Subject: Re: [PATCH 11/13] Changes to show virtual ids to user
Posted by Pavel Emelianov on Fri, 25 May 2007 06:33:22 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Pavel Emelianov <xemul@sw.ru> writes:
>
>> That's true. Sending of signal from parent ns to children
>> is tricky question. It has many solutions, I wanted to
>> discuss which one is better:
>
> With unix domain sockets and the like it is conceivable we get
> a pid transfer from one namespace to another and both namespaces
> are leaf namespaces. I don't remember we can get a leaf to leaf
> transfer when sending signals.

We should not allow any transfer from leaf NS to leaf NS.
Should I explain why?

>> 1. Make an "unused" pid in each namespace and use it when signal
>>   comes from outside. This resembles the way it is done in OpenVZ.
>> 2. Send the signal like it came from the kernel.
>>
>>> In particular we need to know the pid of the source task
>>> in the destination namespace.
>> But the source task is not always visible in dst. In this case
>> we may use pid, that never exists in the destination, just like
>> it was kill run from bash by user.
>
> Quite true. So we have the question how do we name a the pid of
> an unmapped task.
>
> The two practical alternatives I see are:
> - Map the struct pid into the namespace in question.

Bad solution. We will poison the dst namespace

> - Use pid == 0 (as if the kernel had generated the signal).

Not just pid 0, but SI_KERNEL in si_code.

> - Use pid == -1 (to signal an unknown user space task?)

Hm... Strange solution.

> My gut fee is that using pid == 0 is the simplest and most robust
> way to handle it. That way we don't have information about things
> outside the pid namespace leaking in. Of course I don't there may

> be trust issues with reporting a user space process as pid == 0.
>
> The worst case I can see with pid == 0. Is that it would be a bug
> that we can fix later. For other cases it would seem to be a user
> space API thing that we get stuck with for all time.

We cannot trust userspace application to expect some pid other than
positive. All that we can is either use some always-absent pid or
send the signal as SI_KERNEL.

Our experience show that making decisions like above causes random
applications failures that are hard (or even impossible) to debug.

> Eric
>