Subject: Re: [PATCH 4/9] allow killing tasks in your own or child userns
Posted by akpm on Thu, 24 Feb 2011 00:54:30 GMT
View Forum Message <> Reply to Message

On Thu, 24 Feb 2011 00:48:18 +0000
"Serge E. Hallyn" <serge@hallyn.com> wrote:

> Quoting Andrew Morton (akpm@linux-foundation.org):
> > On Thu, 17 Feb 2011 15:03:25 +0000
> > "Serge E. Hallyn" <serge@hallyn.com> wrote:
> >
> > > /*
> > > + * called with RCU read lock from check_kill_permission()
> > > + */
> > > +static inline int kill_ok_by_cred(struct task_struct *t)
> > > +{
> > > + const struct cred *cred = current_cred();
> > > + const struct cred *tcred = __task_cred(t);
> > > +
> > > + if (cred->user->user_ns == tcred->user->user_ns &&
> > > +   (cred->euid == tcred->suid ||
> > > +   cred->euid == tcred->uid ||
> > > +   cred->uid == tcred->suid ||
> > > +   cred->uid == tcred->uid))
> > > + return 1;
> > > +
> > > + if (ns_capable(tcred->user->user_ns, CAP_KILL))
> > > + return 1;
> > > +
> > > + return 0;
> > > +}
> >
> > The compiler will inline this for us.
>
> Is that simply true with everything (worth inlining) nowadays, or is
> there a particular implicit hint to the compiler that'll make that
> happen?

We've basically stopped inlining things nowadays. gcc inlines
aggressively and sometimes we have to use noinline to stop it. Also,
modern gcc's like to ignore the inline directive anwyay, so we have to
resort to __always_inline when we disagree.

_____