Subject: Re: strict isolation of net interfaces
Posted by ebiederm on Fri, 30 Jun 2006 17:58:47 GMT
View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
>>
>>>Serge E. Hallyn wrote:
>>>
>>>>Quoting Cedric Le Goater (clg@fr.ibm.com):
>>>>
>>>>
>>>>>we could work on virtualizing the net interfaces in the host, map them to
>>>>>eth0 or something in the guest and let the guest handle upper network layers
> ?
>>>>>
>>>>>lo0 would just be exposed relying on skbuff tagging to discriminate traffic
>>>>>between guests.
>>>>
>>>>This seems to me the preferable way. We create a full virtual net
>>>>device for each new container, and fully virtualize the device
>>>>namespace.
>>>
>>>I have a few questions about all the network isolation stuff:
>>
>
> It seems these questions are not important.

I'm just trying to get us back to a productive topic.

>> So far I have seen two viable possibilities on the table,
>> neither of them involve multiple names for a network device.
>> layer 3 (filtering the allowed ip addresses at bind time roughly the current
>> vserver).
>>   - implementable as a security hook.
>>   - Benefit no measurable performance impact.
>>   - Downside not many things we can do.
>
> What things ? Can you develop please ? Can you give some examples ?

DHCP, tcpdump,.. Probably a bad way of phrasing it. But there
is a lot more that we can do using a pure layer 2 approach.

>> layer 2 (What appears to applications a separate instance of the network
>> stack).
>>   - Implementable as a namespace.

>
> what about accessing a NFS mounted outside the container ?

As I replied earlier it isn't a problem. If you get to it through the
filesystem namespace it uses the network namespace it was mounted with
for it's connection.

>>   - Each network namespace would have dedicated network devices.
>>   - Benefit extremely flexible.
>
> For what ? For who ? Do you have examples ?

See above.

>>   - Downside since at least the slow path must examine the packet
>>   it has the possibility of slowing down the networking stack.
>
> What is/are the slow path(s) you identified ?

Grr. I put that badly. Basically at least on the slow path you need to
look at a per network namespace data structure. The extra pointer
indirection could slow things down. The point is that we may be
able to have a fast path that is exactly the same as the rest
of the network stack.

If the obvious approach does not work my gut the feeling the
network stack fast path will give us an implementation without overhead.

>> For me the important characteristics.
>> - Allows for application migration, when we take our ip address with us.
>>   In particular it allows for importation of addresses assignments
>>   mad on other machines.
>
> Ok for the two methods no ?

So far.

>> - No measurable impact on the existing networking when the code
>>   is compiled in.
>
> You contradict ...

How so? As far as I can tell this is a basic requirement to get
merged.

>> - Clean predictable semantics.
>
> What that means ? Can you explain, please ?

>> This whole debate on network devices show up in multiple network namespaces
>> is just silly.
>
> The debate is not on the network device show up. The debate is can we have a
> network isolation ___usable for everybody___ not only for the beauty of having
> namespaces and for a system container like.

This subthread talking about devices showing up in multiple namespaces seemed very much exactly on how network devices show up.

> I am not against the network device virtualization or against the namespaces. I
> am just asking if the namespace is the solution for all the network
> isolation. Should we nest layer 2 and layer 3 vitualization into namespaces or
> separate them in order to have the flexibility to choose isolation/performance.

I believe I addressed Herbert Poetzl's concerns earlier. To me the question is can we implement an acceptable layer 2 solution, that distrubutions and other people who do not need isolation would have no problem compiling in by default.

The joy of namespaces is that if you don't want it you don't have to use it. Layer 2 can do everything and is likely usable by everyone iff the performance is acceptable.

>> The only reason for wanting that appears to be better management.
>> We have deeper issues like can we do a reasonable implementation without a
>> network device showing up in multiple namespaces.
>
> Again, I am not against having the network device virtualization. It is a good
> idea.
>
>> I think the reason the debate exists at all is that it is a very approachable
>> topic, as opposed to the fundamentals here.
>> If we can get layer 2 level isolation working without measurable overhead
>> with one namespace per device it may be worth revisiting things. Until
>> then it is a side issue at best.
>
> I agree, so where are the answers of the questions I asked in my previous email
> ? You said you did some implementation of network isolation with and without
> namespaces, so you should be able to answer...

Sorry. More than anything those questions looked retorical and aimed at disarming some of the silliness. I will go back and try and answer those. Fundamentally when we have one namespace that includes network devices, network sockets, and all of the data structures necessary to use them (routing tables and the like) and we have a tunnel device that can connect namespaces the answers are trivial and I though obvious.

Eric