Subject: Re: [RFC] [PATCH 0/3] containers: introduction
Posted by ebiederm on Wed, 10 Jan 2007 22:00:58 GMT
View Forum Message <> Reply to Message

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

> Quoting Eric W. Biederman (ebiederm@xmission.com):
>> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>>
>> > Or we could go ahead and fully implement it in procfs. As you'd said
>> > earlier, that really maps best into what we want. Containerfs was
>> > just much simpler and quicker to implement for demonstrating the semantics.
>>
>> Well for what it is worth I just notices that nfs is currently and automounter
>> that transparently unmounts it's children when you unmount it. I don't think
>> that is quite enough to split /proc into two but it does have some potential
>> when it comes to new features.
>>
>> Using itty bity purpose built file systems if there is an automounter for them
>> because much easier for user space.
>
> I'm not parsing the last sentence.
>
> Are you suggesting that we may be able to stick with a custom fs,
> using autofs to automount it if the symlink /proc/$$/container is
> dereferenced while only a kernel mount of /containers exists?
>
> I suppose a simpler solution is to not define /proc/$$/container,
> but rather just let /container in the containerfs symlink to
> the current process' container. That way you can't reference
> /containers/container unless containerfs is already mounted under
> /containers, and we avoid the problem completely.

I am saying:
autofs is not special. Doing automounting the nfs way
you can add and remove mounts transparently to the user.

A very good use for this would be to mount/unmount things like
/proc/sys/fs/binfmt_misc/.

That technique may have an implication for the design of a container
filesystem.

The result is that if something is more simply implemented as a
separate filesystem, that is a possibility.

Eric

_____

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