Virtual Switching Without a Hypervisor for a More Secure Cloud

Xin Jin
Princeton University

Joint work with Eric Keller(UPenn) and Jennifer Rexford(Princeton)
Public Cloud Infrastructure

• Cloud providers offer computing resources on demand to multiple “tenants”

• Benefits:
  – Public (any one can use)
  – Economies of scale (lower cost)
  – Flexibility (pay-as-you-go)
Server Virtualization

• Multiple VMs run on the same server

• Benefits
  – Efficient use of server resources
  – Backward compatibility

• Examples
  – Xen
  – KVM
  – VMware
Network Virtualization

• Software switches
  – Run in the hypervisor or the control VM (Dom0)
• Benefits: Flexible control at the “edge”
  – Access control
  – Resource and name space isolation
  – Efficient communication between co-located VMs
• Examples
  – Open vSwitch
  – VMware’s vSwitch
  – Cisco’s Nexus 1000v Switch
Security: a major impediment for moving to the cloud!

Let’s take a look at where the vulnerabilities are...
Vulnerabilities in Server Virtualization

- The hypervisor is quite complex
- Large amount of code $\rightarrow$ Bugs (NIST’s National Vulnerability Database)
Vulnerabilities in Server Virtualization

- The hypervisor is an attack surface (bugs, vulnerable)
  
  → Malicious customers attack the hypervisor
Vulnerabilities in Network Virtualization

- Software switch in control VM (Dom0)
- Hypervisor is involved in communication
Vulnerabilities in Network Virtualization

- Software switch is coupled with the control VM
  - e.g., software switch crash can lead to a complete system crash
Dom0 Disaggregation [e.g., SOSP’11]

- Disaggregate control VM (Dom0) into smaller, single-purpose and independent components
- Malicious customer can still attack hypervisor!
NoHype [ISCA’10, CCS’11]

- Eliminate the hypervisor attack surface
- What if I want to use a software switch?

- Pre-allocating memory and cores
- Using hardware virtualized I/O devices
- Hypervisor is only used to boot up and shut down guest VMs.
Software Switching in NoHype

- Bouncing packets through the physical NIC
- Consumes excessive bandwidth on PCI bus and the physical NIC!
Our Solution Overview

- Eliminate the hypervisor attack surface
- Enable software switching in an efficient way
Eliminate the Hypervisor-Guest Interaction

- **Shared memory**
  - Two FIFO buffers for communication
- **Polling only**
  - Do not use event channel; no hypervisor involvement
Limit Damage From a Compromised Switch

- Decouple software switch from Dom0
  - Introduce a Switch Domain (DomS)
- Decouple software switch from the hypervisor
  - Eliminate the hypervisor attack surface
Preliminary Prototype

**Prototype based on**
- Xen 4.1: used to boot up/shut down VMs
- Linux 3.1: kernel module to implement polling/FIFO
- Open vSwitch 1.3
Preliminary Evaluation

- Evaluate the throughput between DomS and a guest VM, compared with native Xen
- Traffic measurement: Netperf
- Configuration: each VM has 1 core and 1GB of RAM
Evaluation on Throughput

• FIFO Size
  – Polling period is fixed to 1ms
  – Reach high throughput with just 256 FIFO pages (Only 1MB)

• Polling Period
  – Shorter polling period, higher throughput
  – CPU resource consumption?
    → Future work
Comparison with Native Xen

- Outperforms native Xen when message size is smaller than 8 KB.
- Future work: incorporate more optimization.
Conclusion and Future Work

• Trend towards software switching in the cloud
• Security in hypervisor and Dom0 is a big concern
• Improve security by enabling software switching without hypervisor involvement

• Future work
  – Detection and remediation of DomS compromise
Thanks!

Q&A