
Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by [Cedric Le Goater](#) on Wed, 13 Dec 2006 15:00:35 GMT
[View Forum Message](#) <> [Reply to Message](#)

Serge E. Hallyn wrote:

> Quoting Cedric Le Goater (clg@fr.ibm.com):

>> Dave Hansen wrote:

>>> On Mon, 2006-12-11 at 16:23 +0100, Cedric Le Goater wrote:

>>>> Even letting the concept of nsproxy escape to user space sounds wrong.

>>>> nsproxy is an internal space optimization. It's not struct container

>>>> and I don't think we want it to become that.

>>>> i don't agree here. we need that, so does openvz, vserver, people working

>>>> on resource management.

>>> I think what those projects need is some way to group tasks. I'm not

>>> sure they actually need nsproxies.

>> not only tasks. ipc, fs, etc.

>>

>>> Two tasks in the same container could very well have different

>>> nsproxies. The nsproxy defines how the pid namespace, and pid<->task

>>> mappings happen for a given task.

>> not only. there are other namespaces in nsproxy.

>

> Right, and as Eric has pointed out, you may well want to use one id to

> refer to several nsproxies - for instance if you are using unshare

> to provide per-user private mount namespaces using pam_namespace.so

> (that's mostly for LSPP systems right now, but I do this on my laptop

> too). All my accounts are in the same 'container', but have different

> mount namespaces, hence different nsproxies.

I think we have definition issue here : what is a 'container' ?

I don't see any issue with the above scenario. unsharing mount namespace results in the creation of a new nsproxy which will require a new identifier in order to find this new mount namespace.

so yes, different mount namespaces, hence different nsproxies, hence different ids if you want to find that new mount namespace.

>>> The init process for a container is

>>> special and might actually appear in more than one pid namespace, while

>>> its children might only appear in one. That means that this init

>>> process's nsproxy can and should actually be different from its

>>> children's. This is despite the fact that they are in the same

>>> container.

>>>

>>> If we really need this 'container' grouping, it can easily be something

>>> pointed to by the nsproxy, but it shouldn't be the nsproxy.

>> ok so let's add a container object, containing a nsproxy and add
>> another indirection ...
>
> No thanks.

exactly.

C.

Containers mailing list
Containers@lists.osdl.org
<https://lists.osdl.org/mailman/listinfo/containers>
