Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Tue, 04 Sep 2007 22:49:21 GMT
View Forum Message <> Reply to Message

On Tue 04-09-07 16:32:10, Serge E. Hallyn wrote:
> Quoting Jan Kara (jack@suse.cz):
> > On Thu 30-08-07 17:14:47, Serge E. Hallyn wrote:
> > > Quoting Jan Kara (jack@suse.cz):
> > > > 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.".
> > >
> > > Since this is over NFS, you handle it the way you would any other time
> > > that user Alice on some other machine managed to do this.
> > I meant this would actually happen over a local filesystem (imagine
> > something like "hostfs" from UML).
>
> Ok, then that is where I was previously suggesting that we use an api to
> report a uid meaningful in bob's context, where we currently (in the
> absense of meaningful mount uids and uid equivalence) tell Bob that root
> was the one who brought him over quota. From a user pov 'nobody' would
> make more sense, but I don't think we want the kernel to know about user
> nobody, right?
  But what is the problem with using the filesystem ids? All virtual
machines in my example should have a notion of those...

> So if the msg weren't broadcast, or netlink sockets were tied to one
> user namespace, we could call a
> int uid_in_user_ns(struct user *, struct user_ns *)
> sending in Alice's user struct and Bob's userns, and use the result in
> the netlink message. Otherwise I'm not sure what is the right answer.
> We just might need the equivalent of 'struct pid' to struct user, or
> persistant global user namespace ids (persistant after user namespace
> destruction, not across reboot) so we can safely send the user_ns * in a
> netlink msg.
  Yes, that could also be a solution.

> > > > 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
> > > >
> > > Right, so long as we're talking about local filesystems that's the way
> > > to go. If a file write was allowed which brought bob over quota,
> > > clearly the person responsible had some uid valid on the filesystem to
> > > allow him to do so.
> > > Fine. So I'll keep UID in the quota netlink protocol with the meaning
> > "the identity of the user for filesystem operations".
> >
> I think that's ok.
>
> Hopefully when that changes to accomodate user namespaces, we can use
> netlink field versioning to make that transition pretty seamless?
  Yes, we'd just assign the attribute a different number and teach
userspace about the new attribute format...

> If not, then we probably should in fact make some decision now so as not
> to change the api.
>
> > > > 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...
> > > >
> > > Ok, now I see. This is again unrelated to user namespaces, it's an
> > > issue regardless.
> > >
> > > Is there no way to just report Alice as the guilty party to Bob on his
> > > machine as (host=nfsserver,uid=1000)?
> > > You know, in fact this contains all the information but it is quite useless
> > for an ordinary user. The message should be understandable to average desktop
> >
> What is the ordinary user going to do about it? If the user didn't set
> up the nfsserver and/or the second client, the only thing he can do is
> report the guilty user to an admin. In which case the tuple
> (host=nfsserver,uid=1000) is exactly the data he needs to report.
  Maybe write him an email or go and bang him with a baseball bat ;)
Seriously, if someone (like admin) is able to find a physical identity of the
guilty user, then we should be able to do this in a software too, shouldn't
we?

      Honza
--

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

_____