

# vPF\_RING User Guide

High Speed Packet Capture On Virtual Machines

Version 1.0.0  
July 2011

© 2011 ntop.org

# 1. Table of Contents

Introduction .....	3
What's New with vPF_RING User's Guide? .....	3
Welcome to vPF_RING .....	3
vPF_RING Installation.....	5
Host side.....	5
<b>Installation Prerequisites .....</b>	<b>5</b>
<b>Patched QEMU Installation .....</b>	<b>5</b>
Guest side.....	5
<b>vNPlug Kernel Module Installation .....</b>	<b>6</b>
<b>vPF_RING Library Installation.....</b>	<b>6</b>
Using vPF_RING .....	7
Checking PF_RING Device Configuration.....	7
Libpfiring and Libpcap .....	7

## 2. Introduction

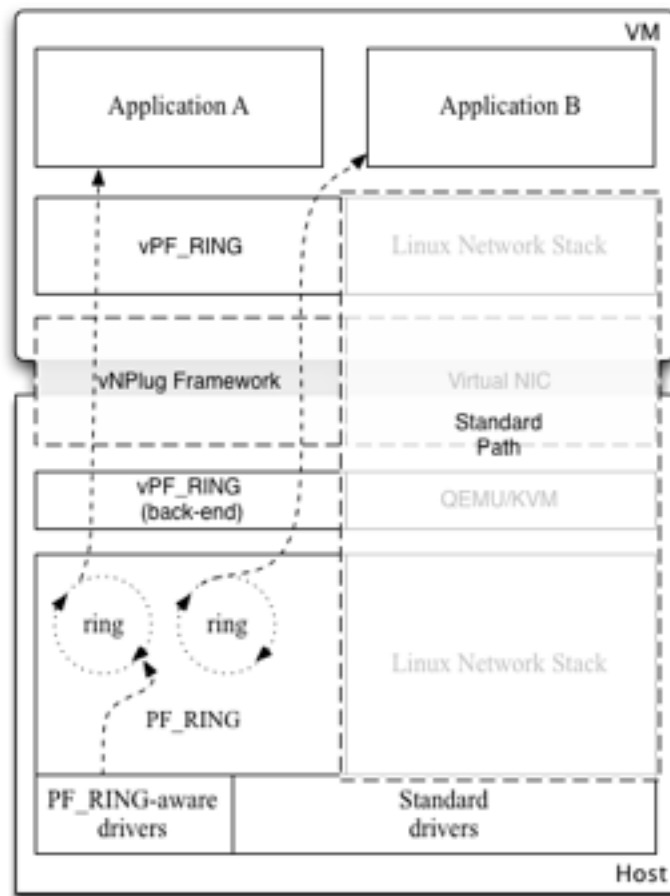
vPF\_RING is a high speed packet capture framework that turns a Virtual Machine running on a commodity PC into an efficient network measurement box.

### 2.1. What's New with vPF\_RING User's Guide?

- Release 1.0 (July 2011)
  - Initial vPF\_RING users guide.

## 3. Welcome to vPF\_RING

vPF\_RING's architecture is depicted in the figure below.



The main building blocks from the bottom are:

- Specialized PF\_RING-aware drivers (optional) (host side) that allow to further enhance packet capture by efficiently copying packets from the driver to PF\_RING without passing through the kernel data structures. For further information please refer to the PF\_RING User's Guide.
- The standard PF\_RING kernel module (host side).

- The vPF\_RING backend (host side), that interacts with the standard PF\_RING module on the host side, and the user-space library on the guest side by means of the vNPlug Framework.
- The vNPlug Framework (host and guest side), that provides a direct mapping of the PF\_RING memory structures on the guest and a reliable communication channel. This framework comes as a QEMU patch on the host side, and a kernel module on the guest side.
- The user-space vPF\_RING library that provides transparent PF\_RING-support to user-space applications on the VM.

Incoming packets are copied by the kernel module on the host side into a memory ring allocated at creation time, and directly read by the user-space applications on the VM.

Applications can issue standard PF\_RING API calls, described in the PF\_RING User's Guide.

## 4. vPF\_RING Installation

Download vPF\_RING as explained in [http://www.ntop.org/products/pf\\_ring/vpf\\_ring](http://www.ntop.org/products/pf_ring/vpf_ring)

The vPF\_RING source code layout is the following:

- README
- guest/
- host/

### 4.1. Host side

#### Installation Prerequisites

The vPF\_RING installation expects that PF\_RING is compiled and installed on the host system.

```
host $ cd <PF_RING>/kernel
host $ make
host # make install
host # insmod pf_ring.ko
host $ cd <PF_RING>/userland/lib
host $ ./configure
host $ make
host # make install
```

Note: if you want to use a PF\_RING-aware drivers with `transparent_mode` or other settings, please refer to the PF\_RING User's Guide.

#### Patched QEMU Installation

Compile and install the patched QEMU.

```
host $ cd <VPF_RING>/host
host $ ./configure
host $ make
host # make install
```

### 4.2. Guest side

In order to run the VM with vNPlug/vPF\_RING support use the “-device vnplug” parameter. Remember to enable the Virtualization support in your BIOS (required by KVM).

```
host # modprobe kvm_intel
host # /usr/local/kvm/bin/qemu-system-x86_64 \
    -hda ubuntu-amd64.img \
    -boot c \
    -m 512 \
    -vnc 0.0.0.0:0 \
    -device vnplug
```

Note: when using the preconfigured VM, you can skip the next sub-sections.

## vNPlug Kernel Module Installation

In order for the vNPlug framework to work properly, you should load the acpihp kernel module (hotplug support), otherwise it won't be able to dynamically map ring memory.

```
guest # modprobe acpihp
```

Compile and install the vNPlug kernel module.

```
guest $ cd <VPF_RING>/guest/kernel
guest $ make
guest # make headers_install
guest # insmod vnplug.ko
```

## vPF\_RING Library Installation

Copy the VPF\_RING user-space module.

```
guest $ cp libpfiring_mod_virtual_XXX.a <PF_RING>/userland/lib/libs/
guest $ cp pfring_mod_virtual.h <PF_RING>/userland/lib/
```

Compile and install the PF\_RING library with VPF\_RING support

```
guest $ cd <PF_RING>/userland/lib
guest $ ./configure
guest $ make
guest # make install
```

Example: compile and run pfcount

```
guest $ cd <PF_RING>/userland/examples
guest $ make pfcount
guest # ./pfcount -i host:eth0
```

## 5.Using vPF\_RING

Before using any PF\_RING application the pf\_ring kernel module should be loaded on the host side.

```
host # insmod <PF_RING>/kernel/pf_ring.ko
```

Note: if you want to use a PF\_RING-aware drivers with transparent\_mode or other settings, please refer to the PF\_RING User's Guide.

On the guest side both the standard hotplug module and the vnplug module should be loaded.

```
guest # modprobe acpihp  
guest # insmod vnplug.ko
```

### 5.1. Checking PF\_RING Device Configuration

As with standard PF\_RING, when a ring is activated a new entry /proc/net/pf\_ring is created on the host.

```
host # cat /proc/net/pf_ring/info  
Version      : 4.7.1  
Ring slots   : 4096  
Slot version  : 13  
Capture TX   : Yes [RX+TX]  
IP Defragment : No  
Socket Mode  : Standard  
Transparent mode : Yes (mode 0)  
Total rings   : 0  
Total plugins : 2
```

### 5.2. Libpfiring and Libpcap

As vPF\_RING results in a standard PF\_RING module which is hidden by the PF\_RING API, both libpfiring and libpcap can be compiled and used as described in the PF\_RING User's Guide.

Note: in order to indicate to the library to use the vPF\_RING module, you need to prepend 'host:' to the device name (e.g. host:ethX@Y).