
Subject: [PATCH 2/4] userns: let copy_ipcns handle setting ipc_ns->user_ns
Posted by [serge](#) on Thu, 24 Feb 2011 00:22:04 GMT

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

To do that, we have to pass in the task_struct of the task which will own the ipc_ns, so we can assign its user_ns.

Changelog:

Feb 23: As per Oleg comment, just pass in tsk. To get the ipc_ns from the nsproxy we need to include nsproxy.h

Signed-off-by: Serge E. Hallyn <serge.hallyn@canonical.com>

```
include/linux/ipc_namespace.h | 7 ++++---
ipc/namespace.c               | 13 ++++++++-----
kernel/nsproxy.c              | 7 +-----
3 files changed, 13 insertions(+), 14 deletions(-)
```

diff --git a/include/linux/ipc_namespace.h b/include/linux/ipc_namespace.h

index 46d2eb4..c079d09 100644

--- a/include/linux/ipc_namespace.h

+++ b/include/linux/ipc_namespace.h

@@ -5,6 +5,7 @@

#include <linux/idr.h>

#include <linux/rwsem.h>

#include <linux/notifier.h>

+#include <linux/nsproxy.h>

/*

* ipc namespace events

@@ -93,7 +94,7 @@ static inline int mq_init_ns(struct ipc_namespace *ns) { return 0; }

#if defined(CONFIG_IPC_NS)

extern struct ipc_namespace *copy_ipcns(unsigned long flags,

- struct ipc_namespace *ns);

+ struct task_struct *tsk);

static inline struct ipc_namespace *get_ipc_ns(struct ipc_namespace *ns)

{

if (ns)

@@ -104,12 +105,12 @@ static inline struct ipc_namespace *get_ipc_ns(struct ipc_namespace *ns)

extern void put_ipc_ns(struct ipc_namespace *ns);

#else

static inline struct ipc_namespace *copy_ipcns(unsigned long flags,

- struct ipc_namespace *ns)

+ struct task_struct *tsk)

{

if (flags & CLONE_NEWIPC)

```

    return ERR_PTR(-EINVAL);

- return ns;
+ return tsk->nsproxy->ipc_ns;
}

static inline struct ipc_namespace *get_ipc_ns(struct ipc_namespace *ns)
diff --git a/ipc/namespace.c b/ipc/namespace.c
index aa18899..3c3e522 100644
--- a/ipc/namespace.c
+++ b/ipc/namespace.c
@@ -15,7 +15,8 @@

#include "util.h"

-static struct ipc_namespace *create_ipc_ns(struct ipc_namespace *old_ns)
+static struct ipc_namespace *create_ipc_ns(struct task_struct *tsk,
+      struct ipc_namespace *old_ns)
{
    struct ipc_namespace *ns;
    int err;
@@ -44,17 +45,19 @@ static struct ipc_namespace *create_ipc_ns(struct ipc_namespace
*old_ns)
    ipcns_notify(IPCNS_CREATED);
    register_ipcns_notifier(ns);

- ns->user_ns = old_ns->user_ns;
- get_user_ns(ns->user_ns);
+ ns->user_ns = get_user_ns(task_cred_xxx(tsk, user)->user_ns);

    return ns;
}

-struct ipc_namespace *copy_ipcs(unsigned long flags, struct ipc_namespace *ns)
+struct ipc_namespace *copy_ipcs(unsigned long flags,
+      struct task_struct *tsk)
{
+ struct ipc_namespace *ns = tsk->nsproxy->ipc_ns;
+
    if (!(flags & CLONE_NEWIPC))
        return get_ipc_ns(ns);
- return create_ipc_ns(ns);
+ return create_ipc_ns(tsk, ns);
}

/*
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index ac8a56e..a05d191 100644

```

```

--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -75,16 +75,11 @@ static struct nsproxy *create_new_namespaces(unsigned long flags,
    goto out_uts;
}

- new_nsp->ipc_ns = copy_ipcs(flags, tsk->nsproxy->ipc_ns);
+ new_nsp->ipc_ns = copy_ipcs(flags, tsk);
    if (IS_ERR(new_nsp->ipc_ns)) {
        err = PTR_ERR(new_nsp->ipc_ns);
        goto out_ipc;
    }
- if (new_nsp->ipc_ns != tsk->nsproxy->ipc_ns) {
-     put_user_ns(new_nsp->ipc_ns->user_ns);
-     new_nsp->ipc_ns->user_ns = task_cred_xxx(tsk, user)->user_ns;
-     get_user_ns(new_nsp->ipc_ns->user_ns);
- }

    new_nsp->pid_ns = copy_pid_ns(flags, task_active_pid_ns(tsk));
    if (IS_ERR(new_nsp->pid_ns)) {
--
1.7.0.4

```

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