Subject: Re: [PATCH 14/15] Destroy pid namespace on init's death
Posted by Sukadev Bhattiprolu on Thu, 02 Aug 2007 07:37:00 GMT
View Forum Message <> Reply to Message

```
Dave Hansen [haveblue@us.ibm.com] wrote:
| On Tue, 2007-07-31 at 23:16 -0700, sukadev@us.ibm.com wrote:
| > Oleg Nesterov [oleg@tv-sign.ru] wrote:
| > | On 07/30, sukadev@us.ibm.com wrote:
| > | >
| > | > --- lx26-23-rc1-mm1.orig/kernel/exit.c 2007-07-26 20:08:16.000000000 -0700
| > | > +++ lx26-23-rc1-mm1/kernel/exit.c 2007-07-30 23:10:30.000000000 -0700
| > | > @@ -915,6 +915,7 @@ fastcall NORET_TYPE void do_exit(long co
| > | >  {
| > | >   struct task_struct *tsk = current;
| > | >   int group_dead;
| > | > + struct pid_namespace *pid_ns = tsk->nsproxy->pid_ns;
| > | >
| > | >   profile_task_exit(tsk);
| > | >
| > | > @@ -925,9 +926,10 @@ fastcall NORET_TYPE void do_exit(long co
| > | >   if (unlikely(!tsk->pid))
| > | >    panic("Attempted to kill the idle task!");
| > | >   if (unlikely(tsk == task_child_reaper(tsk))) {
| > | > -  if (task_active_pid_ns(tsk) != &init_pid_ns)
| > | > -   task_active_pid_ns(tsk)->child_reaper =
| > | > -     init_pid_ns.child_reaper;
| > | > +  if (pid_ns != &init_pid_ns) {
| > | > +   zap_pid_ns_processes(pid_ns);
| > | > +   pid_ns->child_reaper = init_pid_ns.child_reaper;
| > | > +  }
| > | >    else
| > | >     panic("Attempted to kill init!");
| > | >   }
| > |
| > | Just to remind you, this is not right when init is multi-threaded,
| > | we should do this only when the last thread exits.
| >
| > Sorry, I needed to clarify somethings about the multi-threaded init. I
| > got the impresssion that you were sending a patch for the existing bug,
| > and meant to review/clarify in the context of the patch.
| >
| > Anyways, re: requirements for multi-threaded init:
| >
| >  Our current definition of is_container_init() and task_child_reaper()
| >  refer only to the main-thread of the container-init (since they check
| >  for pid_t == 1)
|
| Remember, the "pid" is actually a tgid:
```

|
```
|        asmlinkage long sys_getpid(void)
|        {
|                return current->tgid;
|        }
```
|
| So, there are multiple tasks with a "pid" == 1 with a multithreaded
| init.

Yes, and so am now wondering if is_container_init(), is_global_init()
and the "tsk == task_child_reaper(tsk)" checks be replaced with with
something that covers other threads in the reaper ?

|
| > If the main-thread is exiting and is the last thread in the group,
| > we want terminate other processes in the pid ns (simple case).
| >
| > If the main thread is exiting, but is not the last thread in the
| > group, should we let it exit and let the next thread in the group
| > the reaper of the pid ns ?
|
| Well, what happens with a multithreaded init today?
|
| -- Dave