Subject: [PATCH 6/8] user ns: implement shared mounts
Posted by serue on Tue, 19 Dec 2006 23:01:16 GMT
View Forum Message <> Reply to Message

From: Serge E. Hallyn <serue@us.ibm.com>
Subject: [PATCH 6/8] user ns: implement shared mounts

Implement shared-ns mounts, which allow containers in different user
namespaces to share mounts. Without this, containers can obviously
never even be started.

Here is a sample smount.c (based on Miklos' version) which only
does a bind mount of arg1 onto arg2, but making the destination
a shared-ns mount.

```
int main(int argc, char *argv[])
{
 int type;
 if(argc != 3) {
  fprintf(stderr, "usage: %s src dest", argv[0]);
  return 1;
 }

 fprintf(stdout, "%s %s %s\n", argv[0], argv[1], argv[2]);

 type = MS_SHARE_NS | MS_BIND;
 setfsuid(getuid());

 if(mount(argv[1], argv[2], "none", type, "") == -1) {
  perror("mount");
  return 1;
 }
 return 0;
}
```

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

```
 fs/namespace.c       |   30 ++++++++++++++++++++++++------
 fs/pnode.h           |    1 +
 include/linux/fs.h   |    1 +
 include/linux/mount.h |    1 +
 include/linux/sched.h |    2 ++
 5 files changed, 29 insertions(+), 6 deletions(-)
```

```
diff --git a/fs/namespace.c b/fs/namespace.c
index f85dd73..9cf8463 100644
--- a/fs/namespace.c
+++ b/fs/namespace.c
```

```
@@ -235,7 +235,14 @@ static struct vfsmount *clone_mnt(struct
      int flag)
 {
  struct super_block *sb = old->mnt_sb;
- struct vfsmount *mnt = alloc_vfsmnt(old->mnt_devname);
+ struct vfsmount *mnt;
+
+ if (!(old->mnt_flags & MNT_SHARE_NS)) {
+  if (old->mnt_user_ns != current->nsproxy->user_ns)
+   return ERR_PTR(-EPERM);
+ }
+
+ mnt = alloc_vfsmnt(old->mnt_devname);

  if (mnt) {
   mnt->mnt_flags = old->mnt_flags;
@@ -258,6 +265,10 @@ static struct vfsmount *clone_mnt(struct
   }
   if (flag & CL_MAKE_SHARED)
    set_mnt_shared(mnt);
+  if (flag & CL_SHARE_NS)
+   mnt->mnt_flags |= MNT_SHARE_NS;
+  else
+   mnt->mnt_flags &= ~MNT_SHARE_NS;

   /* stick the duplicate mount on the same expiry list
    * as the original if that was on one */
@@ -369,6 +380,7 @@ static int show_vfsmnt(struct seq_file *
   { MNT_NOSUID, ",nosuid" },
   { MNT_NODEV, ",nodev" },
   { MNT_NOEXEC, ",noexec" },
+  { MNT_SHARE_NS, ",share_userns" },
   { MNT_NOATIME, ",noatime" },
   { MNT_NODIRATIME, ",nodiratime" },
   { MNT_RELATIME, ",relatime" },
@@ -903,11 +915,14 @@ static int do_change_type(struct nameida
 /*
  * do loopback mount.
  */
-static int do_loopback(struct nameidata *nd, char *old_name, int recurse)
+static int do_loopback(struct nameidata *nd, char *old_name, int recurse,
+       int uidns_share)
 {
  struct nameidata old_nd;
  struct vfsmount *mnt = NULL;
  int err = mount_is_safe(nd);
+ int flag = (uidns_share ? CL_SHARE_NS : 0);
+
```

```
  if (err)
   return err;
  if (!old_name || !*old_name)
@@ -926,9 +941,9 @@ static int do_loopback(struct nameidata

  err = -ENOMEM;
  if (recurse)
-  mnt = copy_tree(old_nd.mnt, old_nd.dentry, 0);
+  mnt = copy_tree(old_nd.mnt, old_nd.dentry, flag);
  else
-  mnt = clone_mnt(old_nd.mnt, old_nd.dentry, 0);
+  mnt = clone_mnt(old_nd.mnt, old_nd.dentry, flag);

  if (!mnt || IS_ERR(mnt)) {
   err = mnt ? PTR_ERR(mnt) : -ENOMEM;
@@ -1415,9 +1430,11 @@ long do_mount(char *dev_name, char *dir_
   mnt_flags |= MNT_NODIRATIME;
  if (flags & MS_RELATIME)
   mnt_flags |= MNT_RELATIME;
+ if (flags & MS_SHARE_NS)
+  mnt_flags |= MNT_SHARE_NS;

  flags &= ~(MS_NOSUID | MS_NOEXEC | MS_NODEV | MS_ACTIVE |
-     MS_NOATIME | MS_NODIRATIME | MS_RELATIME);
+     MS_NOATIME | MS_NODIRATIME | MS_RELATIME | MS_SHARE_NS);

  /* ... and get the mountpoint */
  retval = path_lookup(dir_name, LOOKUP_FOLLOW, &nd);
@@ -1432,7 +1449,8 @@ long do_mount(char *dev_name, char *dir_
   retval = do_remount(&nd, flags & ~MS_REMOUNT, mnt_flags,
         data_page);
  else if (flags & MS_BIND)
-  retval = do_loopback(&nd, dev_name, flags & MS_REC);
+  retval = do_loopback(&nd, dev_name, flags & MS_REC,
+       mnt_flags & MNT_SHARE_NS);
  else if (flags & (MS_SHARED | MS_PRIVATE | MS_SLAVE | MS_UNBINDABLE))
   retval = do_change_type(&nd, flags);
  else if (flags & MS_MOVE)
diff --git a/fs/pnode.h b/fs/pnode.h
index d45bd8e..eb62f4c 100644
--- a/fs/pnode.h
+++ b/fs/pnode.h
@@ -22,6 +22,7 @@ #define CL_SLAVE       0x02
 #define CL_COPY_ALL   0x04
 #define CL_MAKE_SHARED   0x08
 #define CL_PROPAGATION   0x10
+#define CL_SHARE_NS   0x20
```

```
 static inline void set_mnt_shared(struct vfsmount *mnt)
 {
diff --git a/include/linux/fs.h b/include/linux/fs.h
index e7268d2..bb801cb 100644
--- a/include/linux/fs.h
+++ b/include/linux/fs.h
@@ -121,6 +121,7 @@ #define MS_PRIVATE (1<<18) /* change to
 #define MS_SLAVE (1<<19) /* change to slave */
 #define MS_SHARED (1<<20) /* change to shared */
 #define MS_RELATIME (1<<21) /* Update atime relative to mtime/ctime. */
+#define MS_SHARE_NS (1<<22) /* ignore user namespaces for permission */
 #define MS_ACTIVE (1<<30)
 #define MS_NOUSER (1<<31)

diff --git a/include/linux/mount.h b/include/linux/mount.h
index acdeca7..28e0964 100644
--- a/include/linux/mount.h
+++ b/include/linux/mount.h
@@ -35,6 +35,7 @@ #define MNT_SHRINKABLE 0x100
 #define MNT_SHARED 0x1000 /* if the vfsmount is a shared mount */
 #define MNT_UNBINDABLE 0x2000 /* if the vfsmount is a unbindable mount */
 #define MNT_PNODE_MASK 0x3000 /* propogation flag mask */
+#define MNT_SHARE_NS 0x4000  /* ignore user namespaces for permission */

 struct vfsmount {
   struct list_head mnt_hash;
diff --git a/include/linux/sched.h b/include/linux/sched.h
index 450fc39..73df38c 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -1599,6 +1599,8 @@ static inline int task_mnt_same_uidns(st
 {
   if (tsk->nsproxy == init_task.nsproxy)
     return 1;
+  if (mnt->mnt_flags & MNT_SHARE_NS)
+    return 1;
   if (mnt->mnt_user_ns == tsk->nsproxy->user_ns)
     return 1;
   return 0;
--
1.4.1
```