Subject: SIOCGIFADDR fails in container
Posted by mose on Wed, 24 Oct 2012 21:38:57 GMT
View Forum Message <> Reply to Message

Debian 6.0.6
Kernel 2.6.32-5-openvz-amd64
vzctl 3.0.24

Container 270 (spaced for readability)
NETIF="
 ifname=eth0,bridge=vzbr0,mac=00:18:51:F4:B3:1D,
  host_ifname=veth270.0,host_mac=00:18:51:E4:88:29,mac_filter=off;
 ifname=eth1,bridge=vzbr1,mac=00:18:51:B0:95:9D,
  host_ifname=veth270.1,host_mac=00:18:51:CB:DE:3D,mac_filter=off"

I am attempting to use perl modules in a container, such as Net::RawIP and Net::Frame::Device, but they are failing when they attempt to query any interface for its MAC address (lo, eth0 or eth1).

A similar problem is described in the post below.

 comments.gmane.org/gmane.linux.kernel.containers.lxc.general /238

A program is included in the post to demonstrate the problem. I can run the program on the host, and if the interface has an IP address, it will return it, but it does not find the MAC address, for some reason. For example,

> ./a.out vzbr0
Getting IP / MAC info for device (vzbr0)
Got IP (167777507) and MAC ()

If I run the program in the container,

./a.out eth0
Getting IP / MAC info for device (eth0)
SIOCGIFADDR failed with code (22) : Invalid argument

Since a patch was submitted as a result of that post, and it was 2 years ago, I'm assuming that the patch is in the kernel I am using. I looked through Net::RawIP and saw that it was using tap to get its information. On the chance this somehow involved the tun device, I followed the steps to make the tun device accessible in the container as described in the wiki.

wiki.openvz.org/VPN_via_the_TUN/TAP_device

Unfortunately, that didn't seem to affect anything. Is there anything else I need to turn on for SIOCGIFADDR and SIOCGIFHWADDR to work, or is this just not possible from a container? ip/ifconfig must have a way to do this, because they display the MAC address for the interface. Any pointers to more information, debugging tips or things to try would be greatly appreciated.

Thanks!

Subject: Re: SIOCGIFADDR fails in container
Posted by mose on Mon, 29 Oct 2012 22:02:12 GMT
View Forum Message <> Reply to Message

Just to provide some closure to this post, there are other ways to determine the MAC address of an interface. I am now using Net::Frame and its siblings, including Net::Frame::Device. Net::Frame::Device is built on the dumb networking library (libdumbnet on Debian where libdnet conflicts with the DECNet library and libdumb conflicts with the dynamic universal music bibliotheque). Of course, now that I have found a successful way to do this, it is no longer needed for the direction this project is going.

Net::Frame::Device uses Net::Libdnet and Net::Libdent6. Unfortunately, several installation tests in the Net::Libdnet and Net::Libdnet6 perl modules fail, because libdumbnet cannot figure out the default path out of the container, at least in this configuration. So, the installation has to be forced. The container has eth0 and eth1. There is only one default gateway, which is for eth0.

I haven't looked at the source code for libdumbnet to see what it is trying to do. All of the other calls seem to work OK. At this point, at least, it is not an issue for what I'm trying to do. I'm happy, and I'm moving on.