Subject: [PATCH 3/8] user ns: add user_namespace ptr to vfsmount
Posted by serue on Tue, 19 Dec 2006 23:00:34 GMT
View Forum Message <> Reply to Message

From: Serge E. Hallyn <serue@us.ibm.com>
Subject: [PATCH 3/8] 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       |    3 +++
 include/linux/mount.h |    2 ++
 include/linux/sched.h |   20 ++++++++++++++++++++
 3 files changed, 25 insertions(+), 0 deletions(-)

diff --git a/fs/namespace.c b/fs/namespace.c
index 5ef336c..9f98a67 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_namespace.h>
 #include <asm/uaccess.h>
 #include <asm/unistd.h>
 #include "pnode.h"
@@ -56,6 +57,7 @@ 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 = get_user_ns(current->nsproxy->user_ns);
   atomic_set(&mnt->mnt_count, 1);
   INIT_LIST_HEAD(&mnt->mnt_hash);
   INIT_LIST_HEAD(&mnt->mnt_child);
@@ -88,6 +90,7 @@ EXPORT_SYMBOL(simple_set_mnt);

 void free_vfsmnt(struct vfsmount *mnt)
 {
+ 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 1b7e178..acdeca7 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
@@ -54,6 +55,7 @@ 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 5a3f630..450fc39 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -83,6 +83,8 @@ #include <linux/resource.h>
 #include <linux/timer.h>
 #include <linux/hrtimer.h>
 #include <linux/task_io_accounting.h>
+#include <linux/nsproxy.h>
+#include <linux/mount.h>

 #include <asm/processor.h>

@@ -1586,6 +1588,24 @@ extern int cond_resched_lock(spinlock_t
 extern int cond_resched_softirq(void);

 /*
+ * Check whether a task and a vfsmnt belong to the same uidns.
+ * Since the initial namespace is exempt from these checks,
+ * return 1 if so.  Also return 1 if the vfsmnt is exempt from
+ * such checking.  Otherwise, if the uid namespaces are different,
+ * return 0.
+ */
+static inline int task_mnt_same_uidns(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