## Subject: Re: namespace and nsproxy syscalls Posted by Herbert Poetzl on Tue, 26 Sep 2006 22:09:48 GMT View Forum Message <> Reply to Message

On Tue, Sep 26, 2006 at 12:17:01PM -0500, Serge E. Hallyn wrote: > Quoting Herbert Poetzl (herbert@13thfloor.at): > > On Tue, Sep 26, 2006 at 07:56:49AM -0500, Serge E. Hallyn wrote: > > Quoting Cedric Le Goater (clg@fr.ibm.com): > > > Hello all. >>>> >>> A while ago, we expressed the need to have a new syscall >>> specific to namespaces, the clone and unshare are good >>> candidates but we are reaching the limit of the clone flags and >>> clone has been hijacked enough. >>> So, I came up with unshare\_ns. the patch for the core feature >>> follows the email. Not much difference with unshare() for >>> the moment but it gives us the freedom to diverge when new >>> namespaces come in. I have faith also! If you feel it's useful, >>> i'll send the full patchset for review on the list. >>> I'd like to discuss of another syscall which would allow >>> a process to bind to a set of namespaces ( == nsproxy == > > > container) : >>>> >>> bind\_ns(ns\_id\_t id, int flags) >>> What about just using a pid instead of introducing some ns id t? >>> I'm guessing that any time you want to bind to some other nsproxy, >>> it will be the nsproxy of a decendent nsproxy, so even if it is in >>> a new pidspace, you will have a pid in your pidspace to reference >>> it. > > > > what about lightweight containers where the process > > creating the namespace(s) goes away after starting > > a few scripts inside the guest? > So long as the scripts are running, those processes have a pid which > could be used. > But I guess your concern is how the sysadmin can know which pids to use, > since he might have only known the pid which started the container? not only, just consider a lightweight guest which does nothing more but 'running' /etc/rc to start services, quite naturally this script is not running very long (a few seconds usually) but you might want to enter the guest namespace at a later time too :)

- > Dunno. Good question. Guess it might imply that either (a) we need
- > namespace id's after all, or (b) we need to keep init processes around
- > even for application conatiners.

that's just a waste of resources ... IMHO it is a little weird to actually consider having an init process 'just' to have a reference for a bunch of namespaces, given that you might want to access them individually, am I missing something?

for me this suggestion sounds like making a dog mandatory for each household, so that when you want to get the younger son on the phone you can refer to him as 'the younger son of the family with the dog charly':) ...

- > > how to avoid having duplicate identifiers when there
- > > is a chance that the same pid will be used again
- > > to create a second namespace?

>

- > Well at least that's simple, the pid will no longer be a valid handle to
- > the first namespace ever since that process died :)

which simply makes it inaccesible which is not what you actually want, sorry ...

best, Herbert

> -serge

Containers mailing list Containers@lists.osdl.org https://lists.osdl.org/mailman/listinfo/containers