Subject: Re: LSM and Containers (was: LSM conversion to static interface)
Posted by serue on Tue, 23 Oct 2007 13:32:10 GMT
View Forum Message <> Reply to Message

Quoting Crispin Cowan (crispin@crispincowan.com):
> Peter, I may be mistaken, but I think you are talking about an entirely
> different issue than the LSM static interface issue, so I've changed the
> subject.
>
> Peter Dolding wrote:
> > You are all focusing on vendors. I am thinking server farm or people
> > running many different distros side by side using containers.
> This right here is a challenging goal.
>
> It completely surprises me that anyone would consider trying to run
> different distros in different containers.

It seems reasonable to me.

> It would especially surprise
> me if one tried to run different kernels in different containers.

That's not just unreasonable, it's impossible :)

> It is my understanding of containers that they are intended to be a
> *lightweight* virtualization technique, giving each container
> effectively a private copy of identical instances of the host OS.
>
> If you want to rent out divergent distros, kernels, etc. then it seems
> to me that heavier virtualization like Xen, KVM, VMware, etc. are the
> right answer, rather than trying to force difficult kernel solutions
> into the container and LSM features into the kernel.

For different kernels, yes, but unless you pick two distros which
require incompatible kernel features (?) I don't see running, say,
gentoo, fedora, and ubuntu under different containers as a problem.

Perhaps the biggest reason not to do that, speaking practically, is that
you miss out on some of the ability to share /usr, /lib, etc readonly
among containers to save overall disk space.

> I call it "difficult" because you would have to build a great big switch
> into the LSM interface, so that each hook is dispatched to the LSM
> module being used by the current container. This will impose some
> complexity and overhead, making each hook slower. Worse,the semantics
> become painfully undefined if a syscall by one container touches an
> object owned by a different container; which LSM gets called to mediate
> the access?

At first my thought was this is worse than dealing with stacker.

But on the other hand, perhaps introducing some sort of 'personality' to objects and subjects, where the personality decides which LSM is invoked for access, can be done more optimally than one would think. It would probably require strict enforcement that two "things" with different personalities can NOT mix, ever.

> What makes a *lot* more sense to me is for individual LSMs to try to
> "containerize" themselves. This is actually the AppArmor plan: we hope
> to eventually support having a private AppArmor policy per container.

Agreed, that had been my assumption. That, and that the configuring of LSM policies inside a container would simply be disabled if say loading a suse container under a fedora host.

> Thus all of the containers on the physical machine will be running
> identical kernels, and all will use AppArmor, but each one can have a
> different AppArmor policy set, so that e.g. my private Apache process
> instance is confined in my container different than your Apache process
> is confined in your container.
>
> I see no barrier to SELinux or SMACK or TOMOYO doing the same thing. But
> I see *big* barriers to trying to support multiple LSM modules in the
> kernel at the same time with each container using the LSM of its choice.

It's not as clear to me how SMACK (or the MLS/MCS portion of selinux) would be handled.

-serge