Subject: Q: How complete is the pid namespace in mainline
Posted by ebiederm on Fri, 26 Oct 2007 05:17:47 GMT
View Forum Message <> Reply to Message

Guys how complete do you fee the pid namespace support is that
has been merged into Linus's tree?

My impression until I started reading through code earlier today
was that the support was just about done except for a couple of
tricky details.

Eric

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

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by Cedric Le Goater on Fri, 26 Oct 2007 08:52:38 GMT
View Forum Message <> Reply to Message

Eric W. Biederman wrote:
> Guys how complete do you fee the pid namespace support is that
> has been merged into Linus's tree?
>
> My impression until I started reading through code earlier today
> was that the support was just about done except for a couple of
> tricky details.

Yes It looks sane.

Here's what I have in mind :

* there are some patches from suka that make sure the pid namespace init
  is not getting abusively killed by one of this children
* the pid cleanup is not complete
 . locks
 . kthread (i should work soon on improving kthread to support
        signals)

IMO, the proc mount shouldn't be under the pid namespace. we will
need that sooner or later.

C.

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by ebiederm on Fri, 26 Oct 2007 09:33:25 GMT
View Forum Message <> Reply to Message

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

> Eric W. Biederman wrote:
>> Guys how complete do you fee the pid namespace support is that
>> has been merged into Linus's tree?
>>
>> My impression until I started reading through code earlier today
>> was that the support was just about done except for a couple of
>> tricky details.
>
> Yes It looks sane.
>
> Here's what I have in mind :
>
> * there are some patches from suka that make sure the pid namespace init
>   is not getting abusively killed by one of this children
> * the pid cleanup is not complete
> . locks
> . kthread (i should work soon on improving kthread to support
>   signals)
>
> IMO, the proc mount shouldn't be under the pid namespace. we will
> need that sooner or later.

I was hoping to get a larger list of unfixed issues.

Currently from my review I have generated about 25 bugfix patches.
Several of them in some fairly obvious places.

I think it is a good base to build on, but it feels to like we still
have a significant ways to go.

I think the assumption that we can use global pid numbers instead
of instead of struct pids is racy, and a serious maintenance problem.
It leads to silent breakage of routines like get_net_ns_by_pid,
and possibly a couple of other places.

I'm really not happy with pid_nr meaning a pid number in the

init_pid_ns and pid_vnr meaning a pid number meaning a pid in the
local pid namespace. But that is just a matter of names so I
don't think it has caused any problems. Short of making it
to easy to get a pid number in the &init_pid_ns.

Eric

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

---

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by Sukadev Bhattiprolu on Fri, 26 Oct 2007 17:17:18 GMT
View Forum Message <> Reply to Message

Eric W. Biederman [ebiederm@xmission.com] wrote:
|
| Guys how complete do you fee the pid namespace support is that
| has been merged into Linus's tree?
|
| My impression until I started reading through code earlier today
| was that the support was just about done except for a couple of
| tricky details.

The only thing that I know is pending is the issue of signalling
container-init. We have not been able to find a clean fix for it.

The problem now is that a process in a child namespace can terminate
its container-init and thereby the entire container. We have a 3-patch
set (Oleg's and mine) that kind of addresses this. The scenario where
the patchset fails is :

- the container-init has a blockable, fatal signal blocked

- a descendant of the container-init posts the fatal signal to
  container-init.

- container-init then unblocks the signal without ignoring or
  handling the signal.

In this case again the container-init can be terminated.

(by fatal I mean a signal whose default action is to terminate the process
SIGKILL is of couse not blockable and is not a problem)

This issue can be addressed in user-space by the container-init - which
should just ignore the fatal signal or setup a handler for it.

Dave had suggested we print a warning the first time a container-init forks()
without a handler for a fatal signal. I was planning on adding that as
patch 4 of the signal patch set and get some feedback.

Suka

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

---

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by ebiederm on Fri, 26 Oct 2007 18:17:33 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com writes:
>
> Dave had suggested we print a warning the first time a container-init forks()
> without a handler for a fatal signal. I was planning on adding that as
> patch 4 of the signal patch set and get some feedback.

Yes. How to cleanly handle signalling of container init is
a tricky one. It does sound like you have made a reasonable start
there.

Suka it is a lot more then that. How much more I'm not certain
of. I suspect the only way to find the rest of the cases is
just go through the code with a fine tooth come and read and look.

So far doing that it has not at all hard for me to find either
bugs or places where the implementation can be improved.

Currently we have little things like kill(-1,...) signalling the
wrong set of processes, and a couple of proc bugs.

That autofs and coda out on the fringe don't quite do the right
thing in the presence of multiple pid namespaces isn't a big
surprise, little details like that are easy to overlook.

We of course still have the kthread issue where we can get
kernel threads trapped and we have kernel threads calling
kill_proc, keeping us from removing it.

There is all cap_set_all which isn't filtering by pid namespace.

Then we have the unix domain sockets that don't appear to do the
right thing when passing credentials across pid namespaces. I

think we may have the same issues with signals as well.

Anyway I can find a lot issues like that without trying very
hard. Which suggests to me that there are issues that I'm missing
that are out there as well.

So it appears there is lots of cleanup work to do.

Eric

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

---

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by Sukadev Bhattiprolu on Fri, 26 Oct 2007 21:29:59 GMT
View Forum Message <> Reply to Message

Eric W. Biederman [ebiederm@xmission.com] wrote:
| sukadev@us.ibm.com writes:
| >
| > Dave had suggested we print a warning the first time a container-init forks()
| > without a handler for a fatal signal. I was planning on adding that as
| > patch 4 of the signal patch set and get some feedback.
|
| Yes. How to cleanly handle signalling of container init is
| a tricky one. It does sound like you have made a reasonable start
| there.
|
| Suka it is a lot more then that. How much more I'm not certain
| of. I suspect the only way to find the rest of the cases is
| just go through the code with a fine tooth come and read and look.

I agree. I did not mean to ignore the kthread conversions and was only
referring to the core pid namespace clone stuff.

|
| So far doing that it has not at all hard for me to find either
| bugs or places where the implementation can be improved.
|
| Currently we have little things like kill(-1,...) signalling the
| wrong set of processes, and a couple of proc bugs.

I just realized the fix for this is in the signal patchset I was
referring to.

https://lists.linux-foundation.org/pipermail/containers/2007-August/006987.html

I notice that you have sent a patch for the kill -1.

The proc_mnt bug Linus found seems to have slipped through when
merging Pavel's and my patches.

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

---

Subject: Re: Q: How complete is the pid namespace in mainline
Posted by ebiederm on Fri, 26 Oct 2007 23:09:26 GMT
View Forum Message <> Reply to Message

sukadev@us.ibm.com writes:

> Eric W. Biederman [ebiederm@xmission.com] wrote:
> | sukadev@us.ibm.com writes:
> | >
> | > Dave had suggested we print a warning the first time a container-init
> forks()
> | > without a handler for a fatal signal. I was planning on adding that as
> | > patch 4 of the signal patch set and get some feedback.
> |
> | Yes. How to cleanly handle signalling of container init is
> | a tricky one. It does sound like you have made a reasonable start
> | there.
> |
> | Suka it is a lot more then that. How much more I'm not certain
> | of. I suspect the only way to find the rest of the cases is
> | just go through the code with a fine tooth come and read and look.
>
> I agree. I did not mean to ignore the kthread conversions and was only
> referring to the core pid namespace clone stuff.

Sure, and that make sense.

> | So far doing that it has not at all hard for me to find either
> | bugs or places where the implementation can be improved.
> |
> | Currently we have little things like kill(-1,...) signalling the
> | wrong set of processes, and a couple of proc bugs.
>
> I just realized the fix for this is in the signal patchset I was
> referring to.
>
> https://lists.linux-foundation.org/pipermail/containers/2007-August/006987.html

>
> I notice that you have sent a patch for the kill -1.

Yes. I'm trying to get out as many simple little bug fixes
as I can.

Sorry for missing the fact you guys had generated some patches
to address this. Still I think mine is a little more comprehensive
and shorter ;)

That bug is on my list of really nasty bugs I want to avoid.

> The proc_mnt bug Linus found seems to have slipped through when
> merging Pavel's and my patches.

I really don't mind a handful of little bugs, it would be
surprising if something hadn't slipped through at this point.

As long as everyone is aware that it is going to take a bit
to find everything and stabilizing it all and everyone keeps
looking we should be fine.

Oh. Do you know if there was a good reason for forcing
the tty, session, and process group of a the first process
in a pid namespace?

Eric

_____