
Subject: Re: Re: [PATCH resend] proc: Fix proc_kill_inodes to kill dentries on all proc superblocks

Posted by [Alexey Dobriyan](#) on Thu, 01 Nov 2007 15:59:38 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thu, Nov 01, 2007 at 06:48:25PM +0300, Pavel Emelyanov wrote:

> Eric W. Biederman wrote:

> > It appears we overlooked support for removing generic proc files
> > when we added support for multiple proc super blocks. Handle
> > that now.

> >

> > Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>

>

> AFAIS this is just making the kill for all the super blocks
> we have.

>

> Acked-by: Pavel Emelyanov <xemul@openvz.org>

bzzrt, this function is executed only for directories and symlinks,
obviously buggy (you can't flip ->f_op on live file) and is going away
after readdir/rmmmod races for directories will be dealt the same way as
for regular file.

```
> > --- a/fs/proc/generic.c
> > +++ b/fs/proc/generic.c
> > @@ -561,28 +561,32 @@ static int proc_register(struct proc_dir_entry * dir, struct
proc_dir_entry * dp
> >     static void proc_kill_inodes(struct proc_dir_entry *de)
> >     {
> >         struct list_head *p;
> > -        struct super_block *sb = proc_mnt->mnt_sb;
> > +        struct super_block *sb;
> >
> >     /*
> >     * Actually it's a partial revoke().
> >     */
> > -        file_list_lock();
> > -        list_for_each(p, &sb->s_files) {
> > -            struct file * filp = list_entry(p, struct file, f_u.fu_list);
> > -            struct dentry * dentry = filp->f_path.dentry;
> > -            struct inode * inode;
> > -            const struct file_operations *fops;
> > -
> > -            if (dentry->d_op != &proc_dentry_operations)
> > -                continue;
> > -            inode = dentry->d_inode;
> > -            if (PDE(inode) != de)
> > -                continue;
```

```
> > - fops = filp->f_op;
> > - filp->f_op = NULL;
> > - fops_put(fops);
> > + spin_lock(&sb_lock);
> > + list_for_each_entry(sb, &proc_fs_type.fs_supers, s_instances) {
> > + file_list_lock();
> > + list_for_each(p, &sb->s_files) {
> > + struct file * filp = list_entry(p, struct file, f_u.fu_list);
> > + struct dentry * dentry = filp->f_path.dentry;
> > + struct inode * inode;
> > + const struct file_operations *fops;
> > +
> > + if (dentry->d_op != &proc_dentry_operations)
> > +   continue;
> > + inode = dentry->d_inode;
> > + if (PDE(inode) != de)
> > +   continue;
> > + fops = filp->f_op;
> > + filp->f_op = NULL;
> > + fops_put(fops);
> > +
> > + file_list_unlock();
> > }
> > - file_list_unlock();
> > + spin_unlock(&sb_lock);
> > }
```

Yup, all this bogo-revoke is going away RSN. you just kicked my lazy ass.

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
