Subject: Re: [PATCH 01/11] SYSCTL: export root and set handling routines
Posted by Stanislav Kinsbursky on Thu, 12 Jan 2012 09:17:16 GMT
View Forum Message <> Reply to Message

>
> Please stop and take a look at /proc/net. If your /proc/net is not a
> symlink please look at a modern kernel.
>
> /proc/<pid>/net reflects the network namespace of the task in question.
>

Ok, I know that.
I know, that if some task with pid N is in other network namespace, then
/proc/<N>/net contents will differ to /proc/selt/net contents.

>> And what do you think about "conteinerization" of /proc contents in the way like
>> "sysfs" was done?
>
> I think the way sysfs is done is a pain in the neck to use. Especially
> in the context of commands like "ip netns exec". With the sysfs model
> there is a lot of extra state to manage.
>
> I totally agree that the way sysfs is done is much better than the way
> /proc/sys is done today. Looking at current can be limiting in the
> general case.
>
> My current preference is the way /proc/net was done.
>

Ok. But this approach still requires some additional data to manage in user
space. I.e. it's really easy to manage container's context using it's fs root,
because container's root is a part of initial configuration. But container's
processed pids numbers in parent context are unpredictable.

>> Implementing /proc "conteinerization" in this way can give us great flexibility.
>> For example, /proc/net (and /proc/sys/sunrpc) depends on mount owner net
>> namespace, /proc/sysvipc depends on mount owner ipc namespace, etc.
>> And this approach doesn't break backward compatibility as well.
>
> The thing is /proc/net is already done.
>
> All I see with making things like /proc/net depend on the context of the
> process that called mount is a need to call mount much more often.
>

/proc/net is a part or /proc. And /proc mount is called per container. So this
is just like it is.

I have some solution I mind, which looks quite simple to implement, doesn't require significant additional state to manage and suits my needs.
Please, consider this.
It's based on sysfs containerization approach, but simplified a lot.
Sysctl's (comparing to sysfs entries) entries are the same for all namespaces.
This actually means, that we don't need any additional infrastructure for managing dentries. All we need to know on read/write operations with sysctl's is the namespaces /proc was mounted from.

Thus if we:

1) replace /proc sb->s_fsdata content from pid_namespace to nsproxy and
2) add link to /proc sb to ctl_table and
3) add ns tag (pid, net, else or none) to ctl_table

then we will have all we need to manage sysctl's content in the way we want.
And looks like this approach doesn't break backward compatibility.

What do you think about it?

--
Best regards,
Stanislav Kinsbursky