Subject: [RFC][PATCH 2/5] Define helper functions
Posted by Sukadev Bhattiprolu on Sat, 10 Mar 2007 04:02:07 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [RFC][PATCH 2/5] Define helper functions

Define some helper functions that will be used in subsequent patches
to move pid namespace out of nsproxy and into pid->pid_nrs list.

Changelog:
 - Remove set_pid_ns() as its no longer needed.
 - Remove set_pid_ns_child_reaper() from use-pid-ns-from-struct-pid-nr
   patch and moved it into this patch
 - Add priv_check_pid_ns()

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Cc: Cedric Le Goater <clg@fr.ibm.com>
Cc: Dave Hansen <haveblue@us.ibm.com>
Cc: Serge Hallyn <serue@us.ibm.com>
Cc: containers@lists.osdl.org
---
 kernel/fork.c | 29 ++++++++++++++++++++++++++++
 kernel/pid.c | 30 ++++++++++++++++++++++++++++++
 2 files changed, 59 insertions(+)

Index: lx26-20-mm2b/kernel/pid.c
===
--- lx26-20-mm2b.orig/kernel/pid.c 2007-03-09 19:00:12.000000000 -0800
+++ lx26-20-mm2b/kernel/pid.c 2007-03-09 19:00:42.000000000 -0800
@@ -327,6 +327,36 @@ static struct pid_namespace *clone_pid_n
  return ns;
 }

+/*
+ * Return the clone-pid-namespace of this pid.
+ *
+ * Clone-pid-namespace of a process is the namespace that the process
+ * did the most recent unshare or clone of CLONE_NEWPID into. If the
+ * process never unshared its pid namespace, then its clone-pid-namespace
+ * is the same as its parent's clone-pid-namespace.
+ */
+struct pid_namespace *pid_ns(struct pid *pid)
+{
+ struct pid_nr* pid_nr;
+ struct pid_namespace *ns;
+ struct hlist_node *head;
+

```
+ if (!pid)
+  return 0;
+
+ rcu_read_lock();
+ /*
+  * Our clone-pid-ns is simply the pid_ns of the first pid_nr
+  * on our pid_nrs list
+  */
+ head = pid->pid_nrs.first;
+ pid_nr = hlist_entry(head, struct pid_nr, node);
+ ns = pid_nr->pid_ns;
+
+ rcu_read_unlock();
+ return ns;
+}
+
 struct pid *alloc_pid(void)
 {
  struct pid *pid;
Index: lx26-20-mm2b/kernel/fork.c
===================================================================
--- lx26-20-mm2b.orig/kernel/fork.c 2007-03-09 19:00:14.000000000 -0800
+++ lx26-20-mm2b/kernel/fork.c 2007-03-09 19:00:42.000000000 -0800
@@ -953,6 +953,35 @@ static inline void rcu_task_init(struct
 static inline void rcu_task_init(struct task_struct *p) {}
 #endif

+static inline int priv_check_pid_ns(unsigned long clone_flags)
+{
+ if (clone_flags & CLONE_NEWPID)
+  if (!capable(CAP_SYS_ADMIN))
+   return -EPERM;
+ return 0;
+}
+
+/*
+ * Make @tsk the child reaper for the clone-pid-ns of the process
+ * identified by @pid
+ */
+static void set_pid_ns_child_reaper(unsigned long clone_flags, struct pid *pid,
+  struct task_struct *tsk)
+{
+ struct pid_namespace *lpid_ns;
+
+ if (!(clone_flags & CLONE_NEWPID))
+  return;
+
+ lpid_ns = pid_ns(pid);
```

```
+ BUG_ON(lpid_ns == &init_pid_ns);
+
+ /* don't need to lock here since we just created the pid ns */
+ lpid_ns->child_reaper = tsk;
+
+ return;
+}
+
 /*
  * This creates a new process as a copy of the old one,
  * but does not actually start it yet.
```

_____

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