

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

>> A better way to put this is that we already have a lock that attach_pid
>> and detach_pid use. So we don't need another one for what should be
>> a very rare case.

>
> I think we'd at least need rcu if we supported unshare.

We might.

>> I don't think we will need to add pids in the new pid namespace for
>> sid and pgrp leaders into the new pid namespace.

>>
>> If we do need to pull in sid and pgrp leaders calling setsid() before
>> the operation won't help (wrong namespace)

>
> I'm pretty sure sid and pgrp leaders are a single task_struct pointer
> each, so setting these to point to yourself before an unshare works.
> But I guess for clone it does not work!

Point.

> I agree with what you say in a later email - just returning '0' if sid or
> pgrp is not in our pid_ns presents no problems, and not having a pgrp on
> which to do kill -pgrp doesn't matter either...

>
> So how about we
>
> 1. remove the setsid requirement before CLONE_NEWPID
> 2. punt on unshare(CLONE_NEWPID) support for now
punt as in skip it for the moment (I agree).

> 3. remove pid->lock saving some space and time for now

Sounds good.

>> and the problem is the same
>> for unshare and clone.

>
> Disagree, but irrelevant if we do the above for now.

I see your point about unshare being a little simpler because we do
allocate the local pids before we call unshare.

Eric

Containers mailing list

Containers@lists.osdl.org

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