Subject: Re: [PATCH] pid: Extend/Fix pid_vnr
Posted by Pavel Emelianov on Thu, 06 Dec 2007 10:02:03 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> pid_vnr returns the user space pid with respect to the pid namespace
> the struct pid was allocated in. What we want before we return a pid
> to user space is the user space pid with respect to the pid namespace
> of current.
>
> pid_vnr is a very nice optimization but because it isn't quite what we
> want it is easy to use pid_vnr at times when we aren't certain the
> struct pid was allocated in our pid namespace.
>
> Currently this describes at least tiocgpgrp and tiocgsid in ttyio.c
> the parent process reported in the core dumps and the parent
> process in get_signal_to_deliver.
>
> So unless the performance impact is huge having an interface that does
> what we want instead of always what we want should be much more
> reliable and much less error prone.
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

Yup. xxx_vnr with the existing meaning turns out to be useless.

Thanks, Eric.

Acked-by: Pavel Emelyanov <xemul@openvz.org>

> ---
> include/linux/pid.h | 14 +++----------
> include/linux/sched.h | 5 ++---
> kernel/pid.c | 6 ++++++
> 3 files changed, 11 insertions(+), 14 deletions(-)
>
> diff --git a/include/linux/pid.h b/include/linux/pid.h
> index 061abb6..b91f473 100644
> --- a/include/linux/pid.h
> +++ b/include/linux/pid.h
> @@ -127,9 +127,8 @@ extern void FASTCALL(free_pid(struct pid *pid));
> * the helpers to get the pid's id seen from different namespaces
> *
> * pid_nr() : global id, i.e. the id seen from the init namespace;
> - * pid_vnr() : virtual id, i.e. the id seen from the namespace this pid
> - * belongs to. this only makes sence when called in the
> - * context of the task that belongs to the same namespace;
> + * pid_vnr() : virtual id, i.e. the id seen from the pid namespace of

```
> + *               current.
>   * pid_nr_ns() : id seen from the ns specified.
>   *
>   * see also task_xid_nr() etc in include/linux/sched.h
> @@ -144,14 +143,7 @@ static inline pid_t pid_nr(struct pid *pid)
>  }
>
>  pid_t pid_nr_ns(struct pid *pid, struct pid_namespace *ns);
> -
> -static inline pid_t pid_vnr(struct pid *pid)
> -{
> - pid_t nr = 0;
> - if (pid)
> -  nr = pid->numbers[pid->level].nr;
> - return nr;
> -}
> +pid_t pid_vnr(struct pid *pid);
>
>  #define do_each_pid_task(pid, type, task)    \
>   do {        \
> diff --git a/include/linux/sched.h b/include/linux/sched.h
> index 1b1e25b..9293114 100644
> --- a/include/linux/sched.h
> +++ b/include/linux/sched.h
> @@ -1286,9 +1286,8 @@ struct pid_namespace;
>   * from various namespaces
>   *
>   * task_xid_nr()     : global id, i.e. the id seen from the init namespace;
> - * task_xid_vnr()    : virtual id, i.e. the id seen from the namespace the task
> - *                     belongs to. this only makes sence when called in the
> - *                     context of the task that belongs to the same namespace;
> + * task_xid_vnr()    : virtual id, i.e. the id seen from the pid namespace of
> + *                     current.
>   * task_xid_nr_ns()  : id seen from the ns specified;
>   *
>   * set_task_vxid()   : assigns a virtual id to a task;
> diff --git a/kernel/pid.c b/kernel/pid.c
> index 21f027c..c507ca7 100644
> --- a/kernel/pid.c
> +++ b/kernel/pid.c
> @@ -442,6 +442,12 @@ pid_t pid_nr_ns(struct pid *pid, struct pid_namespace *ns)
>   return nr;
>  }
>
> +pid_t pid_vnr(struct pid *pid)
> +{
> + return pid_nr_ns(pid, current->nsproxy->pid_ns);
> +}
```

> +EXPORT_SYMBOL_GPL(pid_vnr);
> +
> pid_t task_pid_nr_ns(struct task_struct *tsk, struct pid_namespace *ns)
> {
>   return pid_nr_ns(task_pid(tsk), ns);

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