

RE: [PATCH] Add ioctl to disable bpf timestamping

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/net/2004-09/0090.html>

From: Ed Maste (emaste_at_sandvine.com)

Date: 09/08/04

Date: Wed, 8 Sep 2004 11:40:17 -0400

To: <freebsd-net@FreeBSD.org>, <tcpdump-workers@tcpdump.org>

BMS wrote:

- > *Here's a patch against 5.3 to add a per-instance switch which allows*
- > *the user to specify if captured packets should be timestamped (and,*
- > *if so, whether microtime() or the faster but less accurate*
- > *getmicrotime() call should be used).*

We've implemented this internally on 4.7, and have seen quite impressive results. I have a test case that sends 512 byte packets and has the snap length set to get the whole packet. Using microtime(), I am able to get about 120 kpps to my test app. With no timestamp I can get 200 kpps. Without context the absolute numbers don't mean much but the relative improvement is quite impressive.

Guy Helmer wrote:

- > *I like the idea (I've been using a hack to call getmicrotime()*
- > *in bpf in my own kernels), but I wonder if it would be better as a*
- > *sysctl? Then it wouldn't require changes to libpcap and/or tcpdump,*
- > *and would work with any application.*

I think an ioctl is the right way to do it, since you could have multiple BPF listeners with different requirements. For example, realtime inspection like Snort may not care about timestamps while manual inspection with tcpdump would.

One way to allow the user to control this behaviour on a per-application basis would be to have libpcap check an env var in order to decide if the ioctl should be set.

Ed Maste
Sandvine Inc.

freebsd-net@freebsd.org mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-net>

To unsubscribe, send any mail to "freebsd-net-unsubscribe@freebsd.org"