Subject: Re: [patch 1/1][NETNS][IPV6] protect addrconf from loopback registration
Posted by ebiederm on Mon, 12 Nov 2007 16:40:32 GMT
View Forum Message <> Reply to Message

Daniel Lezcano <dlezcano@fr.ibm.com> writes:

> The loopback is now dynamically allocated. The ipv6 code was written
> considering the loopback is allocated before the ipv6 protocol
> initialization. This is still the case when we don't use multiple
> network namespaces.

You do know that register_netdevice_notifier delivers events
REGISTER and UP events for devices that are already up?

Thinking about it I wonder if unregister_netdevice_notifier should
actually deliver UNREGISTER events. It wouldn't change the ipv6
case as I don't believe you can unregister ipv6.

> In the case of the network namespaces, ipv6 notification handler is
> already setup and active (done by the initial network namespace),
> so when a network namespace is created, a new instance of the
> loopback device, via dynamic allocation, will trigger a REGISTER event
> to addrconf_notify and this one will try to setup the network device
> while the ipv6 protocol is not yet initialized for the network namespace.

Ok. This sounds like a race in ipv6 that should get fixed.

I know last time my patchset covered ipv6 I did send patches for several
reference counting problems. I'm surprised something bad still exists.

Anyway let's not patch around this and fix whatever the real problem.

> Because the ipv6 is relying on the fact that the loopback device will
> not trigger REGISTER/UNREGISTER events, I just protect the addrconf_notify
> function when the loopback register event is triggered.

This can't be the case REGISTER events happen.

> In the case of multiple network namespaces, the usual ipv6 protocol
> initialization will be done after the loopback initialization with
> the subsystem registration mechanism.
>
> Signed-off-by: Daniel Lezcano <dlezcano@fr.ibm.com>
> Signed-off-by: Benjamin Thery <benjamin.thery@bull.net>
> ---
> net/ipv6/addrconf.c | 9 +++++++--
> 1 file changed, 7 insertions(+), 2 deletions(-)

>
> 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) {
>   idev = ipv6_add_dev(dev);
>   if (!idev)
>   return notifier_from_errno(-ENOMEM);

This hunk is clearly bogus.

> @@ -2366,11 +2367,15 @@ static int addrconf_notify(struct notifi
>   /* MTU falled under IPV6_MIN_MTU. Stop IPv6 on this
> interface. */
>
> case NETDEV_DOWN:
> + addrconf_ifdown(dev, 0);
> + break;
> +
> case NETDEV_UNREGISTER:
>   /*
>   * Remove all addresses from this interface.
>   */
> - addrconf_ifdown(dev, event != NETDEV_DOWN);
> + if (!(dev->flags & IFF_LOOPBACK))
> + addrconf_ifdown(dev, 1);

I can see how this could be a problem.
>   break;
>
> case NETDEV_CHANGENAME:
>
> --