Subject: [PATCH 17/20] Allow signalling container-init
Posted by Pavel Emelianov on Fri, 10 Aug 2007 11:48:24 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Only the global-init process must be special - any other container-init
process must be killable to prevent run-away processes in the system.

TODO: Ideally we should allow killing the container-init only from parent
container and prevent it being killed from within the container.
But that is a more complex change and will be addressed by a follow-on
patch. For now allow the container-init to be terminated by any process
with sufficient privileges.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Acked-by: Pavel Emelyanov <xemul@openvz.org>
Cc: Oleg Nesterov <oleg@tv-sign.ru>

---

signal.c | 6 ++----
1 files changed, 2 insertions(+), 4 deletions(-)

```
--- lx26-23-rc1-mm1.orig/kernel/signal.c 2007-08-07 13:52:12.000000000 -0700
+++ lx26-23-rc1-mm1/kernel/signal.c 2007-08-09 17:22:19.000000000 -0700
@@ -1861,11 +1861,9 @@ relock:
    continue;

   /*
-   * Init of a pid space gets no signals it doesn't want from
-   * within that pid space. It can of course get signals from
-   * its parent pid space.
+   * Global init gets no signals it doesn't want.
    */
-  if (current == task_child_reaper(current))
+  if (is_global_init(current))
    continue;

   if (sig_kernel_stop(signr)) {
```