
Subject: [RFC] [PATCH 2/4] uid_ns: replace inode->fsuid checks under fs/
Posted by [serue](#) on Tue, 07 Nov 2006 04:19:39 GMT

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

Replace inode->fsuid in fs/*.c with inode_task_same_uid(), which
will later be used to compare uid namespaces.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>

```
fs/attr.c      | 10 ++++++----
fs/fcntl.c     |  3 +-
fs/generic_acl.c |  2 +-
fs/locks.c     |  4 +--
fs/posix_acl.c |  2 +-
fs/utimes.c    |  4 +--
6 files changed, 14 insertions(+), 11 deletions(-)
```

54f0e4ce61c74cc7419988fbbadd0a3c54e21893

diff --git a/fs/attr.c b/fs/attr.c

index 97de946..b913555 100644

--- a/fs/attr.c

+++ b/fs/attr.c

@ @ -30,20 +30,21 @ @ int inode_change_ok(struct inode *inode,

```
/* Make sure a caller can chown. */
if ((ia_valid & ATTR_UID) &&
- (current->fsuid != inode->i_uid ||
+ (!inode_task_same_uid(inode, current) ||
   attr->ia_uid != inode->i_uid) && !capable(CAP_CHOWN))
    goto error;
```

```
/* Make sure caller can chgrp. */
if ((ia_valid & ATTR_GID) &&
- (current->fsuid != inode->i_uid ||
+ (!inode_task_same_uid(inode, current) ||
   (!in_group_p(attr->ia_gid) && attr->ia_gid != inode->i_gid)) &&
   !capable(CAP_CHOWN))
    goto error;
```

```
/* Make sure a caller can chmod. */
if (ia_valid & ATTR_MODE) {
- if ((current->fsuid != inode->i_uid) && !capable(CAP_FOWNER))
+ if (!inode_task_same_uid(inode, current) &&
+ !capable(CAP_FOWNER))
    goto error;
/* Also check the setgid bit! */
```

```

    if (!in_group_p((ia_valid & ATTR_GID) ? attr->ia_gid :
@@ -53,7 +54,8 @@ int inode_change_ok(struct inode *inode,

/* Check for setting the inode time. */
if (ia_valid & (ATTR_MTIME_SET | ATTR_ETIME_SET)) {
- if (current->fsuid != inode->i_uid && !capable(CAP_FOWNER))
+ if (!inode_task_same_uid(inode, current) &&
+     !capable(CAP_FOWNER))
    goto error;
}
fine:
diff --git a/fs/fcntl.c b/fs/fcntl.c
index 8ba82c9..b1ed443 100644
--- a/fs/fcntl.c
+++ b/fs/fcntl.c
@@ -215,7 +215,8 @@ static int setfl(int fd, struct file * f

/* O_NOATIME can only be set by the owner or superuser */
if ((arg & O_NOATIME) && !(filp->f_flags & O_NOATIME))
- if (current->fsuid != inode->i_uid && !capable(CAP_FOWNER))
+ if (!inode_task_same_uid(inode, current) &&
+     !capable(CAP_FOWNER))
    return -EPERM;

/* required for strict SunOS emulation */
diff --git a/fs/generic_acl.c b/fs/generic_acl.c
index 9ccb789..a6402a9 100644
--- a/fs/generic_acl.c
+++ b/fs/generic_acl.c
@@ -78,7 +78,7 @@ generic_acl_set(struct inode *inode, str

if (S_ISLNK(inode->i_mode))
    return -EOPNOTSUPP;
- if (current->fsuid != inode->i_uid && !capable(CAP_FOWNER))
+ if (inode_task_same_uid(inode, current) && !capable(CAP_FOWNER))
    return -EPERM;
if (value) {
    acl = posix_acl_from_xattr(value, size);
diff --git a/fs/locks.c b/fs/locks.c
index e0b6a80..f5c4787 100644
--- a/fs/locks.c
+++ b/fs/locks.c
@@ -1452,7 +1452,7 @@ int setlease(struct file *filp, long arg
    struct inode *inode = dentry->d_inode;
    int error;

- if ((current->fsuid != inode->i_uid) && !capable(CAP_LEASE))
+ if (inode_task_same_uid(inode, current) && !capable(CAP_LEASE))

```

```

return -EACCES;
if (!S_ISREG(inode->i_mode))
return -EINVAL;
@@ -1486,7 +1486,7 @@ int fcntl_setlease(unsigned int fd, stru
struct inode *inode = dentry->d_inode;
int error;

- if ((current->fsuid != inode->i_uid) && !capable(CAP_LEASE))
+ if (inode_task_same_uid(inode, current) && !capable(CAP_LEASE))
return -EACCES;
if (!S_ISREG(inode->i_mode))
return -EINVAL;
diff --git a/fs/posix_acl.c b/fs/posix_acl.c
index aec931e..dd34bee 100644
--- a/fs/posix_acl.c
+++ b/fs/posix_acl.c
@@ -217,7 +217,7 @@ posix_acl_permission(struct inode *inode
switch(pa->e_tag) {
case ACL_USER_OBJ:
/* (May have been checked already) */
- if (inode->i_uid == current->fsuid)
+ if (inode_task_same_uid(inode, current))
goto check_perm;
break;
case ACL_USER:
diff --git a/fs/utimes.c b/fs/utimes.c
index 558f581..77ad8c9 100644
--- a/fs/utimes.c
+++ b/fs/utimes.c
@@ -61,7 +61,7 @@ asmlinkage long sys_utime(char __user *
if (IS_IMMUTABLE(inode))
goto mnt_drop_write_and_out;

- if (current->fsuid != inode->i_uid &&
+ if (!inode_task_same_uid(inode, current) &&
(error = vfs_permission(&nd, MAY_WRITE)) != 0)
goto mnt_drop_write_and_out;
}
@@ -119,7 +119,7 @@ long do_utimes(int dfd, char __user *fil
if (IS_IMMUTABLE(inode))
goto mnt_drop_write_and_out;

- if (current->fsuid != inode->i_uid &&
+ if (!inode_task_same_uid(inode, current) &&
(error = vfs_permission(&nd, MAY_WRITE)) != 0)
goto mnt_drop_write_and_out;
}
--

```

1.1.6

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
