Subject: Re: [RFC][PATCH] Devices visibility container
Posted by ebiederm on Tue, 25 Sep 2007 13:43:53 GMT
View Forum Message <> Reply to Message

Cedric Le Goater <clg@fr.ibm.com> writes:

> Hello Eric !
>
> Eric W. Biederman wrote:
>> Pavel Emelyanov <xemul@openvz.org> writes:
>>
>>> At KS we have pointed out the need in some container, that allows
>>> to limit the visibility of some devices to task within it. I.e.
>>> allow for /dev/null, /dev/zero etc, but disable (by default) some
>>> IDE devices or SCSI discs and so on.
>>
>> NAK
>>
>> We do not want a control group subsystem for this.
>
> we will need one way to configure the list of available devices from
> user space. Any proposal ?

Proposal 1/2. From the kernel side we have.
dev_ns_add(kdev_t cur_dev, struct dev_ns *target_ns, kdev_t target_kdev)
Which looks up the device and add it to the hash tables in the proper
device namespace, and fires off the appropriate hotplug events.

I guess the easy user space interface would be:
echo <device_ns_pid>:<major>:<minor> > /sys/block/ram0/dev

Although I suspect that we want some restrictions on what
combinations of major and minor numbers are valid.

Despite the fact that my gut says writeable sysfs files were
a bad idea. Since we have them my gut says sysfs the filesystem
of devices is where we need the control files for devices.

>> For the short term we can just drop CAP_SYS_MKNOD.
>
> Sure. Pavel is working on something mid-term ;)

Well. I don't think midterm is mergeable, I do think it is good
for conversation though. I also don't see why what Pavel is doing
can't be implemented as a device namespace.

>> For the long term we need a device namespace for application
>> migration, so they can continue to use devices with the same

>> major+minor number pair after the migration event.
>
> Hmm, yes. I can imagine that for some big database application using
> raw devices but it only means that the same device must be present
> upon restart. I don't see any identifier virtualization issues.

Well there is the classic one. You are migrating to a machine which
is using that major+minor number for a different device already.

Especially in the cases like network block devices or SCSI talking
to SAN, we can talk to the same device and still have a different
major+minor number after migration in the current setup.

I think we can hit similar issues with ttys, loopback devices,
and ramdisks as well.

>> Things like
>> ensuring a call to stat on a given file before and after the migration
>> return the exact same information sounds compelling. So I don't think
>> this is even strictly limited to virtual devices anymore. How many
>> applications are there out there that memorize the stat data on a file
>> and so they can detect if it has changed?
>
> that we need to support of course, otherwise we would break things
> like tail.

Exactly. tail, git, backup software.
All kinds of infrequently run but interesting software.

>> If we need something between those two it may make sense to enhance
>> the LSM or perhaps introduce an alternate set security hooks. Still
>> if we are going to need a full device namespace that may be a little
>> much.
>
> serge's implementation using security hooks should help us choose
> the right approach.

Sure.

Currently I have to agree with Alan Cox that our biggest security
need seems to be a good implementation of revoke in the kernel.
So we can do things like ensure a device is not being used by anyone
else. For removal of character and block devices we may not need a
general thing but it is worth looking at.

Eric

_____
Containers mailing list