
Subject: Re: [patch 3/9] unprivileged mounts: account user mounts

Posted by [Dave Hansen](#) on Tue, 08 Jan 2008 18:18:47 GMT

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

On Tue, 2008-01-08 at 12:35 +0100, Miklos Szeredi wrote:

> plain text document attachment

> (unprivileged-mounts-account-user-mounts.patch)

> From: Miklos Szeredi <mszeredi@suse.cz>

>

> Add sysctl variables for accounting and limiting the number of user

> mounts.

...

> +int nr_user_mounts;

> +int max_user_mounts = 1024;

Just from a containers point of view, I think this is something we'll need to fix up in the near future if it stays in the current form.

Instead of having a global tracking, perhaps we could have a per-user limit tracked in 'struct user'. The plans are to ensure that two containers' users "dave" each have a different 'struct user', so that seems to be a decent place to track it.

Also, is a read-only sysctl really the best way to get the number of user mounts back out of the kernel? What would you use it for?

Do you need any special logic for setting 'max_user_mounts' in the case where it gets set _below_ 'nr_user_mounts'?

> /* /sys/fs */

> struct kobject *fs_kobj;

> EXPORT_SYMBOL_GPL(fs_kobj);

> @@ -477,11 +480,30 @@ static struct vfsmount *skip_mnt_tree(st

> return p;

> }

>

> +static void dec_nr_user_mounts(void)

> +{

> + spin_lock(&vfsmount_lock);

> + nr_user_mounts--;

> + spin_unlock(&vfsmount_lock);

> +}

> +

> static void set_mnt_user(struct vfsmount *mnt)

> {

> BUG_ON(mnt->mnt_flags & MNT_USER);

> mnt->mnt_uid = current->fsuid;

> mnt->mnt_flags |= MNT_USER;

```

> + spin_lock(&vfsmount_lock);
> + nr_user_mounts++;
> + spin_unlock(&vfsmount_lock);
> +}

```

One little nitpick on the patch layout: It's a wee bit difficult to audit how the set function is used vs the clear one when its users don't come until the later patches. It might be worth introducing the users here, too.

```

> +static void clear_mnt_user(struct vfsmount *mnt)
> +{
> + if (mnt->mnt_flags & MNT_USER) {
> +  mnt->mnt_uid = 0;
> +  mnt->mnt_flags &= ~MNT_USER;
> +  dec_nr_user_mounts();
> + }
> }
>
> static struct vfsmount *clone_mnt(struct vfsmount *old, struct dentry *root,
> @@ -542,6 +564,7 @@ static inline void __mntput(struct vfsmo
>  */
>  WARN_ON(atomic_read(&mnt->__mnt_writers));
>  dput(mnt->mnt_root);
> + clear_mnt_user(mnt);
>  free_vfsmnt(mnt);
>  deactivate_super(sb);
> }
> @@ -1306,6 +1329,7 @@ static int do_remount(struct nameidata *
>  else
>  err = do_remount_sb(sb, flags, data, 0);
>  if (!err) {
> + clear_mnt_user(nd->path.mnt);
>  nd->path.mnt->mnt_flags = mnt_flags;
>  if (flags & MS_SETUSER)
>  set_mnt_user(nd->path.mnt);
> Index: linux/include/linux/fs.h
> =====
> --- linux.orig/include/linux/fs.h 2008-01-03 20:52:38.000000000 +0100
> +++ linux/include/linux/fs.h 2008-01-03 21:15:35.000000000 +0100
> @@ -50,6 +50,9 @@ extern struct inodes_stat_t inodes_stat;
>
>  extern int leases_enable, lease_break_time;
>
> +extern int nr_user_mounts;
> +extern int max_user_mounts;
> +
> #ifdef CONFIG_DNOTIFY

```

```

> extern int dir_notify_enable;
> #endif
> Index: linux/kernel/sysctl.c
> =====
> --- linux.orig/kernel/sysctl.c 2008-01-03 17:13:22.000000000 +0100
> +++ linux/kernel/sysctl.c 2008-01-03 21:15:35.000000000 +0100
> @@ -1288,6 +1288,22 @@ static struct ctl_table fs_table[] = {
> #endif
> #endif
> {
> + .ctl_name = CTL_UNNUMBERED,
> + .procname = "nr_user_mounts",
> + .data = &nr_user_mounts,
> + .maxlen = sizeof(int),
> + .mode = 0444,
> + .proc_handler = &proc_dointvec,
> + },
> + {
> + .ctl_name = CTL_UNNUMBERED,
> + .procname = "max_user_mounts",
> + .data = &max_user_mounts,
> + .maxlen = sizeof(int),
> + .mode = 0644,
> + .proc_handler = &proc_dointvec,
> + },
> + {
> .ctl_name = KERN_SETUID_DUMPABLE,
> .procname = "suid_dumpable",
> .data = &suid_dumpable,
>
> --
>

```

```

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

```

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