Subject: [PATCH] Rename is_cgroup_init()
Posted by Sukadev Bhattiprolu on Thu, 04 Oct 2007 16:15:59 GMT
View Forum Message <> Reply to Message

From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Subject: [PATCH] Rename is_cgroup_init()

is_container_init() was accidentally renamed to is_cgroup_init() when
renaming "container" to "control group". This patch restores the
original name.

Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
Acked-by: Serge Hallyn <serue@us.ibm.com>
---
 include/linux/sched.h | 4 ++--
 kernel/capability.c | 2 +-
 kernel/pid.c | 4 ++--
 3 files changed, 5 insertions(+), 5 deletions(-)

Index: 2.6.23-rc8-mm2/include/linux/sched.h
===
--- 2.6.23-rc8-mm2.orig/include/linux/sched.h 2007-10-02 22:05:18.000000000 -0700
+++ 2.6.23-rc8-mm2/include/linux/sched.h 2007-10-03 10:26:12.000000000 -0700
@@ -1303,10 +1303,10 @@ static inline int is_global_init(struct
 }

 /*
- * is_cgroup_init:
+ * is_container_init:
  * check whether in the task is init in its own pid namespace.
  */
-extern int is_cgroup_init(struct task_struct *tsk);
+extern int is_container_init(struct task_struct *tsk);

 extern struct pid *cad_pid;

Index: 2.6.23-rc8-mm2/kernel/capability.c
===
--- 2.6.23-rc8-mm2.orig/kernel/capability.c 2007-09-27 13:20:11.000000000 -0700
+++ 2.6.23-rc8-mm2/kernel/capability.c 2007-10-03 10:26:35.000000000 -0700
@@ -130,7 +130,7 @@ static inline int cap_set_all(kernel_cap
     int found = 0;

     do_each_thread(g, target) {
-        if (target == current || is_cgroup_init(target->group_leader))
+   if (target == current || is_container_init(target->group_leader))
             continue;
         found = 1;

```
      if (security_capset_check(target, effective, inheritable,
Index: 2.6.23-rc8-mm2/kernel/pid.c
===================================================================
--- 2.6.23-rc8-mm2.orig/kernel/pid.c 2007-10-03 10:24:12.000000000 -0700
+++ 2.6.23-rc8-mm2/kernel/pid.c 2007-10-03 10:24:45.000000000 -0700
@@ -81,7 +81,7 @@ struct pid_namespace init_pid_ns = {
 };
 EXPORT_SYMBOL_GPL(init_pid_ns);

-int is_cgroup_init(struct task_struct *tsk)
+int is_container_init(struct task_struct *tsk)
 {
  int ret = 0;
  struct pid *pid;
@@ -94,7 +94,7 @@ int is_cgroup_init(struct task_struct *t

  return ret;
 }
-EXPORT_SYMBOL(is_cgroup_init);
+EXPORT_SYMBOL(is_container_init);

 /*
  * Note: disable interrupts while the pidmap_lock is held as an
```


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

---

Subject: Re: [PATCH] Rename is_cgroup_init()
Posted by akpm on Fri, 05 Oct 2007 23:37:06 GMT
View Forum Message <> Reply to Message

On Thu, 4 Oct 2007 09:15:59 -0700
>
> From: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> Subject: [PATCH] Rename is_cgroup_init()
>
> is_container_init() was accidentally renamed to is_cgroup_init() when
> renaming "container" to "control group". This patch restores the
> original name.
>
> Signed-off-by: Sukadev Bhattiprolu <sukadev@us.ibm.com>
> Acked-by: Serge Hallyn <serue@us.ibm.com>
> ---
> include/linux/sched.h | 4 ++--
> kernel/capability.c | 2 +-
> kernel/pid.c | 4 ++--

```
>  3 files changed, 5 insertions(+), 5 deletions(-)
>
> Index: 2.6.23-rc8-mm2/include/linux/sched.h
> ===================================================================
> --- 2.6.23-rc8-mm2.orig/include/linux/sched.h 2007-10-02 22:05:18.000000000 -0700
> +++ 2.6.23-rc8-mm2/include/linux/sched.h 2007-10-03 10:26:12.000000000 -0700
> @@ -1303,10 +1303,10 @@ static inline int is_global_init(struct
>  }
>
>  /*
> - * is_cgroup_init:
> + * is_container_init:
>   * check whether in the task is init in its own pid namespace.
>   */
> -extern int is_cgroup_init(struct task_struct *tsk);
> +extern int is_container_init(struct task_struct *tsk);
>
>  extern struct pid *cad_pid;
>
> Index: 2.6.23-rc8-mm2/kernel/capability.c
> ===================================================================
> --- 2.6.23-rc8-mm2.orig/kernel/capability.c 2007-09-27 13:20:11.000000000 -0700
> +++ 2.6.23-rc8-mm2/kernel/capability.c 2007-10-03 10:26:35.000000000 -0700
> @@ -130,7 +130,7 @@ static inline int cap_set_all(kernel_cap
>       int found = 0;
>
>       do_each_thread(g, target) {
> -             if (target == current || is_cgroup_init(target->group_leader))
> +             if (target == current || is_container_init(target->group_leader))
>                       continue;
>               found = 1;
>       if (security_capset_check(target, effective, inheritable,
> Index: 2.6.23-rc8-mm2/kernel/pid.c
> ===================================================================
> --- 2.6.23-rc8-mm2.orig/kernel/pid.c 2007-10-03 10:24:12.000000000 -0700
> +++ 2.6.23-rc8-mm2/kernel/pid.c 2007-10-03 10:24:45.000000000 -0700
> @@ -81,7 +81,7 @@ struct pid_namespace init_pid_ns = {
>  };
>  EXPORT_SYMBOL_GPL(init_pid_ns);
>
> -int is_cgroup_init(struct task_struct *tsk)
> +int is_container_init(struct task_struct *tsk)
>  {
>   int ret = 0;
>   struct pid *pid;
> @@ -94,7 +94,7 @@ int is_cgroup_init(struct task_struct *t
>
>   return ret;
```

> }
> -EXPORT_SYMBOL(is_cgroup_init);
> +EXPORT_SYMBOL(is_container_init);
>
> /*
> * Note: disable interrupts while the pidmap_lock is held as an

This patch is splattered across approximately seven different patches - ugly.

I just edited the diffs and changelogs in place and replaced all
occurrences of is_cgroup_init with is_container_init

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