Subject: Re: [RFC][PATCH] Pid namespaces vs locks interaction
Posted by serue on Wed, 12 Dec 2007 18:42:25 GMT
View Forum Message <> Reply to Message

Quoting Vitaliy Gusev (vgusev@openvz.org):
> On 12 December 2007 20:31:15 Serge E. Hallyn wrote:
> > Quoting Vitaliy Gusev (vgusev@openvz.org):
> > > Hello
> > >
> > > On 6 December 2007 18:51:30 Serge E. Hallyn wrote:
> > > > > > fl_pid is used by nfs, fuse and gfs2. For instance nfs keeps in
> > > > > > fl_pid some unique id to identify locking process between hosts - it
> > > > > > is not a process pid.
> > > > >
> > > > > Ok, but so the struct user_flock->fl_pid is being set to the task's
> > > > > virtual pid, while the struct kernel_flock->fl_pid is being set to
> > > > > task->tgid for nfsd use.
> > > > >
> > > > > Why can't nfs just generate a uniqueid from the struct pid when it
> > > > > needs it?
> > > >
> > > > I think it is hard. lockd uses struct nlm_host to get process unique id
> > > > (see __nlm_alloc_pid() function).
> > >
> > > Looks pretty simple though... That whole set of code could even stay
> > > the same except for in __nlm_alloc_pid():
> > >
> > > option 1: compare struct pid* instead of uint32_t pid
> > > option 2: use the "global pid" out of the stored struct pid,
> > >   something like pid->numbers[0].nr.
>
> We can't use process pid. Process pid is circulated! NFS (lockd) needs
> unique process id between hosts which can't repeat oneself.

Ok sorry - by letting this thread sit a few days I lost track of where
we were.

I see now, so you're saying fl_pid for nfs is not in fact a task pid.
It's a magically derived unique id. (And you say it is unique across
all the nfs clients?)

So does the p in fl_pid stand for something, or could we rename it to
fl_id or fl_uniqueid?

Maybe that's too much bother, but so long as we're bothering with a pid
cleanup at all it seems worth it to me. On the other hand maybe
J. Bruce Fields was right and we should accept the fact that the
flock->fl_pid shouldn't be taken too seriously, and leave it be.

-serge

> > > > Fuse just seems to copy the pid to report it to userspace, so it would
> > > > just copy pid_vnr(kernel_flock->pid) into user_flock->fl_pid.
> > > >
> > > > Anyway I haven't looked at all the uses of struct fl_pid, but you
> > > > can always get the pidnr back from the struct pid if needed so there
> > > > should be no problem.
> > > >
> > > > The split definately seems worthwhile to me, so that
> > > > user_flock->fl_pidnr can always be said to be the pid in the acting
> > > > process' namespace, and flock->fl_pid can always be a struct pid,
> > > > rather than having fl_pid sometimes be current->tgid, or sometimes
> > > > pid_vnr(flock->fl_nspid)...
> > > >
> > > > -serge
> > > > -
> > > > To unsubscribe from this list: send the line "unsubscribe
> > > > linux-fsdevel" in the body of a message to majordomo@vger.kernel.org
> > > > More majordomo info at http://vger.kernel.org/majordomo-info.html
> > >
> > > --
> > > Thank,
> > > Vitaliy Gusev
> >
> > -
> > To unsubscribe from this list: send the line "unsubscribe linux-fsdevel" in
> > the body of a message to majordomo@vger.kernel.org
> > More majordomo info at http://vger.kernel.org/majordomo-info.html
>
>
>
> --
> Thank,
> Vitaliy Gusev

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