Subject: Re: Re: [patch 05/10] add "permit user mounts in new namespace" clone flag
Posted by ebiederm on Tue, 17 Apr 2007 19:54:14 GMT
View Forum Message <> Reply to Message

Miklos Szeredi <miklos@szeredi.hu> writes:

```
>> > 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.
```

We are certainly much more complex then the code in plan9 (just
read through it) so I think we have room for improvement.

Just for reference what I saw in plan 9 was:
- No super user checks in it's mount, unmount, or namespace creation paths.
- A flag to deny new mounts but not new bind mounts (for administrative purposes
  the comment said).

Our differences from plan9.
- suid capable binaries. (SUID please go away).
- A history of programs assuming only root could call mount/unmount.

```
>> 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?
```

The practical question is how do we limit what a user can mount and
unmount.

I would contend that at first glance stuffing a user in their own
mount namespace is sufficient, on a system with utilities aware
of the consequences of mount/unmount.

So we may not need a unprivileged mount disable except as a way
to allow an old user space to run a new kernel.

>> 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.

After user space has been updated we always want to allow unprivileged
mounts.

If I get pushed I will say that we need to remove suid exec capability
from user space as well. At which point we don't even need directory
security checks, there is enough benefit there I certainly think
it is worth considering having an entire NOSUID user space.

Removing suid is probably excessive but if it isn't much harder
then sane mount namespace support we should probably consider it.

>> >   - 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?

Ok. I just looked at what user space is doing. The difference is that
what user space is doing predates mount namespaces, and was there as
far as I can tell to keep one user from causing problems for
another user.   If we choose to make mount namespaces to be
the unit of granularity we don't need this capability.

All we have to do is deny unmounts that would confuse a suid
executable. Which mounts are those?

Eric