Subject: Re: Network namespaces a path to mergable code.
Posted by Sam Vilain on Tue, 27 Jun 2006 22:20:39 GMT
View Forum Message <> Reply to Message

Andrey Savochkin wrote:
> On Tue, Jun 27, 2006 at 11:20:40AM -0600, Eric W. Biederman wrote:
>
>> Thinking about this I am going to suggest a slightly different direction
>> for get a patchset we can merge.
>>
>> First we concentrate on the fundamentals.
>> - How we mark a device as belonging to a specific network namespace.
>> - How we mark a socket as belonging to a specific network namespace.
>>
>
> I agree with the direction of your thoughts.
> I was trying to do a similar thing, define clear steps in network
> namespace merging.
>
> My first patchset covers devices but not sockets.
> The only difference from what you're suggesting is ipv4 routing.
> For me, it is not less important than devices and sockets. May be even
> more important, since routing exposes design deficiencies less obvious at
> socket level.
>

It sounds then like it would be a good start to have general socket
namespaces, if it would merge more easily - perhaps then network device
namespaces would fall into place more easily.

AIUI socket namespaces are also necessary for situations where you want
containers to share IP addresses. AIUI, PlanetLab do something like this
with a module atop of VServer already (but read
http://openvz.org/pipermail/devel/2006-June/000666.html for a proper
explanation from Mark Huang)

>> As part of the fundamentals we add a patch to the generic socket code
>> that by default will disable it for protocol families that do not indicate
>> support for handling network namespaces, on a non-default network namespace.
>>
>
> Fine
>
> Can you summarize you objections against my way of handling devices, please?
>
There were many objections, the major one being the patch was too large for certainty of
adequate review.

Quoting what I perceived as a summary from Eric:
> When I went through this, my patchset just added an explicit
> continue if the devices was not in the appropriate namespace.
> I actually prefer the multiple list implementation but at
> the same time I think it is harder to get a clean implementation
> out of it.

You offered to re-do the patch without separate lists - I suggest that
this go ahead. No-one should really care; splitting it out into separate
lists can then be considered a performance optimization for later.

> And what was the typo you referred to in your letter to Kirill Korotaev?
>
I think this is the comment he refers to:
> These hunks should use for_each_netdev(ifp);

Both quotes are from http://lkml.org/lkml/2006/6/26/147

Though, in Kirill's defense, it seems a bit strange to expect him to
raise a fault that was just raised by Eric, in a reply to the message
where he raised it.

Sam.