Subject: [PATCH 1/4] pidns: Remove the child_reaper special case from de_thread.
Posted by ebiederm on Wed, 12 Dec 2007 13:27:00 GMT
View Forum Message <> Reply to Message

This patch inspired by Oleg's leader_pid change for the timer code now
tracks the child_reaper by pid instead of by task_struct, allowing us
to not need to update anything when de_thread is run.

The logic in de_thread that is supposed to let a threaded init call
exec has bit rotted. Since the signal drop code does not look at
child_reaper changing child_reaper early is of no help. Since
zap_other_threads uses sigaddset my previous patch to drop signals
sent to init before queueing them actually fixes this case.

Since the child_reaper never exits we don't have to worry about
reference counting the pid.

By tracking the child_reaper of a pid namespace by it's task group id,
we don't need to do anything when the leading task in the task group
exits.

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
---
 fs/exec.c | 8 --------
 include/linux/pid_namespace.h | 9 ++-------
 init/main.c | 2 +-
 kernel/fork.c | 2 +-
 kernel/pid.c | 10 +++++++++-
 5 files changed, 13 insertions(+), 18 deletions(-)

diff --git a/fs/exec.c b/fs/exec.c
index f73edfc..96b4822 100644
--- a/fs/exec.c
+++ b/fs/exec.c
@@ -770,14 +770,6 @@ static int de_thread(struct task_struct *tsk)
  return -EAGAIN;
 }

- /*
- * child_reaper ignores SIGKILL, change it now.
- * Reparenting needs write_lock on tasklist_lock,
- * so it is safe to do it under read_lock.
- */
- if (unlikely(tsk->group_leader == task_child_reaper(tsk)))
- task_active_pid_ns(tsk)->child_reaper = tsk;
-
 sig->group_exit_task = tsk;
 zap_other_threads(tsk);

```
  read_unlock(&tasklist_lock);
diff --git a/include/linux/pid_namespace.h b/include/linux/pid_namespace.h
index ab13faa..2e67033 100644
--- a/include/linux/pid_namespace.h
+++ b/include/linux/pid_namespace.h
@@ -18,7 +18,7 @@ struct pid_namespace {
  struct kref kref;
  struct pidmap pidmap[PIDMAP_ENTRIES];
  int last_pid;
- struct task_struct *child_reaper;
+ struct pid *child_reaper;
  struct kmem_cache *pid_cachep;
  int level;
  struct pid_namespace *parent;
@@ -75,11 +75,6 @@ static inline void zap_pid_ns_processes(struct pid_namespace *ns)
 #endif /* CONFIG_PID_NS */

 extern struct pid_namespace *task_active_pid_ns(struct task_struct *tsk);
-
-static inline struct task_struct *task_child_reaper(struct task_struct *tsk)
-{
- BUG_ON(tsk != current);
- return tsk->nsproxy->pid_ns->child_reaper;
-}
+extern struct task_struct *task_child_reaper(struct task_struct *tsk);

 #endif /* _LINUX_PID_NS_H */
diff --git a/init/main.c b/init/main.c
index 5c6df6b..57404b6 100644
--- a/init/main.c
+++ b/init/main.c
@@ -828,7 +828,7 @@ static int __init kernel_init(void * unused)
   * assumptions about where in the task array this
   * can be found.
   */
- init_pid_ns.child_reaper = current;
+ init_pid_ns.child_reaper = task_pid(current);

  cad_pid = task_pid(current);

diff --git a/kernel/fork.c b/kernel/fork.c
index a210860..20d26a5 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1305,7 +1305,7 @@ static struct task_struct *copy_process(unsigned long clone_flags,

   if (thread_group_leader(p)) {
    if (clone_flags & CLONE_NEWPID)
```

```
-    p->nsproxy->pid_ns->child_reaper = p;
+    p->nsproxy->pid_ns->child_reaper = pid;

    p->signal->leader_pid = pid;
    p->signal->tty = current->signal->tty;
diff --git a/kernel/pid.c b/kernel/pid.c
index 873c00f..294fc28 100644
--- a/kernel/pid.c
+++ b/kernel/pid.c
@@ -76,7 +76,7 @@ struct pid_namespace init_pid_ns = {
  },
  .last_pid = 0,
  .level = 0,
- .child_reaper = &init_task,
+ .child_reaper = &init_struct_pid,
 };
 EXPORT_SYMBOL_GPL(init_pid_ns);

@@ -484,6 +484,14 @@ struct pid_namespace *task_active_pid_ns(struct task_struct *tsk)
 }
 EXPORT_SYMBOL_GPL(task_active_pid_ns);

+struct task_struct *task_child_reaper(struct task_struct *tsk)
+{
+ struct pid_namespace *ns;
+ BUG_ON(tsk != current);
+ ns = task_active_pid_ns(tsk);
+ return pid_task(ns->child_reaper, PIDTYPE_PID);
+}
+
 /*
  * Used by proc to find the first pid that is greater then or equal to nr.
  *
--
1.5.3.rc6.17.g1911
```

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