

---

Subject: [patch 10/10] allow unprivileged fuse mounts  
Posted by [Miklos Szeredi](#) on Thu, 12 Apr 2007 16:45:51 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Miklos Szeredi <[mszeredi@suse.cz](mailto:mszeredi@suse.cz)>

Use FS\_SAFE for "fuse" fs type, but not for "fuseblk".

FUSE was designed from the beginning to be safe for unprivileged users. This has also been verified in practice over many years. And unprivileged fuse mounts still require a private namespace with user mounts enabled, this is more strict than the current userspace policy.

This will enable future installations to remove the suid-root fusermount utility.

Don't require the "user\_id=" and "group\_id=" options for unprivileged mounts, but if they are present verify them for sanity.

Disallow the "allow\_other" option for unprivileged mounts.

Signed-off-by: Miklos Szeredi <[mszeredi@suse.cz](mailto:mszeredi@suse.cz)>

---

Index: linux/fs/fuse/inode.c

```
=====
--- linux.orig/fs/fuse/inode.c 2007-04-12 13:26:42.000000000 +0200
+++ linux/fs/fuse/inode.c 2007-04-12 14:04:34.000000000 +0200
@@ -311,6 +311,19 @@ static int parse_fuse_opt(char *opt, str
    d->max_read = ~0;
    d->blksize = 512;

+ /*
+  * For unprivileged mounts use current uid/gid. Still allow
+  * "user_id" and "group_id" options for compatibility, but
+  * only if they match these values.
+  */
+ if (!capable(CAP_SYS_ADMIN)) {
+   d->user_id = current->uid;
+   d->user_id_present = 1;
+   d->group_id = current->gid;
+   d->group_id_present = 1;
+ }
+
+ while ((p = strsep(&opt, ",")) != NULL) {
+   int token;
+   int value;
```

```

@@ -339,6 +352,8 @@ static int parse_fuse_opt(char *opt, str
case OPT_USER_ID:
    if (match_int(&args[0], &value))
        return 0;
+   if (d->user_id_present && d->user_id != value)
+   return 0;
    d->user_id = value;
    d->user_id_present = 1;
    break;
@@ -346,6 +361,8 @@ static int parse_fuse_opt(char *opt, str
case OPT_GROUP_ID:
    if (match_int(&args[0], &value))
        return 0;
+   if (d->group_id_present && d->group_id != value)
+   return 0;
    d->group_id = value;
    d->group_id_present = 1;
    break;
@@ -536,6 +553,10 @@ static int fuse_fill_super(struct super_
if (!parse_fuse_opt((char *) data, &d, is_bdev))
    return -EINVAL;

+ /* This is a privileged option */
+ if ((d.flags & FUSE_ALLOW_OTHER) && !capable(CAP_SYS_ADMIN))
+ return -EPERM;
+
    if (is_bdev) {
#ifdef CONFIG_BLOCK
        if (!sb_set_blocksize(sb, d.blksize))
@@ -639,6 +660,7 @@ static struct file_system_type fuse_fs_t
    .fs_flags = FS_HAS_SUBTYPE,
    .get_sb = fuse_get_sb,
    .kill_sb = kill_anon_super,
+ .fs_flags = FS_SAFE,
    };

#ifdef CONFIG_BLOCK

```

---

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

---