From: Eric W. Biederman <ebiederm@xmission.com>

It turns out that the existing assignment in copy_process of the child_reaper can handle the initial assignment of child_reaper we just need to generalize the test in kernel/fork.c

Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
Signed-off-by: Daniel Lezcano <daniel.lezcano@free.fr>

---
#include/linux/pid.h | 11 ++++++++++
init/main.c | 9 ----------
kernel/fork.c | 2 +-  
3 files changed, 12 insertions(+), 10 deletions(-)
diff --git a/include/linux/pid.h b/include/linux/pid.h
index 49f1c2f..efceda0 100644
--- a/include/linux/pid.h
+++ b/include/linux/pid.h
@@ -141,6 +141,17 @@ static inline struct pid_namespace *ns_of_pid(struct pid *pid)
}

/*
+ * is_child_reaper returns true if the pid is the init process
+ * of the current namespace. As this one could be checked before
+ * pid_ns->child_reaper is assigned in copy_process, we check
+ * with the pid number.
+ */
+static inline bool is_child_reaper(struct pid *pid)
+{
+return pid->numbers[pid->level].nr == 1;
+}
+/*
+ * the helpers to get the pid's id seen from different namespaces
+ *
+ * pid_nr() : global id, i.e. the id seen from the init namespace;
diff --git a/init/main.c b/init/main.c
index 33c37c3..793ebfd 100644
--- a/init/main.c
+++ b/init/main.c
@@ -875,15 +875,6 @@ static int __init kernel_init(void * unused)
* init can run on any cpu.
 */
set_cpus_allowed_ptr(current, cpu_all_mask);
/*
 * Tell the world that we're going to be the grim reaper of innocent orphaned children.
 * We don't want people to have to make incorrect assumptions about where in the task array this can be found.
 */
-init_pid_ns.child_reaper = current;

cad_pid = task_pid(current);

diff --git a/kernel/fork.c b/kernel/fork.c
index 25e4291..c9f0784 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1289,7 +1289,7 @@ static struct task_struct *copy_process(unsigned long clone_flags,
 tracehook_finish_clone(p, clone_flags, trace);

 if (thread_group_leader(p)) {
-if (clone_flags & CLONE_NEWPID)
+if (is_child_reaper(pid))
    p->nsproxy->pid_ns->child_reaper = p;

    p->signal->leader_pid = pid;
--
1.7.1

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

Subject: Re: [PATCH 1/3] pid: Remove the child_reaper special case in init/main.c
Posted by Serge E. Hallyn on Tue, 15 Feb 2011 18:37:07 GMT
View Forum Message <> Reply to Message

Quoting Daniel Lezcano (daniel.lezcano@free.fr):
> From: Eric W. Biederman <ebiederm@xmission.com>
> 
> It turns out that the existing assignment in copy_process of the child_reaper can handle the initial assignment of child_reaper we just need to generalize the test in kernel/fork.c
>
> Signed-off-by: Eric W. Biederman <ebiederm@xmission.com>
> Signed-off-by: Daniel Lezcano <daniel.lezcano@free.fr>
Acked-by: Serge E. Hallyn <serge@hallyn.com>

---
include/linux/pid.h |   11 +++++++++++  
init/main.c |   9 -------  
kernel/fork.c | 2 +-  
3 files changed, 12 insertions(+), 10 deletions(-)

diff --git a/include/linux/pid.h b/include/linux/pid.h
index 49f1c2f..efceda0 100644
--- a/include/linux/pid.h
+++ b/include/linux/pid.h
@@ -141,6 +141,17 @@ static inline struct pid_namespace *ns_of_pid(struct pid *pid)
 }

 /* is_child_reaper returns true if the pid is the init process
 + * of the current namespace. As this one could be checked before
 + * pid_ns->child_reaper is assigned in copy_process, we check
 + * with the pid number.
 + */
 +static inline bool is_child_reaper(struct pid *pid)
 +{
 +
 +
 + /* the helpers to get the pid's id seen from different namespaces
 + */
 +
 + pid_nr() : global id, i.e. the id seen from the init namespace;
 diff --git a/init/main.c b/init/main.c
index 33c37c3..793ebfd 100644
--- a/init/main.c
+++ b/init/main.c
@@ -875,15 +875,6 @@ static int __init kernel_init(void * unused)
 
 */
+ set_cpus_allowed_ptr(current, cpu_all_mask);
 */

 - * Tell the world that we're going to be the grim
 - * reaper of innocent orphaned children.
 - *
 - * We don't want people to have to make incorrect
 - * assumptions about where in the task array this
 - * can be found.
 - */

 -init_pid_ns.child_reaper = current;
 >
cad_pid = task_pid(current);
>
diff --git a/kernel/fork.c b/kernel/fork.c
index 25e4291..c9f0784 100644
--- a/kernel/fork.c
+++ b/kernel/fork.c
@@ -1289,7 +1289,7 @@ static struct task_struct *copy_process(unsigned long clone_flags,
 tracehook_finish_clone(p, clone_flags, trace);
>
 if (thread_group_leader(p)) {
- if (clone_flags & CLONE_NEWPID)
+ if (is_child_reaper(pid))
     p->nsproxy->pid_ns->child_reaper = p;
>
     p->signal->leader_pid = pid;
>
---
1.7.1
>
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

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

Page 4 of 4 ---- Generated from OpenVZ Forum