Subject: Re: [RFD] L2 Network namespace infrastructure
Posted by ebiederm on Sat, 23 Jun 2007 17:55:09 GMT
View Forum Message <> Reply to Message

Stephen Hemminger <shemminger@linux-foundation.org> writes:

> On Sat, 23 Jun 2007 08:20:40 -0700
> Ben Greear <greearb@candelatech.com> wrote:
>
>> Patrick McHardy wrote:
>> > Eric W. Biederman wrote:
>> >
>> >> -- The basic design
>> >>
>> >> There will be a network namespace structure that holds the global
>> >> variables for a network namespace, making those global variables
>> >> per network namespace.
>> >>
>> >> One of those per network namespace global variables will be the
>> >> loopback device. Which means the network namespace a packet resides
>> >> in can be found simply by examining the network device or the socket
>> >> the packet is traversing.
>> >>
>> >> Either a pointer to this global structure will be passed into
>> >> the functions that need to reference per network namespace variables
>> >> or a structure that is already passed in (such as the network device)
>> >> will be modified to contain a pointer to the network namespace
>> >> structure.
>> >>
>> >
>> >
>> > I believe OpenVZ stores the current namespace somewhere global,
>> > which avoids passing the namespace around. Couldn't you do this
>> > as well?
>
> Maybe the current namespace should be attached to something else
> like sysfs root? Having multiple namespace indirection possiblities
> leads to interesting cases where current namespace is not correctly
> associated with current sysfs tree or current proc tree, ...

Yes. There are some oddities there.

In my current tree there is code that makes proc and sysfs match the inspecting
process.

I haven't quite solved the inspection problem where we want to look at
the namespace of a different process. But as long as we have clean
code to do the basics that isn't a big leap when we come to it.

I'm not really seeing any problems along this line at this point.

The big problem at this point is code review and merging, and in particular breaking this work up into small enough pieces that they can be digested, successfully code reviewed and merged.

Eric

____