Subject: Re: Re: [PATCH 1/3] Signal semantics for /sbin/init
Posted by Daniel Pittman on Mon, 17 Sep 2007 23:20:34 GMT
View Forum Message <> Reply to Message

Oleg Nesterov <oleg@tv-sign.ru> writes:
> On 09/14, Daniel Pittman wrote:
>> Oleg Nesterov <oleg@tv-sign.ru> writes:
>> > On 09/13, Cedric Le Goater wrote:
>> >> Oleg Nesterov wrote:
>>
>> [...]
>>
>> >> To respect the current init semantic,
>> >
>> > The current init semantic is broken in many ways ;)
>>
>> Yup. They sure are, but they are pretty set in stone by now. :)
>>
>> >> shouldn't we discard any unblockable signal (STOP and KILL) sent by a
>> >> process to its pid namespace init process ? Then, all other signals
>> >> should be handled appropriately by the pid namespace init.
>> >
>> > Yes, I think you are probably right, this should be enough in
>> > practice. After all, only root can send the signal to /sbin/init. On
>> > my machine, /proc/1/status shows that init doesn't have a handler for
>> > non-ignored SIGUNUSED == 31, though.
>> >
>> > But who knows? The kernel promises some guarantees, it is not good to
>> > break them. Perhaps some strange non-standard environment may suffer.
>>
>> In this case "strange non-standard environments" would mean anyone
>> running the 'upstart' daemon from recent Ubuntu -- it depends on the
>> current kernel semantics.
>
> Just curious, could you tell more? What "current kernel semantics" do
> you mean?

The semantics where (inside the container) init is protected from a wide
range of unhandled signals by default.

> Do you mean that the 'upstart' daemon sends the unhandled signal to
> init?

Well, yes and no: upstart does not install a handler for several signals
that the traditional sysvinit package uses -- notably, USR1 and USR2
which can trigger reloading of inittab.

The Debian/Ubuntu init scripts still send that signal to the init

process during boot to ensure compatibility with the traditional init package.

The current kernel semantics ensure that upstart can do nothing and the unhandled signal does it no harm -- expect, under OpenVZ while getting Ubuntu working I found that it was not protected and the init script would simply kill upstart dead.

The upstream developers of upstart feel that the containers should provide the same semantics as a raw init, and given that an unknown number of end users will have their own administration systems that depend on the same assumptions about how init works I tend to agree.

So, upstart never sends itself a signal that it can't handle, but the rest of the OS environment can.

Regards,
Daniel

(I tend to think the default protection was a mistake, too, but historic mistakes are todays standards. :/)
--
Daniel Pittman <daniel@cybersource.com.au> Phone: 03 9621 2377
Level 4, 10 Queen St, Melbourne Web: http://www.cyber.com.au
Cybersource: Australia's Leading Linux and Open Source Solutions Company
_____