

---

Subject: Re: [PATCH 1/1] containers: implement nsproxy containers subsystem  
Posted by [serue](#) on Tue, 05 Jun 2007 13:16:55 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Quoting Pavel Emelianov (xemul@openvz.org):

> Serge E. Hallyn wrote:

> > Quoting Pavel Emelianov (xemul@openvz.org):

> > > Serge E. Hallyn wrote:

> > > > From 190ea72d213393dd1440643b2b87b5b2128dff87 Mon Sep 17 00:00:00 2001

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

> > > > Date: Mon, 4 Jun 2007 14:18:52 -0400

> > > > Subject: [PATCH 1/1] containers: implement nsproxy containers subsystem

> > > >

> > > > When a task enters a new namespace via a clone() or unshare(), a new

> > > > container is created and the task moves into it. This enables

> > I have a design question.

> >

> > How the child that has a new namespace guesses what id

> > this namespace has in containers?

> >

> > parse /proc/\$\$/container

>

> Ok.

>

> > So more likely the parent would have to grab the cloned pid of the

> > child, parse /proc/\$\$/container, then rename the container.

>

> Child can happen to die before this and we'll have an orphaned

> container. I mean, it will be deletable, but its name will be unknown.

Can we address that with a release\_agent?

Or, the program which did the unshare() (or the fn which was cloned())  
can print out it's /proc/getpid()/container before it (optionally resets  
stdout and) executes the actual program to be run in the container.

> Maybe its better to get the containers id from the pid of new task?

Hmm, that seems reasonable. In fact I like it.

Good idea, thanks. I think I'll go ahead and implement that.

-serge

---