Subject: Re: [PATCH] Send quota messages via netlink Posted by serue on Wed, 05 Sep 2007 14:28:02 GMT View Forum Message <> Reply to Message

Quoting Jan Kara (jack@suse.cz): > On Tue 04-09-07 18:48:52, Serge E. Hallyn wrote: > > Quoting Jan Kara (jack@suse.cz): > > > 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): >>>>>>> limagine it so that you have a machine and on it several virtual >>>>>>>> want UIDs to be independent between these virtual machines. That's it, >>>>>>> right? >>>>>>> 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 >>>>>>>>>> wachineA and machineB) listening on the netlink socket. We want to send a >>>>>>>>> The second se >>>>>>>>> Bob to exceed his guotas" 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... > > >> I don't know what you mean by filesystem ids. Do you mean the uid > > stored on the fs? I imagine a network fs could get fancy and store > > something more detailed than the unix uid, based on the user's keys. > > >> Do you mean the inode->i_uid? Nothing wrong with that. Then we just > > assume that either you are in the superblock or mount's user namespace > > (depending on how we implement it, probably superblock), or can figure > > out what that is. > I meant the identity the process uses to access the filesystem (to > identify the user who caused the limit excess) and also the identity stored

> in the quota file (to identify whose quota was exceeded).

> Anyway, any identity more complicated than just a number needs changes in

> both quota file format and filesystems so at that moment, we can also

> change the netlink interface...

>

> > Sure, and in many ways. But if working with NFS, as far as I know the

> > most common way to solve it is to enforce a common /etc/passwd across

> > all the valid NFS clients :)

> Then one wonders whether user namespaces are really what users want ;).

Absolutely.

You use nfs to share filesystems among separate machines that you want to have look similar.

You use user namespaces to pretend one machine is a bunch of separate machines. So if you're just going to split up your machine into 5 vms and then have them all share disk over nfs, you may just want to keep it as one machine :)

Ideally each vm would have completely separate disk space, so file access across user namespaces wouldn't happen. More realistically, file trees will be shared read-only - i.e. /lib, /usr, etc. Some of that can be handled simply using read-only bind mounts. We'd like to allow users to create vm's as well, so then we want uid 500 in the initial user namespace to be uid 0 in a newly created user namespace.

So what Eric and I are worried about are corner cases and admin mistakes, not regular function.

(And again I really do think we'll want to tie netlink sockets to a user namespace, not a network namespace, so there may be no issue at all so long as proper filesystem access checks are implemented so that every action on some filesystem is done with credentials valid in that filesystems' user namespace)

-serge

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