Subject: Re: [PATCH 8/9] signal: Drop signals before sending them to init.
Posted by Oleg Nesterov on Sun, 16 Dec 2007 15:52:44 GMT
View Forum Message <> Reply to Message

On 12/13, Eric W. Biederman wrote:
>
> Oleg Nesterov <oleg@tv-sign.ru> writes:
>
> > OK, if we change the semantics for /sbin/init signals we can avoid
> > a lot of problems,
>
> Yes. Otherwise we must track the source of the signals.

Yes.

> > Well, I am not sure about "explain" though. Unless I missed something
> > this makes the semantics a bit special.
>
> Well the semantics are a bit special for init period. I just
> make them special in a slightly different way.

Yes.

But still I personally can't agree with the new behaviour.

> > Suppose that init does sigtimedwait() but the handler == SIG_DFL.
>
> Yes that is a bit surprising. However it is still easy to explain.
> The signal is never enqueued so sigtimedwait never gets the chance
> to do anything with it.

But this precisely means that sigtimedwait() is just broken for init.

Another example. /sbin/init execs, and tries to make sure it doesn't
miss the important signal (say, SIGCHLD). So it blocks SIGCHLD, execs,
and the new program installs the handler. However, the handler == SIG_DFL
right after exec, so signal could be missed.

Eric, I think the blocked signal should be enqueued. If application
blocks the signal, this is the strong indication it does care about
it, we shouldn't throw it out.

Yes, this also has surprises. If init unblocks the signal without
installing the handler, it could be killed. But this can happen with
your patch as well. sig_init_drop() returns false if we have a handler,
but this races with sys_rt_sigaction() which can set SIG_DFL, so init
could be killed.

IOW, I still have a strong feeling that this patch

 http://marc.info/?l=linux-kernel&m=118753610515859

is better, and more correct. That said, this all is very subjective,
I can't "prove" this of course.

> Interestingly enough this is not a problem
> for the current sysvinit.
>
> sysvinit does this at start up:
> /*
> * Ignore all signals.
> */
> for(f = 1; f <= NSIG; f++)
> SETSIG(sa, f, SIG_IGN, SA_RESTART);

Yes sure, the "good" init shouldn't have any problems with any approach.

Oleg.