

---

Subject: Re: [patch 0/8] unprivileged mount syscall  
Posted by [serue](#) on Mon, 16 Apr 2007 01:11:53 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Miklos Szeredi (miklos@szeredi.hu):

> > > Agreed on desired behavior, but not on chroot sufficing. It actually  
> > > sounds like you want exactly what was outlined in the OLS paper.  
> > >  
> > > Users still need to be in a different mounts namespace from the admin  
> > > user so long as we consider the deluser and backup problems  
> > >  
> > > I don't think it matters, because /share/\$USER duplicates a part or  
> > > the whole of the user's namespace.  
> > >  
> > > So backup would have to be taught about /share anyway, and deluser  
> > > operates on /home/\$USER and not on /share/\*, so there shouldn't be any  
> > > problem.  
> >  
> > In what I was thinking of, /share/\$USER is bind mounted to  
> > ~\$USER/share, so it would have to be done in a private namespace in  
> > order for deluser to not be tricked.  
>  
> But /share/\$USER is surely not bind mounted to ~\$USER/share in the  
> \_global\_ namespace, is it? I can't see any sense in that.

No it's not, only in the private namespace.

> > > There's actually very little difference between rbind+chroot, and  
> > > CLONE\_NEWNS. In a private namespace:  
> > >  
> > > 1) when no more processes reference the namespace, the tree will be  
> > > disbanded  
> > >  
> > > 2) the mount tree won't be accessible from outside the namespace  
> >  
> > But it *can* be, if properly set up. That's part of the point of the  
> > example in the OLS paper. When a user logs in, sshd clones a new  
> > namespace, then bind-mounts /share/\$USER into ~\$USER/share. So assuming  
> > that /share/\$USER was --make-shared'd, it and ~\$USER are now in the  
> > same peer group, and any changes made by the user under ~\$USER will  
> > be reflected back into /share/\$USER.  
>  
> I acknowledge, that it can be done. My point was that it can be done  
> more simply without using CLONE\_NS.

Seems like a matter of preference, but I see what you're saying.

> > > Wanting a persistent namespace contradicts 1).

> >  
> > Not necessarily, see above.  
> >  
> > > Wanting a per-user (as opposed to per-session) namespace contradicts  
> > > 2). The namespace `_has_` to be accessible from outside, so that a new  
> > > session can access/copy it.  
> >  
> > Again, I *think* you are wrong that private namespace contradicts this  
> > requirement.  
>  
> I'm not saying there's any contradiction, I'm saying `rbind+chroot` is a  
> better fit.

Ok, I see.

> I haven't yet heard a single reason why a per-session namespace with  
> parts shared per-user is better than just a per-user namespace.

In fact I suspect we could show that they are functionally equivalent  
(for your purposes) by drawing the fs tree and peer groups from  
`current->fs->root` on up for both methods.

And not using private namespaces leaves the admin (at least for now)  
better able to diagnose the state of the system.

-serge

---

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

---