Subject: Re:  Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by Miklos Szeredi on Tue, 17 Apr 2007 18:36:04 GMT
View Forum Message <> Reply to Message

> > I'm still not sure, what your problem is.
>
> My problem right now is that I see a serious complexity escalation in
> the user interface that we must support indefinitely.
>
> I see us taking a nice powerful concept and seriously watering it down.
> To some extent we have to avoid confusing suid applications.  (I would
> so love to remove the SUID bit...).
>
> I'm being contrary to ensure we have a good code review.

OK.  And it's very much appreciated :)

> I have heard it said that there are two kinds of design.  Something
> so simple it obviously has no deficiencies.  Something so complex it has
> no obvious deficiencies.  I am very much afraid we are slipping the
> mount namespace into the latter category of work.  Which is a bad
> bad thing for core OS feature.

I've tried to make this unprivileged mount thing as simple as
possible, and no simpler.  If we can make it even simpler, all the
better.

> > With the v3 of the usermounts patchset, by default, user mounts are
> > disabled, because the "allow unpriv submounts" flag is cleared on all
> > mounts.
> >
> > There are several options available to sysadmins and distro builders
> > to enable user mounts in a secure way:
> >
> >  - pam module, which creates a private namespace, and sets "allow
> >    unpriv submounts" on the mounts within the namespace
> >
> >  - pam module, which rbinds / onto /mnt/ns/$USER, and chroots into
> >    /mnt/ns/$USER, then sets the "allow unpriv submounts" on the
> >    mounts under /mnt/ns/$USER.
>
> In part this really disturbs me because we now have two mechanisms for
> controlling the scope of what a user can do.

You mean rbind+chroot and clone(CLONE_NS)?  Yes, those are two
different mechanisms achieving very similar results.  But what has
this to do with unprivileged mounts?

> A flag or a new namespace.  Two mechanisms to accomplish the same
> thing sound wrong, and hard to manage.

The flag permitting the unprivileged mounts (which we now agreed to
name "allowusermnt") is used in both cases.

Just creating a new namespace doesn't always imply that you want to
allow user mounts inside, does it?  These are orthogonal features.

> >   - sysadmin creates /mnt/usermounts writable to all users, with
> >     sticky bit (same as /tmp), does "mount --bind /mnt/usermounts
> >     /mnt/usermounts" and sets the "allow unpriv submounts" on
> >     /mnt/usermounts.
> >
> > All of these are perfectly safe wrt userdel and backup (assuming it
> > doesn't try back up /mnt).
>
> I also don't understand at all the user= mount flag and options.

The "user=UID" or (or MNT_USER flag) serves multiple purposes:

 - help mount(8) move away from /etc/mtab
 - allow unprivileged umounts
 - account user mounts

> All it seemed to be used for was adding permissions to unmount.  In user
> space to deal with the lack of any form of untrusted mounts I can understand
> this.  In kernel space this seems to be more of a problem.

Why is handling unprivileged mounts in kernel different from handling
them in userspace in this respect?

Miklos

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