Subject: Re: [patch] unprivileged mounts update
Posted by serue on Wed, 25 Apr 2007 17:20:12 GMT
View Forum Message <> Reply to Message

Quoting H. Peter Anvin (hpa@zytor.com):
> Miklos Szeredi wrote:
> >
> > Andrew, please skip this patch, for now.
> >
> > Serge found a problem with the fsuid approach: setfsuid(nonzero) will
> > remove filesystem related capabilities. So even if root is trying to
> > set the "user=UID" flag on a mount, access to the target (and in case
> > of bind, the source) is checked with user privileges.
> >
> > Root should be able to set this flag on any mountpoint, _regardless_
> > of permissions.
> >
>
> Right, if you're using fsuid != 0, you're not running as root

Sure, but what I'm not clear on is why, if I've done a
prctl(PR_SET_KEEPCAPS, 1) before the setfsuid, I still lose the
CAP_FS_MASK perms. I see the special case handling in
cap_task_post_setuid(). I'm sure there was a reason for it, but
this is a piece of the capability implementation I don't understand
right now.

I would send in a patch to make it honor current->keep_capabilities,
but I have a feeling there was a good reason not to do so in the
first place.

> (fsuid is
> the equivalent to euid for the filesystem.)

If it were really the equivalent then I could keep my capabilities :)
after changing it.

> I fail to see how ruid should have *any* impact on mount(2). That seems
> to be a design flaw.

May be, but just using fsuid at this point stops me from enabling user
mounts under /share if /share is chmod 000 (which it is).

thanks,
-serge

_____
Containers mailing list
Containers@lists.linux-foundation.org