Subject: Re: [PATCH 9/9] signal: Ignore signals sent to the pid namespace init
Posted by ebiederm on Thu, 13 Dec 2007 18:16:21 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> writes:

```
> On 12/12, Eric W. Biederman wrote:
>>
>> -static int is_sig_init(struct task_struct *tsk)
>> +static int is_sig_init(struct task_struct *init, struct pid *sender)
>>  {
>> - if (likely(!is_global_init(tsk->group_leader)))
>> + if (!is_container_init(init))
>> +  return 0;
>> +
>> + if (!sender)
>> +  sender = task_tgid(current);
>
> What if this signal is sent from the interrupt and sender == NULL?
>
>> +
>> + if (!pid_in_pid_ns(sender, task_active_pid_ns(init)))
>>    return 0;
>
> In that case the result of the above check can be wrong, no?
```

Yes. If we are not in process context (in_interrupt) we do infer
the sender incorrectly. Duh. I saw something in earlier patches
people had posted didn't understand it, and didn't get an answer
when I asked about it. I guess I should have thought about that
corner case and looked a little harder.

In this case the sender would always be the kernel. Oleg do you
know which signals are sent from interrupt context and through
which signal sending entry points?

I'm inclined to say:

```
if (!sender && in_interrupt())
 sender = &init_struct_pid;
```

Which will cause the signal to be dropped if SIG_DFL for the real init
(same pid namespace), and otherwise cause the signal to be sent. Which
sight unseen feels like the right thing.

It worries me that we are likely going through check_kill_permission and
all of the rest.

I expect instead of testing for in_interrupt I should be doing something

like the siginfo test in check_kill_permssion. And just lumping all of
the kernel related signals into the same bin.

I place this one on the backburner of my thoughts and come back to it in
a bit. I am close enough that it should be a simple straight forward code
change whatever the outcome.

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