
Subject: [PATCH 3/5] Use struct pid parameter in copy_process()
Posted by [Sukadev Bhattiprolu](#) on Tue, 13 Mar 2007 04:44:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [PATCH 3/5] 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.

Changelog:

- Fixed Badari Pulavarty's comments and passed in &init_struct_pid from fork_idle().
- Fixed Eric Biederman's comments and simplified this patch and used a new patch to remove the likely(pid) check.

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
Acked-by: Eric W. Biederman <ebiederm@xmission.com>

kernel/fork.c | 11 ++++++-----
1 file changed, 6 insertions(+), 5 deletions(-)

Index: lx26-21-rc3-mm2/kernel/fork.c

```
=====
--- lx26-21-rc3-mm2.orig/kernel/fork.c 2007-03-12 17:16:39.000000000 -0700
+++ lx26-21-rc3-mm2/kernel/fork.c 2007-03-12 17:17:48.000000000 -0700
@@ -966,7 +966,7 @@ static struct task_struct *copy_process(
     unsigned long stack_size,
     int __user *parent_tidptr,
     int __user *child_tidptr,
-    int pid)
+    struct pid *pid)
 {
     int retval;
     struct task_struct *p = NULL;
@@ -1033,7 +1033,7 @@ 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(pid);
```

```

INIT_LIST_HEAD(&p->children);
INIT_LIST_HEAD(&p->sibling);
@@ -1265,7 +1265,7 @@ static struct task_struct *copy_process(
    list_add_tail_rcu(&p->tasks, &init_task.tasks);
    __get_cpu_var(process_counts)++;
}
- attach_pid(p, PIDTYPE_PID, find_pid(p->pid));
+ attach_pid(p, PIDTYPE_PID, pid);
  nr_threads++;
}

@@ -1336,7 +1336,8 @@ 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);

@@ -1364,7 +1365,7 @@ 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>
