Subject: dev_change_netns on a tunnel device
Posted by Sapan Bhatia on Mon, 18 Jun 2007 05:01:28 GMT
View Forum Message <> Reply to Message

Hi Eric,

Are the current semantics of tunnel devices (ipip, ip_gre etc.) with
respect to changing the current netns correct? These devices have an
ip_tunnel structure associated with each net_device, which is
currently a global (not per_net). The result is that you can set up in
container X a tunnel with endpoints associated with a device in
container Y in a different network namespace. This feature seems
useful, because you can hook up containers on 2 machines with one of
these tunnels, without using etun (or like) devices, but intuitively,
it seems that the tunnel module should be listening for a
DEV_UNREGISTER and reset it when the device is migrated

Sapan

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

Subject: Re: dev_change_netns on a tunnel device
Posted by ebiederm on Tue, 19 Jun 2007 17:07:46 GMT
View Forum Message <> Reply to Message

"Sapan Bhatia" <sapan.bhatia@gmail.com> writes:

> Hi Eric,
>
> Are the current semantics of tunnel devices (ipip, ip_gre etc.) with
> respect to changing the current netns correct? These devices have an
> ip_tunnel structure associated with each net_device, which is
> currently a global (not per_net). The result is that you can set up in
> container X a tunnel with endpoints associated with a device in
> container Y in a different network namespace. This feature seems
> useful, because you can hook up containers on 2 machines with one of
> these tunnels, without using etun (or like) devices, but intuitively,
> it seems that the tunnel module should be listening for a
> DEV_UNREGISTER and reset it when the device is migrated

Good question. Examining the ipip case here is my conclusion.

- Because the directing of packets is based on ip address and multiple
  network namespaces are allowed to use the same ip addresses then
  the decode needs to take the network namespace into account.

- The only thing that would prevent the migration semantics from being
  correct is if you could manipulate a migrated tunnel in such a way that you
  could do something nasty to the source namespace.

  Since a tunnel change command is also a tunnel rename command that should
  force any ipip tunnel into using ip addresses from the current
  namespace, which makes it safe, and thus unable to affect the source
  namespace.

So semantically I believe the tunnel semantics are essentially
correct. However I believe there are several places where the code
needs to be updated to correctly implement those semantics.

Eric

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

---

Subject: Re: dev_change_netns on a tunnel device
Posted by Sapan Bhatia on Tue, 19 Jun 2007 19:01:40 GMT
View Forum Message <> Reply to Message

> - Because the directing of packets is based on ip address and multiple
>   network namespaces are allowed to use the same ip addresses then
>   the decode needs to take the network namespace into account.

Hm, but a tunnel device can belong to only one namespace, so shouldn't this
get handled automatically, since packet flow upstream of the tunnel is
handled by the container route table?

So eg. if we have tunnel ipip0 in container X, with endpoints bound to eth0
in init_net, then encapsulated packets intercepted by the tunnel show up in
on ipip0, which if my understanding is correct, should be able to have an
address collision with a device in another container, since the other
address doesn't show up in the local route table.

- The only thing that would prevent the migration semantics from being
>   correct is if you could manipulate a migrated tunnel in such a way that
> you
>   could do something nasty to the source namespace.
>
  Since a tunnel change command is also a tunnel rename command that should
>   force any ipip tunnel into using ip addresses from the current
>   namespace, which makes it safe, and thus unable to affect the source

> namespace.

One example of a safety violation is that in GRE, the key participates in
routing, and one might be able to set it so that a container sneaks its
packets into another tunnel that it doesn't own. Don't know if 'ip tunnel'
can be used to do other bad things.

There might also be some applications that assume the semantics in which the
backend and frontend are in the same container. eg. the OSPF daemon of
QUAGGA for some reason binds to the 'local' endpoint of a GRE tunnel instead
of the address assigned to the interface, and inside a container, fails with
a "No such device". We're looking into this right now.

So semantically I believe the tunnel semantics are essentially
> correct. However I believe there are several places where the code
> needs to be updated to correctly implement those semantics.
>
> Eric
>

Ok.

Thank you,
Sapan

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

---

Subject: Re: dev_change_netns on a tunnel device
Posted by ebiederm on Tue, 19 Jun 2007 19:37:40 GMT
View Forum Message <> Reply to Message

"Sapan Bhatia" <sapan.bhatia@gmail.com> writes:

> - Because the directing of packets is based on ip address and multiple
> network namespaces are allowed to use the same ip addresses then
> the decode needs to take the network namespace into account.
>
>
> Hm, but a tunnel device can belong to only one namespace, so shouldn't this get
> handled automatically, since packet flow upstream of the tunnel is handled by
> the container route table?
>
> So eg. if we have tunnel ipip0 in container X, with endpoints bound to eth0 in

> init_net, then encapsulated packets intercepted by the tunnel show up in on
> ipip0, which if my understanding is correct, should be able to have an address
> collision with a device in another container, since the other address doesn't
> show up in the local route table.

But you can have a collision in the local route table. 127.0.0.1 is the
common case here but other cases are also allowed. It requires a pretty
sophisticated setup to trigger problems in this area though.

> - The only thing that would prevent the migration semantics from being
> correct is if you could manipulate a migrated tunnel in such a way that
> you
> could do something nasty to the source namespace.
>
> Since a tunnel change command is also a tunnel rename command that should
> force any ipip tunnel into using ip addresses from the current
> namespace, which makes it safe, and thus unable to affect the source
> namespace.
>
>
> One example of a safety violation is that in GRE, the key participates in
> routing, and one might be able to set it so that a container sneaks its packets
> into another tunnel that it doesn't own. Don't know if 'ip tunnel' can be used
> to do other bad things.

If there is something like that we should certainly handle the migration and
become an unconfigured tunnel or simply become a non-migratable network device.
That code exists for lo right no.

That means we should probably generate a new instance of the reference tunnel
device in each network namespace.

> There might also be some applications that assume the semantics in which the
> backend and frontend are in the same container. eg. the OSPF daemon of QUAGGA
> for some reason binds to the 'local' endpoint of a GRE tunnel instead of the
> address assigned to the interface, and inside a container, fails with a "No
> such device". We're looking into this right now.

Odd.

> So semantically I believe the tunnel semantics are essentially
> correct. However I believe there are several places where the code
> needs to be updated to correctly implement those semantics.
>
> Eric

I really haven't audited the tunnels in much detail. Which is so it
probably makes sense to make the non-movable until we can update the

code to do the right thing, whatever that is.

Eric

_____

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