Subject: Re: [RFC][PATCH] Pid namespaces vs locks interaction
Posted by serue on Thu, 06 Dec 2007 15:51:30 GMT
View Forum Message <> Reply to Message

Quoting Vitaliy Gusev (vgusev@openvz.org):
> On 6 December 2007 17:53:40 Serge E. Hallyn wrote:
> > Quoting Vitaliy Gusev (vgusev@openvz.org):
> > > Hello!
> > >
> > > I am working on pid namespaces vs locks interaction and want to evaluate
> > > the idea.
> > > fcntl(F_GETLK,..) can return pid of process for not current pid namespace
> > > (if process is belonged to the several namespaces). It is true also for
> > > pids in /proc/locks. So correct behavior is saving pointer to the struct
> > > pid of the process lock owner.
> > > --
> > > Thank,
> > > Vitaliy Gusev
> > >
> > > diff --git a/fs/locks.c b/fs/locks.c
> > > index 8b8388e..d2d3d75 100644
> > > --- a/fs/locks.c
> > > +++ b/fs/locks.c
> > > @@ -125,6 +125,7 @@
> > > #include <linux/syscalls.h>
> > > #include <linux/time.h>
> > > #include <linux/rcupdate.h>
> > > +#include <linux/pid_namespace.h>
> > >
> > > #include <asm/semaphore.h>
> > > #include <asm/uaccess.h>
> > > @@ -185,6 +186,7 @@ void locks_init_lock(struct file_lock *fl)
> > >   fl->fl_fasync = NULL;
> > >   fl->fl_owner = NULL;
> > >   fl->fl_pid = 0;
> > > + fl->fl_nspid = NULL;
> > >
> > The idea seems right, but why are you keeping fl->fl_pid around?
> >
> > Seems like the safer thing to do would be to have a separate
> > struct user_flock, with an integer pid, for communicating to userspace,
> > and a struct flock, with struct pid, for kernel use? Then fcntl_getlk()
> > and fcntl_setlk() do the appropriate conversions.
>
> fl_pid is used by nfs, fuse and gfs2. For instance nfs keeps in fl_pid some
> unique id to identify locking process between hosts - it is not a process
> pid.

Ok, but so the struct user_flock->fl_pid is being set to the task's virtual pid, while the struct kernel_flock->fl_pid is being set to task->tgid for nfsd use.

Why can't nfs just generate a uniqueid from the struct pid when it needs it?

Fuse just seems to copy the pid to report it to userspace, so it would just copy pid_vnr(kernel_flock->pid) into user_flock->fl_pid.

Anyway I haven't looked at all the uses of struct fl_pid, but you can always get the pidnr back from the struct pid if needed so there should be no problem.

The split definately seems worthwhile to me, so that user_flock->fl_pidnr can always be said to be the pid in the acting process' namespace, and flock->fl_pid can always be a struct pid, rather than having fl_pid sometimes be current->tgid, or sometimes pid_vnr(flock->fl_nspid)...

-serge

_____