Subject: [PATCH 9/28] [PREP 9/14] Masquerade the siginfo when sending a pid to a foreign namespace
Posted by Pavel Emelianov on Fri, 15 Jun 2007 16:07:23 GMT
View Forum Message <> Reply to Message

When user send signal from (say) init namespace to any task in a sub namespace the siginfo struct must not carry the sender's pid value, as this value may refer to some task in the destination namespace and thus may confuse the application.

The consensus was to pretend in this case as if it is the kernel who sends the signal.

The pid_ns_accessible() call is introduced to check this pid-to-ns accessibility.

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

```
 include/linux/pid.h |   10 ++++++++++
 kernel/signal.c     |   18 +++++++++++++++++++
 2 files changed, 28 insertions(+)

--- ./include/linux/pid.h.sigmasqerade 2007-06-15 15:00:59.000000000 +0400
+++ ./include/linux/pid.h 2007-06-15 15:08:39.000000000 +0400
@@ -141,6 +141,16 @@ static inline pid_t pid_nr_ns(struct pid
 {
  return pid_nr(pid);
 }
+
+/*
+ * checks whether the pid actually lives in the namespace ns, i.e. it was
+ * created in this namespace or it was moved there.
+ */
+
+static inline int pid_ns_accessible(struct pid_namespace *ns, struct pid *pid)
+{
+ return 1;
+}
 #else
 #endif

--- ./kernel/signal.c.sigmasqerade 2007-06-15 15:02:29.000000000 +0400
+++ ./kernel/signal.c 2007-06-15 15:06:04.000000000 +0400
@@ -1124,6 +1124,22 @@ EXPORT_SYMBOL_GPL(kill_pid_info_as_uid);
  * is probably wrong.  Should make it like BSD or SYSV.
  */
```

```
+static inline void masquerade_siginfo(struct pid_namespace *src_ns,
+  struct pid *tgt_pid, struct siginfo *info)
+{
+ if (tgt_pid != NULL && !pid_ns_accessible(src_ns, tgt_pid)) {
+  /*
+   * current namespace is not seen from the taks we
+   * want to send the signal to, so pretend as if it
+   * is the kernel who does this to avoid pid messing
+   * by the target
+   */
+
+  info->si_pid = 0;
+  info->si_code = SI_KERNEL;
+ }
+}
+
 static int kill_something_info(int sig, struct siginfo *info, int pid_nr)
 {
  int ret;
@@ -1149,9 +1165,11 @@ static int kill_something_info(int sig,
   ret = count ? retval : -ESRCH;
  } else if (pid_nr < 0) {
   pid = find_vpid(-pid_nr);
+  masquerade_siginfo(current->nsproxy->pid_ns, pid, info);
   ret = kill_pgrp_info(sig, info, pid);
  } else {
   pid = find_vpid(pid_nr);
+  masquerade_siginfo(current->nsproxy->pid_ns, pid, info);
   ret = kill_pid_info(sig, info, pid);
  }
  rcu_read_unlock();
```

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