Subject: Re: [netns] sysfs: issues porting shadow directories on top of 2.6.21-mm2
Posted by ebiederm on Wed, 20 Jun 2007 00:19:06 GMT
View Forum Message <> Reply to Message

Benjamin Thery <benjamin.thery@bull.net> writes:

> Hi Eric,
>
> For the past few weeks, I've been trying to port your netns patchset on top of
> 2.6.21-mm2. It took me a lot more time than I first expected to have something
> working.

Ok. Interesting. There are a few pieces missing to make it work on the latest -mm.
The removal of the doubly linked lists..

> I started the port based on your latest public git repository tag:
> "netns/v2.6.21-rc6-netns17".
>
> I met a few difficulties during the port the worst being porting the shadow
> directories patches on top of Greg's sysfs patches.
>
> Greg modified a lot of things in sysfs and I had to "rewrite"/adapt most of your
> "sysfs: Implement sysfs manged shadow directory support" patch. My knowledge of
> sysfs approaching zero, the result isn't that great.
>
> Any chance you've updated the patchset for a recent version of the -mm kernel?

I'm looking at it.

> Here are some issues I have with the sysfs part of the netns patchset:
>
> * The first thing I'm not sure to understand in your patch is how shadow dirs
> and there "real" counterpart are supposed to be linked (via dentry and via
> sysfs_dirent).
>
> Is it something like:
>
> /sys/class/net/ ("real" net class)
> /sys/class/net-shadow1/
> /sys/class/net-shadow2/
>
> or:
>
> /sys/class/net/
> /sys/class/net/net-shadow1/
> /sys/class/net/net-shadow2/

In the pure sysfs dirent data structures it does look like this

so I don't think your ``hack'' patch is a hack.

> In add_shadow_dir(), it seems the shadow dentry parent is "class" :
> shadow = d_alloc(dir->d_parent, &dir->d_name);
> and the shadow sysfs_dirent parent is the real "net":
> sysfs_make_dirent(dir->d_fsdata, ...);
>
> On 2.6.21-mm2, if I attach the dentry to "class" (dir->d_parent) as you did
> initially, then the shadow directory lookup "fails": it always returns the same
> shadow dir, whatever network namespace is current. Indeed,
> sysfs_shadow_follow_link() is never called with a SYSFS_DIR dentry, but always
> directly with a SYSFS_SHADOW one, and the tag comparison is never done.

> In add_shadow_dir(), I changed the d_alloc() call and passed dir instead of
> dir->d_parent, and it "solved" the issue: sysfs_shadow_follow_link() is called
> with the SYSFS_DIR dentry, and the shadow dir lookup is done.

Hmm.

> * I also have some issues with symlinks.
>
> Indeed, the way symlinks are "resolved" have changed.

Yes. The they resolve to sysfs_dirents instead of kobjects. From an implementation standpoint it should not make a big difference.

> Symlinks paths aren't
> resolved anymore using kobject linking but uses sysfs_dirent instead. So I had
> to use a dirty hack to skip shadow directories in fs/sysfs/symlink.c:
> fill_object_path()/object_path_length().

I'm not certain it is that dirty but yes. That change is needed.

Eric