Subject: Re: Re: Network virtualization/isolation
Posted by Daniel Lezcano on Thu, 30 Nov 2006 16:38:16 GMT
View Forum Message <> Reply to Message

Vlad Yasevich wrote:
> Daniel Lezcano wrote:
>> Brian Haley wrote:
>>> Eric W. Biederman wrote:
>>>> I think for cases across network socket namespaces it should
>>>> be a matter for the rules, to decide if the connection should
>>>> happen and what error code to return if the connection does not
>>>> happen.
>>>>
>>>> There is a potential in this to have an ambiguous case where two
>>>> applications can be listening for connections on the same socket
>>>> on the same port and both will allow the connection. If that
>>>> is the case I believe the proper definition is the first socket
>>>> that we find that will accept the connection gets the connection.
>> No. If you try to connect, the destination IP address is assigned to a
>> network namespace. This network namespace is used to leave the listening
>> socket ambiguity.
>>> Wouldn't you want to catch this at bind() and/or configuration time and
>>> fail? Having overlapping namespaces/rules seems undesirable, since as
>>> Herbert said, can get you "unexpected behaviour".
>> Overlapping is not a problem, you can have several sockets binded on the
>> same INADDR_ANY/port without ambiguity because the network namespace
>> pointer is added as a new key for sockets lookup, (src addr, src port,
>> dst addr, dst port, net ns pointer). The bind should not be forced to a
>> specific address because you will not be able to connect via 127.0.0.1.
>
> So, all this leads to me ask, how to handle 127.0.0.1?
>
> For L2 it seems easy. Each namespace gets a tagged lo device.
> How do you propose to do it for L3, because disabling access to loopback is
> not a valid option, IMO.

There are 2 options:

1 - Dmitry Mishin proposed to use the l2 mechanism and reinstantiate a new loopback device, I didn't tested that yet, perhaps there are issues with non-127.0.0.1 loopback traffic and routes creation, I don't know.

2 - add the pointer of the network namespace who has originated the packet into the skbuff when the traffic is for 127.0.0.1, so when the packet arrive to IP, it has the namespace destination information because source == destination. I tested it and it works fine without noticeable overhead and this can be done with a very few lines of code.

-- Daniel