Subject: Re: [PATCH] cgroups: implement device whitelist lsm (v3)
Posted by serue on Mon, 17 Mar 2008 14:08:53 GMT
View Forum Message <> Reply to Message

Quoting Casey Schaufler (casey@schaufler-ca.com):
>
> --- Stephen Smalley <sds@epoch.ncsc.mil> wrote:
>
> >
> > ...
> > > 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.
>
> I don't intend that Smack be thought of as a complete security model.
> Smack implements Mandatory Access Control, but leaves the privilege
> mechanism (root and/or capabilities) to the whims of others. Similarly
> Smack does not do DAC (unlike SELinux with MCS) although "owned rules"
> has been proposed as an additional feature. I certainly wouldn't
> want every new facility that comes in to require multiple versions
> that depend on the other LSMs involved. It's true that today's LSM is
> optimized for the only LSM that existed a year ago, and that was a
> monolythic security model.
>
> > > > > 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.
>
> I share this view, which add credibility to the claim that it's
> not popular. (smiley)

>
> > 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,
>
> It does seem as if a separate Linux Privilege Module framework
> might be a better scheme. It would be very easy to pull out, and
> simple to create the obvious LPMs:
>
> - Traditional root
>   hooks look like "return (euid == 0) ? 0 : -EACCES;"
> - No access check at all
>   hooks look like "return 0;"
> - Root or capabilities
>   hooks look like "return (euid == 0 || capable(xxx)) ? 0 : -EACCES;"
> - Pure capabilities
>   hooks look like "return capable(xxx) ? 0 : -EACCES;"
>
> > 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.
>
> Plus, if SELinux wants to abandon capabilities they can add thier own
> scheme or insist the user use the noop LPM and do whatever they like
> in the LSM. Smack has no intention of mucking with the privilege
> mechanism, and will happily go along with whatever the rest of the
> system wants to use, although the noop LSM seems a bit pointless in
> that case.
>
> > The original promise was that LSM would allow kernels to be built that
> > shed capabilities altogether,
>
> I don't remember that, but it's been a long time so it could be true.
>
> > 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).
>
> This is why Smack is sticking to MAC rather than trying to be a
> wholistic security policy mechanism. To quote the prophet, "God

> created the world in 7 days, but then, He didn't have an install
> base".
>
> > So I wouldn't point to capabilities as a counter example to James' point
> > - they are actually a supporting example.
>
> In particular, capabilities are not an access control mechanism,
> they are a privilege mechanism. A lot of discussion about LSM has
> centered around the appropriate charactoristics of an LSM, and
> these discussions always assume that the LSM in question is
> exactly an access control mechanism. If we split the LSM into
> a LACM for access control and an LPM for privilege management
> maybe we can eliminate the most contentious issues.
>
> Does anyone know why that would be stoopid before I whack out
> patches?

No I'd like to see those patches. It would ideally allow LSM to become
purely restrictive and LPM to be purely empowering, presumably making
the resulting hook sets easier to review and maintain. The LPM wouldn't
(I assume) gain any *new* hook points so we wouldn't be adding any new
places for hooks to be overriden by a rootkit.

-serge