Subject: Re: Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by Miklos Szeredi on Thu, 19 Apr 2007 08:36:15 GMT
View Forum Message <> Reply to Message

> > > As I said earlier, I see a case where two mounts that are peers of each
> > > other can become un-identical if we dont propagate the "allowusermnt".
> > >
> > > As a practical example.
> > >
> > > /tmp and /mnt are peers of each other.
> > > /tmp has its "allowusermnt" flag set, which has not been propagated
> > > to /mnt.
> > >
> > > now a normal-user mounts an ext2 file system under /tmp at /tmp/1
> > >
> > > unfortunately the mount wont appear under /mnt/1
> >
> > Argh, that is not true. That's what I've been trying to explain to
> > you all along.
>
> I now realize you did, but I failed to catch it. sorry :-(

It is my fault also. I will add better description to the patch
headers.

> > The propagation will be done _regardless_ of the flag. The flag is
> > only checked for the parent of the _requested_ mount. If it is
> > allowed there, the mount, including any propagations are allowed. If
> > it's denied, then obviously it's denied everywhere.
> >
> > > and in case if you allow the mount to appear under /mnt/1, you will
> > > break unpriviledge mounts semantics which promises: a normal user will
> > > not be able to mount at a location that does not allow user-mounts.
> >
> > No, it does not promise that. The flag just promises, that the user
> > cannot _request_ a mount on the parent mount.
>
> ok. if the ability for a normal user to mount something *indirectly*
> under a mount that has its 'allowusermnt flag' unset,
> is acceptable under the definition of 'allowusermnt', i guess my only
> choice is to accept it. :-)

It is the only "sane" way I think.

Maybe the MS_SETFLAGS/MS_CLEARFLAGS mount operation could have an
option for propagating the given flag(s). Then it's really up to the
sysadmin, to determine in each case what the desired behavior is.

Yes, you are right, that most of the time propagating the "allowusermnt" between peers may be the right thing to do. But for example propagating it from master to slave may not be a good thing at all. So it shouldn't be for the kernel to decide.

Thanks,
Miklos