Subject: [PATCH 5/8] user ns: prepare copy_tree, copy_mnt, and their callers to handle errs
Posted by serue on Tue, 19 Dec 2006 23:01:06 GMT
View Forum Message <> Reply to Message

From: Serge E. Hallyn <serue@us.ibm.com>
Subject: [PATCH 5/8] user ns: prepare copy_tree, copy_mnt, and their callers to handle errs

With shareduserns and non-shareduserns mounts, it will be possible
for clone_mnt to return -EPERM if a namespace tries to bind
mount a non-shareduserns vfsmnt from another user namespace.
But currently they only return NULL, which is interpreted as
-ENOMEM. Update the callers to handle other errors.

Signed-off-by: Serge E. Hallyn <serue@us.ibm.com>
---
 fs/namespace.c | 20 ++++++++++++--------
 fs/pnode.c | 5 ++---
 2 files changed, 15 insertions(+), 10 deletions(-)

```
diff --git a/fs/namespace.c b/fs/namespace.c
index 9f98a67..f85dd73 100644
--- a/fs/namespace.c
+++ b/fs/namespace.c
@@ -709,8 +709,9 @@ struct vfsmount *copy_tree(struct vfsmou
   return NULL;

  res = q = clone_mnt(mnt, dentry, flag);
- if (!q)
-  goto Enomem;
+ if (!q || IS_ERR(q)) {
+  return q;
+ }
  q->mnt_mountpoint = mnt->mnt_mountpoint;

  p = mnt;
@@ -731,8 +732,9 @@ struct vfsmount *copy_tree(struct vfsmou
    nd.mnt = q;
    nd.dentry = p->mnt_mountpoint;
    q = clone_mnt(p, p->mnt_root, flag);
-   if (!q)
-    goto Enomem;
+   if (!q || IS_ERR(q)) {
+    goto Error;
+   }
    spin_lock(&vfsmount_lock);
    list_add_tail(&q->mnt_list, &res->mnt_list);
    attach_mnt(q, &nd);
```

```
@@ -740,7 +742,7 @@ struct vfsmount *copy_tree(struct vfsmou
   }
  }
  return res;
-Enomem:
+Error:
  if (res) {
   LIST_HEAD(umount_list);
   spin_lock(&vfsmount_lock);
@@ -748,7 +750,7 @@ Enomem:
   spin_unlock(&vfsmount_lock);
   release_mounts(&umount_list);
  }
- return NULL;
+ return q;
 }

 /*
@@ -928,8 +930,10 @@ static int do_loopback(struct nameidata
  else
   mnt = clone_mnt(old_nd.mnt, old_nd.dentry, 0);

- if (!mnt)
+ if (!mnt || IS_ERR(mnt)) {
+  err = mnt ? PTR_ERR(mnt) : -ENOMEM;
   goto out;
+ }

  err = graft_tree(mnt, nd);
  if (err) {
@@ -1466,7 +1470,7 @@ struct mnt_namespace *dup_mnt_ns(struct
  /* First pass: copy the tree topology */
  new_ns->root = copy_tree(mnt_ns->root, mnt_ns->root->mnt_root,
      CL_COPY_ALL | CL_EXPIRE);
- if (!new_ns->root) {
+ if (!new_ns->root || IS_ERR(new_ns->root)) {
   up_write(&namespace_sem);
   kfree(new_ns);
   return NULL;
diff --git a/fs/pnode.c b/fs/pnode.c
index 56aacea..1821c95 100644
--- a/fs/pnode.c
+++ b/fs/pnode.c
@@ -187,8 +187,9 @@ int propagate_mnt(struct vfsmount *dest_

   source =  get_source(m, prev_dest_mnt, prev_src_mnt, &type);

-  if (!(child = copy_tree(source, source->mnt_root, type))) {
```

```
-   ret = -ENOMEM;
+   child = copy_tree(source, source->mnt_root, type);
+   if (!child || IS_ERR(child)) {
+    ret = child ? PTR_ERR(child) : -ENOMEM;
    list_splice(tree_list, tmp_list.prev);
    goto out;
   }
--
1.4.1
```

_____

Containers mailing list
Containers@lists.osdl.org
https://lists.osdl.org/mailman/listinfo/containers