Subject: Re: [patch 2/6] [Network namespace] Network device sharing by view
Posted by Herbert Poetzl on Tue, 27 Jun 2006 23:07:23 GMT
View Forum Message <> Reply to Message

On Tue, Jun 27, 2006 at 10:29:39AM -0600, Eric W. Biederman wrote:
> Herbert Poetzl <herbert@13thfloor.at> writes:
>
> > On Tue, Jun 27, 2006 at 01:54:51PM +0400, Kirill Korotaev wrote:
> >> >>My point is that if you make namespace tagging at routing time, and
> >> >>your packets are being routed only once, you lose the ability
> >> >>to have separate routing tables in each namespace.
> >> >
> >> >
> >> >Right. What is the advantage of having separate the routing tables ?
> >
> >> it is impossible to have bridged networking, tun/tap and many other
> >> features without it. I even doubt that it is possible to introduce
> >> private netfilter rules w/o virtualization of routing.
> >
> > why? iptables work quite fine on a typical linux
> > system when you 'delegate' certain functionality
> > to certain chains (i.e. doesn't require access to
> > _all_ of them)
> >
> >> The question is do we want to have fully featured namespaces which
> >> allow to create isolated virtual environments with semantics and
> >> behaviour of standalone linux box or do we want to introduce some
> >> hacks with new rules/restrictions to meet ones goals only?
> >
> > well, soemtimes 'hacks' are not only simpler but also
> > a much better solution for a given problem than the
> > straight forward approach ...
>
> Well I would like to see a hack that qualifies.

> I watched the linux-vserver irc channel for a while and almost
> every network problem was caused by the change in semantics
> vserver provides.

the problem here is not the change in semantics compared
to a real linux system (as there basically is none) but
compared to _other_ technologies like UML or QEMU, which
add the need for bridging and additional interfaces, while
Linux-VServer only focuses on the IP layer ...

> In this case when you allow a guest more than one IP your hack
> while easy to maintain becomes much more complex.

why? a set of IPs is quite similar to a single IP (which
is actually a subset), so no real change there, only
IP_ANY means something different for a guest ...

> Especially as you address each case people care about one at a time.

hmm?

> In one shot this goes the entire way. Given how many people miss that
> you do the work at layer 2 than at layer 3 I would not call this the
> straight forward approach. The straight forward implementation yes,
> but not the straight forward approach.

seems I lost you here ...

> > for example, you won't have multiple routing tables
> > in a kernel where this feature is disabled, no?
> > so why should it affect a guest, or require modified
> > apps inside a guest when we would decide to provide
> > only a single routing table?
> >
> >> From my POV, fully virtualized namespaces are the future.
> >
> > the future is already there, it's called Xen or UML, or QEMU :)
>
> Yep. And now we need it to run fast.

hmm, maybe you should try to optimize linux for Xen then,
as I'm sure it will provide the optimal virtualization
and has all the features folks are looking for (regarding
virtualization)

I thought we are trying to figure a light-weight subset
of isolation and virtualization technologies and methods
which make sense to have in mainline ...

> >> It is what makes virtualization solution usable (w/o apps
> >> modifications), provides all the features and doesn't require much
> >> efforts from people to be used.
> >
> > and what if they want to use virtualization inside
> > their guests? where do you draw the line?
>
> The implementation doesn't have any problems with guests inside
> of guests.
>
> The only reason to restrict guests inside of guests is because
> the we aren't certain which permissions make sense.

well, we have not even touched the permission issues yet

best,
Herbert

> Eric