Subject: - remove-find_attach_pid.patch removed from -mm tree
Posted by Andrew Morton on Thu, 01 Feb 2007 01:25:08 GMT
View Forum Message <> Reply to Message

The patch titled
    Remove find_attach_pid()
has been removed from the -mm tree. Its filename was
    remove-find_attach_pid.patch

This patch was dropped because an updated version will be merged

---
Subject: Remove find_attach_pid()
From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Remove find_attach_pid() interface and have callers of attach_pid() pass in
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 W. Biederman <ebiederm@xmission.com>
Cc: <containers@lists.osdl.org>
Signed-off-by: Andrew Morton <akpm@osdl.org>
---

 fs/exec.c | 2 +-
 include/linux/pid.h | 11 ++---------
 kernel/exit.c | 4 ++--
 kernel/fork.c | 13 ++++++++-----
 kernel/pid.c | 6 ++++++
 kernel/sys.c | 2 +-
 6 files changed, 20 insertions(+), 18 deletions(-)

diff -puN fs/exec.c~remove-find_attach_pid fs/exec.c
--- a/fs/exec.c~remove-find_attach_pid
+++ a/fs/exec.c
@@ -701,7 +701,7 @@ static int de_thread(struct task_struct
    */
   detach_pid(tsk, PIDTYPE_PID);
   tsk->pid = leader->pid;
- find_attach_pid(tsk, PIDTYPE_PID, tsk->pid);
+ attach_pid(tsk, PIDTYPE_PID, find_pid(tsk->pid));
   transfer_pid(leader, tsk, PIDTYPE_PGID);
   transfer_pid(leader, tsk, PIDTYPE_SID);
   list_replace_rcu(&leader->tasks, &tsk->tasks);
diff -puN include/linux/pid.h~remove-find_attach_pid include/linux/pid.h

```
--- a/include/linux/pid.h~remove-find_attach_pid
+++ a/include/linux/pid.h
@@ -84,18 +84,11 @@ extern struct pid *find_get_pid(int nr);
 extern struct pid *find_ge_pid(int nr);

 /*
- * attach_pid(), find_attach_pid() and detach_pid() must be called with the
- * tasklist_lock write-held.
+ * attach_pid() and detach_pid() must be called with the tasklist_lock
+ * write-held.
  */
 extern int FASTCALL(attach_pid(struct task_struct *task, enum pid_type type,
     struct pid *pid));
-
-static inline int find_attach_pid(struct task_struct *task, enum pid_type type,
-    int nr)
-{
- return attach_pid(task, type, find_pid(nr));
-}
-
 extern void FASTCALL(detach_pid(struct task_struct *task, enum pid_type));
 extern void FASTCALL(transfer_pid(struct task_struct *old,
       struct task_struct *new, enum pid_type));
diff -puN kernel/exit.c~remove-find_attach_pid kernel/exit.c
--- a/kernel/exit.c~remove-find_attach_pid
+++ a/kernel/exit.c
@@ -300,12 +300,12 @@ void __set_special_pids(pid_t session, p
  if (process_session(curr) != session) {
   detach_pid(curr, PIDTYPE_SID);
   set_signal_session(curr->signal, session);
-  find_attach_pid(curr, PIDTYPE_SID, session);
+  attach_pid(curr, PIDTYPE_SID, find_pid(session));
  }
  if (process_group(curr) != pgrp) {
   detach_pid(curr, PIDTYPE_PGID);
   curr->signal->pgrp = pgrp;
-  find_attach_pid(curr, PIDTYPE_PGID, pgrp);
+  attach_pid(curr, PIDTYPE_PGID, find_pid(pgrp));
  }
 }

diff -puN kernel/fork.c~remove-find_attach_pid kernel/fork.c
--- a/kernel/fork.c~remove-find_attach_pid
+++ a/kernel/fork.c
@@ -1243,16 +1243,19 @@ static struct task_struct *copy_process(
    __ptrace_link(p, current->parent);

   if (thread_group_leader(p)) {
```

```
+   pid_t pgid = process_group(current);
+   pid_t sid = process_session(current);
+
    p->signal->tty = current->signal->tty;
-   p->signal->pgrp = process_group(current);
-   set_signal_session(p->signal, process_session(current));
-   find_attach_pid(p, PIDTYPE_PGID, process_group(p));
-   find_attach_pid(p, PIDTYPE_SID, process_session(p));
+   p->signal->pgrp = pgid;
+   set_signal_session(p->signal, sid);
+   attach_pid(p, PIDTYPE_PGID, find_pid(pgid));
+   attach_pid(p, PIDTYPE_SID, find_pid(sid));

    list_add_tail_rcu(&p->tasks, &init_task.tasks);
    __get_cpu_var(process_counts)++;
   }
-  find_attach_pid(p, PIDTYPE_PID, p->pid);
+  attach_pid(p, PIDTYPE_PID, find_pid(p->pid));
   nr_threads++;
  }

diff -puN kernel/pid.c~remove-find_attach_pid kernel/pid.c
--- a/kernel/pid.c~remove-find_attach_pid
+++ a/kernel/pid.c
@@ -247,6 +247,9 @@ struct pid * fastcall find_pid(int nr)
 }
 EXPORT_SYMBOL_GPL(find_pid);

+/*
+ * attach_pid() must be called with the tasklist_lock write-held.
+ */
 int fastcall attach_pid(struct task_struct *task, enum pid_type type,
     struct pid *pid)
 {
@@ -259,6 +262,9 @@ int fastcall attach_pid(struct task_stru
  return 0;
 }

+/*
+ * detach_pid() must be called with the tasklist_lock write-held.
+ */
 void fastcall detach_pid(struct task_struct *task, enum pid_type type)
 {
  struct pid_link *link;
diff -puN kernel/sys.c~remove-find_attach_pid kernel/sys.c
--- a/kernel/sys.c~remove-find_attach_pid
+++ a/kernel/sys.c
@@ -1487,7 +1487,7 @@ asmlinkage long sys_setpgid(pid_t pid, p
```

```
  if (process_group(p) != pgid) {
   detach_pid(p, PIDTYPE_PGID);
   p->signal->pgrp = pgid;
-  find_attach_pid(p, PIDTYPE_PGID, pgid);
+  attach_pid(p, PIDTYPE_PGID, find_pid(pgid));
  }

  err = 0;
```
_

Patches currently in -mm which might be from sukadev@us.ibm.com are

remove-find_attach_pid.patch
statically-initialize-struct-pid-for-swapper.patch
explicitly-set-pgid-sid-of-init.patch

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