Subject: [patch 14/22] elevate mount count for extended attributes
Posted by Cedric Le Goater on Thu, 07 Jun 2007 15:25:41 GMT
View Forum Message <> Reply to Message

From: Dave Hansen <hansendc@us.ibm.com>

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

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

---
 fs/nfsd/nfs4proc.c | 7 ++++++-
 fs/xattr.c | 16 ++++++++++++++--
 2 files changed, 20 insertions(+), 3 deletions(-)

Index: 2.6.22-rc4-mm2-robindmount/fs/nfsd/nfs4proc.c
===
--- 2.6.22-rc4-mm2-robindmount.orig/fs/nfsd/nfs4proc.c
+++ 2.6.22-rc4-mm2-robindmount/fs/nfsd/nfs4proc.c
@@ -626,14 +626,19 @@ nfsd4_setattr(struct svc_rqst *rqstp, st
    return status;
   }
  }
+ status = mnt_want_write(cstate->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(cstate->current_fh.fh_export->ex_mnt);
  return status;
 }

Index: 2.6.22-rc4-mm2-robindmount/fs/xattr.c
===
--- 2.6.22-rc4-mm2-robindmount.orig/fs/xattr.c
+++ 2.6.22-rc4-mm2-robindmount/fs/xattr.c
@@ -11,6 +11,7 @@
 #include <linux/slab.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>
@@ -32,8 +33,6 @@ xattr_permission(struct inode *inode, co
   * filesystem  or on an immutable / append-only inode.
   */
  if (mask & MAY_WRITE) {
-  if (IS_RDONLY(inode))
-   return -EROFS;
   if (IS_IMMUTABLE(inode) || IS_APPEND(inode))
    return -EPERM;
  }
@@ -236,7 +235,11 @@ 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;
 }
@@ -251,7 +254,11 @@ 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;
 }
@@ -267,9 +274,14 @@ 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;
 }

--
```