Subject: Re: [PATCH] cgroups: implement device whitelist lsm (v3)
Posted by Stephen Smalley on Fri, 14 Mar 2008 17:41:56 GMT
View Forum Message <> Reply to Message

On Fri, 2008-03-14 at 09:32 -0500, Serge E. Hallyn wrote:
> Quoting Stephen Smalley (sds@epoch.ncsc.mil):
> >
> > On Fri, 2008-03-14 at 21:17 +1100, James Morris wrote:
> > > On Thu, 13 Mar 2008, Serge E. Hallyn wrote:
> > >
> > > > Implement a cgroup using the LSM interface to enforce open and mknod
> > > > on device files.
> > >
> > > Actually, I'm not sure that the LSM approach in general is best here.
> > >
> > > The LSM model is that standard DAC logic lives in the core kernel, and
> > > that extended security logic (e.g. MAC) is called after DAC via hooks.
> > > cgroups has introduced new security logic of its own, which is arguably
> > > "standard DAC" when cgroups is enabled.
> > >
> > > I can understand Greg not wanting this security logic in the core kernel,
> > > but it is specific to cgroups (which itself is security model agnostic)
> > > and does not stand alone as a distinct security framework.
>
> I completely disagree. We have two separate frameworks in the kernel,
> one to enforce generic additional security stuff, and one to track
> tasks. When I need a feature which tracks tasks to do some security
> tasks, it seems obvious that I would use both, just like to enforce a
> certain type of MAC I end up using both netfilter and LSM through
> selinux.

Depends on whether you think LSM hooks are like netfilter hooks (i.e.
fine for each module to just implement a few here and there, then
combine resulting modules), or whether they are about implementing
complete security models (ala SELinux or Smack). As they currently
exist, they aren't very well suited to the former - they impose a cost
on all hooked operations in order to hook any at all, as has been a
concern for your device controller.

> > > The fact that all existing LSMs need to invoke exactly the same code is an
> > > indicator that it doesn't belong in LSM.
>
> No, that's like saying capabilities don't belong in LSM because all LSMS
> need to invoke it the same way. What it is an indicator of is that
> there are (not-quite-)orthogonal pieces of security which users might
> want to use together.

Likely not a popular view, but capabilities don't belong in LSM. Look

at them: the capability state is still directly embedded in the
relevant kernel data structures, various bits of capability specific
logic and interfaces remain in the core kernel, they don't present a
complete security model (just an auxiliary to some other model like DAC
or Smack for privilege purposes), they use only a small subset of the
hooks, they force LSM to violate its usual restrictive-only paradigm to
support capable(), CONFIG_SECURITY=n still has to invoke the capability
functions, and all of the other LSMs do need to call it the same way to
keep Linux working as expected for applications and users.

The original promise was that LSM would allow kernels to be built that
shed capabilities altogether, but in practice no one seems to do that as
both users and applications expect them to exist in Linux. In fact, the
possibility of not having capabilities present has caused problems that
have led to the dummy module being turned more and more into a clone of
the capabilities module (actually managing and testing the capability
bits rather than just uid == 0 as originally).

So I wouldn't point to capabilities as a counter example to James' point
- they are actually a supporting example.

> As I told stephen I hope to provide the enhanced selinux support for
> devices, and at that point perhaps you won't want to support
> SELINUX+CGROUPS_DEV anymore.
>
> Now that's just my opinion and it doesn't count for much. I'll do
> whatever everyone can agree on, but will wait for Paul's opinion about
> adding cgroup hooks next to the two security hooks.
>
> > > Moving this logic into LSM means that instead of the cgroups security
> > > logic being called from one place in the main kernel (where cgroups
> > > lives), it must be called identically from each LSM (none of which are
> > > even aware of cgroups), which I think is pretty obviously the wrong
> > > solution.
> > >
> > > This is baggage which comes with cgroups -- please don't push it into LSM
> > > to try and hide that.
> >
> > I agree with the above, and would further note that I would expect the
> > SELinux solution to the problem would be done not by stacking with or
> > calling this device whitelist lsm but instead by introducing the ability
> > to bind security labels to devices within the kernel (independent of the
> > particular device node(s) in the filesystem used to access that device)
> > and applying permission checks on those device labels when processes
> > attempt to create or access those devices (again independent of the
> > particular device node used to access them). That keeps the policy
> > integrated and analyzable and avoids an external dependency.
>

> Agreed.
>
> -serge
--
Stephen Smalley
National Security Agency

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