Subject: [PATCH 11/25] elevate mount count for extended attributes
Posted by Dave Hansen on Mon, 11 Dec 2006 22:30:09 GMT
View Forum Message <> Reply to Message

This basically audits the callers of xattr_permission(), which
calls permission() and can perform writes to the filesystem.

Signed-off-by: Dave Hansen <haveblue@us.ibm.com>
---

 lxc-dave/fs/nfsd/nfs4proc.c | 7 ++++++-
 lxc-dave/fs/xattr.c | 14 ++++++++++++++
 2 files changed, 20 insertions(+), 1 deletion(-)

diff -puN fs/nfsd/nfs4proc.c~10-24-elevate-mount-count-for-extended-attributes fs/nfsd/nfs4proc.c
--- lxc/fs/nfsd/nfs4proc.c~10-24-elevate-mount-count-for-extended-attributes 2006-12-11 14:22:03.000000000 -0800
+++ lxc-dave/fs/nfsd/nfs4proc.c 2006-12-11 14:22:03.000000000 -0800
@@ -626,14 +626,19 @@ nfsd4_setattr(struct svc_rqst *rqstp, st
    return status;
   }
  }
+ status = mnt_want_write(current_fh->fh_export->ex_mnt);
+ if (status)
+ return status;
  status = nfs_ok;
  if (setattr->sa_acl != NULL)
   status = nfsd4_set_nfs4_acl(rqstp, &cstate->current_fh,
          setattr->sa_acl);
  if (status)
- return status;
+ goto out;
  status = nfsd_setattr(rqstp, &cstate->current_fh, &setattr->sa_iattr,
     0, (time_t)0);
+out:
+ mnt_drop_write(current_fh->fh_export->ex_mnt);
  return status;
 }

diff -puN fs/xattr.c~10-24-elevate-mount-count-for-extended-attributes fs/xattr.c
--- lxc/fs/xattr.c~10-24-elevate-mount-count-for-extended-attributes 2006-12-11 14:22:03.000000000 -0800
+++ lxc-dave/fs/xattr.c 2006-12-11 14:22:03.000000000 -0800
@@ -12,6 +12,7 @@
 #include <linux/smp_lock.h>
 #include <linux/file.h>
 #include <linux/xattr.h>
+#include <linux/mount.h>

```
 #include <linux/namei.h>
 #include <linux/security.h>
 #include <linux/syscalls.h>
@@ -237,7 +238,11 @@ sys_setxattr(char __user *path, char __u
  error = user_path_walk(path, &nd);
  if (error)
   return error;
+ error = mnt_want_write(nd.mnt);
+ if (error)
+  return error;
  error = setxattr(nd.dentry, name, value, size, flags);
+ mnt_drop_write(nd.mnt);
  path_release(&nd);
  return error;
 }
@@ -252,7 +257,11 @@ sys_lsetxattr(char __user *path, char __
  error = user_path_walk_link(path, &nd);
  if (error)
   return error;
+ error = mnt_want_write(nd.mnt);
+ if (error)
+  return error;
  error = setxattr(nd.dentry, name, value, size, flags);
+ mnt_drop_write(nd.mnt);
  path_release(&nd);
  return error;
 }
@@ -268,9 +277,14 @@ sys_fsetxattr(int fd, char __user *name,
  f = fget(fd);
  if (!f)
   return error;
+ error = mnt_want_write(f->f_vfsmnt);
+ if (error)
+  goto out_fput;
  dentry = f->f_path.dentry;
  audit_inode(NULL, dentry->d_inode);
  error = setxattr(dentry, name, value, size, flags);
+ mnt_drop_write(f->f_vfsmnt);
+out_fput:
  fput(f);
  return error;
 }
_
```

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers