

---

Subject: Re: Screamm.. commit f400e198b2ed26ce55b22a1412ded0896e7516ac  
Posted by [ebiederm](#) on Thu, 29 Mar 2007 13:57:15 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

"Serge E. Hallyn" <serue@us.ibm.com> writes:

```
> Yup. Looks like ambiguous naming once again hid some real (future)
> bugs. This is of course safe so far in mainline, but needs to be split
> into
>
> static inline int is_global_init(struct task_struct *tsk)
> {
>     return (tsk == &init_task);
> }
>
> and
>
> static inline int
> is_container_init(struct task_struct *task, struct pid_namespace *ns)
> {
>     return (__pid_nr(task, ns) == 1);
> }
```

Conceptually yes. The implementation of `is_global_init` is just wrong.  
`&init_task` is the first processors idle thread.

`is_container_init` looks correct but I don't know if the `ns` parameter makes any sense.

> Where the latter is needed in, for instance, `kernel/capability.c`.

Yes.

I think more clear cut examples could be made. It isn't clear to me why we skip `pid == 1` in `kernel/capability.c`

I believe a good example is that inside a container you should not be able to send `pid == 1` a signal it doesn't have a handler for. While from outside the container we need that capability.

Eric

---

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

---