Subject: Re: [PATCH] Make access to task's nsproxy liter
Posted by serue on Fri, 10 Aug 2007 14:26:15 GMT
View Forum Message <> Reply to Message

```
Quoting Oleg Nesterov (oleg@tv-sign.ru):
> On 08/10, Oleg Nesterov wrote:
> >
> > On 08/10, Serge E. Hallyn wrote:
> > >
> > > Quoting Pavel Emelyanov (xemul@openvz.org):
> > > > +/*
> > > > + * the namespaces access rules are:
> > > > + *
> > > > + *  1. only current task is allowed to change tsk->nsproxy pointer or
> > > > + *     any pointer on the nsproxy itself
> > > > + *
> > > > + *  2. when accessing (i.e. reading) current task's namespaces - no
> > > > + *     precautions should be taken - just dereference the pointers
> > > > + *
> > > > + *  3. the access to other task namespaces is performed like this
> > > > + *     rcu_read_lock();
> > > > + *     nsproxy = task_nsproxy(tsk);
> > > > + *     if (nsproxy != NULL) {
> > > > + *             / *
> > > > + *               * work with the namespaces here
> > > > + *               * e.g. get the reference on one of them
> > > > + *               * /
> > > > + *     } / *
> > > > + *         * NULL task_nsproxy() means that this task is
> > > > + *         * almost dead (zombie)
> > > > + *         * /
> > > > + *     rcu_read_unlock();
> > >
> > > And lastly, I guess that the caller to switch_task_namespaces() has
> > > to ensure that new_nsproxy either (1) is the init namespace, (2) is a
> > > brand-new namespace to which noone else has a reference, or (3) the
> > > caller has to hold a reference to the new_nsproxy across the call to
> > > switch_task_namespaces().
> > >
> > > As it happens the current calls fit (1) or (2).  Again if we happen to
> > > jump into the game of switching a task into another task's nsproxy,
> > > we'll need to be mindful of (3) so that new_nsproxy can't be tossed into
> > > the bin between
> > >
> > >  if (new)
> > >   get_nsproxy(new);
> >
> > 4) Unless tsk == current, get_task_namespaces(tsk) and get_nsproxy(tsk)
```

> >   are racy even if done under rcu_read_lock().
>
> (sorry for noise, but I'm afraid I was not clear again...)
>
> This looks OK, we don't do get_nsproxy(not_a_current), but perhaps it is
> not immediately obvious that we shouldn't.

Yes, agreed, the code as it stands is fine.

I'm only warning that several people want the ability to enter another task's namespace (and other people are squarely against it :), and if/when we try to implement that again, then simply using switch_task_namespaces() will not suffice. The caller will have to grab an extra reference to the new nsproxy (as you imply, I guess under the task_lock(target)), call switch_task_namespaces(), then drop the extra reference. (Or implement a whole new helper.)

I don't even know whether this warrants a warning or not. Hopefully anyone who'll try to implement that will be able to deduce that for themselves. But still I usually like to see such things warned against...

-serge