Subject: Re: [patch 2/6] [Network namespace] Network device sharing by view
Posted by Herbert Poetzl on Mon, 26 Jun 2006 20:02:25 GMT
View Forum Message <> Reply to Message

On Mon, Jun 26, 2006 at 01:35:15PM -0600, Eric W. Biederman wrote:
> Herbert Poetzl <herbert@13thfloor.at> writes:
>
> > On Mon, Jun 26, 2006 at 10:40:59AM -0600, Eric W. Biederman wrote:
> >> Daniel Lezcano <dlezcano@fr.ibm.com> writes:
> >>
> >> >> Then you lose the ability for each namespace to have its own
> >> >> routing entries. Which implies that you'll have difficulties with
> >> >> devices that should exist and be visible in one namespace only
> >> >> (like tunnels), as they require IP addresses and route.
> >> >
> >> > I mean instead of having the route tables private to the namespace, the
> > routes
> >> > have the information to which namespace they are associated.
> >>
> >> Is this an implementation difference or is this a user visible
> >> difference? As an implementation difference this is sensible, as it is
> >> pretty insane to allocate hash tables at run time.
> >>
> >> As a user visible difference that affects semantics of the operations
> >> this is not something we want to do.
> >
> > well, I guess there are even more options here, for
> > example I'd like to propose the following idea, which
> > might be a viable solution for the policy/isolation
> > problem, with the actual overhead on the setup part
> > not the hot pathes for packet and connection handling
> >
> > we could use the multiple routing tables to provide
> > a single routing table for each guest, which could
> > be used inside the guest to add arbitrary routes, but
> > would allow to keep the 'main' policy on the host, by
> > selecting the proper table based on IPs and guest tags
> >
> > similar we could allow to have a separate iptables
> > chain for each guest (or several chains), which are
> > once again directed by the host system (applying the
> > required prolicy) which can be managed and configured
> > via normal iptable interfaces (both on the guest and
> > host) but actually provide at least to layers
>
> I have real concerns about the complexity of the route you
> have described.
>

> > note: this does not work for hierarchical network
> > contexts, but I do not see that the yet proposed
> > implementations would do, so I do not think that
> > is of concern here ...
>
> Well we are hierarchical in the sense that a parent
> can have a different network namespace then a child.
> So recursive containers work fine. So this is like
> the uts namespace or the ipc namespace rather than
> like the pid namespace.

yes, but you will not be able to apply policy on
the parent, restricting the child networking in a
proper way without jumping through hoops ...

> I really do not believe we have a hotpath issue, if this
> is implemented properly. Benchmarks of course need to be taken,
> to prove this.

I'm fine with proper testing and good numbers here
but until then, I can only consider it a prototype

> There are only two places a sane implementation should show issues.
> - When the access to a pointer goes through a pointer to find
>   that global variable.
> - When doing a lookup in a hash table we need to look at an additional
>   field to verify a hash match. Because having a completely separate
>   hash table is likely too expensive.
>
> If that can be shown to really slow down packets on the hot path I am
> willing to consider other possibilities. Until then I think we are on
> path to the simplest and most powerful version of building a network
> namespace usable by containers.

keep in mind that you actually have three kinds
of network traffic on a typical host/guest system:

 - traffic between unit and outside
   - host traffic should be quite minimal
   - guest traffic will be quite high

 - traffic between host and guest
   probably minimal too (only for shared services)

 - traffic between guests
   can be as high (or even higher) than the
   outbound traffic, just think web guest and
   database guest

> The routing between network namespaces does have the potential to be
> more expensive than just a packet trivially coming off the wire into a
> socket.

IMHO the routing between network namespaces should
not require more than the current local traffic
does (i.e. you should be able to achieve loopback
speed within an insignificant tolerance) and not
nearly the time required for on-wire stuff ...

> However that is fundamentally from a lack of hardware. If the
> rest works smarter filters in the drivers should enable to remove the
> cost.
>
> Basically it is just a matter of:
> if (dest_mac == my_mac1) it is for device 1.
> If (dest_mac == my_mac2) it is for device 2.
> etc.

hmm, so you plan on providing a different MAC for
each guest? how should that be handled from the
user PoV? you cannot simply make up MACs as you
go, and, depending on the network card, operation
in promisc mode might be slower than for a given
set (maybe only one) MAC, no?

> At a small count of macs it is trivial to understand it will go
> fast for a larger count of macs it only works with a good data
> structure. We don't hit any extra cache lines of the packet,
> and the above test can be collapsed with other routing lookup tests.

well, I'm absolutely not against flexibility or
full virtualization, but the proposed 'routing'
on the host effectively doubles the time the
packet spends in the network stack(s), so I can
not believe that this approach would not add
(significant) overhead to the hot path ...

best,
Herbert

> Eric