Subject: Re: [patch 1/1][NETNS][IPV6] protect addrconf from loopback registration
Posted by Daniel Lezcano on Mon, 12 Nov 2007 17:01:22 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> "Denis V. Lunev" <den@sw.ru> writes:
>
>>> Index: linux-2.6-netns/net/ipv6/addrconf.c
>>> ===
>>> --- linux-2.6-netns.orig/net/ipv6/addrconf.c
>>> +++ linux-2.6-netns/net/ipv6/addrconf.c
>>> @@ -2272,7 +2272,8 @@ static int addrconf_notify(struct notifi
>>>
>>>   switch(event) {
>>>   case NETDEV_REGISTER:
>>> - if (!idev && dev->mtu >= IPV6_MIN_MTU) {
>>> + if (!(dev->flags & IFF_LOOPBACK) &&
>>> +   !idev && dev->mtu >= IPV6_MIN_MTU) {
>
> It is idev being true here for the loopback device that would
> prevent things not missing the REGISTER event.
>
> Hmm. But we do call ipv6_add_dev on loopback and now the loopback
> device is practically guaranteed to be the first device so we can
> probably just remove the special case in addrconf_init.
>
> Anyway Daniels patch makes increasingly less sense the more I look
> at it.

Let me try to clarify:

  * when the init network namespace is created, the loopback is created
first, before ipv6, and the notifier call chain for ipv6 is not setup,
so the protocol does not receive the REGISTER event

  * when the init network namespace is destroyed during shutdown, the
loopback is not unregistered, so there is no UNREGISTER event

  * when we create a new network namespace, a new instance of the
loopback is created and a NETDEV_REGISTER is sent to ipv6 because the
notifier call chain has been setup by the init netns (while ipv6
protocol is not yet configured for the namespace which is being created)

  * when the network namespace exits, the loopback is unregistered after
the ipv6 protocol but the NETDEV_UNREGISTER is sent to addrconf_notify
while the ipv6 protocol has been destroyed.

The objective of the patch is to discard these events because they were never taken into account and they are not expected to be receive by ipv6 protocol.

_____