
Subject: [PATCH 2/2] Propagate the long_clone_arg up to the
create_new_namespaces
Posted by [Pavel Emelianov](#) on Wed, 16 Jan 2008 13:00:57 GMT
[View Forum Message](#) <> [Reply to Message](#)

The first user for the long_clone_arg is the namespaces code,
so pull the extended argument up to the namespaces creation
function.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>
Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
Reviewed-by: Serge Hallyn <serue@us.ibm.com>

```
diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
index 0e66b57..ad93b19 100644
--- a/include/linux/nsproxy.h
+++ b/include/linux/nsproxy.h
@@ -62,12 +62,13 @@ static inline struct nsproxy *task_nsproxy(struct task_struct *tsk)
     return rcu_dereference(tsk->nsproxy);
 }
```

```
-int copy_namespaces(unsigned long flags, struct task_struct *tsk);
+int copy_namespaces(unsigned long flags, struct task_struct *tsk,
+ struct long_clone_arg *carg);
void exit_task_namespaces(struct task_struct *tsk);
void switch_task_namespaces(struct task_struct *tsk, struct nsproxy *new);
void free_nsproxy(struct nsproxy *ns);
int unshare_nsproxy_namespaces(unsigned long, struct nsproxy **,
- struct fs_struct *);
+ struct fs_struct *, struct long_clone_arg *carg);
```

```
static inline void put_nsproxy(struct nsproxy *ns)
{
```

```
diff --git a/kernel/fork.c b/kernel/fork.c
index 19873c7..5e85567 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1160,7 +1199,7 @@ static struct task_struct *copy_process(unsigned long clone_flags,
     goto bad_fork_cleanup_signal;
     if ((retval = copy_keys(clone_flags, p)))
         goto bad_fork_cleanup_mm;
- if ((retval = copy_namespaces(clone_flags, p)))
+ if ((retval = copy_namespaces(clone_flags, p, carg)))
     goto bad_fork_cleanup_keys;
     retval = copy_thread(0, clone_flags, stack_start, stack_size, p, regs);
     if (retval)
```

```

@@ -1705,7 +1753,7 @@ asmlinkage long sys_unshare(unsigned long unshare_flags)
    if ((err = unshare_semundo(unshare_flags, &new_ulist)))
        goto bad_unshare_cleanup_fd;
    if ((err = unshare_nsproxy_namespaces(unshare_flags, &new_nsproxy,
-   new_fs)))
+   new_fs, carg)))
        goto bad_unshare_cleanup_semundo;

    if (new_fs || new_mm || new_fd || new_ulist || new_nsproxy) {
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index f5d332c..5fd4a03 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -48,7 +48,8 @@ static inline struct nsproxy *clone_nsproxy(struct nsproxy *orig)
    * leave it to the caller to do proper locking and attach it to task.
    */
    static struct nsproxy *create_new_namespaces(unsigned long flags,
-   struct task_struct *tsk, struct fs_struct *new_fs)
+   struct task_struct *tsk, struct fs_struct *new_fs,
+   struct long_clone_arg *carg)
    {
        struct nsproxy *new_nsp;
        int err;
@@ -119,7 +120,8 @@ out_ns:
    * called from clone. This now handles copy for nsproxy and all
    * namespaces therein.
    */
-int copy_namespaces(unsigned long flags, struct task_struct *tsk)
+int copy_namespaces(unsigned long flags, struct task_struct *tsk,
+   struct long_clone_arg *carg)
    {
        struct nsproxy *old_ns = tsk->nsproxy;
        struct nsproxy *new_ns;
@@ -131,7 +133,8 @@ int copy_namespaces(unsigned long flags, struct task_struct *tsk)
    get_nsproxy(old_ns);

    if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC |
-   CLONE_NEWUSER | CLONE_NEWPID | CLONE_NEWNET)))
+   CLONE_NEWUSER | CLONE_NEWPID | CLONE_NEWNET |
+   CLONE_LONGARG)))
        return 0;

    if (!capable(CAP_SYS_ADMIN)) {
@@ -139,7 +142,7 @@ int copy_namespaces(unsigned long flags, struct task_struct *tsk)
    goto out;
    }

-   new_ns = create_new_namespaces(flags, tsk, tsk->fs);

```

```

+ new_ns = create_new_namespaces(flags, tsk, tsk->fs, carg);
if (IS_ERR(new_ns)) {
    err = PTR_ERR(new_ns);
    goto out;
@@ -179,19 +182,20 @@ void free_nsproxy(struct nsproxy *ns)
 * On success, returns the new nsproxy.
 */
int unshare_nsproxy_namespaces(unsigned long unshare_flags,
- struct nsproxy **new_nsp, struct fs_struct *new_fs)
+ struct nsproxy **new_nsp, struct fs_struct *new_fs,
+ struct long_clone_arg *carg)
{
    int err = 0;

    if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC |
-     CLONE_NEWUSER | CLONE_NEWNET)))
+     CLONE_NEWUSER | CLONE_NEWNET | CLONE_LONGARG)))
        return 0;

    if (!capable(CAP_SYS_ADMIN))
        return -EPERM;

    *new_nsp = create_new_namespaces(unshare_flags, current,
-     new_fs ? new_fs : current->fs);
+     new_fs ? new_fs : current->fs, carg);
    if (IS_ERR(*new_nsp)) {
        err = PTR_ERR(*new_nsp);
        goto out;

```

Containers mailing list
Containers@lists.linux-foundation.org
<https://lists.linux-foundation.org/mailman/listinfo/containers>
