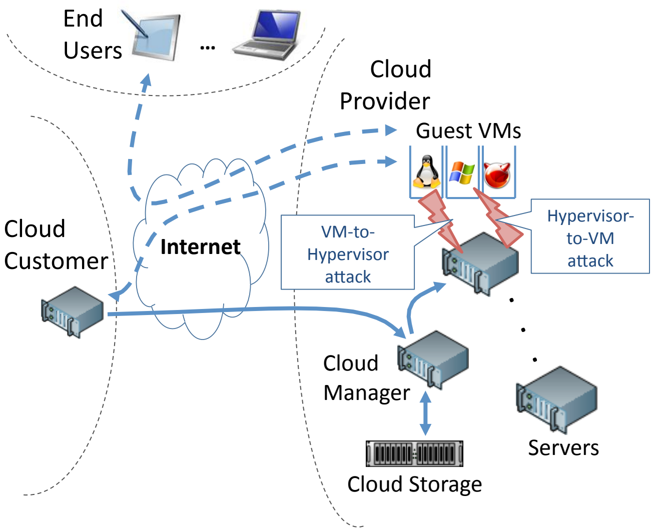
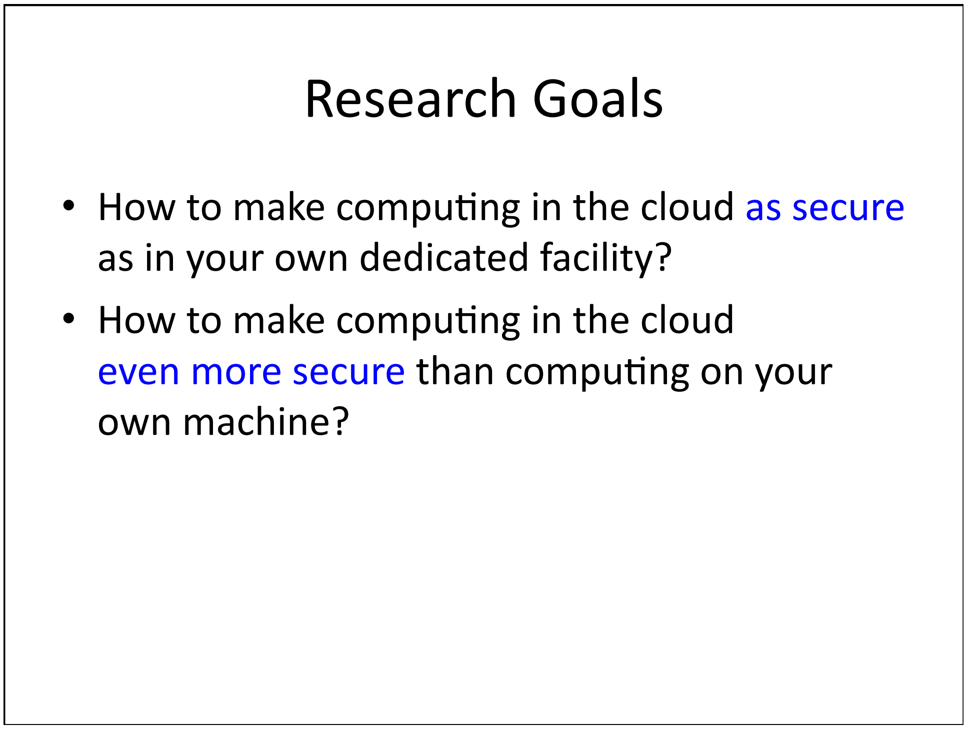
Hardware enhanced Security in Cloud Computing

Ruby B. Lee
Princeton University

ARO workshop on Cloud Security,
March 11, 2013

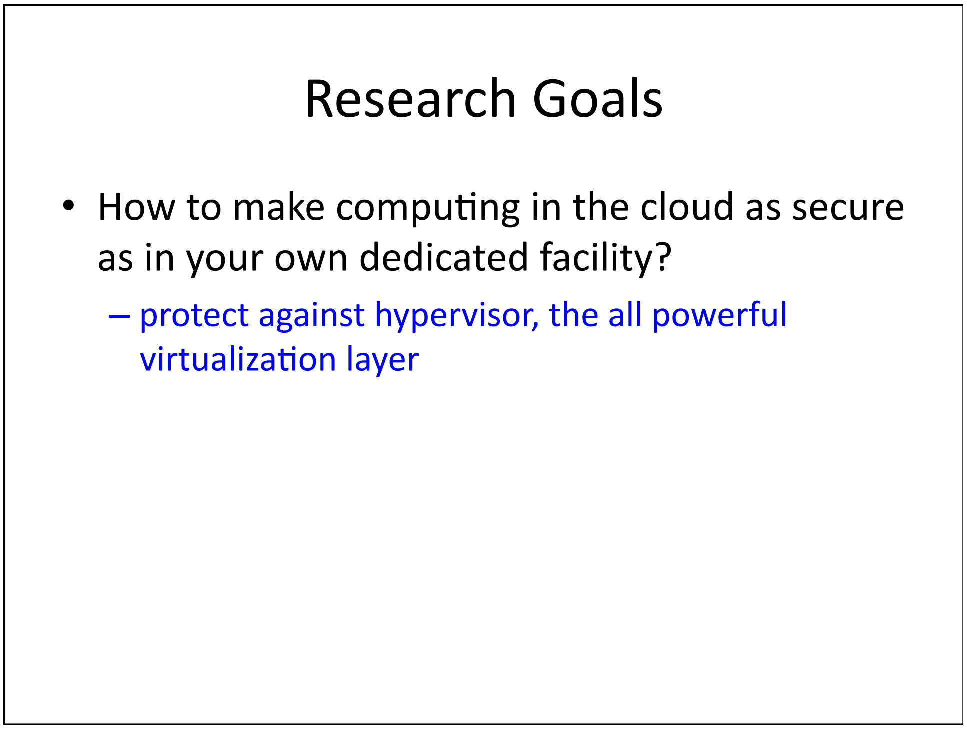
Cloud Computing (Public IaaS)





Research Goals

- How to make computing in the cloud as secure as in your own dedicated facility?
- How to make computing in the cloud even more secure than computing on your own machine?



Research Goals

- How to make computing in the cloud as secure as in your own dedicated facility?
  - protect against hypervisor, the all powerful virtualization layer

Research Directions

- Harden existing hypervisor
- Protect Virtual Machines even from a compromisable commodity hypervisor
  - e.g., Hyperwall
- Remove the hypervisor at VM runtime,
  - e.g., NoHype

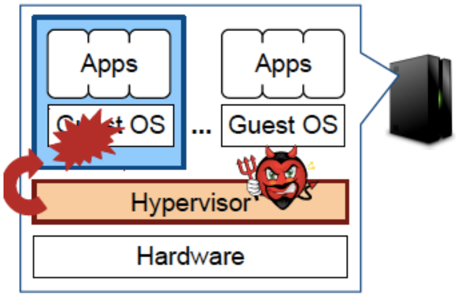
Threat Model:
Compromisable Hypervisor

What if hypervisor itself is compromised?

- How can hardware protect against confidentiality and integrity breaches against a Virtual Machine by an untrusted hypervisor?
  - Retain hypervisor for management
  - Use hardware access control to prevent hypervisor (and DMA) from accessing a VM's memory after it has been allocated

Hypervisor-secure virtualization, e.g., Hyperwall architecture

- Do not trust hypervisor but retain it for management
- Hardware protects VMs from hypervisor-level attackers
- Hardware enables trust evidence attestation

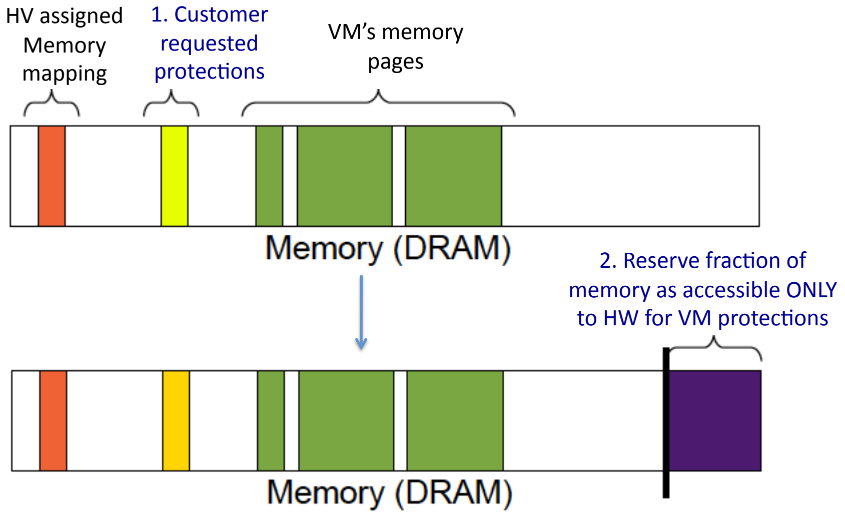


Jakub Szefer and Ruby B. Lee, "Architectural Support for Hypervisor-Secure Virtualization," Intl. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS), March 2012.

What to protect from Hypervisor?

- Protect Virtual Machine's memory from hypervisor and DMA
  - Stores data, code, state, cipher keys
  - Gateway to Networking and Storage
- Secure communication between customer and Virtual Machine in cloud
- Attest trust evidence for SW/HW platform
- Protect VM state on interrupts
  - and on VM init, VM terminate

Customer-specified VM memory protections, hardware enforced



HV assigned Memory mapping

1. Customer requested protections

VM's memory pages

Memory (DRAM)

2. Reserve fraction of memory as accessible ONLY to HW for VM protections

Memory (DRAM)

Confidentiality and Integrity Protections (CIP) for each machine memory page

- Keep VM protections (CIP) in hardware-only accessible DRAM
- For each VM page, 3 bits for HW access control:
  - Unassigned
  - Assigned, no restrictions
  - Assigned, DENY hypervisor access
  - Assigned, DENY DMA access
  - Assigned, DENY Hypervisor and DMA access
- Need to look up CIP tables only on TLB miss
- Fast hardware-enforced access control against untrusted hypervisor and DMA

Protect VM state on Suspend_Resume

- VM's memory protection enforced by Hardware even when VM suspended
- Processor registers hold VM state
- Accessible on interrupts by hypervisor
  - Can breach confidentiality and integrity
- Protect per-VM protection information
    - State capture on interrupt
    - Initial VM state
    - Requested protections
    - Collected trust evidence
- Encrypt and hash general-purpose registers

NoHype: Hypervisor-free Virtualization

- Utilizes Hardware trend: Manycore Chips
- Software trend: Virtualization
- IaaS Cloud Computing

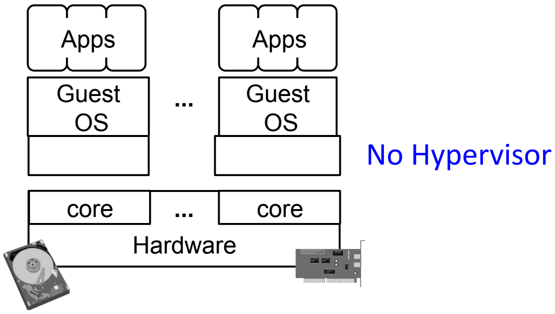
Jakub Szefer, Eric Keller, Ruby B. Lee and Jennifer Rexford, "Eliminating the Hypervisor Attack Surface for a More Secure Cloud," Computer and Communications Security (CCS), October 2011.

E. Keller, J. Szefer, J. Rexford, and R.B. Lee, "NoHype: Virtualized cloud infrastructure without the virtualization," Intl. Symp. on Computer Arch. (ISCA 2010), June 2010.

13

NoHype: remove hypervisor at runtime

- Hypervisor initiates VM and pre-allocates resources
- Remove need for hypervisor at Runtime
- Hypervisor comes in to terminate VM



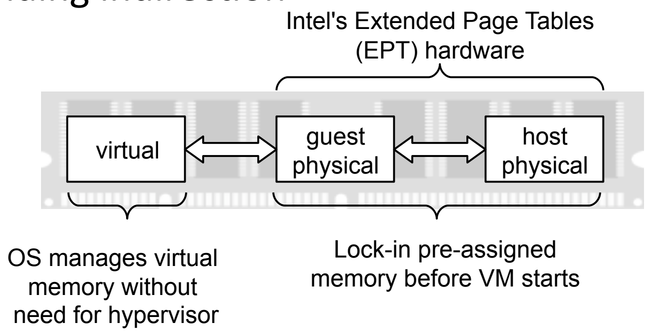
PRINCETON UNIVERSITY

NoHype: Hypervisor-Free Virtualization

Hypervisor Functions	**NoHype Solution**
Scheduling virtual machines	One VM per core (manycore processors)
Managing memory	Pre-allocate memory with processor support
Emulating I/O devices	Direct access to SRIOV virtualized devices
Networking	hardware Ethernet switches
Managing virtual machines	Decouple VM management from VM operation (IaaS)

NoHype on today's hardware

- Pre-allocating memory and cores
- Using hardware virtualized I/O devices
- Short-circuiting the system discovery process
- Avoiding indirection

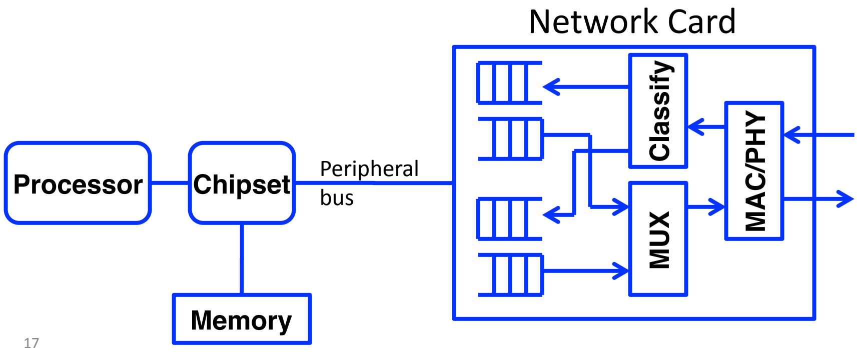

Intel's Extended Page Tables (EPT) hardware
virtual
guest physical
host physical
OS manages virtual memory without need for hypervisor
Lock-in pre-assigned memory before VM starts

NoHype

Use HW-Virtualized Devices

for VM access to Networking and Storage devices

- Per-VM physical device doesn't scale
- Multiple queues on device (per-VM queue)
  - Multiple memory ranges mapping to different queues
  - Static memory partitioning for HW-enforced access control



Research Goals

- How to make computing in a Virtual Machine in the cloud even more secure than computing on your own machine?
  - Protect from Guest OS and other Apps inside VM
- How?
  - HW-SW co-design of minimalist TCB comprising trustworthy hypervisor & processor
  - Protect Apps in VM using Software security monitors/mechanisms (in same address space) which are themselves protected

Hardware-enhanced Access Control under different Threat Models

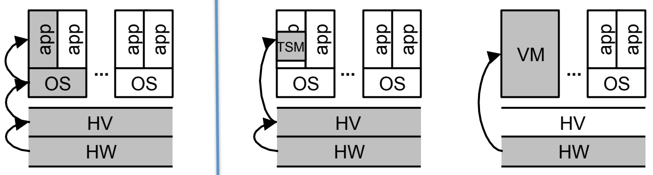
Today

Tomorrow? *Layer-skipping trust chains with HW trust anchors*

TPM-like

Fine-grained, e.g., Bastion

Protect from HV e.g., Hyperwall



Trusted HV, Trusted OS

Trusted HV, Untrusted OS

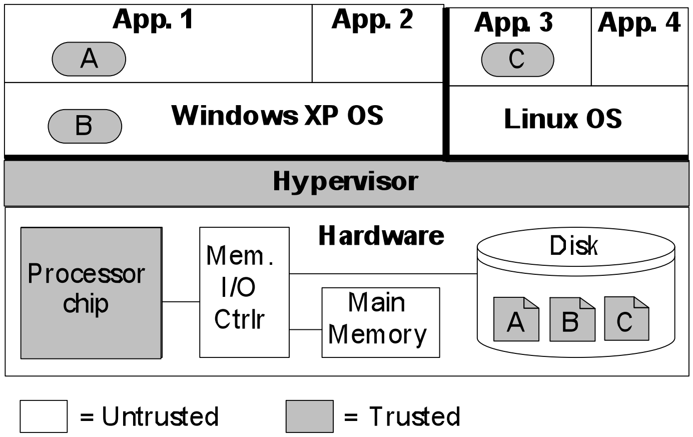
Untrusted HV

▨ trusted □ untrusted

Bastion's Architectural Strategy

- What is a flexible, general-purpose solution for providing security protections?
  - Use software for flexibility
  - Use hardware to protect these software protection mechanisms

Feasibility Example: Bastion architecture



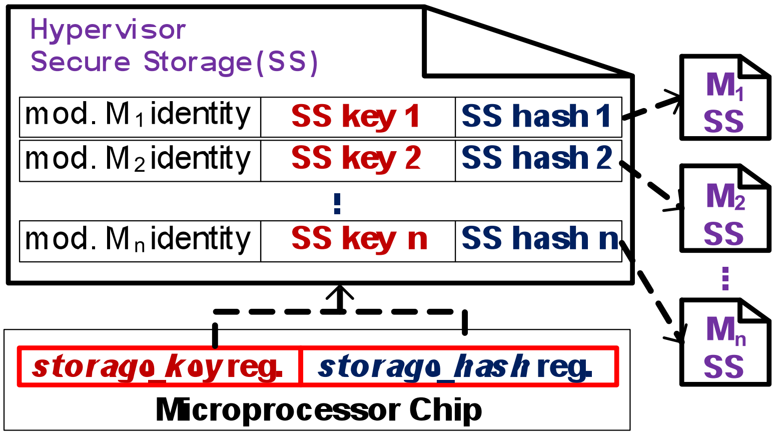
21

Bastion: security mechanisms

- Hypervisor Protection
  - Secure Launch of Hypervisor
  - Protecting Hypervisor at Runtime
- Trusted Software Module Protection
  - Secure Launch
  - Secure Virtual Memory Mapping
  - Secure Physical Memory
  - Secure Inter-Module Control Flow
- Trusted Computing Primitives
  - Secure Storage
    - sealed to each Trusted Software Module
  - Tailored Attestation

Scalable Secure Storage (SS)

sealed to each Trusted Software Module or Hypervisor



mod. M_1 identity	SS key 1	SS hash 1
mod. M_2 identity	SS key 2	SS hash 2
mod. M_n identity	SS key n	SS hash n

Research Directions

- Harden existing hypervisor
- Protect Virtual Machines from a compromisable commodity hypervisor
  - e.g., Hyperwall
- Remove the hypervisor at VM runtime,
  - e.g., NoHype
- Design trustworthy & trusted hypervisor and hardware TCB
- Protect application from guest OS and other apps inside VM

Summary

- HW-SW foundations can make Cloud Computing as secure, or more secure, than dedicated computers
- Proof of concept architectures:
  - Hyperviseor-Secure Virtualization, e.g., Hyperwall
  - Hypervisor-Free Virtualization, e.g., NoHype
  - Bastion hardware-hypervisor TCB protects VM's Trusted Software Modules, which in turn, protect apps and data within a VM

Future Research

- Design minimal, provable, SW-HW co-designed hypervisor-processor TCB
- Verifiable construction of Trusted Software Modules for security monitors and policy managers
- Availability of cloud services
- Information Leakage in Cloud
- Extend Bastion to manycore processors
- Enable migration with NoHype and Hyperwall
- Trust evidence, security verification, secure clients