
Subject: Re: [patch] unprivileged mounts update
Posted by [ebiederm](#) on Wed, 25 Apr 2007 17:21:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

Miklos Szeredi <miklos@szeredi.hu> writes:

```
>> From: Miklos Szeredi <mszeredi@suse.cz>
>>
>> - refine adding "nosuid" and "nodev" flags for unprivileged mounts:
>>   o add "nosuid", only if mounter doesn't have CAP_SETUID capability
>>   o add "nodev", only if mounter doesn't have CAP_MKNOD capability
>>
>> - allow unprivileged forced unmount, but only for FS_SAFE filesystems
>>
>> - allow mounting over special files, but not symlinks
>>
>> - for mounting and umounting check "fsuid" instead of "ruid"
>
> Andrew, please skip this patch, for now.
>
> Serge found a problem with the fsuid approach: setfsuid(nonzero) will
> remove filesystem related capabilities. So even if root is trying to
> set the "user=UID" flag on a mount, access to the target (and in case
> of bind, the source) is checked with user privileges.
```

I do have a major problem with this patchset though. We still have the unnecessary concept of user mounts. That seems only needed now for the /proc/mounts output.

All mounts should have an owner. Prior to the unprivileged mount work root owns all mounts.

```
> Root should be able to set this flag on any mountpoint, _regardless_
> of permissions.
```

We don't need a flag, and thinking of it in the context of a flag is clearly the wrong thing. Yes if we have the proper capability we should be able to explicitly specify the owner of the mount

```
> It is possible to restore filesystem capabilities after setting fsuid,
> but the interfaces are rather horrible at all levels. mount(8) can
> probably live with these, but I'm not sure that using "fsuid" over
> "ruid" has enough advantages to force this.
>
> Why did we want to use fsuid, exactly?
```

```
- Because ruid is completely the wrong thing we want mounts owned
  by whomever's permissions we are using to perform the mount.
```

There are two basic cases.

- Mounting a filesystem as who we are.

This can use fsuid with no problems. If we are suid to root to perform the mount by default we want root to own the mount so that is correct.

- Mounting a filesystem as another user.

This is the tricky case rare case needed in setup. If we aren't jumping through too many hoops to make it work when using fsuid it sounds like the right thing here as well.

How hard is it to set fsuid to a different value? I.e. What hoops does root have to jump through.

Further when using fsuid we don't need an extra flag to mount.

Plus things are a little more consistent with the rest of the linux/unix interface.

Now I can see doing something like using a special flag and not using fsuid for the one case where we explicitly want to mount a filesystem as someone else. However if only user space has to special case this (as it does anyway) and we don't have to special case it in the kernel. So much the better.

Eric

Containers mailing list

Containers@lists.linux-foundation.org

<https://lists.linux-foundation.org/mailman/listinfo/containers>
