Subject: Re: The issues for agreeing on a virtualization/namespaces implementation.
Posted by ebiederm on Thu, 09 Feb 2006 05:41:10 GMT

View Forum Message <> Reply to Message

Kyle Moffett <mrmacman_g4@mac.com> writes:

> On Feb 07, 2006, at 17:06, Eric W. Biederman wrote:
>> I think I can boil the discussion down into some of the fundamental questions
>> that we are facing.
>>
>> Currently everyone seems to agree that we need something like my namespace
>> concept that isolates multiple resources.
>>
>> We need these for
>> UIDS
>> FILESYSTEM
>
> I have one suggestion for this (it also covers capabilities to a certain
> extent). Could we use the kernel credentials system to abstract away the
> concept of a single UID/GID? We currently have uid, euid, gid, egid, groups,
> fsid. I'm thinking that there would be virtualized UID tables to determine
> ownership of processes/SHM/etc.
>
> Each process would have a (uid_container,uid) pair (or similar) as its "uid"
> and likewise for gid. Then the ability to send signals to any given
> (uid_container,uid) or (gid_container,gid) pair would be given by keys in the
> kernel keyring indexed by the "uid_container" part and containing the "uid"
> part (or maybe just a pointer).
>
> Likewise the filesystem access could be virtualized by using uid and gid keys
> in the kernel keyring indexed by vfsmount (Not superblock, so that it would be
> possible to have different UID representations on different mounts/parts of the
> same filesystem).
>
> I'm guessing that the performance implications of the above would not be quite
> so nice, as it would put a lot of code in the fastpath, but I would guess that
> it might be possible to use the existing fields for processes without any
> virtualization needs.

At least for signal sending it looks like it would be easier to just compare
the pointers to struct user. At least in that context it looks like it
would be as cheap as what we are doing now. I just don't know where
to find a struct user for the euid, or is it the normal uid.

Eric