

---

Subject: Re: [PATCH 8/9] signal: Drop signals before sending them to init.  
Posted by [Oleg Nesterov](#) on Thu, 13 Dec 2007 16:25:02 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

On 12/12, Eric W. Biederman wrote:

>  
> By making the rule (for init dropping signals):  
> When sending a signal to init, the presence of a signal handler that  
> is not SIG\_DFL allows the signal to be sent to init. If the signal  
> is not sent it is silently dropped without becoming pending.

But isn't it better to modify sig\_ignore() and handle\_stop\_signal()  
instead? This way we seem to need less changes,

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

(the patch above itself is not complete and a bit obsolete)

> The only noticeable user space difference from today's init is that it  
> no longer needs to worry about signals becoming pending when it has  
> them marked as SIG\_DFL and blocked.

Ugh. I have to apologize again. I got a fever, and it turns out I just  
can't read English.

So, do you mean we can ignore the problems with the signals which are  
currently blocked by /sbin/init?

I personally agree, but I'm not sure I understand this right.

```
> +static int sig_init_drop(struct task_struct *tsk, int sig)
> +{
> + /* All signals for which init has a SIG_DFL handler are
> +  * silently dropped without being sent.
> + */
> + if (!is_sig_init(tsk))
> + return 0;
> +
> + return (tsk->sigband->action[sig-1].sa.sa_handler == SIG_DFL);
> +}
```

What if /sbin/init has a handler, but before this signal is delivered  
/sbin/init does signal(SIG\_DFL) ? We should modify so\_sigaction() to  
prevent this. Note again the patch above.

Oleg.

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

---