

---

Subject: [PATCH 7/9] add a user namespace owner of ipc ns

Posted by [serge](#) on Thu, 17 Feb 2011 15:03:49 GMT

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

---

Changelog:

Feb 15: Don't set new ipc->user\_ns if we didn't create a new ipc\_ns.

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

---

```
include/linux/ipc_namespace.h | 3 +++
ipc/msgutil.c                 | 3 +++
ipc/namespace.c               | 9 ++++++---
kernel/nsproxy.c              | 5 +++++
4 files changed, 18 insertions(+), 2 deletions(-)
```

diff --git a/include/linux/ipc\_namespace.h b/include/linux/ipc\_namespace.h

index 5195298..46d2eb4 100644

--- a/include/linux/ipc\_namespace.h

+++ b/include/linux/ipc\_namespace.h

```
@@ -24,6 +24,7 @@ struct ipc_ids {
    struct idr ipcs_idr;
};
```

```
+struct user_namespace;
```

```
struct ipc_namespace {
```

```
    atomic_t count;
```

```
    struct ipc_ids ids[3];
```

```
@@ -56,6 +57,8 @@ struct ipc_namespace {
```

```
    unsigned int mq_msg_max; /* initialized to DFLT_MSGMAX */
```

```
    unsigned int mq_msgsize_max; /* initialized to DFLT_MSGSIZEMAX */
```

```
+ /* user_ns which owns the ipc ns */
```

```
+ struct user_namespace *user_ns;
```

```
};
```

```
extern struct ipc_namespace init_ipc_ns;
```

diff --git a/ipc/msgutil.c b/ipc/msgutil.c

index f095ee2..d91ff4b 100644

--- a/ipc/msgutil.c

+++ b/ipc/msgutil.c

```
@@ -20,6 +20,8 @@
```

```
DEFINE_SPINLOCK(mq_lock);
```

```
+extern struct user_namespace init_user_ns;
```

```
+
```

```
/*
```

```

* The next 2 defines are here bc this is the only file
* compiled when either CONFIG_SYSVIPC and CONFIG_POSIX_MQUEUE
@@ -32,6 +34,7 @@ struct ipc_namespace init_ipc_ns = {
    .mq_msg_max      = DFLT_MSGMAX,
    .mq_msgsize_max  = DFLT_MSGSIZEMAX,
#endif
+ .user_ns = &init_user_ns,
};

atomic_t nr_ipc_ns = ATOMIC_INIT(1);
diff --git a/ipc/namespace.c b/ipc/namespace.c
index a1094ff..aa18899 100644
--- a/ipc/namespace.c
+++ b/ipc/namespace.c
@@ -11,10 +11,11 @@
#include <linux/slab.h>
#include <linux/fs.h>
#include <linux/mount.h>
+#include <linux/user_namespace.h>

#include "util.h"

-static struct ipc_namespace *create_ipc_ns(void)
+static struct ipc_namespace *create_ipc_ns(struct ipc_namespace *old_ns)
{
    struct ipc_namespace *ns;
    int err;
@@ -43,6 +44,9 @@ static struct ipc_namespace *create_ipc_ns(void)
    ipcns_notify(IPCNS_CREATED);
    register_ipcns_notifier(ns);

+ ns->user_ns = old_ns->user_ns;
+ get_user_ns(ns->user_ns);
+
    return ns;
}

@@ -50,7 +54,7 @@ struct ipc_namespace *copy_ipcs(unsigned long flags, struct
ipc_namespace *ns)
{
    if (!(flags & CLONE_NEWIPC))
        return get_ipc_ns(ns);
- return create_ipc_ns();
+ return create_ipc_ns(ns);
}

/*
@@ -105,6 +109,7 @@ static void free_ipc_ns(struct ipc_namespace *ns)

```

```

    * order to have a correct value when recomputing msgmni.
    */
    ipcns_notify(IPCNS_REMOVED);
+ put_user_ns(ns->user_ns);
}

/*
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 034dc2e..b6dbff2 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -85,6 +85,11 @@ static struct nsproxy *create_new_namespaces(unsigned long flags,
    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>

---