Subject: Re: [RFC][PATCH 2/7] VPIDs: pid/vpid conversions
Posted by ebiederm on Thu, 09 Feb 2006 01:36:33 GMT
View Forum Message <> Reply to Message

Alexey Kuznetsov <kuznet@ms2.inr.ac.ru> writes:

> Hello!
>
>> In capability.c it does for_each_thread or something like that. It is
>> very similar to cap_set_pg. But in a virtual context all != all :)
>
> Do you mean that VPID patch does not include this? Absolutely.
> VPIDs are not to limit access, the patch virtualizes pids, rather
> than deals with access policy.
>
> Take the whole openvz. Make patch -R < vpid_patch. The result is perfectly
> working openvz. Only pids are not virtual, which does not matter. Capisco?

Not quite.

sys_kill knows about three kinds of things referred to by pid.
- individual processes (positive pid)
- process groups (pid < -1 or pid == 0)
- The group of all processes.

When you have multiple instances of the same pid on the box
The group of all processes becomes the group of all processes I
can see. At least that was my impression.

>> I think for people doing migration a private pid space in some form is
>> necessary,
>
> Not "private", but "virtual". VPIDs are made only for migration, not for fun.
>
> And word "private" is critical, f.e. for us preserving some form of pid
> space is critical. It is very sad, but we cannot do anything with this,
> customers will not allow to change status quo.

Ok. I'm not quite certain what the difference is. In OpenVZ it appears
to be something significant. In most conversations it is not.

>> My problem with the vpid case and it's translate at the kernel
>> boundary is that boundary is huge
>
> Believe me, it is surprizingly small.

Well the number of places you need to translate may be small.
But the number of lines of code is certainly large.

Every driver, every ioctl, every other function that could
be abused ioctl.

And if the values are ever stored instead of used immediately
you can get a context mismatch.

Eric