
Subject: [PATCH 2/4] namespace container: move nsproxy setting code

Posted by [serue](#) on Mon, 19 Feb 2007 22:16:00 GMT

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

From: Serge E. Hallyn <serue@us.ibm.com>

Subject: [PATCH 2/4] namespace container: 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.

Changelog:

Feb 14: move swap_nsproxies call into ns_container.c so as to leave nsproxy knowledge out of container.c

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

```
include/linux/nsproxy.h | 20 ++++++++
kernel/fork.c            |  7 +-----
kernel/ns_container.c    |  4 +++-
kernel/nsproxy.c         |  2 +-
4 files changed, 23 insertions(+), 10 deletions(-)
```

225efc9fea0771d283d22c393d948492162c84d4

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/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..23fac0e 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;
@@ -21,8 +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)
{
+ swap_nsproxies(tsk, nsproxy);
  return container_clone(tsk, &ns_subsys);
}

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

```
