Suricata - Support #1787
FreeBSD netmap: [ERRCODE: SC_ERR_NETMAP_READ(264)] - Invalid polling request
05/20/2016 11:16 AM - Victor Julien

Status: Closed
Priority: Normal
Assignee: Aleksey Katargin
Category:
Affected Versions:
Label:

Description
This may be a user issue, as I'm not very familiar with FreeBSD & netmap.

I'm running FreeBSD 10.3 with netmap on a 2 core system. So I would expect 2 worker threads.

netmap:
  # To specify OS endpoint add plus sign at the end (e.g. "eth0+")
  - interface: igb0
    # Number of receive threads. "auto" uses number of RSS queues on interface.
    threads: auto

I start with:

./src/suricata --netmap -v -l /tmp/ -c suricata.yaml

Immediately it starts giving this 'invalid polling request error':

[100147] 20/5/2016 -- 16:04:50 - (util-device.c:218) <Info> (LiveBuildDeviceListCustom) -- Adding interface igb0 from config file
[100147] 20/5/2016 -- 16:04:50 - (runmode-netmap.c:194) <Info> (ParseNetmapConfig) -- Using 1 threads for interface igb0
[100147] 20/5/2016 -- 16:04:50 - (util-runmodes.c:321) <Info> (RunModeSetLiveCaptureWorkersForDevice) -- Going to use 1 thread(s)
[100147] 20/5/2016 -- 16:04:50 - (runmode-netmap.c:194) <Info> (ParseNetmapConfig) -- Using 1 threads for interface igb0
[100147] 20/5/2016 -- 16:04:50 - (util-runmodes.c:321) <Info> (RunModeSetLiveCaptureWorkersForDevice) -- Going to use 1 thread(s)
[100147] 20/5/2016 -- 16:04:50 - (flow-manager.c:720) <Info> (FlowManagerThreadSpawn) -- using 1 flow manager threads
[100147] 20/5/2016 -- 16:04:50 - (flow-manager.c:884) <Info> (FlowRecyclerThreadSpawn) -- using 1 flow recycler threads
[100147] 20/5/2016 -- 16:04:50 - (tm-threads.c:2164) <Notice> (TmThreadWaitOnThreadInit) -- all 2 packet processing threads, 4 management threads initialized, engine started.

AFAICS there should be 2 RSS queues:

# dmesg |grep igb
igb0: <Intel(R) PRO/1000 Network Connection, Version = 2.5.3-k> port 0x2000-0x201f mem 0xe2e00000-
0xe2e1ffff,0xe2e40000-0xe2e43fff irq 16 at device 0.0 on pci
igb0: Using MSIX interrupts with 3 vectors
igb0: Ethernet address: 90:e2:ba:01:7c:38
igb0: Bound queue 0 to cpu 0
igb0: Bound queue 1 to cpu 1
igb0: netmap queues/slots: TX 2/1024, RX 2/1024
igb0: Using MSIX interrupts with 3 vectors
igb0: Ethernet address: 90:e2:ba:01:7c:38
igb0: Bound queue 0 to cpu 0
igb0: Bound queue 1 to cpu 1
igb0: link state changed to UP
igb0: permanently promiscuous mode enabled

If I set 'threads: 1' everything works.

History
#1 - 05/20/2016 11:19 AM - Victor Julien
Actually, looking at Suricata's output above something seems wrong with the runmode setup. It doesn't set up 2 worker mode, but a 2 x 1 worker somehow? Not sure if the effect is different, but this output at least looks suspicious to me.

[100147] 20/5/2016 -- 16:04:50 - (runmode-netmap.c:194) <Info> (ParseNetmapConfig) -- Using 1 threads for interface igb0
[100147] 20/5/2016 -- 16:04:50 - (util-runmodes.c:321) <Info> (RunModeSetLiveCaptureWorkersForDevice) -- Going to use 1 thread(s)
Total memory 2461696
[100147] 20/5/2016 -- 16:04:50 - (runmode-netmap.c:194) <Info> (ParseNetmapConfig) -- Using 1 threads for interface igb0
[100147] 20/5/2016 -- 16:04:50 - (util-runmodes.c:321) <Info> (RunModeSetLiveCaptureWorkersForDevice) -- Going to use 1 thread(s)

#2 - 05/25/2016 01:29 AM - Aleksey Katargin
Latest master source spawn duplicate worker threads.

netmap:
  # To specify OS endpoint add plus sign at the end (e.g. "eth0+")
  - interface: eth2
  # Number of receive threads. "auto" uses number of RSS queues on interface.
  threads: auto
  # You can use the following variables to activate netmap tap or IPS mode.
  # If copy-mode is set to ips or tap, the traffic coming to the current
  # interface will be copied to the copy-iface interface. If 'tap' is set, the
  # copy is complete. If 'ips' is set, the packet matching a 'drop' action
  # will not be copied.
  # To specify the OS as the copy-iface (so the OS can route packets, or forward
  # to a service running on the same machine) add a plus sign at the end
  # (e.g. "copy-iface: eth0+"). Don't forget to set up a symmetrical eth0+ -> eth0
  # for return packets. Hardware checksumming must be 'off' on the interface if
  # using an OS endpoint (e.g. 'ifconfig eth0 -rxcsum -txcsum -rxcsum6 -txcsum6' for FreeBSD
  # or 'ethtool -K eth0 tx off rx off' for Linux).
  copy-mode: tap
copy-iface: eth3
  # Set to yes to disable promiscuous mode
  disable-promisc: no
  # Choose checksum verification mode for the interface. At the moment
  # of the capture, some packets may be with an invalid checksum due to
  # offloading to the network card of the checksum computation.
  # Possible values are:
  # - yes: checksum validation is forced
  # - no: checksum validation is disabled
  # - auto: suricata uses a statistical approach to detect when
  # checksum off-loading is used.
  # Warning: 'checksum-validation' must be set to yes to have any validation
  # checksum-checks: auto
  # BPF filter to apply to this interface. The pcap filter syntax apply here.
  #bpf-filter: port 80 or udp

06/12/2020
# Breakpoint 3, RunModeSetLiveCaptureWorkers (ConfigParser=0x5344a3 <ParseNetmapConfig>, ModThreadsCount=0x5354bd <NetmapConfigGetThreadsCount>, recv_mod_name=0x601ea5 "ReceiveNetmap", decode_mod_name=0x601e98 "DecodeNetmap", thread_name=0x602a53 "W", live_dev=0x0)
   at util-runmodes.c:374

374        int nlive = LiveGetDeviceCount();
(gdb) n
378        for (ldev = 0; ldev < nlive; ldev++) {
(gdb) print nlive
$1 = 4
(gdb)

#3 - 06/01/2016 07:14 AM - Victor Julien

- Status changed from New to Closed

Thanks Aleksey, that was the issue indeed. Fixed in #1768 (https://github.com/inliniac/suricata/pull/2128)