Subject: Re: [PATCH 2/2] hijack: update task_alloc_security
Posted by serue on Thu, 29 Nov 2007 15:38:15 GMT
View Forum Message <> Reply to Message

Quoting Crispin Cowan (crispin@crispincowan.com):
> Serge E. Hallyn wrote:
> > Quoting Crispin Cowan (crispin@crispincowan.com):
> >
> >> Is there to be an LSM hook, so that modules can decide on an arbitrary
> >> decision of whether to allow a hijack? So that this "do the right
> >> SELinux" thing can be generalized for all LSMs to do the right thing.
> >>
> > Currently:
> >
> > 1. the permission is granted through ptrace
> > 2. the lsm knows a hijack is going in security_task_alloc()
> >   when task != current
> >
> > so the lsm has all the information it needs. But I have no objection
> > to a separate security_task_hijack() hook if you find the ptrace hook
> > insufficient.
> >
> I find that ptrace, specifically CAP_SYS_PTRACE, is overloaded. AppArmor
> is having problems because we have to choose between granting
> cap_sys_ptrace, or not allowing the process to read /proc/pid/self &
> such like. So there, the problem is that we have to grant too much power
> to a process to just let it read some /proc stuff about itself.
>
> Here the problem appears to be the other way. cap_sys_ptrace is powerful
> enough to mess with other user's processes on the system, but if ptrace
> gives you hijack, then that seems to give you the power to control
> processes in someone else's namespace.

The user namespace patchset I'm working on right now to start having
signals respect user namespaces introduces CAP_NS_OVERRIDE. Once that
is in, then hijack would require CAP_NS_OVERRIDE|CAP_SYS_PTRACE.

Of course, since we're considering only allowing HIJACK_NS which is
only allowed into a different namespace, hijack would then always
require CAP_NS_OVERRIDE...

Does that suffice?

If you still prefer an LSM hook, we can add that.

thanks,
-serge

_____

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