Linux KVM Virtual Traffic Monitoring

East-West traffic visibility
Scott Harvey

Director of Engineering at APCON
Responsible for teams located in Oregon and Texas covering:
• Datacenter monitoring systems
• Virtual network monitoring
• Traffic capture and applications
• Security bypass tap/switch
• Optical network taps
Session Topic: Linux KVM traffic monitoring

Monitoring Virtualized Networks

Goal: A deeper understanding of how data moves in virtualized environments and how to monitor them.

- More data center traffic is virtualized
- Let’s discuss how virtualized networks work
- And east-west traffic monitoring technologies
Linux KVM Virtual Traffic Monitoring

Agenda

- About APCON
- Virtualization value proposition
- Types of virtualization
- Types of virtual switching
- The orchestration layer
- Challenges of monitoring virtualized networks
- Virtual monitoring techniques
- Example virtual monitoring solution
About APCON

Global Network Monitoring Leader

• Stable 20+ year old private company
• Fortune 1000 customer base
• Virtual and physical network monitoring
• High availability: enterprise class architecture
  • Small 1RU up to large 504 ports in 14RU chassis
  • Virtual network monitoring solutions
• Global deployments in 40 countries
What is Virtualization?

- Virtualization software is enabling several operating systems and applications to run on one physical server or “host.”

- Each self-contained virtual machine (VM) is isolated from the others, and uses as much of the host’s computing resources as is configured.
What does it look like?

Virtualization Defined
For those more visually inclined...

Traditional Architecture

Virtual Architecture
Why is it important?

Linux and Microsoft Virtualization Trends

Figure 5. Polls Asking "Which Vendor/Framework Will Be the Most Important to Your Private Cloud Strategy?" December 2011 and December 2014, Gartner Data Center Conference, Las Vegas

Source: Gartner (March 2015)
Virtualization value proposition

- Up to 80 percent greater utilization of every server or “host”
- Reductions in hardware by a ratio of 10:1 or better
- Capital and operations expenses cut by half, with annual savings of more than $1,500 for each server virtualized
- Robust, affordable high availability

Source: VMware®
https://www.vmware.com/products/vsphere-operations-management/gettingstarted
Linux KVM Virtual Traffic Monitoring

Types of virtualization

• Brand variety

• Type 1 vs type 2 hypervisors
Virtual Network Bridge

What is it?

• A virtual network bridge is software that enables multiple VM’s to share a physical NIC.

Benefits:

• A simple way for VM applications to access and share the LAN
Virtual Switch/Bridge

What is it?

• A virtual switch (vswitch) or bridge is a software application that allows communication between virtual machines as well as the physical network

Benefits:

• Helps ease deployment of virtual machines, and provides similar capabilities of physical Ethernet switches

• Provides standard Ethernet based intelligent communication between virtual machines and the physical network
Virtual Distributed Switch

What is it?

• Centralizes network provisioning, administration and monitoring using data center-wide network aggregation

Benefits:

• Aggregation of per-host virtual switches presented and controlled as a single distributed switch through single configuration at the Datacenter level

• The vDS abstracts configuration of individual virtual switches and enables centralized provisioning, administration, and monitoring
What does it look like?

Open vSwitch

VMware vDS
Linux KVM Virtual Traffic Monitoring

The orchestration layer

- The orchestration “layer” allows for planned automation and provisioning tasks within a virtual environment

- **Commercial:**
  - CSC ServiceMesh Agility
  - Flexiant
  - IBM SmartCloud
  - HP Operations Orchestration
  - VMware vCenter Orchestrator
  - Oracle Nimbula

- **Open-Source:**
  - Abiquo
  - CloudStack
  - Eucalyptus
  - OpenStack
  - Puppet / Chef
The Visibility Challenge – Blind Spots!

**Before Migration**

- Web and SQL apps are heavy talkers
- Orchestration migrates SQL app
- Migrating top talkers increases efficiency

**After Migration**

- Blind spots created
- Intra-VM is significant traffic
- Requires virtual monitoring

Heavy Traffic

Blind Spot!
The Enterprise Virtualization Challenge

- Intra-VM network traffic is a blind spot
- Elasticity of virtual environment complicates monitoring
- Complexity to monitor any traffic, across an evolving network, and ensure diagnostic tool compatibility
- Lack of unified virtual and physical monitoring
- Tool manufacturers provide proprietary, vendor specific solution
Virtual monitoring techniques

Let’s discuss three approaches

1. Use vswitch tap/mirror
2. Custom VM or agent application
3. Use native traffic control
Use vswitch tap/mirror

Most vswitch’s include basic tap/mirror function that can replicate all traffic to a SPAN/ER-SPAN port

PROS:
• Simple approach to use vswitch SPAN/ER-SPAN port

CONS:
• Mirroring all VM traffic across LAN maybe too much bandwidth
• Replicating traffic burdens vswitch and server performance
• Mirroring many VMs may not scale due to bandwidth
Linux KVM Virtual Traffic Monitoring

Custom VM or agent application

Installing a vendor specific VM monitoring application or software agent on every server is a common approach.

PROS:
• Custom application on the server assists with monitoring
• Can reduce backhaul traffic using filters or sending meta data

CONS:
• Consumes high cost server VM, processor and memory resources
• Increases maintenance across all servers
• Often not practical to install custom VMs on all servers
• Server application may not have network-wide visibility
Linux KVM Virtual Traffic Monitoring

Use native traffic control

Linux KVM has built in traffic control functions that can be accessed to setup virtual taps and traffic filters

PROS:
• Requires no dedicated server resource; minimal vswitch impact
• Direct efficient access using standard Linux KVM files
• Filter traffic-of-interest for monitoring
• Reduces LAN bandwidth used for monitoring

CONS:
• Requires some advanced knowledge of Linux CLI tools
There are two types of Linux soft switches
  - Linux bridge
  - OpenVSwitch

For every virtual guest port, a “vnet” tap port is created. Each tap port is accessible via the hypervisor.

Linux kernel network control mechanisms can be used to steer traffic from internal to external interfaces.

GRE encapsulation can be used to route traffic to remote endpoint.
Linux KVM Virtual Machine Example

- Virtual server with 4 virtual machines
  1. Production Network – Internal network within the server. “East <-> West” traffic not visible from traditional monitoring tools
  2. Maintenance Network – NAT behind physical interface EM1
  3. A network interface for a GRE tunnel to tap select traffic from internal networks
### How does this look on CLI?

- **When a VM is initiated virtual interfaces are created**

```bash
virsh domiflist rhel7.0

<table>
<thead>
<tr>
<th>Interface</th>
<th>Type</th>
<th>Source</th>
<th>Model</th>
<th>MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>vnet0</td>
<td>bridge</td>
<td>VMmaint</td>
<td>virtio</td>
<td>52:54:00:7f:81:6b</td>
</tr>
<tr>
<td>vnet1</td>
<td>network</td>
<td>VMPro</td>
<td>virtio</td>
<td>52:54:00:8d:20:b7</td>
</tr>
</tbody>
</table>

brctl show (Linux bridge)

<table>
<thead>
<tr>
<th>bridge name</th>
<th>bridge id</th>
<th>STP enabled</th>
<th>interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMPro</td>
<td>8000.5254001cbbc4f</td>
<td>yes</td>
<td>VMPro-nic</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>vnet1</td>
</tr>
<tr>
<td>VMmaint</td>
<td>8000.782bb04183e</td>
<td>no</td>
<td>em1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>vnet0</td>
</tr>
</tbody>
</table>
```
Linux bridge – create GRE tunnel

- Gretap is used to encapsulate layer 2 frames for traffic forwarding and is supported on all Linux distributions.
- Requires local IP and remote IP. GRE Key, TTL optional
- CLI commands to setup a GRE encapsulation:

```bash
ip link add l2gre0 type gretap \ 
remote 172.20.16.11 \ 
local 172.20.15.11 key 1
ip link set l2gre0 up
```
Traffic control (TC) is a Linux utility to tap and filter vnet interfaces.

Example: mirror and forward all traffic from vnet0 to tunnel l2gre0 via “pass all” filter.

```bash
#setting ingress port
tc qdisc add dev vnet0 ingress

#setting egress port
tc filter add dev vnet0 parent ffff:
protocol all \
  u32 match u8 0 0 \
  action mirred egress mirror dev l2gre0
```
Linux bridge - filtering traffic using TC

- Select traffic of interest by applying a filter on the virtual tap
- TC filtering is done using a U32 type filter, example below is filtering on IP protocol for ICMP packets

```bash
> tc filter add dev vnet0 parent ffff: protocol ip pref 3
  basic handle 0x1 \n  u32(00010000/00ff0000 at 0) \n  action order 1: mirred (egress mirror to device l2gre0)
pipe\n  index 2491 ref 1 bind 1

> 
```
Linux bridge - filtering traffic using TC

- Adding multiple conditional filters can be tedious to type
- Example of two condition filter compares IP source to “192.168.100.1” (0xC0A8641) and TCP destination port to “7777”

```c
filter protocol ip pref 1 basic handle 0x1
(
    u32(c0a86401/ffffffff at 12)
) AND (;
    cmp(u8 at 9 layer 1 mask 0xff eq 6)
    AND cmp(u16 at 2 layer 2 mask 0xffffffff eq 7777)
)
action order 1: mirred (Egress Mirror to device l2gre0) pipe
    index 2483 ref 1 bind 1
```
OpenVSwitch - creating port mirror over GRE

- Monitor vnet create mirrors over GRE using OVS commands

```bash
ifconfig br0 192.168.1.200 netmask 255.255.255.0
ovs-vsctl add-br br0
ovs-vsctl add-port br0 vnet1
ovs-vsctl add-port br0 vnet0
ovs-vsctl add-port br0 gre0 \  
  -- set interface gre0 type=gre  
  options:remote_ip=192.168.1.100,ttl=255,key=5000\  
  -- --id=@p get port gre0 \  
  -- --id=@m create mirror name=m0 select-all=true output-port=@p \  
  -- set bridge br0 mirrors=@m
```
More advanced topics for a different discussion

- Open vSwitch
  - VXLAN (Virtual Extensible LAN)
  - STT (Stateless Transport Tunneling)
  - GeNeVE (Generic Network Encapsulation)
  - IPsec

- Linux/KVM accelerations
  - Intel DPDK (Data Plane Development Kit)
  - PCI-SIG SR-IOV (Single Root – I/O Virtualization)
  - PF_RING
  - Netmap
Linux KVM Virtual Traffic Monitoring

Example of virtual monitoring
## Linux KVM Virtual Traffic Monitoring

![Diagram of TitanXR software](image)

### IntellaTap-VE Servers

<table>
<thead>
<tr>
<th>Server</th>
<th>Virtual Machine</th>
<th>Internal Network</th>
<th>Interface</th>
<th>IP Address</th>
<th>Tap Status</th>
<th>Filter</th>
<th>Tunnel End Point</th>
<th>Tunnel Status</th>
<th>Tunnel Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server01</td>
<td>pub1.webserver</td>
<td>net1</td>
<td>vnet0</td>
<td>10.10.164.100</td>
<td>Up</td>
<td>web traffic</td>
<td>10.10.166.12</td>
<td>Up</td>
<td>16.6 Mbps</td>
</tr>
<tr>
<td>Server01</td>
<td>pub2.webserver</td>
<td>net1</td>
<td>vnet1</td>
<td>10.10.164.120</td>
<td>Up</td>
<td>web traffic</td>
<td>10.10.166.12</td>
<td>Up</td>
<td>16.6 Mbps</td>
</tr>
<tr>
<td>Server02</td>
<td>shoretel.voip</td>
<td>net1</td>
<td>vnet2</td>
<td>10.10.165.10</td>
<td>Up</td>
<td>tcp traffic</td>
<td>10.10.166.14</td>
<td>Up</td>
<td>5.5 Mbps</td>
</tr>
<tr>
<td>Server02</td>
<td>shoretel2.voip</td>
<td>net1</td>
<td>vnet3</td>
<td>10.10.165.11</td>
<td>Down</td>
<td></td>
<td>10.10.166.14</td>
<td>Up</td>
<td>5.5 Mbps</td>
</tr>
</tbody>
</table>

### Server01 and 02

<table>
<thead>
<tr>
<th>Server</th>
<th>Virtual Machine</th>
<th>Internal Network</th>
<th>Interface</th>
<th>IP Address</th>
<th>Tap Status</th>
<th>Filter</th>
<th>Tunnel End Point</th>
<th>Tunnel Status</th>
<th>Tunnel Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server01</td>
<td>pub1.webserver</td>
<td>net1</td>
<td>vnet0</td>
<td>10.10.164.100</td>
<td>Up</td>
<td>web traffic</td>
<td>10.10.166.12</td>
<td>Up</td>
<td>16.6 Mbps</td>
</tr>
<tr>
<td>Server01</td>
<td>pub2.webserver</td>
<td>net1</td>
<td>vnet1</td>
<td>10.10.164.120</td>
<td>Up</td>
<td>web traffic</td>
<td>10.10.166.12</td>
<td>Up</td>
<td>16.6 Mbps</td>
</tr>
<tr>
<td>Server02</td>
<td>shoretel.voip</td>
<td>net1</td>
<td>vnet2</td>
<td>10.10.165.10</td>
<td>Up</td>
<td>tcp traffic</td>
<td>10.10.166.14</td>
<td>Up</td>
<td>5.5 Mbps</td>
</tr>
<tr>
<td>Server02</td>
<td>shoretel2.voip</td>
<td>net1</td>
<td>vnet3</td>
<td>10.10.165.11</td>
<td>Down</td>
<td></td>
<td>10.10.166.14</td>
<td>Up</td>
<td>5.5 Mbps</td>
</tr>
</tbody>
</table>
Virtual network visibility is important in today’s data center
Discussed how data moves in virtual networks
Reviewed some east-west traffic monitoring techniques
Advantages of traffic control based monitoring for Linux KVM
IntellaTap-VE KVM product to seamlessly manage virtual taps in KVM environments.

For more information on APCON virtual monitoring visit www.apcon.com
Linux KVM Monitoring Solutions

SPEAKER

Scott Harvey
Director of Engineering
Automate vnet monitoring setup (3 steps)

### 3 Steps

1. Create GRE
2. Tap vnet
3. Create Filter
Configuring filters

- Graphical simpler version of the complex two condition CLI filter