## Subject: Re: [RFC] [PATCH 0/3] containers: introduction Posted by Paul Menage on Fri, 12 Jan 2007 19:03:48 GMT

View Forum Message <> Reply to Message

On 1/12/07, Serge E. Hallyn <serue@us.ibm.com> wrote:

>

- > I agree, so long as "provided requirements aren't too different" is
- > replaced by "provided there is commonality to be merged." Differences in
- > lifetime rules and fs behavior could make it pounding a round peg into
- > a square hole...

Well, at a minimum they have the commonality of "track this set of processes, and all their children", and report how the set changes.

>

- > We were thinking that each container directory would have a file
- > representing each namespace in the nsproxy. To enter only a few
- > namespaces out of an existing namespace container, then, you could
- > create a directory for a new namespace container, link the namespaces
- > you want out of other containers, then enter the container (presumably
- > by doing 'echo (container path) > /proc/\$\$/ns container)'

>

- > So in some ways that's actually closer to what you currently have
- > than the default container creation rules.

Very similar, yes - the only difference would be that in my model you'd do

echo \$\$ > (container path)/tasks

But I thought that Eric (and others?) objected to the idea of being able to move a task into an existing namespace/container on the grounds of race conditions, etc?

If you were able to go with this model rather than the unshare/clone model, that would be a lot simpler than implementing a new clone model for containers. It would also make your namespace work more useful/flexible, I think - there are definitely cases where I want to be able to add a new process into an existing container/namespace/virtual server, which isn't possible with the unshare/clone model unless the root process in the container is written to be able to spawn new processes for you.

Paul