Subject: Re: [PATCH 10/12] L2 network namespace: playing with pass-through device
Posted by Vlad Yasevich on Wed, 13 Dec 2006 15:10:18 GMT
View Forum Message <> Reply to Message

Hi Daniel

Daniel Lezcano wrote:
> Herbert Poetzl wrote:
>> On Tue, Dec 12, 2006 at 04:50:50PM +0100, Daniel Lezcano wrote:
>>> Dmitry Mishin wrote:
>>>> On Tuesday 12 December 2006 17:19, Daniel Lezcano wrote:
>>>>> Dmitry Mishin wrote:
>>>>>
>>>>>>>> Why do yo need to have a child list and sibling list ?
>>>>>>> Because of the level2<->level3 hierarchy, for example.
>>>>>> This hierarchy doesn't exist with ns->parent ? Do you have an example
>>>>>> when the hierarchy should be used ? I mean when we need to browse from
>>>>>> l2 -> l3 ?
>>>>> For example, to check that new ifaddr is already used by child l3 namespace.
>>>>> The devinet isolation does already do that, you can not add a new ifaddr
>>>>> if it already exists. Do you have another example ?
>>>> Could devinet isolation provide ifaddrs list with namespaces?
>>>> What will be with child namespaces if you decide to destroy parent namespace?
>>>> If we decide to destroy them, than how we could get their list?
>>>> It is a question of flexibility and easy management.
>>>> Why do you want to remove this code?
>>> I don't want to especially remove this code, I just want to understand
>>> what it does and why. If it appears to be useless, let's remove it, if
>>> it appears to be useful, let's keep it.
>>>
>>> By the way, what is the meaning on destroying the namespaces directly,
>>> is it not the kref mechanism which needs to do that ? For example, if
>>> you create a l2 namespace and after you create l3 namespaces. You want
>>> to destroy the l2 namespace, the l2 namespace should stay "zombie" until
>>> all the l3 namespaces exit. If you need to wipe out all the namespaces,
>>> you should destroy all the related namespaces' ressources, like killing
>>> all processes inside it. The namespaces will "put" their respective kref
>>> and will trigger the freeing of the ressources.
>> networking (mostly sockets) will probably require
>> some mechanism to 'zap' them, ignoring the defined
>> timeouts. otherwise the spaces could hang around
>> for quite a while waiting for some response, which
>> might never come ...
>
> Yes, exact. We will need a specific socket cleanup by namespace in order
> to do network migration. This is the only case I see to 'zap' the sockets.
> The sockets should never be flushed in other cases. For example, you

> launch an application into a network namespace, it sends 10MB to a peer
> and exits. The network namespace should stay "alive" until all orphans
> sockets have flushed their buffers to the peer. This behavior is
> perfectly handled by the kref mechanism because sock_release will "put"
> the network namespace and that will trigger the network namespace
> destruction.
>

Are you saying that you can't see the reason to be able to shutdown/destroy a
given container. What if it's misbehaving or has been compromised???

I would think an administrator, should be able to shutdown/destroy a
given container or namespace from above or outside of such container/namespace
if it's warranted. If this case, if we destroy an L2 namespace, L3 children
should probably be cleaned up as well.

Thanks
-vlad

_____