Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by ebiederm on Fri, 08 Dec 2006 20:57:38 GMT
View Forum Message <> Reply to Message

"Serge E. Hallyn" <serue@us.ibm.com> writes:

> Quoting Eric W. Biederman (ebiederm@xmission.com):
>> clg@fr.ibm.com writes:
>>
>> > From: Cedric Le Goater <clg@fr.ibm.com>
>> >
>> > This patch adds a hashtable of nsproxy using the nsproxy as a key.
>> > init_nsproxy is hashed at init with key 0. This is considered to be
>> > the 'host' nsproxy.
>>
>> NAK. Which namespace do these ids live in?
>>
>> It sounds like you are setting up to make the 'host' nsproxy special
>> and have special rules. That also sounds wrong.
>>
>> 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.
>>
>> Eric
>
> So would you advocate referring to containers just by the pid of a
> process containing the nsproxy, and letting userspace maintain a mapping
> of id's to containers through container create/enter commands? Or is
> there some other way you were thinking of doing this?

There are two possible ways.
1) Just use a process using the namespace.
   This is easiest to implement.

2) Have a struct pid reference in the namespace itself, and probably
   an extra pointer in struct pid to find it.
   This is the most stable, because fork/exit won't affect which pid you need
   to use.

Beyond that yes it seems to make sense to let user space maintain any mapping
of containers to ids.

Eric

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