Subject: Re: [patch 3/9] unprivileged mounts: account user mounts
Posted by Miklos Szeredi on Tue, 08 Jan 2008 19:18:38 GMT
View Forum Message <> Reply to Message

> On Tue, 2008-01-08 at 12:35 +0100, Miklos Szeredi wrote:
> > plain text document attachment
> > (unprivileged-mounts-account-user-mounts.patch)
> > From: Miklos Szeredi <mszeredi@suse.cz>
> >
> > Add sysctl variables for accounting and limiting the number of user
> > mounts.
> ...
> > +int nr_user_mounts;
> > +int max_user_mounts = 1024;
>
> Just from a containers point of view, I think this is something we'll
> need to fix up in the near future if it stays in the current form.
>
> Instead of having a global tracking, perhaps we could have a per-user
> limit tracked in 'struct user'. The plans are to ensure that two
> containers' users "dave" each have a different 'struct user', so that
> seems to be a decent place to track it.

At one time there was a per-user accounting patch, but it was dropped,
because it was deemed an unnecessary additional compexity.

max_user_mounts is analogue to files_stat.max_files (which is a sysctl
tunable also), and it's purpose is really to make sure that a user is
not able to create an insane number of mounts, and not to acurately
limit normal usage.

So I'm not sure a per-container or per-user count is really needed.

> Also, is a read-only sysctl really the best way to get the number of
> user mounts back out of the kernel? What would you use it for?

Just to check, why I got that (EPERM or whatever) error for the mount
command.

> Do you need any special logic for setting 'max_user_mounts' in the case
> where it gets set _below_ 'nr_user_mounts'?

No, I don't think such corner cases really matter in this case.

> > /* /sys/fs */
> > struct kobject *fs_kobj;
> > EXPORT_SYMBOL_GPL(fs_kobj);
> > @@ -477,11 +480,30 @@ static struct vfsmount *skip_mnt_tree(st

> >   return p;
> > }
> >
> > +static void dec_nr_user_mounts(void)
> > +{
> > + spin_lock(&vfsmount_lock);
> > + nr_user_mounts--;
> > + spin_unlock(&vfsmount_lock);
> > +}
> > +
> > static void set_mnt_user(struct vfsmount *mnt)
> > {
> >   BUG_ON(mnt->mnt_flags & MNT_USER);
> >   mnt->mnt_uid = current->fsuid;
> >   mnt->mnt_flags |= MNT_USER;
> > + spin_lock(&vfsmount_lock);
> > + nr_user_mounts++;
> > + spin_unlock(&vfsmount_lock);
> > +}
>
> One little nitpick on the patch layout: It's a wee bit difficult to
> audit how the set function is used vs the clear one when its users don't
> come until the later patches. It might be worth introducing the users
> here, too.

Yeah, maybe some of the patches should be folded together. If a
resubmit is necessary I'll look into that.

Miklos