Subject: Re: Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by Ram Pai on Mon, 16 Apr 2007 08:47:00 GMT
View Forum Message <> Reply to Message

>
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> > Quoting Miklos Szeredi (miklos@szeredi.hu):
> >> From: Miklos Szeredi <mszeredi@suse.cz>
> >>
> >> If CLONE_NEWNS and CLONE_NEWNS_USERMNT are given to clone(2) or
> >> unshare(2), then allow user mounts within the new namespace.
> >>
> >> This is not flexible enough, because user mounts can't be enabled
> for
> >> the initial namespace.
> >>
> >> The remaining clone bits also getting dangerously few...
> >>
> >> Alternatives are:
> >>
> >>   - prctl() flag
> >>   - setting through the containers filesystem
> >
> > Sorry, I know I had mentioned it, but this is definately my least
> > favorite approach.
> >
> > Curious whether are any other suggestions/opinions from the
> containers
> > list?
>
> Given the existence of shared subtrees allowing/denying this at the
> mount
> namespace level is silly and wrong.
>
> If we need more than just the filesystem permission checks can we
> make it a mount flag settable with mount and remount that allows
> non-privileged users the ability to create mount points under it
> in directories they have full read/write access to.

Also for bind-mount and remount operations the flag has to be propagated down its propagation tree. Otherwise a unpriviledged mount in a shared mount wont get reflected in its peers and slaves, leading to unidentical shared-subtrees.

RP

>
> I don't like the use of clone flags for this purpose but in this
> case the shared subtress are a much more fundamental reasons for not
> doing this at the namespace level.
>
> Eric
> _____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containe rs

---

Subject: Re: Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by Miklos Szeredi on Mon, 16 Apr 2007 09:32:47 GMT
View Forum Message <> Reply to Message

> > Given the existence of shared subtrees allowing/denying this at the
> > mount
> > namespace level is silly and wrong.
> >
> > If we need more than just the filesystem permission checks can we
> > make it a mount flag settable with mount and remount that allows
> > non-privileged users the ability to create mount points under it
> > in directories they have full read/write access to.
>
> Also for bind-mount and remount operations the flag has to be propagated
> down its propagation tree. Otherwise a unpriviledged mount in a shared
> mount wont get reflected in its peers and slaves, leading to unidentical
> shared-subtrees.

That's an interesting question. Do we want shared mounts to be
totally identical, including mnt_flags? It doesn't look as if
do_remount() guarantees that currently.

Miklos

Subject: Re: Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by Ram Pai on Mon, 16 Apr 2007 09:49:01 GMT
View Forum Message <> Reply to Message

On Mon, 2007-04-16 at 11:32 +0200, Miklos Szeredi wrote:
> > > Given the existence of shared subtrees allowing/denying this at the
> > > mount

> > > namespace level is silly and wrong.
> > >
> > > If we need more than just the filesystem permission checks can we
> > > make it a mount flag settable with mount and remount that allows
> > > non-privileged users the ability to create mount points under it
> > > in directories they have full read/write access to.
> > >
> > Also for bind-mount and remount operations the flag has to be propagated
> > down its propagation tree. Otherwise a unpriviledged mount in a shared
> > mount wont get reflected in its peers and slaves, leading to unidentical
> > shared-subtrees.
>
> That's an interesting question. Do we want shared mounts to be
> totally identical, including mnt_flags? It doesn't look as if
> do_remount() guarantees that currently.

Depends on the semantics of each of the flags. Some flags like of the
read/write flag, would not interfere with the propagation semantics
AFAICT. But this one certainly seems to interfere.

RP

> Miklos
