
Subject: Re: [RFC] [PATCH 0/4] uid_ns: introduction
Posted by [ebiederm](#) on Thu, 09 Nov 2006 13:26:13 GMT
[View Forum Message](#) <> [Reply to Message](#)

"Serge E. Hallyn" <serue@us.ibm.com> writes:

> So from your pov the same objection would apply to tagging vfsmounts,
> or not?

No. The issue is that the NFS server merges different mounts to the same nfs server into the same superblock.

> What is the scenario where the caching is broken? It can't be multiple
> clients accessing the same NFS export from the same NFS service container,
> since that would just be an erroneous setup, right?

>

>> > As I recall there are two basic issues.

>> >

>> > Putting the default on the mount structure instead of the superblock
>> > for filesystems that are not uid namespaces aware sounded reasonable,
>> > and allowed certain classes of sharing between namespaces where they
>> > agreed on a subset of the uids (especially for read-only data).

>>

>> yes, that is especially interesting for --bind mounts

>> when you 'know' that you will dedicate a certain

>> sub-tree to one context/guest

>

> Ok, so you wouldn't object to a patch which tagged vfsmounts?

>

> I guess a NULL vfsmnt->user_ns pointer would mean ignore user_ns and
> only apply uid checks (useful for ro bind mount of /usr into multiple
> containers).

Bind mounts are peculiar. But I think as long as you charged the to the context in which they happen (don't do the bind until after you switch the user_ns. You should be fine.

> That of course wouldn't preclude also tagging inodes in later patches.

>

> If you do object, then I can jump straight to tagging inodes with a
> container, though that seems more likely to interfere conceptually
> with any filesystems which are uid namespace aware.

I'm pretty certain tagging inodes is the wrong approach. You want a callback that allows the filesystem to make that determination, a uid namespace aware filesystem.

Remote filesystems will be able to do things like tell you a particular file is owned by "user@domain" which can get translated into a uid, uid_ns pair.

Where tagging the inode becomes a problem is when things like joe@domain1 is fred@domain2, and treats those two users the same. I don't know if anything actually supports that today but that is an interesting case to handle.

Eric

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
