Subject: [RFC] [PATCH 4/4] uid_ns: Add filesystem uid checks
Posted by serue on Tue, 07 Nov 2006 04:20:32 GMT
View Forum Message <> Reply to Message

With user namespaces, two users in different namespaces might
have the same uid, but should not have access to each others
files.

Add a user_namespace pointer to the superblock, and check that
whenever the uid is checked.

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

---

 fs/super.c | 4 ++++
 include/linux/fs.h | 6 +++++-
 2 files changed, 9 insertions(+), 1 deletions(-)

b47f5fefa827b613c885fe851a8ce2642c2cb135
diff --git a/fs/super.c b/fs/super.c
index 95214f0..33354a8 100644
--- a/fs/super.c
+++ b/fs/super.c
@@ -37,6 +37,7 @@
 #include <linux/idr.h>
 #include <linux/kobject.h>
 #include <linux/mutex.h>
+#include <linux/user.h>
 #include <asm/uaccess.h>

@@ -81,6 +82,8 @@ static struct super_block *alloc_super(s
    * lock ordering than usbfs:
    */
   lockdep_set_class(&s->s_lock, &type->s_lock_key);
+ s->s_user_ns = current->nsproxy->user_ns;
+ get_user_ns(current->nsproxy->user_ns);
   down_write(&s->s_umount);
   s->s_count = S_BIAS;
   atomic_set(&s->s_active, 1);
@@ -109,6 +112,7 @@ out:
  */
 static inline void destroy_super(struct super_block *s)
 {
+ put_user_ns(s->s_user_ns);
  security_sb_free(s);
  kfree(s);

```
 }
diff --git a/include/linux/fs.h b/include/linux/fs.h
index 699c7b5..6aac556 100644
--- a/include/linux/fs.h
+++ b/include/linux/fs.h
@@ -279,6 +279,7 @@ extern int dir_notify_enable;
 #include <linux/sched.h>
 #include <linux/mutex.h>
 #include <linux/kevent.h>
+#include <linux/nsproxy.h>

 #include <asm/atomic.h>
 #include <asm/semaphore.h>
@@ -294,6 +295,7 @@ struct kstatfs;
 struct vm_area_struct;
 struct vfsmount;
 struct pagevec;
+struct user_namespace;

 extern void __init inode_init(unsigned long);
 extern void __init inode_init_early(void);
@@ -982,6 +984,7 @@ struct super_block {
  unsigned char  s_blocksize_bits;
  unsigned char  s_dirt;
  unsigned long long s_maxbytes; /* Max file size */
+ struct user_namespace *s_user_ns;
  struct file_system_type *s_type;
  struct super_operations *s_op;
  struct dquot_operations *dq_op;
@@ -2160,7 +2163,8 @@ static inline void free_secdata(void *se
 static inline int inode_task_same_uid(struct inode *ino,
   struct task_struct *tsk)
 {
- return (ino->i_uid == tsk->fsuid);
+ return (ino->i_uid == tsk->fsuid &&
+  ino->i_sb->s_user_ns == current->nsproxy->user_ns);
 }
 #endif /* __KERNEL__ */
 #endif /* _LINUX_FS_H */
--
1.1.6
```