
Subject: [PATCH 3/13] Introdution of config option and clone flag
Posted by [Pavel Emelianov](#) on Thu, 24 May 2007 12:42:04 GMT
[View Forum Message](#) <> [Reply to Message](#)

The config option is CONFIG_PID_NS. The flag is CLONE_NEWPIDS.

As I have already said - cloning of pid namespace from fork()
is not allowed - use unshare for this.

Signed-off-by: Pavel Emelianov <xemul@openvz.org>

```
diff --git a/init/Kconfig b/init/Kconfig
index 2a46e35..59e4625 100644
--- a/init/Kconfig
+++ b/init/Kconfig
@@ -127,6 +127,16 @@ config SWAP_PREFETCH
    Workstations and multiuser workstation servers will most likely want
    to say Y.
```

```
+config PID_NS
+ bool "Pid namespaces"
+ default n
+ help
+  Enable pid namespaces support. When on task is allowed to unshare
+  its pid namespace from parent and become its init. After this task
+  all its children will see only the tasks from this namespace.
+  However tasks from parent namespace see all the tasks in the system.
+  Only one level of nesting is allowed. Tasks cannot leave the namespace.
+
```

```
config SYSVIPC
    bool "System V IPC"
    ---help---
```

```
diff --git a/include/linux/sched.h b/include/linux/sched.h
index d4de6d8..7743a11 100644
--- a/include/linux/sched.h
+++ b/include/linux/sched.h
@@ -26,6 +26,7 @@
#define CLONE_STOPPED 0x02000000 /* Start in stopped state */
#define CLONE_NEWUTS 0x04000000 /* New utsname group? */
#define CLONE_NEWIPC 0x08000000 /* New ipcns */
+#define CLONE_NEWPIDS 0x10000000 /* New pids */
```

```
/*
 * Scheduling policies
diff --git a/kernel/fork.c b/kernel/fork.c
index d7207a1..3ab517c 100644
```

```

--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1606,7 +1612,7 @@ asmlinkage long sys_unshare(unsigned lon
    err = -EINVAL;
    if (unshare_flags & ~(CLONE_THREAD|CLONE_FS|CLONE_NEWNS|CLONE_SIGHAND|
        CLONE_VM|CLONE_FILES|CLONE_SYSVSEM|
-   CLONE_NEWUTS|CLONE_NEWIPC))
+   CLONE_NEWUTS|CLONE_NEWIPC|CLONE_NEWPIDS))
        goto bad_unshare_out;

    if ((err = unshare_thread(unshare_flags)))
diff --git a/kernel/nsproxy.c b/kernel/nsproxy.c
index 1bc4b55..9bcc047 100644
--- a/kernel/nsproxy.c
+++ b/kernel/nsproxy.c
@@ -110,6 +110,9 @@ int copy_namespaces(int flags, struct ta

    get_nsproxy(old_ns);

+ if (flags & CLONE_NEWPIDS)
+ return -EINVAL;
+
    if (!(flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))
        return 0;

@@ -154,7 +157,8 @@ int unshare_nsproxy_namespaces(unsigned
    struct nsproxy *old_ns = current->nsproxy;
    int err = 0;

- if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS | CLONE_NEWIPC)))
+ if (!(unshare_flags & (CLONE_NEWNS | CLONE_NEWUTS |
+   CLONE_NEWIPC | CLONE_NEWPIDS)))
        return 0;

#ifdef CONFIG_IPC_NS
@@ -166,6 +170,10 @@ int unshare_nsproxy_namespaces(unsigned
    if (unshare_flags & CLONE_NEWUTS)
        return -EINVAL;
#endif
+
+#ifndef CONFIG_PID_NS
+ if (unshare_flags & CLONE_NEWPIDS)
+ return -EINVAL;
+#endif

    if (!capable(CAP_SYS_ADMIN))
        return -EPERM;

```
