Subject: Re: [DRAFT] Container mini-summit notes v0.01
Posted by ebiederm on Sat, 08 Sep 2007 20:26:08 GMT
View Forum Message <> Reply to Message

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

>> > then you should have taken CAP_SYS_MKNOD away from the container.
>>
>> no serge,
>>
>> we want the container to be able to mknod()
>
> Someone give me one good reason why this is needed.

The picture that I see is still fuzzy, so I cannot say exactly what for a device namespace needs to take. The practical issues is that we have virtual devices that when we migrate people will want to continue using. ptys are the common case here, but there are loop devices and other virtual devices.

Doing things like changing the major/minor numbers on a device we currently have open during migration could be painful.

For non-virtual devices we can treat it as a device hot plug event, because we really cannot continue with the device open. For the virtual devices we can do better and so it is quite likely that we want to.

This isn't an important issue until we get to the point of dealing with migration however.

>> >> Or mounts it from somewhere outside.
>> >
>> > and CAP_SYS_MOUNT
>>
>> and that also.
>
> Same here. Restricting containers to user mounts - which include
> a great deal of things including fuse loopback etc - should be fine.

The last I looked at user mounts they implied nosuid and nodev.

Which leads to an interesting implication. sys_mknod support in a container does not appear to be fundamental, while device namespaces so we can keep virtual devices at their same major/minor numbers looks fundamental.

> But again, if everyone but me agrees on this, we can try to focus on
> this instead of devpts this year.  Cedric, was this mentioned at the
> kernel summit?  Was there any reaction to this idea?

We didn't go into much technical detail a kernel summit.  The goal was to stick to topic that were of general interest to most of the group.  Which was mostly kernel process related.  We did talk about our high level objectives and the biggest question was when will the container work be done?  No real objections were answered.

So for technical details we still need to discuss them on the appropriate mailing lists.

> This of course is also something that could be implemented pretty simply
> as a container subsys defining the security_mknod hook, with the
> whitelist defined through the task container interface.

Something to mention.  I keep thinking for the isolation aspects of this it may make sense to refactor the code behind the security hooks to be a table based implementation like netfilter.  Allowing code from multiple parties to be used together instead of the current all or nothing paradigm.

>> > Anyway if people really all agree on a per-container device whitelist,
>> > I won't object.  Just seems like overkill to me.
>> >>> Whereas devpts you do need namespaces for.
>> >>> -serge

The practical question is what do we need to do to migrate applications that are using virtual devices.

>> let's get back on the mailing list !

Back.

Eric

_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

## Subject: Re: [DRAFT] Container mini-summit notes v0.01
## Posted by serue on Mon, 10 Sep 2007 14:18:34 GMT

View Forum Message <> Reply to Message

Quoting Eric W. Biederman (ebiederm@xmission.com):

> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
> >> > then you should have taken CAP_SYS_MKNOD away from the container.
> >>
> >> no serge,
> >>
> >> we want the container to be able to mknod()
> >
> > Someone give me one good reason why this is needed.
>
> The picture that I see is still fuzzy, so I cannot say exactly what
> for a device namespace needs to take.  The practical issues is that we
> have virtual devices that when we migrate people will want to continue
> using.  ptys are the common case here, but there are loop devices
> and other virtual devices.
>
> Doing things like changing the major/minor numbers on a device
> we currently have open during migration could be painful.
>
> For non-virtual devices we can treat it as a device hot plug
> event, because we really cannot continue with the device open.
> For the virtual devices we can do better and so it is quite likely
> that we want to.
>
> This isn't an important issue until we get to the point of dealing
> with migration however.

Sorry, I was focusing on the virtual server needs.

devpts is it's own fs so I was fully expecting to make it mountable
multiple times so a container can have it's own /dev/pts/0.  So what
other virtual devices would we want to be able to rec-reate for a
migrated application?  (I wonder (a) what gregkh will say about having
a device namespace, and (b) what the sysfs implications will be)

> >> >> Or mounts it from somewhere outside.
> >> >
> >> > and CAP_SYS_MOUNT
> >>
> >> and that also.
> >
> > Same here.  Restricting containers to user mounts - which include
> > a great deal of things including fuse loopback etc - should be fine.
>
> The last I looked at user mounts they implied nosuid and nodev.
>
> Which leads to an interesting implication. sys_mknod support in
> a container does not appear to be fundamental, while device namespaces

> so we can keep virtual devices at their same major/minor numbers looks
> fundamental.
>
> > But again, if everyone but me agrees on this, we can try to focus on
> > this instead of devpts this year.  Cedric, was this mentioned at the
> > kernel summit?  Was there any reaction to this idea?
>
> We didn't go into much technical detail a kernel summit.  The goal
> was to stick to topic that were of general interest to most of the
> group.  Which was mostly kernel process related.  We did talk about
> our high level objectives and the biggest question was when will the
> container work be done?  No real objections were answered.
>
> So for technical details we still need to discuss them on the appropriate
> mailing lists.
>
> > This of course is also something that could be implemented pretty simply
> > as a container subsys defining the security_mknod hook, with the
> > whitelist defined through the task container interface.
>
> Something to mention.  I keep thinking for the isolation aspects of this
> it may make sense to refactor the code behind the security hooks to
> be a table based implementation like netfilter.  Allowing code from
> multiple parties to be used together instead of the current all or
> nothing paradigm.
>
> >> > Anyway if people really all agree on a per-container device whitelist,
> >> > I won't object.  Just seems like overkill to me.
> >> >>> Whereas devpts you do need namespaces for.
> >> >>> -serge
>
> The practical question is what do we need to do to migrate applications
> that are using virtual devices.
>
> >> let's get back on the mailing list !
>
> Back.

Excellent.

> Eric

-serge
_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [DRAFT] Container mini-summit notes v0.01
Posted by ebiederm on Mon, 10 Sep 2007 16:09:58 GMT
View Forum Message <> Reply to Message

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

> Sorry, I was focusing on the virtual server needs.
>
> devpts is it's own fs so I was fully expecting to make it mountable
> multiple times so a container can have it's own /dev/pts/0.  So what
> other virtual devices would we want to be able to rec-reate for a
> migrated application?  (I wonder (a) what gregkh will say about having
> a device namespace, and (b) what the sysfs implications will be)

Depends.  There are things like the loop device that could be interesting.
There may be some others.  I haven't looked at it enough detail to get
beyond the fact that in some sense it isn't just limited to pts devices.

A multimount devpts is interesting though.

Eric

_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

Subject: Re: [DRAFT] Container mini-summit notes v0.01
Posted by Oren Laadan on Wed, 26 Sep 2007 20:14:21 GMT
View Forum Message <> Reply to Message

(sorry from the delay, been away :)

Eric W. Biederman wrote:
> "Serge E. Hallyn" <serue@us.ibm.com> writes:
>
>> Sorry, I was focusing on the virtual server needs.
>>
>> devpts is it's own fs so I was fully expecting to make it mountable
>> multiple times so a container can have it's own /dev/pts/0.  So what
>> other virtual devices would we want to be able to rec-reate for a
>> migrated application?  (I wonder (a) what gregkh will say about having
>> a device namespace, and (b) what the sysfs implications will be)
>
> Depends.  There are things like the loop device that could be interesting.
> There may be some others.  I haven't looked at it enough detail to get
> beyond the fact that in some sense it isn't just limited to pts devices.

>
> A multimount devpts is interesting though.

Devices I had to deal with (in zap) so far - to be able to ckpt/restart
(and migrate) a desktop session:

* /dev/rtc  (e.g. for mplayer)

* /dev/dsp

* /dev/random ?  (to isolate entropy pools ?)

* virtual consoles - e.g. in zap, an X server that uses a virtual device
runs inside a pod/container/VE (and X per-se requires a virtual console)

* virtual terminals - e.g. in zap we allow access to a pod from the host
without a need to run 'sshd' inside and setup a network in the pod. (Then
with a suitable utility and network access to the host, this also allows
sort of remote (a-la serial) console access).
>From inside the pod it looks like /dev/tty{1,2,..}, so one can run 'getty'
processes inside the pod. From the outside (for the admin, e.g.) it is an
extended /dev/tty that has an extra ioctl to multiplex access, so the
admin (program) can ask to be connected to tty X of pod Y, and it will
connect to that console (like connecting via serial line).
The main advantage is that as a virtual device it can be migrated (with
its buffers, if not empty, as they reside inside the pod) so upon restart
they go with the 'getty' processes that use them. The (old) admin will
see the line dropped, and the (new) admin after the migration can connect
at the new machine.

Oren.

_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

---

## Subject: Re: [DRAFT] Container mini-summit notes v0.01
## Posted by Sukadev Bhattiprolu on Thu, 18 Oct 2007 00:52:16 GMT

View Forum Message <> Reply to Message

Oren Laadan [orenl@cs.columbia.edu] wrote:
|
| (sorry from the delay, been away :)
|
| Eric W. Biederman wrote:

| > "Serge E. Hallyn" <serue@us.ibm.com> writes:
| >
| >> Sorry, I was focusing on the virtual server needs.
| >>
| >> devpts is it's own fs so I was fully expecting to make it mountable
| >> multiple times so a container can have it's own /dev/pts/0.  So what
| >> other virtual devices would we want to be able to rec-reate for a
| >> migrated application?  (I wonder (a) what gregkh will say about having
| >> a device namespace, and (b) what the sysfs implications will be)
| >
| > Depends.  There are things like the loop device that could be interesting.
| > There may be some others.  I haven't looked at it enough detail to get
| > beyond the fact that in some sense it isn't just limited to pts devices.
| >
| > A multimount devpts is interesting though.
|
| Devices I had to deal with (in zap) so far - to be able to ckpt/restart
| (and migrate) a desktop session:
|
| * /dev/rtc  (e.g. for mplayer)
|
| * /dev/dsp
|
| * /dev/random ?  (to isolate entropy pools ?)
|
| * virtual consoles - e.g. in zap, an X server that uses a virtual device
| runs inside a pod/container/VE (and X per-se requires a virtual console)
|
| * virtual terminals - e.g. in zap we allow access to a pod from the host
| without a need to run 'sshd' inside and setup a network in the pod. (Then
| with a suitable utility and network access to the host, this also allows
| sort of remote (a-la serial) console access).
| >From inside the pod it looks like /dev/tty{1,2,..}, so one can run 'getty'
| processes inside the pod. From the outside (for the admin, e.g.) it is an
| extended /dev/tty that has an extra ioctl to multiplex access, so the
| admin (program) can ask to be connected to tty X of pod Y, and it will
| connect to that console (like connecting via serial line).

This sounds really interesting. Were these devices part of a complete
device namespace ?  IOW, does say /dev/tty2 in each pods have the same
major/minor number (4,2) ? Does each '/dev/tty2' have a separate entry
in sysfs ?

| The main advantage is that as a virtual device it can be migrated (with
| its buffers, if not empty, as they reside inside the pod) so upon restart
| they go with the 'getty' processes that use them. The (old) admin will
| see the line dropped, and the (new) admin after the migration can connect

| at the new machine.
|
| Oren.
|
|
| _______________________________________________
| Containers mailing list
| Containers@lists.linux-foundation.org
| https://lists.linux-foundation.org/mailman/listinfo/containers

_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers

## Subject: Re: [DRAFT] Container mini-summit notes v0.01
## Posted by Oren Laadan on Tue, 30 Oct 2007 04:35:15 GMT

View Forum Message <> Reply to Message

sukadev@us.ibm.com wrote:
> Oren Laadan [orenl@cs.columbia.edu] wrote:
> |
> | (sorry from the delay, been away :)
> |
> | Eric W. Biederman wrote:
> | > "Serge E. Hallyn" <serue@us.ibm.com> writes:
> | >
> | >> Sorry, I was focusing on the virtual server needs.
> | >>
> | >> devpts is it's own fs so I was fully expecting to make it mountable
> | >> multiple times so a container can have it's own /dev/pts/0.  So what
> | >> other virtual devices would we want to be able to rec-reate for a
> | >> migrated application?  (I wonder (a) what gregkh will say about having
> | >> a device namespace, and (b) what the sysfs implications will be)
> | >
> | > Depends.  There are things like the loop device that could be interesting.
> | > There may be some others.  I haven't looked at it enough detail to get
> | > beyond the fact that in some sense it isn't just limited to pts devices.
> | >
> | > A multimount devpts is interesting though.
> |
> | Devices I had to deal with (in zap) so far - to be able to ckpt/restart
> | (and migrate) a desktop session:
> |
> | * /dev/rtc  (e.g. for mplayer)
> |
> | * /dev/dsp
> |

> | * /dev/random ?  (to isolate entropy pools ?)
> |
> | * virtual consoles - e.g. in zap, an X server that uses a virtual device
> | runs inside a pod/container/VE (and X per-se requires a virtual console)
> |
> | * virtual terminals - e.g. in zap we allow access to a pod from the host
> | without a need to run 'sshd' inside and setup a network in the pod. (Then
> | with a suitable utility and network access to the host, this also allows
> | sort of remote (a-la serial) console access).
> | >From inside the pod it looks like /dev/tty{1,2,..}, so one can run 'getty'
> | processes inside the pod. From the outside (for the admin, e.g.) it is an
> | extended /dev/tty that has an extra ioctl to multiplex access, so the
> | admin (program) can ask to be connected to tty X of pod Y, and it will
> | connect to that console (like connecting via serial line).
>
> This sounds really interesting. Were these devices part of a complete
> device namespace ?  IOW, does say /dev/tty2 in each pods have the same
> major/minor number (4,2) ? Does each '/dev/tty2' have a separate entry
> in sysfs ?

yes, they are virtualization-aware (keep in mind that this was done
before the recent work on namespaces), by having the open() method check
in which pod (namespace) it is called and act accordingly. So /dev/zty2
(zty stands for zap-tty) has the same maj/min in all pods. while at this
moment it is not integrated with sysfs, I see no reason not to do so.

>
>
> | The main advantage is that as a virtual device it can be migrated (with
> | its buffers, if not empty, as they reside inside the pod) so upon restart
> | they go with the 'getty' processes that use them. The (old) admin will
> | see the line dropped, and the (new) admin after the migration can connect
> | at the new machine.
> |
> | Oren.
> |
> |
> | _______________________________________________
> | Containers mailing list
> | Containers@lists.linux-foundation.org
> | https://lists.linux-foundation.org/mailman/listinfo/containers

_______________________________________________
Containers mailing list
Containers@lists.linux-foundation.org
https://lists.linux-foundation.org/mailman/listinfo/containers