Subject: Re: LSM and Containers
Posted by serue on Thu, 25 Oct 2007 01:44:13 GMT
View Forum Message <> Reply to Message

Quoting Peter Dolding (oiaohm@gmail.com):
> On 10/25/07, Crispin Cowan <crispin@crispincowan.com> wrote:
> > Peter Dolding wrote:
> > > The other thing you have not though of and is critical. If LSM is the
> > > same LSM across all containers. What happens if that is breached and
> > > tripped to disable. You only want to loss one container to a breach
> > > not the whole box and dice in one hit. Its also the reason why my
> > > design does not have a direct link between controllers. No cascade
> > > threw system to take box and dice.
> > >
> > Sorry, but I totally disagree.
> >
> > If you obtain enough privilege to disable the LSM in one container, you
> > also obtain enough privilege to disable *other* LSMs that might be
> > operating in different containers. This is a limitation of the
> > Containers feature, not of LSM.
> >
> That is not a Container feature. If you have enough privilege does
> not mean you can. Root user in a Container does not mean you can play
> with other containers applications. There is a security split at the
> container edge when doing Virtual Servers what by using one LSM you
> are disregarding.
>
> Simple point if one LSM is disabled in a container it can only get the
> max rights of that Container. So cannot see the other LSM's on the
> system bellow it. Reason also why in my model its the same layout if
> there is 1 or 1000 stacked so attack cannot tell how deep they are in
> and if there is anything to be gained by digging. You have to break

You're sometimes hard to parse, but here are a few basic facts within
which to constrain our discussions:

1. LSMs are a part of the kernel. The entire kernel is in the
   same trusted computing base
2. containers all run on the same kernel
3. whether an lsm is compromised, or a tty driver, or anything
   else which is in the TCB, all containers are compromised
4. it is very explicitly NOT a goal to hide from a container
   the fact that it is in a container. So your 'cannot tell how
   deep they are' is not a goal.

If you want to be able to 'plug' lsms in per container, by all means
feel free to write a proof of concept. It is kind of a cool idea. But
be clear about what you'll gain: You allow the container admin to

contrain data access within his container in the way he chooses using the model with which he is comfortable. It does nothing to protect one container from another, does nothing to protect against kernel exploits, and absolutely does nothing to protect a container from the 'host'.

Also please keep in mind that the container security framework is not only not yet complete, it's pretty much not started. My own idea for how to best do it are outlined in emails which are in the containers list archive. But in terms of LSM they follow the idea Crispin outlines, namely that the LSMs support containers themselves. And, in a proces in a container started without CAP_NS_OVERRIDE (which does not yet exist :) in its capability bounding set will only be able to access files in another container as DAC user 'other' (i.e if perms are 754, it will get read access, if 750, then none), even if it has CAP_DAC_OVERRIDE. (unless it gets an authorization key for the owning user in the target namespace, but *that's* probably *years* off)

-serge

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