WeGet letters

by Michael W Lucas

Dear Computer Touching Advice Columnist,

FreeBSD's jails have a few different ways to network. How do I decide which to use?

—Baffled and Distressed

Dear BAD,

I'm writing this at BSDCan 2025. Two stories above us, someone is using a hammer drill. Traversing the concrete building shifts the vibration so that it exactly matches the resonance frequency of human teeth. Turns out Mister Hammer Drill needed to demonstrate that he was doing work so that he'd make his fifty bucks for a day's labor. I gave him sixty bucks to knock it off and go home. Do not fear, this unexpected expense did not impact my gelato fund; I embezzled the cash from the con.

This is my problem only because I failed to sidestep quickly enough and they made me con chair, which shows that certain parts of the community need better methods of leader selection. Being drafted has left me baffled and distressed.

The point is, that we are all baffled and distressed. We're merely baffled and distressed about different things.

Jail networking is one of those topics that seems simple until you touch it. No matter what virtualization system you use, jails or bhyve or some lesser system, you face similar decisions. Jails can either share their host's networking stack or have their own stack. On a shared stack, a jail can have its own address or be bound to a private address on a loopback interface and access the outside world via NAT. You could give a jail a full network stack with VNET. All are disasters, but each disaster possesses unique qualities that might make it preferable.

Jail networking is one of those topics that seems simple until you touch it.

Remember, a jail is a lie.

A jail thinks it is a full operating system and expects to have full control over whatever network it accesses. If you assign an IP (public or private) to a jail, the jail thinks it fully con-

trols that address. The jail's `rc.conf` includes no information on configuring interfaces. With a shared stack, the host can stomp all over the jail's networking. Want to run sshd(8) on your jails? By default, the SSH daemon listens on every IP address attached to the system. You must reconfigure it to change that behavior. That's not hard, but it's easily overlooked. Similarly, if you want multiple jails running the same database server? They can't all use the same port and IP address. The amount of administrative tracking you must perform is directly proportional to the number of jails sharing that network stack.

Given all this, why not give each jail its own networking stack? Because it's more complicated. You must create a bridge on the host and then create a virtual interface for every jail. One end of the interface goes into the jail; the other gets plugged into the bridge. Every complication is a bright shiny chance for failure.

The good part of VNET is that each jail truly controls its IP addresses. You must set it in the jail's `rc.conf` and can set up each service to monopolize that address.

A VNET jail isn't exactly like a host. The most common interface is epair(4). You cannot capture packets on an epair; if you want to debug the network or capture netflow, you must do so from the host. You might think of netflow as a network administrator thing, but that's not true. Netflow solves nothing but identifies everything. If I had written my netflow book using nfdump rather than flow tools, it would still be useful, but I have a talent for poor choices such as making a career in system administration. I'm not saying that a knowledge of promiscuous mode, fdisk, or window scaling leads to misery, but I will say that happy folks have no understanding of any of them. Anyway, you don't get promiscuous mode on an epair interface. Plan ahead.

Both shared stack jails and VNET jails can use private or public addresses. If you use a private address, the jail can't access the outside world without outbound NAT. Incoming connections must be mapped through the NAT. As per Rule of System Administration number 19, "the purpose of any private addressing scheme is to increase administrative overhead."

You might reasonably ask what I chose to do. My hosting environment has a limited number of public IPv4 addresses and an entire IPv6 /64. I could use a bastardization NAT/direct access system, but, instead, I declare that certain jails are IPv6-only and can only access the modern parts of the Internet. I have no problem declaring that certain people are unworthy of connecting to my private package server that nobody outside my /64 can download from.

Using only public addresses is much easier. Unfortunately, IPv4 addresses currently retail for about $55 USD each with a minimum order of 256, and I can't embezzle enough because BSDCan doesn't have that kind of money.

Anyone know how EuroBSDCon's finances are? Do they need a con chair? My new assistant is great with a hammer drill.

Have a question for Michael?
Send it to letters@freebsdjournal.org

Despite his best efforts, **MICHAEL W LUCAS** (https://mwl.io) has been unable to persuade the *FreeBSD Journal* board to fire him from this column. Yet. His most recent books are *Run Your Own Mail Server* and *Laserblasted*. His next book is a second edition of *Networking for System Administrators*, at https://mwl.io/n4sa2e