Subject: Re: [PATCH RFC] cgroup_clone: use pid of newly created task for new cgroup
Posted by serue on Thu, 12 Jun 2008 00:37:42 GMT
View Forum Message <> Reply to Message

Quoting Paul Menage (menage@google.com):
> On Wed, Jun 11, 2008 at 8:46 AM, Serge E. Hallyn <serue@us.ibm.com> wrote:
> >
> > From f0635c20e9e9643fa9a90dd7e29b7855ff32ad40 Mon Sep 17 00:00:00 2001
> > From: Serge Hallyn <serge@us.ibm.com>
> > Date: Wed, 11 Jun 2008 10:41:37 -0500
> > Subject: [PATCH 1/1] cgroup_clone: use pid of newly created task for new cgroup
> >
> > cgroup_clone creates a new cgroup with the pid of the task. This works
> > correctly for unshare, but for clone cgroup_clone is called from
> > copy_namespaces inside copy_process, which happens before the new pid
> > is created. As a result, the new cgroup was created with current's pid.
> > This patch:
> >
> > 1. Moves the call inside copy_process to after the new pid
> > is created
> > 2. Passes the struct pid into ns_cgroup_clone (as it is not
> > yet attached to the task)
> > 3. Passes a name from ns_cgroup_clone() into cgroup_clone()
> > so as to keep cgroup_clone() itself simpler
> > 4. Uses pid_vnr() to get the process id value, so that the
> > pid used to name the new cgroup is always the pid as it
> > would be known to the task which did the cloning or
> > unsharing. I think that is the most intuitive thing to
> > do. This way, task t1 does clone(CLONE_NEWPID) to get
> > t2, which does clone(CLONE_NEWPID) to get t3, then the
> > cgroup for t3 will be named for the pid by which t2 knows
> > t3.
> >
> > (Thanks to Dan Smith for finding the main bug)

Seems this bug was also reported on May 21 by Daniel Lezcano. I'm going
to have to blame an over-active left middle finger for hitting the d key
without reading it...

-serge

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