Subject: Re: [PATCH] vt: Rework the console spawning variables.
Posted by Oleg Nesterov on Mon, 11 Sep 2006 01:05:34 GMT

View Forum Message <> Reply to Message

On 09/10, Eric W. Biederman wrote:
>
> Ok. I think I see the where the confusion is. We were looking
> at different parts of the puzzle. But I we need to resolve this
> to make certain I didn't do something clever and racy.

Yes, I think we misunderstood each other :)

> As for the rest of your suggestion it would not be hard to be able to
> follow a struct pid pointer in an rcu safe way, and we do in the pid
> hash table. In other contexts so far I always have other variables
> that need to be updated in concert, so there isn't a point in coming
> up with a lockless implementation. I believe vt_pid is the only
> case that I have run across where this is a problem and I have
> at least preliminary patches for every place where signals are
> sent.
>
> Updating this old code is painful.

No, no, we shouldn't change the old code, it is fine.

Just in case, to avoid any possible confusion.

put_pid(pid) has the following restrictions. The caller should ensure
that any other possible reference to this pid "owns" it (did get_pid()).

So we can add a new helper, put_pid_rcu(). It is ok if this pid is used
in parallel under rcu_read_lock() without bumping pid->count. Contrary,
the only restriction those users must not call get_pid(pid).

But yes, you are right, I don't see an immediate usage of put_pid_rcu().

Oleg.