Subject: Re: [PATCH] Send quota messages via netlink
Posted by ebiederm on Wed, 29 Aug 2007 18:31:52 GMT
View Forum Message <> Reply to Message

Jan Kara <jack@suse.cz> writes:

>> I suspect the namespace virtualisation guys would be interested in a new
>> interface which is sending current->user->uid up to userspace. uids are
>> per-namespace now. What are the implications? (cc's added)

>   I know there's something going on in this area but I don't know any
> details. If somebody has some advice what should be passed into userspace
> so that user/group can be idenitified, it is welcome.

For non networking stuff netlink is a pain to use in this area.

Although if we are very careful we may be ok. But this requires
some thinking through.

In principle the uid that corresponds to a struct user depends
on which user namespace you are in.

Now there is a cheap trick we can play. A traditional filesystem
belongs to exactly one user namespace. So we can return the uid
in the filesystems user namespace.

Wait you are returning current->user->uid? Shouldn't we return
the user who's quota is exceeded? I.e. if alice owns a file
and makes it world writable. And bob writes to the file wouldn't
that file still be billed to alice's quota? So shouldn't we complain
about alice and not bob?

Anyway if the goal is to return a user who maps to the filesystem we
can just always return uids in the filesystems uid namespace.

Although if filesystems start supporting multiple user namespaces
natively we might have a challenge on our hands.

Let me see if I can think of a concrete example here.

We have a nfs server with quotas.
We have clients who mount the nfs filesystem without synchronizing
their /etc/password files, so we have separate user namespaces.

What are the ways to make this work?
- Everyone who has right access to the NFS mount on all
  machines must have their uid synchronized across all machines
  (the easiest case).

- Each different kernel has a mapping from it's local uids to
  the uids of the nfs filesystem. (ick if we do much more the
  root squash).

- The nfs filesystem knows about the situation and remembers the
  uid source (the uid namespace) as well as the uid when storing
  owners of files. NFSv4 allows for this by treating users
  as user@domain.

Generally synchronizing uid namespaces (with possibly a root squash
exception) is the sanest and simplest thing to do in a case like this,
but it isn't always what is done.

As long as we are returning the filesystems idea of users we
shouldn't have to worry much about uid namespaces. However
for non-traditional filesystems that don't store the user
as just a uid, say 9p and NFSv4, this implies that we want
to use the filesystems string identifier. However I don't think
the quota system supports these filesystems yet. So that
isn't an issue just yet.

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.

Eric

Subject: Re: [PATCH] Send quota messages via netlink
Posted by Jan Kara on Wed, 29 Aug 2007 19:26:53 GMT

View Forum Message <> Reply to Message

On Wed 29-08-07 12:31:52, Eric W. Biederman wrote:
> Jan Kara <jack@suse.cz> writes:
>
> >> I suspect the namespace virtualisation guys would be interested in a new
> >> interface which is sending current->user->uid up to userspace. uids are
> >> per-namespace now. What are the implications? (cc's added)
>
> >   I know there's something going on in this area but I don't know any
> > details. If somebody has some advice what should be passed into userspace
> > so that user/group can be idenitified, it is welcome.

>
> For non networking stuff netlink is a pain to use in this area.
>
> Although if we are very careful we may be ok. But this requires
> some thinking through.
>
> In principle the uid that corresponds to a struct user depends
> on which user namespace you are in.
>
> Now there is a cheap trick we can play. A traditional filesystem
> belongs to exactly one user namespace. So we can return the uid
> in the filesystems user namespace.
>
> Wait you are returning current->user->uid? Shouldn't we return
> the user who's quota is exceeded? I.e. if alice owns a file
> and makes it world writable. And bob writes to the file wouldn't
> that file still be billed to alice's quota? So shouldn't we complain
> about alice and not bob?

Yes, the quota will still be billed to Alice and originally we complained only about Alice. Now, we are actually passing identities of two users: The one who actually caused the quota to be exceeded and the one whose quota is exceeded. Userspace app can then decide what to do with the information... For example it makes sence to display the message to both Alice and Bob in the case you've described...

> Anyway if the goal is to return a user who maps to the filesystem we
> can just always return uids in the filesystems uid namespace.
>
> Although if filesystems start supporting multiple user namespaces
> natively we might have a challenge on our hands.
>
> Let me see if I can think of a concrete example here.
>
> We have a nfs server with quotas.
> We have clients who mount the nfs filesystem without synchronizing
> their /etc/password files, so we have separate user namespaces.
>
> What are the ways to make this work?
> - Everyone who has right access to the NFS mount on all
>   machines must have their uid synchronized across all machines
>   (the easiest case).
>
> - Each different kernel has a mapping from it's local uids to
>   the uids of the nfs filesystem. (ick if we do much more the
>   root squash).
>
> - The nfs filesystem knows about the situation and remembers the
>   uid source (the uid namespace) as well as the uid when storing

>   owners of files. NFSv4 allows for this by treating users
>   as user@domain.
>
> Generally synchronizing uid namespaces (with possibly a root squash
> exception) is the sanest and simplest thing to do in a case like this,
> but it isn't always what is done.
>
> As long as we are returning the filesystems idea of users we
> shouldn't have to worry much about uid namespaces. However
> for non-traditional filesystems that don't store the user
> as just a uid, say 9p and NFSv4, this implies that we want
> to use the filesystems string identifier. However I don't think
> the quota system supports these filesystems yet. So that
> isn't an issue just yet.

OK, quota kind of works for NFSv4 - we simply enforce quotas on the server on a traditional filesystem and there are some RPC calls to get quota status. For 9p, it does not work. But we should probably design the interface generic enough so that it accommodates those untraditional cases anyway.

> 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?

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