Subject: Re: [PATCH 1/5] net: Modify all rtnetlink methods to only work in the initial namespace
Posted by Daniel Lezcano on Wed, 10 Oct 2007 14:05:26 GMT
View Forum Message <> Reply to Message

Denis V. Lunev wrote:

```
> Eric W. Biederman wrote:
>> Before I can enable rtnetlink to work in all network namespaces
>> I need to be certain that something won't break.  So this
>> patch deliberately disables all of the rtnletlink methods in everything
>> except the initial network namespace.  After the methods have been
>> audited this extra check can be disabled.
>>
> [...]
>>  static int br_dump_ifinfo(struct sk_buff *skb, struct netlink_callback *cb)
>>  {
>> + struct net *net = skb->sk->sk_net;
>>   struct net_device *dev;
>>   int idx;
>>
>
> I've read some code today greping 'init_net.loopback_dev' and found
> interesting non-trivial for me issue.
>
> Network namespace is extracted from the packet in two different ways in
> TCP. This is a socket for outgoing path and a device for incoming.
> Though, there are some places called uniformly both from incoming and
> outgoing path.
>
> Typical example is netfilters. They are called uniformly all around the
> code. The prototype is the following:
>
> static unsigned int reject6_target(struct sk_buff **pskb,
>                                    const struct net_device *in,
>                                    const struct net_device *out,
>                                    unsigned int hooknum,
>                                    const struct xt_target *target,
>                                    const void *targinfo);
>
```

Thanks Denis for auditing the code.

As far as I see, struct net_device *in is NULL for outgoing traffic and
struct net_device *out is NULL for ingress traffic. Except for the
FORWARD rules where both are filled. If we are following network
namespace semantic, we should not have two network devices belonging to
two differents namespaces, right ?
In this case, the following line of code should be sufficient to

retrieve the network namespace, no ?

```
struct net *net = in?in->nd_net:out->nd_net;
```

> So, we are bound to the following options:
> - perform additional non-uniform hacks around to place 'struct net' into
>   other and other structures like xt_target
> - add 7th parameter here and over
> - introduce an skb_net field in the 'struct sk_buff' making all code
>   uniform, at least when we have an skb
>
> I think that this is not the last place with such a parameter list and
> we should make a decision at this point when the code in not mainline yet.
>
> As far as I understand, netfilters are not touched by the Eric and we
> can face some non-trivial problems there.

In Eric's git tree:
http://git.kernel.org/?p=linux/kernel/git/ebiederm/linux-2.6-netns.git

There are some modifications concerning
net/ipv4/netfiler/iptable_filter.c and at the ipt_hook function, there is:

```
struct net *net = (in?in:out)->nd_net;
```

> So, if my point about uniformity is valid, this patchset looks wrong and
> should be re-worked :(

As Eric said, we want to build the network namespace step by step,
taking care of not breaking the init network namespace.

If you want to make iptables per namespace or catch problems before the
code goes to Dave's tree, IMHO it will be more convenient to post to
containers@ the patches against netns49, where the modifications will be
in a network namespace big picture.

Regards.

-- Daniel