Subject: Re: [PATCH 1/4] proc: fix NULL ->i_fop oops
Posted by Stephen Smalley on Tue, 20 Nov 2007 15:05:05 GMT
View Forum Message <> Reply to Message

On Mon, 2007-11-19 at 12:51 +0000, Christoph Hellwig wrote:
> On Fri, Nov 16, 2007 at 06:06:51PM +0300, Alexey Dobriyan wrote:
> > proc_kill_inodes() can clear ->i_fop in the middle of vfs_readdir resulting in
> > NULL dereference during "file->f_op->readdir(file, buf, filler)".
> >
> > The solution is to remove proc_kill_inodes() completely:
> > a) we don't have tricky modules implementing their tricky readdir hooks which
> >   could keeping this revoke from hell.
> > b) In a situation when module is gone but PDE still alive, standard readdir
> >   will return only "." and "..", because pde->next was cleared by
> >   remove_proc_entry().
> > c) the race proc_kill_inode() destined to prevent is not completely fixed, just
> >   race window made smaller, because vfs_readdir() is run without sb_lock held and
> >   without file_list_lock held. Effectively, ->i_fop is cleared at random moment,
> >   which can't fix properly anything.
>
> Nice, getting rid of this is a very good step formwards. Unfortunately
> we have another copy of this junk in
> security/selinux/selinuxfs.c:sel_remove_entries() which would need the
> same treatment.

Can't just be dropped completely for selinux - we need a way to drop
obsolete entries from the prior policy when we load a new policy.

Is the only real problem here the clearing of f_op? If so, we can
likely remove that from sel_remove_entries() without harm, and fix the
checks for it to use something more reliable.

--
Stephen Smalley
National Security Agency