Subject: Re: [PATCH 1/2][INET] Fix potential kfree on vmalloc-ed area of request_sock_queue
Posted by davem on Thu, 15 Nov 2007 00:09:05 GMT
View Forum Message <> Reply to Message

From: Eric Dumazet <dada1@cosmosbay.com>
Date: Wed, 14 Nov 2007 20:42:38 +0100

> On Wed, 14 Nov 2007 21:08:29 +0300
> Pavel Emelyanov <xemul@openvz.org> wrote:
>
> > The request_sock_queue's listen_opt is either vmalloc-ed or
> > kmalloc-ed depending on the number of table entries. Thus it
> > is expected to be handled properly on free, which is done in
> > the reqsk_queue_destroy().
> >
> > However the error path in inet_csk_listen_start() calls
> > the lite version of reqsk_queue_destroy, called
> > __reqsk_queue_destroy, which calls the kfree unconditionally.
> >
> > Fix this and move the __reqsk_queue_destroy into a .c file as
> > it looks too big to be inline.
> >
> > Signed-off-by: Pavel Emelyanov <xemul@openvz.org>
> >
> > ---
> >
>
> > +void __reqsk_queue_destroy(struct request_sock_queue *queue)
> > +{
> > + struct listen_sock *lopt = reqsk_queue_yank_listen_sk(queue);
>
> WARNING : lopt can be NULL here (or else the locking in reqsk_queue_yank_listen_sk() would be useless ?)
>
> kfree(NULL) was ok, not NULL->nr_table_entries :)

I think for the error recovery case he is trying to fix all of this
locking is unnecessary and we know lopt is not NULL.

Pavel can you rework your fix a bit to deal with this?

Thanks a lot.