Subject: Re: [PATCH RESEND 2/2] Fix some kallsyms_lookup() vs rmmod races
Posted by Alexey Dobriyan on Mon, 19 Mar 2007 10:14:25 GMT
View Forum Message <> Reply to Message

On Sat, Mar 17, 2007 at 08:37:18PM +1100, Rusty Russell wrote:
> On Fri, 2007-03-16 at 12:51 +0100, Ingo Molnar wrote:
> > * Alexey Dobriyan <adobriyan@sw.ru> wrote:
> >
> > > [cc'ing folks whose proc files are affected]
> > >
> > > kallsyms_lookup() can call module_address_lookup() which iterates over
> > > modules list without module_mutex taken. Comment at the top of
> > > module_address_lookup() says it's for oops resolution so races are
> > > irrelevant, but in some cases it's reachable from regular code:
> >
> > looking at the problem from another angle: wouldnt this be something
> > that would benefit from freeze_processes()/unfreeze_processes(), and
> > hence no locking would be required?
>
> Actually, the list manipulation is done with stop_machine for this
> reason.

mmm, my changelog is slightly narrow than it should be.

 Non-emergency code is traversing modules list.
 It finds "struct module *".
 module is removed.
 "struct module *" is now meaningless, but still dereferenced.

How would all this refrigerator stuff would help? It wouldn't,

 Non-emergency code is traversing modules list.
 It finds "struct module *".
 Everything is freezed.
 Module is removed.
 Everything is unfreezed.
 "struct module *" is now meaningless, but still dereferenced.

> Alexey, is preempt enabled in your kernel?

Yes. FWIW,

CONFIG_PREEMPT=y
CONFIG_PREEMPT_BKL=y
CONFIG_DEBUG_PREEMPT=y

I very much agree with proto-patch which _copies_ all relevant
information into caller-supplied structure, keeping module_mutex private.

Time to split it sanely.