Subject: Re: [patch -mm 08/17] nsproxy: add hashtable
Posted by Herbert Poetzl on Sat, 09 Dec 2006 04:11:10 GMT
View Forum Message <> Reply to Message

On Fri, Dec 08, 2006 at 01:57:38PM -0700, Eric W. Biederman wrote:
> "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?

well, I gave a similar answer in another email,
so I fully agree with the NAK here ...

> >> 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.

while I agree that nsproxy is definitely the wrong
point to tie a 'context' too, as it can contain a
mixture of spaces from inside and outside a context,

and it would require to forbid doing things like
clone() with the space flags, both inside and outside
a 'container' to allow to use them for actual vps
applications, I think that we have to have some kind
of handle to tie specific sets of namespaces too

that 'can' be an nsproxy or something different, but
I'm absolutely unhappy with tying it to a process,
as I already mentioned several times, that lightweight
'containers' do not use/have an init process, and no
single process might survive the entire life span of
that 'container' ...

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

I agree with that, but we need something to move
around between the various spaces ...

for example, Linux-VServer ties the namespaces to
the context structure (atm) which allows userspace
to set and enter specific spaces of a guest context
(I assume OpenVZ does similar)

HTC,
Herbert

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