posted by ebiederm on Mon, 01 Oct 2007 08:45:07 GMT
View Forum Message <> Reply to Message

"Denis V. Lunev" <den@sw.ru> writes:

> 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.

Yes. That is what I saw in practice as well.
Thanks for confirming this.

It happened to reproducible because I had a dhcp client asking
for a list of links in parallel with the actual link coming up
during boot.

Looking at netlink_unicast and netlink_broadcast I am generally
convinced that we can remove the call of sk_data_ready in
rtnl_unlock.   I think those are the only two possible paths
through there and I don't see how we could miss a processing a
packet on the way through there.

What would be nice is if we could figure out how to eliminate
this race. As that would allow netlink packets to be processed
synchronously and we could actually use current for security
checks, and for getting the context of the calling process.

Right now we are 99% of the way there but because of the above
race the code must all be written as if netlink packets were coming
in completely asynchronously. Which is unfortunate and a pain.

Eric

_____

Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers