

---

Subject: [RFC PATCH 2/4] namespace containers: move nsproxy setting code  
Posted by [serue](#) on Mon, 12 Feb 2007 22:22:45 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: "Serge E. Hallyn" <[serue@us.ibm.com](mailto:serue@us.ibm.com)>  
Subject: [RFC PATCH 2/4] namespace containers: move nsproxy setting code

Move nsproxy setting code from clone and unshare into container\_clone.  
Containers will need to do this for namespace entering functionality, so  
go ahead and move all setting of tsk->nsproxy there for simplicity/  
consistency.

The clone path (at kernel/nsproxy.c:copy\_namespaces()) should be  
cleaned up:

1. The kfree(new\_ns) on error at bottom may not be safe,  
if the nscont->nsproxy has already been set to it.  
However if it has been set, then container\_clone() should  
have succeeded, so this *\*should\** not be possible.
2. This path is taking a few extra copies - it sets the  
tsk->nsproxy to the new nsproxy early, then the  
swap\_nsproxies() function copies it again. This should  
be cleaned up, but at least it is currently correct.

Best thing would be to create a common helper for the unshare  
and clone cases.

Signed-off-by: Serge E. Hallyn <[serue@us.ibm.com](mailto:serue@us.ibm.com)>

---

```
include/linux/container.h | 3 +-  
include/linux/nsproxy.h | 20 ++++++++  
kernel/container.c | 6 +++++  
kernel/fork.c | 7 +-----  
kernel/ns_container.c | 5 +++--  
kernel/nsproxy.c | 2 +-  
6 files changed, 30 insertions(+), 13 deletions(-)
```

```
16e2718ad6e2764b4abc61513b5202ceff5413f3  
diff --git a/include/linux/container.h b/include/linux/container.h  
index db2fc27..ef2ec57 100644  
--- a/include/linux/container.h  
+++ b/include/linux/container.h  
@@ -223,7 +223,8 @@ struct container_subsys {  
};
```

```

int container_register_subsys(struct container_subsys *subsys);
-int container_clone(struct task_struct *tsk, struct container_subsys *ss);
+int container_clone(struct task_struct *tsk, struct container_subsys *ss,
+    struct nsproxy *nsproxy);

static inline struct container_subsys_state *container_subsys_state(
    struct container *cont,
diff --git a/include/linux/nsproxy.h b/include/linux/nsproxy.h
index 0255e27..d11eb09 100644
--- a/include/linux/nsproxy.h
+++ b/include/linux/nsproxy.h
@@ -58,10 +58,26 @@ static inline void exit_task_namespaces(
    put_nsproxy(ns);
}
}
+
+static inline void swap_nsproxies(struct task_struct *tsk, struct nsproxy *nsproxy)
+{
+ struct nsproxy *oldnsp;
+
+ task_lock(tsk);
+ oldnsp = tsk->nsproxy;
+ tsk->nsproxy = nsproxy;
+ get_nsproxy(nsproxy);
+ task_unlock(tsk);
+ put_nsproxy(oldnsp);
+}
+
+
#ifdef CONFIG_CONTAINER_NS
-int ns_container_clone(struct task_struct *tsk);
+int ns_container_clone(struct task_struct *tsk, struct nsproxy *nsproxy);
#else
-static inline int ns_container_clone(struct task_struct *tsk) { return 0; }
+static inline int ns_container_clone(struct task_struct *tsk, struct nsproxy *nsproxy) {
+ swap_nsproxies(tsk, nsproxy);
+ return 0;
+}
#endif

#endif
diff --git a/kernel/container.c b/kernel/container.c
index 0606753..fc559ce 100644
--- a/kernel/container.c
+++ b/kernel/container.c
@@ -55,6 +55,7 @@
#include <linux/time.h>
#include <linux/backing-dev.h>
#include <linux/sort.h>

```

```

+#include <linux/nsproxy.h>

#include <asm/uaccess.h>
#include <asm/atomic.h>
@@ -1584,7 +1585,8 @@ static void get_unused_name(char *buf) {
    * container_clone - duplicate the current container and move this
    * task into the new child
    */
-int container_clone(struct task_struct *tsk, struct container_subsys *subsys)
+int container_clone(struct task_struct *tsk, struct container_subsys *subsys,
+    struct nsproxy *nsproxy)
{
    struct dentry *dentry;
    int ret = 0;
@@ -1593,6 +1595,8 @@ int container_clone(struct task_struct *
    struct inode *inode;
    int h;

+ swap_nsproxies(tsk, nsproxy);
+
    /* We shouldn't be called by an unregistered subsystem */
    BUG_ON(subsys->subsys_id < 0);

diff --git a/kernel/fork.c b/kernel/fork.c
index b1a3d6c..4ebdd53 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1663,7 +1663,7 @@ asmlinkage long sys_unshare(unsigned lon
    err = -ENOMEM;
    goto bad_unshare_cleanup_ipc;
}
- err = ns_container_clone(current);
+ err = ns_container_clone(current, new_nsproxy);
    if (err)
        goto bad_unshare_cleanup_dupns;
}
@@ -1673,11 +1673,6 @@ asmlinkage long sys_unshare(unsigned lon

    task_lock(current);

- if (new_nsproxy) {
-     current->nsproxy = new_nsproxy;
-     new_nsproxy = old_nsproxy;
- }
-
    if (new_fs) {
        fs = current->fs;
        current->fs = new_fs;

```

```

diff --git a/kernel/ns_container.c b/kernel/ns_container.c
index c90485d..d60d4f5 100644
--- a/kernel/ns_container.c
+++ b/kernel/ns_container.c
@@ -7,6 +7,7 @@
#include <linux/module.h>
#include <linux/container.h>
#include <linux/fs.h>
+#include <linux/nsproxy.h>

struct nscont {
    struct container_subsys_state css;
@@ -21,9 +22,9 @@ static inline struct nscont *container_n
    struct nscont, css);
}

-int ns_container_clone(struct task_struct *tsk)
+int ns_container_clone(struct task_struct *tsk, struct nsproxy *nsproxy)
{
- return container_clone(tsk, &ns_subsys);
+ return container_clone(tsk, &ns_subsys, nsproxy);
}

/*
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 1123ab2..6312ef8 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -111,7 +111,7 @@ int copy_namespaces(int flags, struct ta
    if (err)
        goto out_pid;

- err = ns_container_clone(tsk);
+ err = ns_container_clone(tsk, new_ns);
    if (err)
        goto out_container;
out:
--
1.1.6

```

---