
Subject: Re: [RFC PATCH 3/4] namespace containers: add nsproxy to nscont struct
Posted by [serue](#) on Tue, 13 Feb 2007 16:34:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

Quoting Paul Menage (menage@google.com):

```
> On 2/12/07, Serge E. Hallyn <serue@us.ibm.com> wrote:
> >--- a/kernel/container.c
> >+++ b/kernel/container.c
> >@@ -1581,6 +1581,14 @@ static void get_unused_name(char *buf) {
> >     sprintf(buf, "node%d", atomic_inc_return(&namecnt));
> > }
> >
> >+/* XXX need to create include/linux/ns_container.h and move this there */
> >+#ifdef CONFIG_CONTAINER_NS
> >+void ns_set_nsproxy_from_task(struct container *cont, struct task_struct
> >+*tsk);
> >+#else
> >+static inline void
> >+ns_set_nsproxy_from_task(struct container *cont, struct task_struct *tsk)
> >+{ }
> >+#endif
> >+
>
> Can't this just go in include/linux/nsproxy.h ?
```

I didn't want nsproxy.h knowing about containers if it didn't have to.
I'd rather have ns_container.h knowing about nsproxy and not the other way around.

```
> >
> >+ /* mark the auto-created container with the new namespace list */
> >+ ns_set_nsproxy_from_task(child, tsk);
> >+
>
> This should be in ns_container_clone() or some (possibly new)
> subsystem callback, rather than embedded in the generic containers
> code.
```

Would you complain if I change kernel/container.c:container_clone() to return the container *, so that ns_container_clone() doesn't need to deduce that from tsk? So something like the following (not even compiled, just to show what I'd be doing):

```
diff --git a/kernel/container.c b/kernel/container.c
index 4608a12..8895137 100644
--- a/kernel/container.c
+++ b/kernel/container.c
```

```

@@ -1593,7 +1593,7 @@ ns_set_nsproxy_from_task(struct container
 * 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,
+struct container *container_clone(struct task_struct *tsk, struct container_subsys *subsys,
    struct nsproxy *nsproxy)
{
    struct dentry *dentry;
diff --git a/kernel/ns_container.c b/kernel/ns_container.c
index 921c600..4291b18 100644
--- a/kernel/ns_container.c
+++ b/kernel/ns_container.c
@@ -23,9 +23,22 @@ static inline struct nscont *container_n
    struct nscont, css);
}

+static void ns_set_nsproxy_from_task(struct container *cont, struct task_struct *tsk)
+{
+ struct nscont *ns = container_nscont(cont);
+ ns->nsproxy = tsk->nsproxy;
+ get_nsproxy(ns->nsproxy);
+}
+
int ns_container_clone(struct task_struct *tsk, struct nsproxy *nsproxy)
{
- return container_clone(tsk, &ns_subsys, nsproxy);
+ struct container *c;
+
+ c = container_clone(tsk, &ns_subsys, nsproxy);
+ if (IS_ERR(c))
+ return ERR_PTR(c);
+ ns_set_nsproxy_from_task(c, tsk);
+ return 0;
}

/*
@@ -95,14 +108,6 @@ static void ns_create_nsproxy(struct nsc
    get_nsproxy(ns->nsproxy);
}

-/* called from container_clone */
-void ns_set_nsproxy_from_task(struct container *cont, struct task_struct *tsk)
-{
- struct nscont *ns = container_nscont(cont);
- ns->nsproxy = tsk->nsproxy;
- get_nsproxy(ns->nsproxy);
-}

```

```
-  
void ns_attach(struct container_subsys *ss, struct container *cont,  
               struct container *old_cont, struct task_struct *tsk)  
{  
  

```
