
Subject: [PATCH] attach_pid() with struct pid parameter
Posted by [Sukadev Bhattiprolu](#) on Thu, 11 Jan 2007 13:04:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>

Implement a new version of attach_pid() with a struct pid parameter and wrap find_attach_pid() around it. attach_pid() would also be used in subsequent container patches.

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: containers@lists.osdl.org

include/linux/pid.h | 28 ++++++-----

kernel/pid.c | 7 +++----

2 files changed, 20 insertions(+), 15 deletions(-)

Index: lx26-20-rc2-mm1/include/linux/pid.h

=====

--- lx26-20-rc2-mm1.orig/include/linux/pid.h 2007-01-11 04:44:06.674046656 -0800

+++ lx26-20-rc2-mm1/include/linux/pid.h 2007-01-11 04:44:56.820423248 -0800

@@ -72,17 +72,6 @@ extern struct task_struct *FASTCALL(get_
extern struct pid *get_task_pid(struct task_struct *task, enum pid_type type);

/*

- * find_attach_pid() and detach_pid() must be called with the tasklist_lock

- * write-held.

- */

-extern int FASTCALL(find_attach_pid(struct task_struct *task,
- enum pid_type type, int 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));

-

-/*

* look up a PID in the hash table. Must be called with the tasklist_lock

* or rcu_read_lock() held.

*/

@@ -94,6 +83,23 @@ extern struct pid *FASTCALL(find_pid(int
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.
+ */
+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));
+
+extern struct pid *alloc_pid(void);
+extern void FASTCALL(free_pid(struct pid *pid));

```

Index: lx26-20-rc2-mm1/kernel/pid.c

```

=====
--- lx26-20-rc2-mm1.orig/kernel/pid.c 2007-01-11 04:44:06.674046656 -0800
+++ lx26-20-rc2-mm1/kernel/pid.c 2007-01-11 04:44:56.821423096 -0800
@@ -247,14 +247,13 @@ struct pid * fastcall find_pid(int nr)
 }
 EXPORT_SYMBOL_GPL(find_pid);

-int fastcall find_attach_pid(struct task_struct *task, enum pid_type type,
- int nr)
+int fastcall attach_pid(struct task_struct *task, enum pid_type type,
+ struct pid *pid)
 {
 struct pid_link *link;
- struct pid *pid;

 link = &task->pids[type];
- link->pid = pid = find_pid(nr);
+ link->pid = pid;
 hlist_add_head_rcu(&link->node, &pid->tasks[type]);

 return 0;

```

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>

Subject: Re: [PATCH] attach_pid() with struct pid parameter

Posted by [serue](#) on Thu, 11 Jan 2007 15:46:02 GMT

[View Forum Message](#) <> [Reply to Message](#)

Could you also add a comment above both `find_attach_pid()` and `attach_pid()` saying they are always called with the `tasklist_lock` write-held? Keeps each patch reader from having to go verify that...

thanks,
-serge

Quoting Sukadev Bhattiprolu (sukadev@us.ibm.com):

```
>
> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
>
> Implement a new version of attach_pid() with a struct pid parameter and
> wrap find_attach_pid() around it. attach_pid() would also be used in
> subsequent container patches.
>
> 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: containers@lists.osdl.org
> ---
> include/linux/pid.h | 28 ++++++-----
> kernel/pid.c        | 7 +++---
> 2 files changed, 20 insertions(+), 15 deletions(-)
>
> Index: lx26-20-rc2-mm1/include/linux/pid.h
> =====
> --- lx26-20-rc2-mm1.orig/include/linux/pid.h 2007-01-11 04:44:06.674046656 -0800
> +++ lx26-20-rc2-mm1/include/linux/pid.h 2007-01-11 04:44:56.820423248 -0800
> @@ -72,17 +72,6 @@ extern struct task_struct *FASTCALL(get_
> extern struct pid *get_task_pid(struct task_struct *task, enum pid_type type);
>
> /*
> - * find_attach_pid() and detach_pid() must be called with the tasklist_lock
> - * write-held.
> - */
> -extern int FASTCALL(find_attach_pid(struct task_struct *task,
> -    enum pid_type type, int 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));
> -
> -/*
> - * look up a PID in the hash table. Must be called with the tasklist_lock
```

```

> * or rcu_read_lock() held.
> */
> @@ -94,6 +83,23 @@ extern struct pid *FASTCALL(find_pid(int
> 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.
> + */
> +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));
> +
> extern struct pid *alloc_pid(void);
> extern void FASTCALL(free_pid(struct pid *pid));
>
> Index: lx26-20-rc2-mm1/kernel/pid.c
> =====
> --- lx26-20-rc2-mm1.orig/kernel/pid.c 2007-01-11 04:44:06.674046656 -0800
> +++ lx26-20-rc2-mm1/kernel/pid.c 2007-01-11 04:44:56.821423096 -0800
> @@ -247,14 +247,13 @@ struct pid * fastcall find_pid(int nr)
> }
> EXPORT_SYMBOL_GPL(find_pid);
>
> -int fastcall find_attach_pid(struct task_struct *task, enum pid_type type,
> - int nr)
> +int fastcall attach_pid(struct task_struct *task, enum pid_type type,
> + struct pid *pid)
> {
> struct pid_link *link;
> - struct pid *pid;
>
> link = &task->pids[type];
> - link->pid = pid = find_pid(nr);
> + link->pid = pid;
> hlist_add_head_rcu(&link->node, &pid->tasks[type]);
>
> return 0;

```

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