
Subject: [patch -mm 3/4] add pid_namespace to nsproxy
Posted by [Cedric Le Goater](#) on Fri, 08 Sep 2006 17:00:07 GMT
[View Forum Message](#) <> [Reply to Message](#)

Add a notion of pid namespace to nsproxy (and by extension, to task_struct).
Currently there is only one pid namespace, init_pid_ns and all tasks belong
to this pid namespace. When a new task is created, it inherits its parent's
pid namespace (in copy_process()).

This is based on Eric Biederman's patch: <http://lkml.org/lkml/2006/2/6/285>

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Signed-off-by: Cedric Le Goater <clg@fr.ibm.com>
Cc: Eric Biederman <ebiederm@xmission.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>
Cc: containers@lists.osdl.org

```
include/linux/init_task.h | 2 ++
include/linux/nsproxy.h   | 2 ++
include/linux/pid_namespace.h | 15 ++++++
kernel/nsproxy.c          | 12 ++++++
4 files changed, 31 insertions(+)
```

Index: 2.6.18-rc6-mm1/include/linux/init_task.h

```
=====
--- 2.6.18-rc6-mm1.orig/include/linux/init_task.h
+++ 2.6.18-rc6-mm1/include/linux/init_task.h
@@ -7,6 +7,7 @@
#include <linux/utsname.h>
#include <linux/lockdep.h>
#include <linux/ipc.h>
+#include <linux/pid_namespace.h>

#define INIT_FDTABLE \
{ \
@@ -72,6 +73,7 @@
```

```
extern struct nsproxy init_nsproxy;
#define INIT_NS_PROXY(nsproxy) { \
+ .pid_ns = &init_pid_ns, \
. count = ATOMIC_INIT(1), \
. nslock = SPIN_LOCK_UNLOCKED, \
. uts_ns = &init_uts_ns, \
Index: 2.6.18-rc6-mm1/include/linux/nsproxy.h
```

```
=====
--- 2.6.18-rc6-mm1.orig/include/linux/nsproxy.h
+++ 2.6.18-rc6-mm1/include/linux/nsproxy.h
```

```

@@ -7,6 +7,7 @@
struct mnt_namespace;
struct uts_namespace;
struct ipc_namespace;
+struct pid_namespace;

/*
 * A structure to contain pointers to all per-process
@@ -23,6 +24,7 @@ struct ipc_namespace;
struct nsproxy {
    atomic_t count;
    spinlock_t nslock;
+ struct pid_namespace *pid_ns;
    struct uts_namespace *uts_ns;
    struct ipc_namespace *ipc_ns;
    struct mnt_namespace *mnt_ns;
Index: 2.6.18-rc6-mm1/include/linux/pid_namespace.h
=====
--- 2.6.18-rc6-mm1.orig/include/linux/pid_namespace.h
+++ 2.6.18-rc6-mm1/include/linux/pid_namespace.h
@@ -5,6 +5,7 @@
#include <linux/mm.h>
#include <linux/threads.h>
#include <linux/pid.h>
+#include <linux/nsproxy.h>

struct pidmap {
    atomic_t nr_free;
@@ -20,4 +21,18 @@ struct pid_namespace {

extern struct pid_namespace init_pid_ns;

+static inline void get_pid_ns(struct pid_namespace *ns)
+{
+}
+
+static inline int copy_pid_ns(int flags, struct task_struct *tsk)
+{
+ tsk->nsproxy->pid_ns = &init_pid_ns;
+ return 0;
+}
+
+static inline void put_pid_ns(struct pid_namespace *ns)
+{
+}
+
#endif /* _LINUX_PID_NS_H */
Index: 2.6.18-rc6-mm1/kernel/nsproxy.c

```

```

=====
--- 2.6.18-rc6-mm1.orig/kernel/nsproxy.c
+++ 2.6.18-rc6-mm1/kernel/nsproxy.c
@@ -19,6 +19,7 @@
#include <linux/init_task.h>
#include <linux/mnt_namespace.h>
#include <linux/utsname.h>
+#include <linux/pid_namespace.h>

struct nsproxy init_nsproxy = INIT_NSPROXY(init_nsproxy);

@@ -68,6 +69,8 @@ struct nsproxy *dup_namespaces(struct ns
    get_uts_ns(ns->uts_ns);
    if (ns->ipc_ns)
        get_ipc_ns(ns->ipc_ns);
+   if (ns->pid_ns)
+       get_pid_ns(ns->pid_ns);
}

    return ns;
@@ -111,10 +114,17 @@ int copy_namespaces(int flags, struct ta
    if (err)
        goto out_ipc;

+   err = copy_pid_ns(flags, tsk);
+   if (err)
+       goto out_pid;
+
out:
    put_nsproxy(old_ns);
    return err;

+out_pid:
+   if (new_ns->ipc_ns)
+       put_ipc_ns(new_ns->ipc_ns);
out_ipc:
    if (new_ns->uts_ns)
        put_uts_ns(new_ns->uts_ns);
@@ -135,5 +145,7 @@ void free_nsproxy(struct nsproxy *ns)
    put_uts_ns(ns->uts_ns);
    if (ns->ipc_ns)
        put_ipc_ns(ns->ipc_ns);
+   if (ns->pid_ns)
+       put_pid_ns(ns->pid_ns);
    kfree(ns);
}

--

```

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
