Subject: [PATCH] Use struct pid parameter in copy_process()
Posted by Sukadev Bhattiprolu on Thu, 22 Feb 2007 23:29:58 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [PATCH] Use struct pid parameter in copy_process()

Modify copy_process() to take a struct pid * parameter instead of a pid_t.
This simplifies the code a bit and also avoids having to call find_pid()
to convert the pid_t to a struct pid.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Cc: Cedric Le Goater <clg@fr.ibm.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>
Cc: Eric Biederman <ebiederm@xmission.com>
Cc: containers@lists.osdl.org
---
 kernel/fork.c | 39 ++++++++++++++++++++++++---------------
 1 file changed, 19 insertions(+), 20 deletions(-)

Index: lx26-20-mm1/kernel/fork.c
===
--- lx26-20-mm1.orig/kernel/fork.c 2007-02-20 14:03:17.000000000 -0800
+++ lx26-20-mm1/kernel/fork.c 2007-02-20 14:22:00.000000000 -0800
@@ -965,7 +965,7 @@ static struct task_struct *copy_process(
      unsigned long stack_size,
      int __user *parent_tidptr,
      int __user *child_tidptr,
-     int pid)
+   struct pid *spid)
 {
  int retval;
  struct task_struct *p = NULL;
@@ -1032,7 +1032,7 @@ static struct task_struct *copy_process(
  p->did_exec = 0;
  delayacct_tsk_init(p); /* Must remain after dup_task_struct() */
  copy_flags(clone_flags, p);
- p->pid = pid;
+ p->pid = pid_nr(spid);

  INIT_LIST_HEAD(&p->children);
  INIT_LIST_HEAD(&p->sibling);
@@ -1248,24 +1248,22 @@ static struct task_struct *copy_process(
   }
  }

- if (likely(p->pid)) {

```
-  add_parent(p);
-  tracehook_init_task(p);
-
-  if (thread_group_leader(p)) {
-   p->signal->tty = current->signal->tty;
-   p->signal->pgrp = pid_nr(task_pgrp(current));
-   set_signal_session(p->signal,
-      pid_nr(task_session(current)));
-   attach_pid(p, PIDTYPE_PGID, task_pgrp(current));
-   attach_pid(p, PIDTYPE_SID, task_session(current));
+ add_parent(p);
+ tracehook_init_task(p);

-   list_add_tail_rcu(&p->tasks, &init_task.tasks);
-   __get_cpu_var(process_counts)++;
-  }
-  attach_pid(p, PIDTYPE_PID, find_pid(p->pid));
-  nr_threads++;
+ if (thread_group_leader(p)) {
+  p->signal->tty = current->signal->tty;
+  p->signal->pgrp = pid_nr(task_pgrp(current));
+  set_signal_session(p->signal,
+     pid_nr(task_session(current)));
+  attach_pid(p, PIDTYPE_PGID, task_pgrp(current));
+  attach_pid(p, PIDTYPE_SID, task_session(current));
+
+  list_add_tail_rcu(&p->tasks, &init_task.tasks);
+  __get_cpu_var(process_counts)++;
   }
+ attach_pid(p, PIDTYPE_PID, spid);
+ nr_threads++;

  total_forks++;
  spin_unlock(&current->sighand->siglock);
@@ -1334,7 +1332,8 @@ struct task_struct * __cpuinit fork_idle
  struct task_struct *task;
  struct pt_regs regs;

- task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL, 0);
+ task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL,
+     &init_struct_pid);
  if (!IS_ERR(task))
   init_idle(task, cpu);

@@ -1362,7 +1361,7 @@ long do_fork(unsigned long clone_flags,
   return -EAGAIN;
  nr = pid->nr;
```

```
- p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, nr);
+ p = copy_process(clone_flags, stack_start, regs, stack_size, parent_tidptr, child_tidptr, pid);
  /*
   * Do this prior waking up the new thread - the thread pointer
   * might get invalid after that point, if the thread exits quickly.
```

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH] Use struct pid parameter in copy_process()
Posted by ebiederm on Fri, 23 Feb 2007 06:40:58 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com writes:

> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> Subject: [PATCH] Use struct pid parameter in copy_process()
>
> Modify copy_process() to take a struct pid * parameter instead of a pid_t.
> This simplifies the code a bit and also avoids having to call find_pid()
> to convert the pid_t to a struct pid.

I would recommend doing this in 2 steps:
- One patch to kill the likely(p->pid).
- And another to kill change the pid argument.

The indentation change makes it really hard to see what
the change in pid argument buys.

This also needs to be part of the patchset that adds a dummy
struct pid to init, to make the dependency clear.

Also given that you change the type there is no need to change
the name of the pid parameter to copy process, and the spid
name just looks strange.

Eric

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH] Use struct pid parameter in copy_process()
Posted by Sukadev Bhattiprolu on Fri, 23 Feb 2007 23:26:57 GMT

Eric W. Biederman [ebiederm@xmission.com] wrote:
| sukadev@us.ibm.com writes:
|
| > From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
| > Subject: [PATCH] Use struct pid parameter in copy_process()
| >
| > Modify copy_process() to take a struct pid * parameter instead of a pid_t.
| > This simplifies the code a bit and also avoids having to call find_pid()
| > to convert the pid_t to a struct pid.
|
| I would recommend doing this in 2 steps:
| - One patch to kill the likely(p->pid).
| - And another to kill change the pid argument.

Yes. I can break that up into two patches, but I missed and Badari
pointed the other caller to copy_process()

```
struct task_struct * __cpuinit fork_idle(int cpu)
{
	struct task_struct *task;
	struct pt_regs regs;

	task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL, 0);
	if (!IS_ERR(task))
		init_idle(task, cpu);

	return task;
}
```

Now this is passing a null struct pid which would not be good
if I remove the if (likely(p->pid)) check in copy_process().

Does this copy_process() mean there can be multiple tasks with
pid_t == 0 (one per cpu on an SMP system) ?

Can we simply attach all those tasks to init_struct_pid by passing
in &init_struct_pid to the above copy_process() ?

|
| The indentation change makes it really hard to see what
| the change in pid argument buys.

Right.

|
| This also needs to be part of the patchset that adds a dummy
| struct pid to init, to make the dependency clear.

Ok.

|
| Also given that you change the type there is no need to change
| the name of the pid parameter to copy process, and the spid
| name just looks strange.

Ok.
|
| Eric

_____
Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers

Subject: Re: [PATCH] Use struct pid parameter in copy_process()
Posted by ebiederm on Sat, 24 Feb 2007 03:29:02 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com writes:

> Yes. I can break that up into two patches, but I missed and Badari
> pointed the other caller to copy_process()
>
> struct task_struct * __cpuinit fork_idle(int cpu)
> {
> struct task_struct *task;
> struct pt_regs regs;
>
> task = copy_process(CLONE_VM, 0, idle_regs(&regs), 0, NULL, NULL, 0);
> if (!IS_ERR(task))
> init_idle(task, cpu);
>
> return task;
> }
>
> Now this is passing a null struct pid which would not be good
> if I remove the if (likely(p->pid)) check in copy_process().
>
> Does this copy_process() mean there can be multiple tasks with
> pid_t == 0 (one per cpu on an SMP system) ?

Yes. The idle processes.

> Can we simply attach all those tasks to init_struct_pid by passing
> in &init_struct_pid to the above copy_process() ?

Yes.

Eric

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers