Subject: Screamm.. commit f400e198b2ed26ce55b22a1412ded0896e7516ac
Posted by ebiederm on Thu, 29 Mar 2007 03:25:48 GMT
View Forum Message <> Reply to Message

This is just to vent.

I was clearly not auditing patches well enough earlier and the above
patch got modified since the version I wrote initially. Adding a
few addition is_init calls where what we care about test is not
is the real init process of the system (so we should treat it with
care) but is this the pid == 1 in my pid namespace.

So now we (or maybe me) are going to have to through all of those
is_init references and undo the changes for the ones that don't
apply to the global init.

Guys, please, please be more careful.

The one that set me off is the one in will_become_orphaned_pgrp.

Eric

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

Subject: Re: Screamm.. commit f400e198b2ed26ce55b22a1412ded0896e7516ac
Posted by serue on Thu, 29 Mar 2007 13:13:02 GMT
View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):
>
> This is just to vent.
>
> I was clearly not auditing patches well enough earlier and the above
> patch got modified since the version I wrote initially. Adding a
> few addition is_init calls where what we care about test is not
> is the real init process of the system (so we should treat it with
> care) but is this the pid == 1 in my pid namespace.
>
> So now we (or maybe me) are going to have to through all of those
> is_init references and undo the changes for the ones that don't
> apply to the global init.
>
> Guys, please, please be more careful.
>
> The one that set me off is the one in will_become_orphaned_pgrp.

>
> Eric

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);
}
```

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

-serge

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

---

Subject: Re: Screamm.. commit f400e198b2ed26ce55b22a1412ded0896e7516ac
Posted by serue on Thu, 29 Mar 2007 13:18:08 GMT
View Forum Message <> Reply to Message

Quoting Serge E. Hallyn (serue@us.ibm.com):
> Quoting Eric W. Biederman (ebiederm@xmission.com):
> >
> > This is just to vent.
> >
> > I was clearly not auditing patches well enough earlier and the above
> > patch got modified since the version I wrote initially. Adding a
> > few addition is_init calls where what we care about test is not
> > is the real init process of the system (so we should treat it with
> > care) but is this the pid == 1 in my pid namespace.
> >
> > So now we (or maybe me) are going to have to through all of those
> > is_init references and undo the changes for the ones that don't
> > apply to the global init.
> >

> > Guys, please, please be more careful.
> >
> > The one that set me off is the one in will_become_orphaned_pgrp.
> >
> > Eric

By the way thanks for catching this. Clearly it slipped right past
the rest of us!

-serge

> 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);
> }
>
> Where the latter is needed in, for instance, kernel/capability.c.
>
> -serge

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