Subject: Re: [patch -mm 1/5] mqueue namespace : add struct mq_namespace
Posted by serue on Wed, 03 Oct 2007 13:59:55 GMT
View Forum Message <> Reply to Message

Quoting Cedric Le Goater (clg@fr.ibm.com):
> sukadev@us.ibm.com wrote:
> > Cedric Le Goater [clg@fr.ibm.com] wrote:
> > |
> > | >> however, we have an issue with the signal notification in __do_notify()
> > | >> we could kill a process in a different pid namespace.
> > | >
> > | > So I took a quick look at the code as it is (before this patchset)
> > | > and the taking a reference to a socket and the taking a reference to
> > | > a struct pid should do the right thing when we intersect with other
> > | > namespaces. It certainly does not look like a fundamental issue.
> >
> > |
> > | right. this should be covered when the pid namespace signal handling is
> > | complete. kill_pid_info() should fail to send a signal to a sibling or
> > | a parent pid namespace.
> > |
> > | I guess we should add a WARNING() to say that we're attempting to do so.
> >
> > Just want to clarify how a signal is sent to a parent ns.
> >
> > A process P1 sets itself up to be notified when a message arrives
> > on a queue.
> >
> > P1 then clones P2 with CLONE_NEWPID.
> >
> > P2 writes to the message queue and thus signals P1
> >
> > What should the semantics be here ?
> >
> > I guess it makes less sense for two namespaces to be dependent on the same
> > message queue this way. But, if P2 writes to the queue, technically, the
> > queue is not empty, so P1 should be notified, no ?
> >
> > This sounds similar to the SIGIO signal case (F_SETOWN). My understanding
> > was that we would notify whoever was set to receive the notification, even
> > if they were in a parent ns (again my reasoning was its based on the state
> > of a file).
>
> yes
>
> > IOW, should we change kill_pid_info() ? If the caller can 'see' the
> > 'struct pid' they can signal it. The expectation was that callers would
> > call find_vpid() and thus only see processes in their namespace.

>
> I think we have to decide on some limitations with signals

Yes we do, but

> and make sure
> that we cannot send a signal to a sibling pid namespace.

I think you and Eric (and I) are disagreeing about those limitations.
You take it for granted that a sibling pidns is off limits for signals.
But the signal wasn't sent using a pid, but using a file (in SIGIO
case). So since the fs was shared, the signal should be sent. An
event happened, and the receiver wants to know about it.

> This can occur
> in some special namespaces unshare configuration which should never be used
> but to make sure, let's add a big WARNING when we detect such a pid namespace
> violation.
>
> If it is what you mean, I agree :)
>
> Thanks,
>
> C.
> ____
> Containers mailing list
> Containers@lists.linux-foundation.org
> https://lists.linux-foundation.org/mailman/listinfo/containers

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