Subject: Re: [PATCH 1/3] [IPV6]: Event type in addrconf_ifdown is mis-used.
Posted by den on Sun, 23 Mar 2008 08:13:16 GMT
View Forum Message <> Reply to Message

On Sat, 2008-03-22 at 17:38 -0700, David Miller wrote:
> From: "Denis V. Lunev" <den@openvz.org>
> Date: Tue, 18 Mar 2008 17:35:23 +0300
>
> > addrconf_ifdown is broken in respect to the usage of how parameter. This
> > function is called with (event != NETDEV_DOWN) and (2) on the IPv6 stop.
> > It the latter case inet6_dev from loopback device should be destroyed.
> >
> > Signed-off-by: Denis V. Lunev <den@openvz.org>
>
> The code purposefully treats "2" specially because when IPV6 routes
> are destroyed they are changed to point to the loopback device's
> inet6_dev object.
>
> This allows statistic bumping code to not have to check if it has a
> NULL inet6_dev pointer or not, because that's now impossible.
>
> Since ipv6 is not unloadable, addrconf_cleanup(), and thus the
> "how == 2" case can only occur when ipv6 fails to load properly.
> The only real consequence of this bug is that if ipv6 fails
> to load properly, a subsequent successfull load of ipv6 will
> leak the loopback device's inet6_dev object, which isn't that
> much of a big deal.
>
> I understand that for namespaces you have to deal with multiple
> loopback devices, but you'll need to solve that problem while
> still handling the wish of the ipv6 stack for inet6_dev objects
> of loopback devices to be permanent and guarenteed to always
> be around for the sake of statistics bumping.

First, this behaviour is broken for a namespace right now in the 2.6.26
tree. inet6_dev pointer will be NULL for a loopback inside the
namespace. The case is simple. Just remove all INET6 addresses from a
loopback device inside a VE. This will call

```
  inet6_addr_del
    addrconf_ifdown(dev, 1);
      if (dev == init_net.loopback_dev && how == 1)
            how = 0;
```

the condition will be false and how will not be changed here.

Pls note, that ip6_dst_ifdown deals with a namespace loopback rather
than init_net loopback to track referrences of the namespace objects.
This allows us to catch refcounting bugs smoothly (see patch 3 in the
set).

That's why I have extended a special "2" case to really destroy
inet6_dev to have a way to destroy it. Generic code should not suffer
from this from my POW.

> I thus can't apply any of these patches until those issues are
> resolved.

IMHO special "2" case was intended to have a stub to unload the module
in the future.

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