
Subject: Re: [RFD] L2 Network namespace infrastructure

Posted by [serue](#) on Mon, 25 Jun 2007 15:23:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Jeff Garzik (jeff@garzik.org):

> Eric W. Biederman wrote:

> >Jeff Garzik <jeff@garzik.org> writes:

> >

> >>David Miller wrote:

> >>>I don't accept that we have to add another function argument

> >>>to a bunch of core routines just to support this crap,

> >>>especially since you give no way to turn it off and get

> >>>that function argument slot back.

> >>>

> >>>To be honest I think this form of virtualization is a complete

> >>>waste of time, even the openvz approach.

> >>>

> >>>We're protecting the kernel from itself, and that's an endless

> >>>uphill battle that you will never win. Let's do this kind of

> >>>stuff properly with a real minimal hypervisor, hopefully with

> >>>appropriate hardware level support and good virtualized device

> >>>interfaces, instead of this namespace stuff.

> >>Strongly seconded. This containerized virtualization approach just

> >>bloats up

> >>the kernel for something that is inherently fragile and IMO less secure --

> >>protecting the kernel from itself.

> >>

> >>Plenty of other virt approaches don't stir the code like this, while

> >>simultaneously providing fewer, more-clean entry points for the

> >>virtualization

> >>to occur.

> >

> >Wrong. I really don't want to get into a my virtualization approach is

> >better

> >then yours. But this is flat out wrong.

>

> >99% of the changes I'm talking about introducing are just:

> >- variable

> >+ ptr->variable

> >

> >There are more pieces mostly with when we initialize those variables but

> >that is the essence of the change.

>

> >You completely dodged the main objection. Which is OK if you are

> >selling something to marketing departments, but not OK

>

> >Containers introduce chroot-jail-like features that give one a false

> >sense of security, while still requiring one to "poke holes" in the

> illusion to get hardware-specific tasks accomplished.
>
> The capable/not-capable model (i.e. superuser / normal user) is still
> being secured locally, even after decades of work and whitepapers and
> audits.
>
> You are drinking Deep Kool-Aid if you think adding containers to the
> myriad kernel subsystems does anything besides increasing fragility, and
> decreasing security. You are securing in-kernel subsystems against
> other in-kernel subsystems.

No we're not. As the name 'network namespaces' implies, we are introducing namespaces for network-related variables. That's it.

We are not trying to protect in-kernel subsystems from each other. In fact we're not even trying to protect userspace process from each other. Though that will in part come free when user processes can't access each other's data because they are in different namespaces. But using an LSM like selinux or a custom one to tag and enforce isolation would still be encouraged.

> superuser/user model made that difficult
> enough... now containers add exponential audit complexity to that. Who
> is to say that a local root does not also pierce the container model?

At the moment it does.

> >And as opposed to other virtualization approaches so far no one has been
> >able to measure the overhead. I suspect there will be a few more cache
> >line misses somewhere but they haven't shown up yet.
> >
> >If the only use was strong isolation which Dave complains about I would
> >concur that the namespace approach is inappropriate. However there are
> >a lot other uses.
>
> Sure there are uses. There are uses to putting the X server into the
> kernel, too. At some point complexity and featuritis has to take a back
> seat to basic sanity.

Generally true, yes.

-serge

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