

---

Subject: [PATCH 3/3] Masquerade sender and limit system-wide signals  
Posted by [Sukadev Bhattiprolu](#) on Thu, 30 Aug 2007 06:22:10 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>  
Subject: [RFC][PATCH] Masquerade sender if in ancestor ns

If sender of a signal does not have a pid\_t in the active pid namespace of the caller (eg if sender is in an ancestor namespace), pretend that the signal originated from the kernel.

Also, system-wide signals (kill -s sig -1) should only apply to the active namespace of the sender and its descendant namespaces. Specifically it should not affect processes in ancestor or unrelated namespaces.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>

---

kernel/signal.c | 21 +++++  
1 file changed, 21 insertions(+)

Index: 2.6.23-rc3-mm1/kernel/signal.c

=====

--- 2.6.23-rc3-mm1.orig/kernel/signal.c 2007-08-29 17:38:01.000000000 -0700

+++ 2.6.23-rc3-mm1/kernel/signal.c 2007-08-29 17:38:01.000000000 -0700

@@ -691,6 +691,19 @@ static void handle\_stop\_signal(int sig,  
{  
}

+static void masquerade\_sender(struct task\_struct \*t, struct sigqueue \*q)

+{

+ /\*

+ \* If the sender does not have a pid\_t in the receiver's active

+ \* pid namespace, set si\_pid to 0 and pretend signal originated

+ \* from the kernel.

+ \*/

+ if (!pid\_ns\_equal(t)) {

+ q->info.si\_pid = 0;

+ q->info.si\_code = SI\_KERNEL;

+ }

+}

+

static int send\_signal(int sig, struct siginfo \*info, struct task\_struct \*t,  
struct sigpending \*signals)

{

@@ -742,6 +755,7 @@ static int send\_signal(int sig, struct s  
copy\_siginfo(&q->info, info);

break;

```

    }
+ masquerade_sender(t, q);
    } else if (!is_si_special(info)) {
        if (sig >= SIGRTMIN && info->si_code != SI_USER)
            /*
@@ -1175,6 +1189,7 @@ EXPORT_SYMBOL_GPL(kill_pid_info_as_uid);
static int kill_something_info(int sig, struct siginfo *info, int pid)
{
    int ret;
+
    rcu_read_lock();
    if (!pid) {
        ret = kill_pgrp_info(sig, info, task_pgrp(current));
@@ -1184,6 +1199,12 @@ static int kill_something_info(int sig,

    read_lock(&tasklist_lock);
    for_each_process(p) {
+ /*
+  * System-wide signals only apply to pid namespace
+  * of sender.
+  */
+ if (!task_descendant_pid_ns(p, current))
+ continue;
        if (p->pid > 1 && !same_thread_group(p, current)) {
            int err = group_send_sig_info(sig, info, p);
            ++count;

```

---

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

---