

## Re: [RFC] FUSE permission modell (Was: fuse review bits)

*Source:* <http://linux.derkeiler.com/Mailing-Lists/Kernel/2005-04/3051.html>

---

*From:* Miklos Szeredi ([miklos\\_at\\_szeredi.hu](mailto:miklos_at_szeredi.hu))

*Date:* 04/11/05

To: dan@debian.org

Date: Mon, 11 Apr 2005 21:56:29 +0200

> > > *Root squashing is actually a much less obnoxious restriction. It means*  
> > > *that local uid 0 doesn't automatically correspond to remote uid 0.*  
> >  
> > *I don't agree that it's less obnoxious. Root squashing and a*  
> > *restricted directory (-rwx-----) would have exactly the same affect:*  
> > *root is denied all access.*  
>  
> *That's considerably less obnoxious, because such directories are*  
> *comparatively rare; most files, root can still read. There are still*  
> *a couple unintuitive cases where root has less privelege than a*  
> *particular non-root user, of course. But your model gives root*  
> *normally fewer privileges than the user that mounted th e FS.*

That is exactly the intended effect. If I'm at my work machine (where I'm not an admin unfortunately) and I mount my home machine with sshfs (because FUSE is installed fortunately :), then I bloody well don't want the sysadmin or some automated script of his to go mucking under the mountpoint.

> > > *But why does the kernel need to know anything about this? Why can't*  
> > > *the userspace library present the permissions appropriately to the*  
> > > *kernel?*  
> >  
> > *That is exactly what you should do if you use the default\_permissions*  
> > *options. You set the file mode, and the kernel checks the permission.*  
>  
> *So why not make default\_permissions a feature of the userspace?*

I don't understand. The userspace can't enforce the permissions. That can only be done by the kernel. The "default\_permissions" option tells the kernel to enforce permissions based on file mode. If the option is missing, then the kernel does not enforce those permissions.

> > > *I'm going to be pretty confused if I see a mode 666 file that I*  
> > > *can't even read. So will various programs.*

Re: [RFC] FUSE permission modell (Was: fuse review bits)

> >  
> > *How would you get such I file? I don't understand.*  
>  
> *The permissions exposed by the FUSE layer apparently don't correspond*  
> *to what local users can do with them. That's the problem here. It may*  
> *be that I'm completely misunderstanding you – but from what you've*  
> *described, the userspace daemon can mark a file's permissions as 666,*  
> *and then with allow\_other and allow\_root off no one else will be able*  
> *to read it, despite those permissions.*

Other users won't be able to read even the attributes, so I don't see it as a problem. It will be a special "no go" directory for anyone except the mount owner.

> > > *Except for the allow\_root bits, I think that having userspace handle*  
> > > *the issue entirely would cover both objections.*  
> >  
> > *If I want to allow unprivileged users to be able to mount their*  
> > *filesystems, then handling everything in userspace is not an option.*  
> > *For example if you could mount a filesystem in which files have*  
> > *user=root instead of your own user ID, you could probably confuse some*  
> > *applications running as root, and cause information leak. That's*  
> > *exactly why allow\_root and allow\_other are disabled for normal users.*  
> >  
> > *The only safe option that I can imagine is that the kernel will reset*  
> > *the user and group fields of the file attributes. This would again*  
> > *require a kernel option, but would be far less useful IMO.*  
>  
> *I think we've got a boundary problem here. You are exposing some*  
> *arbitrary, user-supplied values in the permissions, and then performing*  
> *sanity checks at access time; I'm suggesting performing the sanity*  
> *checking on the other side, when the permissions are supplied to the*  
> *kernel by the daemon.*

Well the sanity check on the "server" side is always enforced. You can't "trick" sftp or ftp to not check permissions. So checking on the "client" side too (where the fuse daemon is running) makes no sense, does it?

> *Why would it be less useful to show files that have been "created" by a*  
> *user as owned by that user? Or files that the user has requested no*  
> *other users be able to write as unwritable by group/other? Sure, it*  
> *makes your tarfs a little less mapped onto the tar file. But that's*  
> *one of the recurring objections to implementing archivers as*  
> *filesystems: the ownership in the archive is \_not\_ relevant to the*  
> *mounted copy.*

So this objection is now dealt with. Is that a problem?

Thanks,  
Miklos

Linux-Kernel: Re: [RFC] FUSE permission modell (Was: fuse review bits)

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to [majordomo@vger.kernel.org](mailto:majordomo@vger.kernel.org)

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>