
Subject: Re: [PATCH] containers: define a namespace container subsystem
Posted by [serue](#) on Fri, 02 Feb 2007 23:35:09 GMT

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

Quoting Paul Menage (menage@google.com):

> On 1/30/07, Serge E. Hallyn <serue@us.ibm.com> wrote:

```
> >
> >It behaves pretty differently from other subsystems implemented
> >so far, which could either be seen as evidence that it doesn't
> >belong as a subsystem, or, more likely, that the container
> >subsystem approach is quite flexible.
>
> The latter - the container system is meant to be able to support more
> than just resource controllers.
>
> >+
> >+          char *name)
> >+{
> >+    struct dentry *d = container_get_dentry(parent->dentry, name);
> >+    return d ? d->d_fsd : NULL;
> >+}
> >+
> >+#define NS_CONT_MODE (S_IFDIR | S_IRUGO | S_IXUGO | S_IWUSR)
> >+int container_switch(struct task_struct *tsk)
> >+{
> >+    int h;
> >+    struct container *cur_cont, *new_cont;
> >+    char path[20];
> >+    struct qstr name;
> >+    struct dentry *dentry;
> >+    int ret;
> >+    char *pathbuf = NULL;
> >+    char buffer[20];
> >+
> >+    /* check if nsproxy subsys is registered */
> >+    if (ns_container_subsys_idx == -1)
> >+        return 0;
> >+
> >+    printk(KERN_NOTICE "%s: ns_container subsys registered\n",
> >+FUNCTION__);
> >+    /* check if nsproxy subsys is mounted in some hierarchy */
> >+    rcu_read_lock();
> >+    h = rcu_dereference(subsys[ns_container_subsys_idx]->hierarchy);
> >+    rcu_read_unlock();
> >+    if (h == 0) {
> >+        /* do we mount the nsproxy subsys, or just skip
> >+           * creating a container? I think we just skip
> >+           * it.
>
```

> I'd say that we should try to create a fresh hierarchy with just the
> nsproxy subsystem on it. Otherwise if someone tries to mount the
> nsproxy subsystem later, we end up with some namespaces with no
> container.

Yes, but if we automatically create a fresh hierarchy with just the nsproxy subsystem on it, then if you do any unsharing during boot or login, which with pam_namespace on LSPP systems is very possible, then you'll never be able to manually mount a hierarchy with an nsproxy subsystem and a resource controller in the same hierarchy.

Whereas having the topmost container contain a bunch of nsproxies really isn't a problem I don't think.

> This looks great - I'll incorporate it or something like it in my next
> patch set.

Great. Note though that I just found a little buglet - the following patch is needed for the !CONFIG_CONTAINERS case :)

thanks,
-serge

Subject: [PATCH] fix !config_containers compile

when container_switch was changed from returning void to int,
the !config_containers inline version was not updated.

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

include/linux/container.h | 2 +-
1 files changed, 1 insertions(+), 1 deletions(-)

```
6adc0fbab23e7f971750c3eb6c244d1008a636c0
diff --git a/include/linux/container.h b/include/linux/container.h
index 15b0446..a9075ac 100644
--- a/include/linux/container.h
+++ b/include/linux/container.h
@@ -222,7 +222,7 @@ static inline void container_exit(struct
```

```
static inline void container_lock(void) {}
static inline void container_unlock(void) {}
-static inline void container_switch(struct task_struct *tsk) { }
+static inline int container_switch(struct task_struct *tsk) { }

#endif /* !CONFIG_CONTAINERS */
```

--

1.1.6

Containers mailing list

Containers@lists.osdl.org

<https://lists.osdl.org/mailman/listinfo/containers>
