Subject: Re: [PATCH 2/13] Small preparations for namespaces
Posted by serue on Fri, 25 May 2007 13:55:59 GMT
View Forum Message <> Reply to Message

Quoting Pavel Emelianov (xemul@openvz.org):
> Serge E. Hallyn wrote:
> > Quoting Pavel Emelianov (xemul@openvz.org):
> >> Serge E. Hallyn wrote:
> >>> Quoting Pavel Emelianov (xemul@openvz.org):
> >>>> This includes #ifdefs in get/put_pid_ns and rewriting
> >>>> the child_reaper() function to the more logical view.
> >>>>
> >>>> This doesn't fit logically into any other patch so
> >>>> I decided to make it separate.
> >>>>
> >>>> Signed-off-by: Pavel Emelianov <xemul@openvz.org>
> >>>>
> >>>> ---
> >>>>
> >>>> diff --git a/include/linux/pid_namespace.h b/include/linux/pid_namespace.h
> >>>> index 169c6c2..7af7191 100644
> >>>> --- a/include/linux/pid_namespace.h
> >>>> +++ b/include/linux/pid_namespace.h
> >>>> @@ -26,7 +26,9 @@ extern struct pid_namespace init_pid_ns;
> >>>>
> >>>> static inline void get_pid_ns(struct pid_namespace *ns)
> >>>> {
> >>>> +#ifdef CONFIG_PID_NS
> >>>>   kref_get(&ns->kref);
> >>>> +#endif
> >>>> }
> >>>>
> >>>> extern struct pid_namespace *copy_pid_ns(int flags, struct pid_namespace *ns);
> >>>> @@ -34,12 +36,15 @@ extern void free_pid_ns(struct kref *kre
> >>>>
> >>>> static inline void put_pid_ns(struct pid_namespace *ns)
> >>>> {
> >>>> +#ifdef CONFIG_PID_NS
> >>>>   kref_put(&ns->kref, free_pid_ns);
> >>>> +#endif
> >>>> }
> >>>>
> >>>> static inline struct task_struct *child_reaper(struct task_struct *tsk)
> >>>> {
> >>>> - return init_pid_ns.child_reaper;
> >>>> + BUG_ON(tsk != current);
> >>>> + return tsk->nsproxy->pid_ns->child_reaper;
> >>>> }

> >>>>
> >>>> #endif /* _LINUX_PID_NS_H */
> >>> This can't be bisect-safe, right? You can't just use
> >>> tsk->nsproxy->pid_ns, as you've pointed out yourself.
> >> I can :) See - I have a proving BUG_ON() here.
> >
> > I didn't know BUG_ON()'s actually warded off bugs :)
>
> It does not, but it says to code reader that this call
> expects something special. In this case - tsk is expected
> to be current always. And it is.

I don't think that's sufficient.

It's been awhile so I'm fuzzy on the details, but I think we only fixed
the race by always returning init_pid_ns instead of tsk->nsproxy_pid_ns,
and tsk being current is not safe.

> > You've tested this with the infamous NFS testcase?
>
> What testcase do you mean?

http://lkml.org/lkml/2007/1/17/65

> > I don't see *why* it would work for you, but if you claim it does, I
> > guess you'd know better than I :)
>
> I don't get you here. I've checked that the task passed to
> child_reaper is current always. This BUG_ON prevents later
> code from passing arbitrary task to it.

I don't think that's enough.

thanks,
-serge