
Subject: [RFC] [PATCH 1/3] user ns: add user_namespace ptr to vfsmount
Posted by [serue](#) on Wed, 15 Nov 2006 17:41:21 GMT
[View Forum Message](#) <> [Reply to Message](#)

From: Serge Hallyn <serue@us.ibm.com>
Subject: [RFC] [PATCH 1/3] user ns: add user_namespace ptr to vfsmount

Add user_namespace ptr to vfsmount, and define a helper to compare it
to the task's user_ns.

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

```
fs/namespace.c      | 4 ++++
include/linux/mount.h | 2 ++
include/linux/sched.h | 13 ++++++++
3 files changed, 19 insertions(+), 0 deletions(-)
```

```
diff --git a/fs/namespace.c b/fs/namespace.c
index e199769..cfd4584 100644
--- a/fs/namespace.c
+++ b/fs/namespace.c
@@ -25,6 +25,7 @@ #include <linux/namei.h>
#include <linux/security.h>
#include <linux/mount.h>
#include <linux/ramfs.h>
+#include <linux/user.h>
#include <asm/uaccess.h>
#include <asm/unistd.h>
#include "pnode.h"
@@ -56,6 +57,8 @@ struct vfsmount *alloc_vfsmnt(const char
struct vfsmount *mnt = kmem_cache_alloc(mnt_cache, GFP_KERNEL);
if (mnt) {
    memset(mnt, 0, sizeof(struct vfsmount));
+ mnt->mnt_user_ns = current->nsproxy->user_ns;
+ get_user_ns(mnt->mnt_user_ns);
    atomic_set(&mnt->mnt_count, 1);
    atomic_set(&mnt->mnt_writers, 0);
    INIT_LIST_HEAD(&mnt->mnt_hash);
@@ -109,6 +112,7 @@ void free_vfsmnt(struct vfsmount *mnt)
if (mnt->mnt_sb)
    list_del(&mnt->mnt_sb_list);
spin_unlock(&vfsmount_lock);
+ put_user_ns(mnt->mnt_user_ns);
    kfree(mnt->mnt_devname);
    kmem_cache_free(mnt_cache, mnt);
}
diff --git a/include/linux/mount.h b/include/linux/mount.h
index 8402137..827793f 100644
```

```

--- a/include/linux/mount.h
+++ b/include/linux/mount.h
@@ -21,6 +21,7 @@ struct super_block;
 struct vfsmount;
 struct dentry;
 struct mnt_namespace;
+struct user_namespace;

#define MNT_NOSUID 0x01
#define MNT_NODEV 0x02
@@ -57,6 +58,7 @@ struct vfsmount {
 struct list_head mnt_slave; /* slave list entry */
 struct vfsmount *mnt_master; /* slave is on master->mnt_slave_list */
 struct mnt_namespace *mnt_ns; /* containing namespace */
+ struct user_namespace *mnt_user_ns; /* namespace for uid interpretation */
 int mnt_pinned;
};

```

```

diff --git a/include/linux/sched.h b/include/linux/sched.h
index 47fbd23..b72357d 100644

```

```

--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -82,6 +82,8 @@ #include <linux/param.h>
#include <linux/resource.h>
#include <linux/timer.h>
#include <linux/hrtimer.h>
+#include <linux/nsproxy.h>
+#include <linux/mount.h>

```

```

#include <asm/processor.h>

```

```

@@ -1583,6 +1585,17 @@ extern int cond_resched(void);
extern int cond_resched_lock(spinlock_t * lock);
extern int cond_resched_softirq(void);

```

```

+static inline int task_mnt_same_uid(struct task_struct *tsk,
+ struct vfsmount *mnt)
+{
+ if (tsk->nsproxy == init_task.nsproxy)
+ return 1;
+ if (mnt->mnt_user_ns == tsk->nsproxy->user_ns);
+ return 1;
+ return 0;
+}
+
+
+/*
+ * Does a critical section need to be broken due to another

```

* task waiting?:

--

1.4.1

Containers mailing list

Containers@lists.osdl.org

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