

---

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 forwards. 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

---