Subject: Re: Re: [RFD] L2 Network namespace infrastructure
Posted by dev on Wed, 27 Jun 2007 15:38:35 GMT
View Forum Message <> Reply to Message

Jeff Garzik wrote:
> 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

It is a pure illusion that one kind of virtualization is better then the other one.
Look, maybe *hardware* virtualization and a small size of the hypervisor
make you feel safer, however you totally forget about all the emulation drivers etc.
which can have bugs and security implications as well and maybe triggerable
from inside VM.

> 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 concepts of users in any OS (and in Linux in particular) give people
a false sense of security as well!
I know a bunch of examples of how one user can crash/DoS/abuse another one
e.g. on RHEL5 or mainstream.
But no one have problems with that and no one thinks that multiuserness
is a step-backward or maybe someone does?!

Yes, there are always bugs and sometimes security implications,
but people leave fine with it when it is fixed.

> 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. 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?

containers do the only thing:
make sure that objects from one context are not visible to another one.

If containers are not used - everything returns to the case as it is now, i.e.
everything is global and globally visible and no auditing overhead at all.
So if you are not interested in containers - the code auditing
won't be noticably harder for you.

>>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.

I agree about sanity, however I totally disagree about complexity
you talk about.
Bugs we face/fix in Linux kernel which are found with the help of
that kind of virtualization makes me believe that Linux kernel
only wins from it.

Thanks,
Kirill

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