Subject: Re: [patch 0/8] unprivileged mount syscall
Posted by Ram Pai on Tue, 10 Apr 2007 08:38:04 GMT
View Forum Message <> Reply to Message

On Mon, 2007-04-09 at 22:10 +0200, Miklos Szeredi wrote:
> > > The one in pam-0.99.6.3-29.1 in opensuse-10.2 is totally broken. Are
> > > you interested in the details? I can reproduce it, but forgot to note
> > > down the details of the brokenness.
> >
> > I don't know how far removed that is from the one being used by redhat,
> > but assuming it's the same, then redhat-lspp@redhat.com will be
> > very interested.
>
> OK.
>
> > > - user namespace setup: what if user has multiple sessions?
> > >
> > > 1) namespaces are shared? That's tricky because the session needs to
> > > be a child of a namespace server, not of login. I'm not sure PAM
> > > can handle this
> > >
> > > 2) or mounts are copied on login? That's not possible currently,
> > > as there's no way to send a mount between namespaces. Also it's
> > > tricky to make sure that new mounts are also shared
> >
> > See toward the end of the 'shared subtrees' OLS paper from last year for
> > a suggestion on how to let users effectively 'log in to' an existing
> > private mounts ns.
>
> This?
>
> 1. create a new namespace
> 2. bind /share/$USER to /share
> 3. for each pair ($who, $what) such that
> /share/$USER/$who/$what exists, look
> in /share/$who/allowed for "peer $what
> $USER" or "slave $what $USER". If the
> former is found, rbind /share/$who/$what
> on /share/$USER/$who/$what; if the
> latter is found, do the same and
> follow with marking subtree under
> /share/$USER/$who/$what as slave.
> 4. rbind /share/$USER to /share
> 5. mark subtree under /share as private.
> 6. umount -l /share
>
> Well, someone please explain using short words, because I don't
> understand at all.

I am trying to re-construct Viro's thoughts. I think the steps outlined above; though not accurate, are still insightful.

The idea is -- there is one master namespace, which has under /share, a replica of the mount tree of namespaces belonging to all users.

for example if there are two users A and B, then in the master namespace under /share you will find /share/A and /share/B, each reflecting the mount tree for the namespaces belonging to user-A and user-B respectively.

Note: /share is a shared mount-tree, which means it can propagate mount events.

Everytime the user logs on the machine, a new namespace is created which is the clone of the master namespace. In this new namespace, the /share/$user is made the root of the namespace. Also if other users have allowed part of their namespace available to this user, than those mounts are also brought under this namespace. And finally the entire tree under /share is unmounted.

Note, though multiple namespaces can exist simultaneously for the same user, the user is provided the illusion of per-process-namespace since all the namespaces look identical.

I am trying to rewrite the steps outlined above, which may or may not reflect Viro's thoughts, but certainly reflect my reconstruction of viro's thoughts.

1. clone the master namespace.

2. in the new namespace

```
 move the tree under /share/$me to /
       for each ($user, $what, $how) {
          move /share/$user/$what to /$what
    if ($how == slave) {
             make the mount tree under /$what as slave
          }
       }
```

3. in the new namespace make the tree under
      /share as private and unmount /share

RP

>
> Thanks,
> Miklos

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