Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Thu, 30 Aug 2007 22:18:25 GMT
View Forum Message <> Reply to Message

On Thu 30-08-07 14:10:10, Serge E. Hallyn wrote:
> Quoting Jan Kara (jack@suse.cz):
> > On Wed 29-08-07 15:06:43, Eric W. Biederman wrote:
> > > Jan Kara <jack@suse.cz> writes:
> > > >> However I'm still confused about the use of current->user. If that
> > > >> is what we really want and not the user who's quota will be charged
> > > >> it gets to be a really trick business, because potentially the uid
> > > >> we want to deliver varies depending on who opened the netlink socket.
> > > > I see it's a complicated matter :). What I need to somehow pass to
> > > > userspace is something (and I don't really care whether it will be number,
> > > > string or whatever) that userspace can read and e.g. find a terminal
> > > > window or desktop the affected user has open and also translate the
> > > > identity to some user-understandable name (average user Joe has to
> > > > understand that he should quickly cleanup his home directory ;).
> > > > Thinking more about it, we could probably pass a string to userspace in
> > > > the format:
> > > > <namespace type>:<user identification>
> > > >
> > > > So for example we can have something like:
> > > > unix:1000 (traditional unix UIDs)
> > > > nfs4:joe@machine
> > > >
> > > > The problem is: Are we able to find out in which "namespace type" we are
> > > > and send enough identifying information from a context of unpriviledged
> > > > user?
> > >
> > > Ok. This provides enough context to understand what you are trying to do.
> > > You do want the unix user id, not the filesystem notion. Because you
> > > are looking for the user.
> > >
> > > So we have to figure out how to do the hard thing which is look at
> > > who opened our netlink broadcast see if they are in the same user
> > > namespace as current->user. Which is a pain and we don't currently
> > > have the infrastructure for.
> > There can be arbitrary number of listeners (potentially from different
> > namespaces if I understand it correctly) listening to broadcasts. So I
>
> Currently that is true, but i think isolating netlink sockets is going
> to have to be done pretty soon.
>
> On the one hand cloning a new netlink socket ns when you unshare
> CLONE_NEWNET may seem 'obvious', but I think doing so when you unshare
> CLONE_NEWUSER make much more sense considering netlink's use for audit
> and now for quota.

>
> > think we should pass some universal identifier rather than try to find out
>
> Even with isolating netlink we still may want to send out an identifier.
> However, just as with mounts extensions we're printing out the memory
> address of vfsmounts, we might just want to print out the memory address
> of the userns. It's not universal, but should be good enough.

Maybe before proceeding further with the discussion I'd like to understand following: What are these user namespaces supposed to be good for?

I imagine it so that you have a machine and on it several virtual machines which are sharing a filesystem (or it could be a cluster). Now you want UIDs to be independent between these virtual machines. That's it, right?

Now to continue the example: Alice has UID 100 on machineA, Bob has UID 100 on machineB. These translate to UIDs 1000 and 1001 on the common filesystem. Process of Alice writes to a file and Bob becomes to be over quota. In this situation, there would be probably two processes (from machineA and machineB) listening on the netlink socket. We want to send a message so that on Alice's desktop we can show a message: "You caused Bob to exceed his quotas" and of Bob's desktop: "Alice has caused that you are over quota.".

Because there may be is not a notion of Bob on machineA or of Alice on machineB, we are in trouble, right? What I like the most is to use the filesystem identities (as you suggested in some other email). I. e. because both Alice and Bob share a filesystem, identities of both have to make sense to it (for example for purposes of permission checking). So we can probably send via netlink these (in our example ids 1000 and 1001) and hope that inside machineA and machineB there will be a way to translate these identities to names "Alice" and "Bob". So that user can understand what is happenning. Does this sound plausible?

If we go this route, then we only need a kernel function, that will for a pair ($filesystem, $task) return indentity of that $task used for operations on $filesystem...

Honza

--
Jan Kara <jack@suse.cz>
SuSE CR Labs

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