CSIE 5310

Virtualization Security

Prof. Shih-Wei Li

Department of Computer Science and Information Engineering
National Taiwan University

Agenda

- **Virtual machine based rootkits**
- Intro to virtualization-based secure systems
- Improving hypervisor security
  - Problems
  - New Hypervisor Design

Virtual Machine based Rootkit

- Rootkit: a type of malware that gives unprivileged software/user access to privileged operations of a computer
  - Hide its existence and malicious operations from detection
- Researchers from Microsoft first came up with virtual machine based rootkit (**VMBR**) in their S&P 06 paper:
  - SubVirt: Implementing malware with virtual machines (https://www.microsoft.com/en-us/research/publication/subvirt-implementing-malware-with-virtual-machines/)

VMBR

- Install VMBR underneath an existing operating system during boot time
  - Hoist the OS into a malicious VM to host malicious services
- Hard to remove or detect — the hoisted OS has no access to VMBR
- The paper implements POC for Windows XP and Linux

VMBR

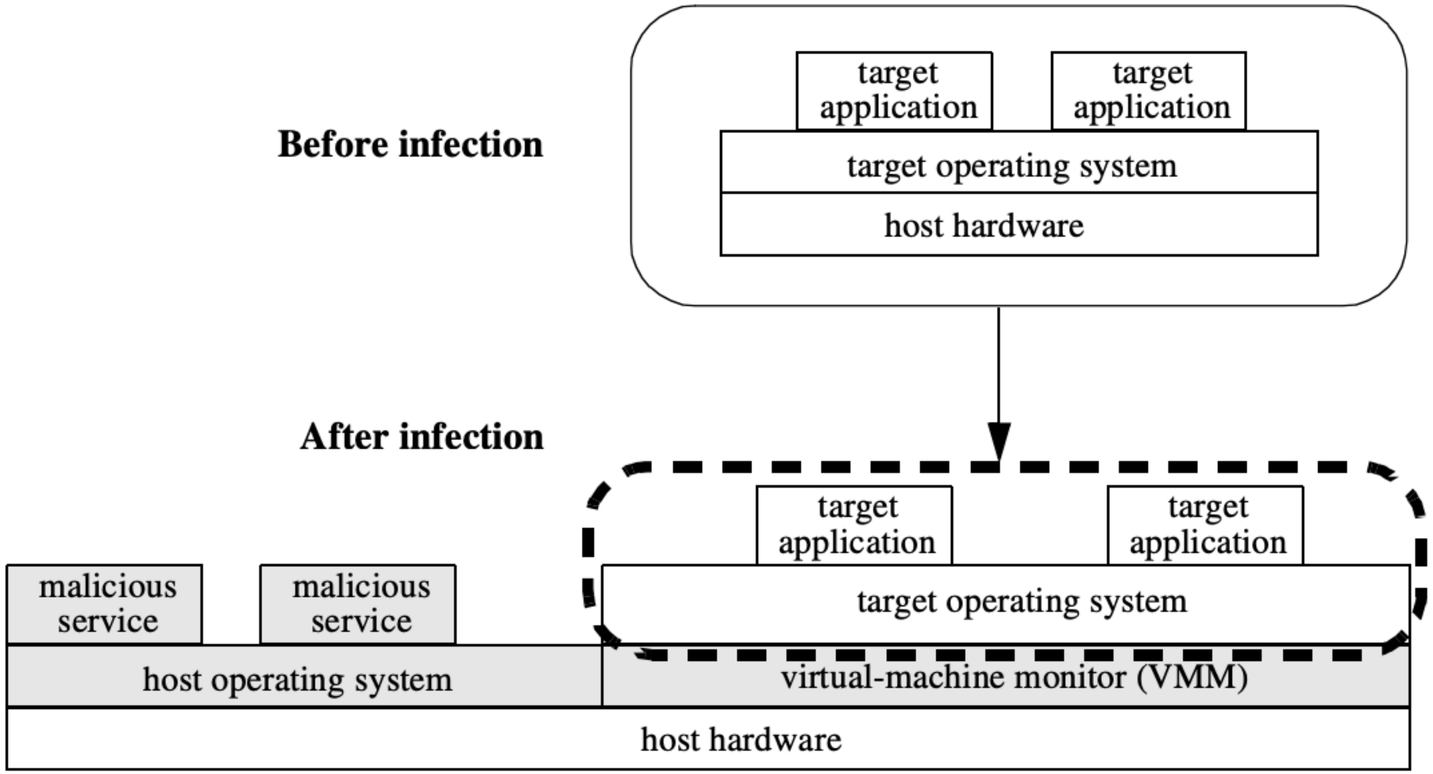


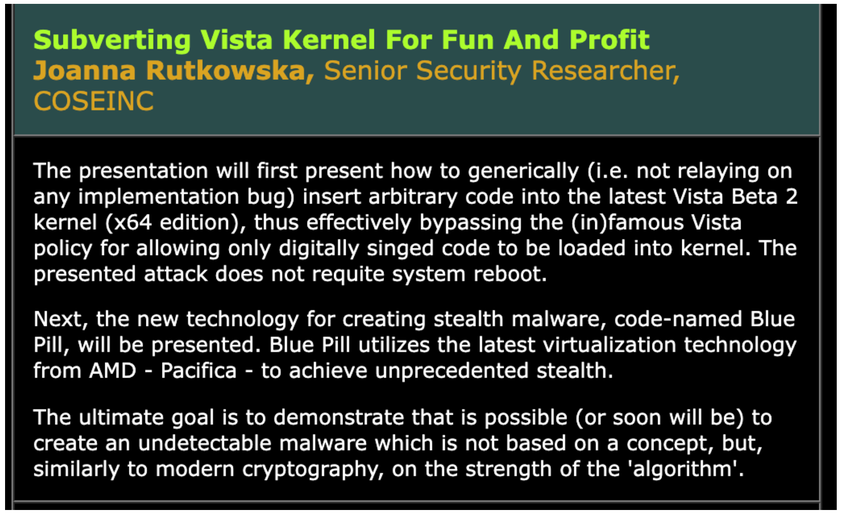
Figure 2. This figure shows how an existing target system can be moved to run inside a virtual machine provided by a virtual-machine monitor. The grey portions of the figure show the components of the VMBR.

VMBR: Malicious Services

- ***Launch malicious services:*** spam relays, host phishing web servers, and perform DoS attack
- ***Observe data/events from the targeted OS:*** keylogging or intercepting network packets
- ***Modify execution of the targeted OS:*** change the behavior of the applications, modify network communications, etc.

Blue Pill

- Rootkit based on x86 virtualization (rely on AMD's virtualization features)
  - Concept is originated from VMBR; runs a thin hypervisor below the OS
- Demoed by Joanna Rutkowska at Black Hat 06 to subvert Windows Vista



Subverting Vista Kernel For Fun And Profit
Joanna Rutkowska, Senior Security Researcher, COSEINC

The presentation will first present how to generically (i.e. not relaying on any implementation bug) insert arbitrary code into the latest Vista Beta 2 kernel (x64 edition), thus effectively bypassing the (in)famous Vista policy for allowing only digitally singed code to be loaded into kernel. The presented attack does not requite system reboot.

Next, the new technology for creating stealth malware, code-named Blue Pill, will be presented. Blue Pill utilizes the latest virtualization technology from AMD - Pacifica - to achieve unprecedented stealth.

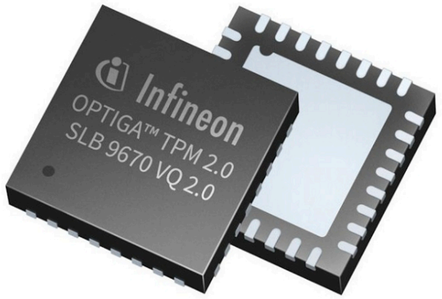
The ultimate goal is to demonstrate that is possible (or soon will be) to create an undetectable malware which is not based on a concept, but, similarly to modern cryptography, on the strength of the 'algorithm'.

Blue Pill: Detection

- Claimed as 100% undetectable: the VMM can fool any detection mechanisms
- Researchers in Black Hat 07 wanted to test Blue Pill on their tool
  - Did not happen — Blue Pill's author requested for $384,000 in funding as a prerequisite for entering the competition

https://www.blackhat.com/presentations/bh-usa-07/Rutkowska/Presentation/bh-usa-07-rutkowska.pdf

Countermeasures

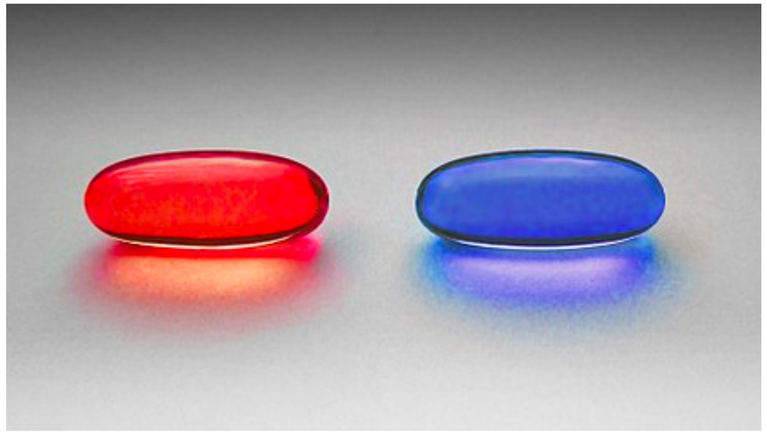


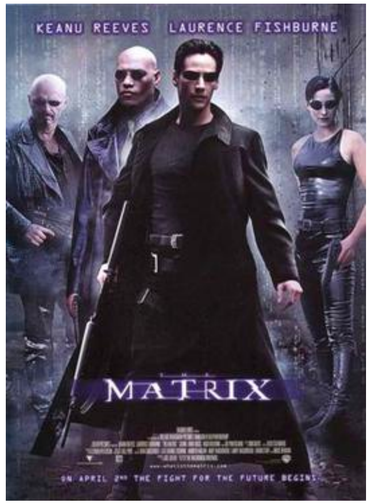
- Leverage Trusted Platform Module (TPM)
  - Attest the integrity of software using known hashes
  - Attest during machine boot to ensure only trusted software components can execute — prevents an attacker from installing VMBR

Red Pill

- A technique to detect the presence of a hypervisor (an anti-VM approach) developed by Joanna Rutkowska
  - Tested on VMware Workstation 4 and Virtual PC 2004
- Leverage the *sidt* instruction:
  - Stores the contents of the interrupt descriptor table register (IDTR)
  - The instruction can be executed in non-privileged mode (ring3) when running in a VM — the VMM returns the contents of the IDTR

Blue Pill and Red Pill





Agenda

- Virtual machine based rootkits
- **Intro to virtualization-based secure systems**
- Improving hypervisor security
  - Problems
  - Software & Hardware Approach

The CIA Triad of Security

- *Confidentiality:* protects information from unauthorized access. Keep sensitive data confidential and guarantee its access by authorized individuals with relevant rights.
- *Integrity:* maintains the information's, consistency, accuracy, and authenticity, with no modifications by an unauthorized party.
- *Availability:* ensures that authorized individuals can access data with minimum interruptions. Provides information, infrastructure, and applications when individuals need them.

Review: Hypervisor Properties

- Isolation
  - Software in one VM cannot access or modify the software (and its data) running in the hypervisor or a separate VM
- Inspection
  - The hypervisor has access to all the state and resources of a VM
  - States: CPU state (registers), memory state, and I/O device state (e.g. contents of storage devices or register state of I/O controllers)
- Interposition
  - A hypervisor interposes on certain VM operations (ex: executing sensitive instructions)

VM Security Benefits

- Isolate and contain faults in virtual machines
  - Sandboxing software in VMs
    - Prevent software crashes in a given VM from affecting other VMs
- It is harder for an attacker from one VM to compromise other VMs; much easier for an application to compromise other applications running on a shared OS kernel

VM Security Benefits

- Virtual machine based security systems
  - Rely on a trusted hypervisor to protect software running in a VM

Virtual machine based security systems

Virtual Machine Introspection:	**Kernel Protection:**	**Application Protection:**
[Livewire 03]	[SecVisor 07]	[Terra 03]
[VMWatcher 07]	[NICKEL 08]	[Proxos 06]
	[Lares 08]	[SP3 08]
	[HookSafe 09]	[Overshadow 08]
	[SIM 09]	[TrustVisor 10]
	[OSck 11]	[InkTag 13]
	[SecPod 15]	[SeCage 15]

Virtual Machine Introspection (VMI)

- Rely on a hypervisor to monitor runtime state of a virtual machine:
  - Forensic (malware) analysis
  - Intrusion detection

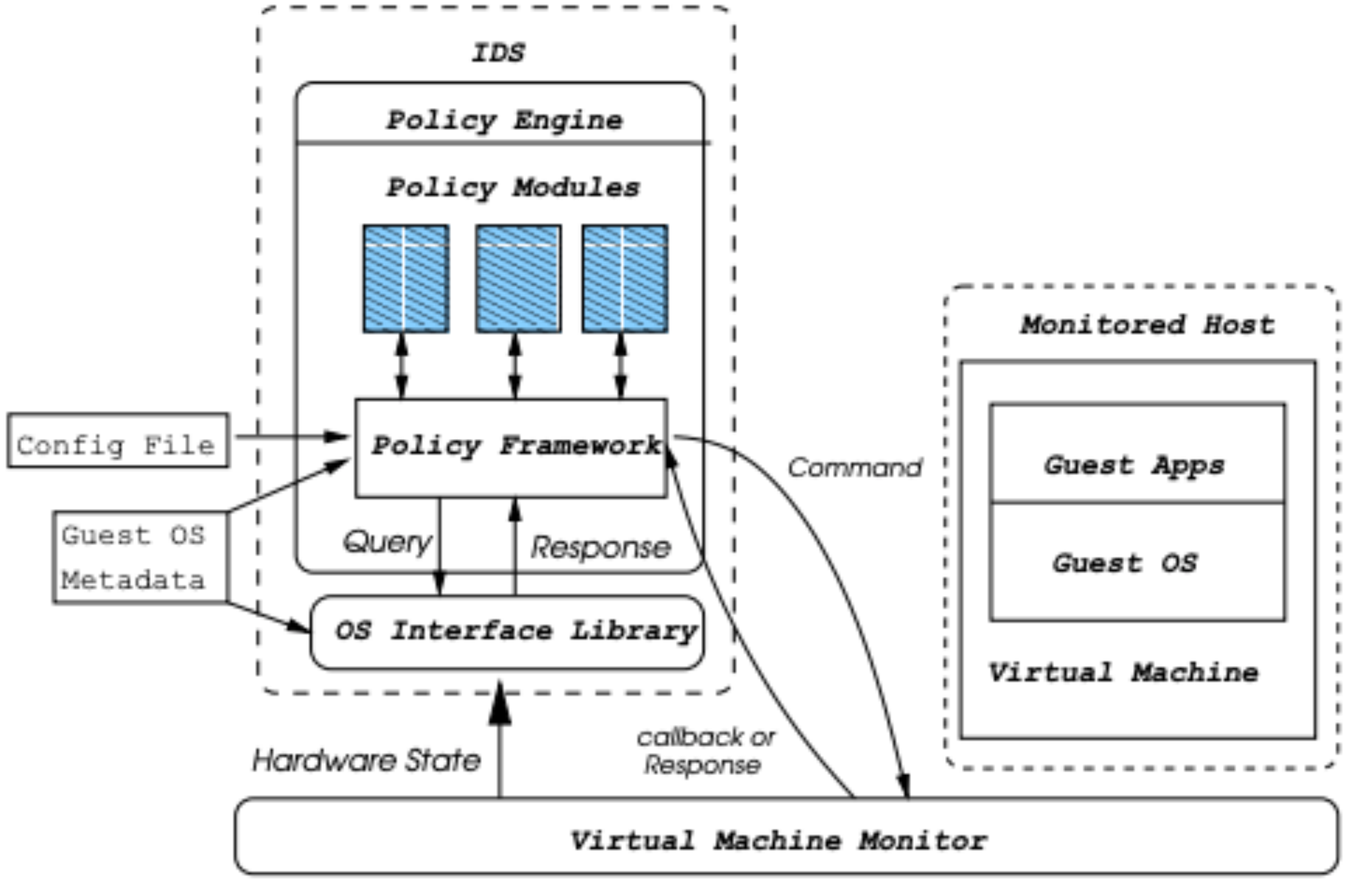
Virtual machine based security systems

Virtual Machine Introspection:	**Kernel Protection:**	**Application Protection:**
[Livewire 03]	[SecVisor 07]	[Terra 03]
[VMWatcher 07]	[NICKEL 08]	[Proxos 06]
	[Lares 08]	[SP3 08]
	[HookSafe 09]	[Overshadow 08]
	[SIM 09]	[TrustVisor 10]
	[OSck 11]	[InkTag 13]
	[SecPod 15]	[SeCage 15]

Virtual Machine Introspection (VMI)

- Garfinkel and Rosemblum proposed a VMI based intrusion detection system (IDS) called Livewire in their NDSS 2003 paper
  - Goal is to detect anomaly in the monitored software
  - Do not protect software from getting compromised!
  - Isolate the monitored software in a VM to protect the IDS

VMI based IDS

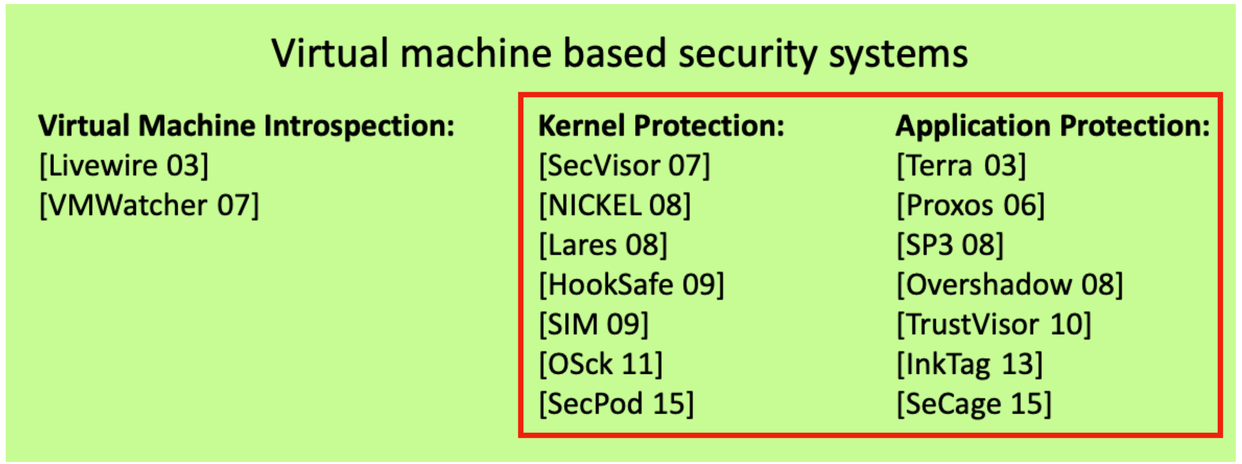


VMI based IDS: Example of Policy Modules

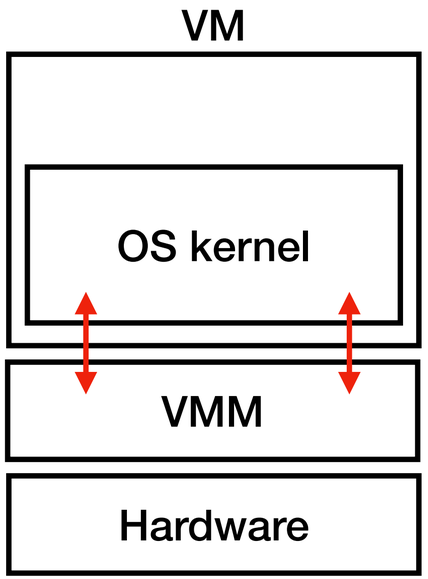
- Check user program integrity by comparing it against known good hashes
  - Scan VM storage and memory for known malicious program
- What else can VMI do?

VM based Software Protection

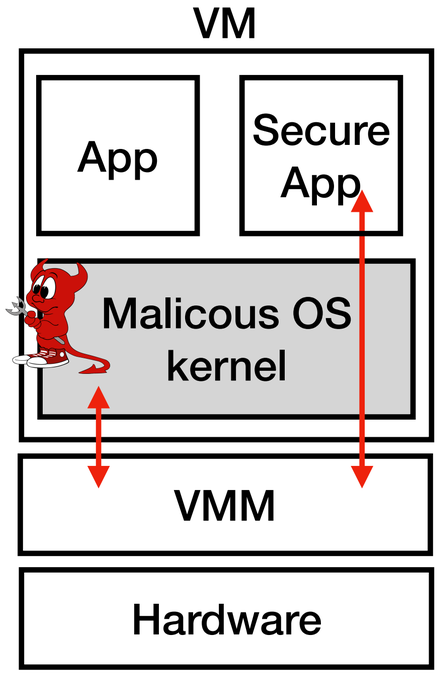
- Rely on a hypervisor to secure the software within a VM
  - The hypervisor interposes VM execution to protect VM software
  - Assume different threat models (integrity or confidentiality protection)
- Two types:
  - VM Kernel protection
  - VM Application protection



VM based Software Protection



VM Kernel protection



VM App protection

Example: SecVisor

- Example: SecVisor [SOSP 07]
  - **Goal:** protect kernel integrity
  - **Threat model:**
    - Assume attacker cannot compromise SecVisor; assume the attacker cannot physically steal the machine
    - Assume the attacker is capable of: (1) arbitrarily modifying existing kernel code or adding new code, (2) loading malicious device drivers, (3) controlling devices to make malicious DMA against the kernel code
  - **Policy:** kernel code read-only; kernel data non-executable; etc.
  - **Mechanism:** SecVisor, a tiny hypervisor, that runs the kernel in a VM to enforce security policy

Trusted Computing Base (TCB)

- According to Butler Lampson et al. from "*Authentication in Distributed Systems: Theory and Practice*", the TCB of a computer system is:

A small amount of software and hardware that security depends on and that we distinguish from a much larger amount that can misbehave without affecting security.

Trusted Computing Base (TCB)

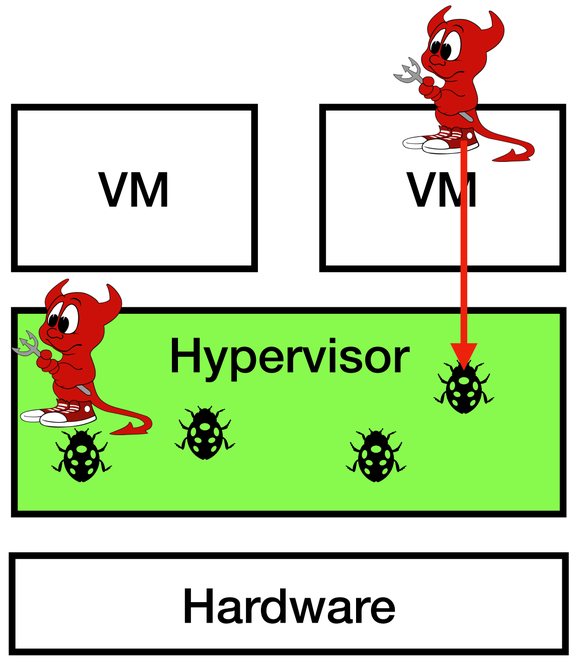
- The TCB usually enforces security policies for the system
  - Bugs inside the TCB could jeopardize the security of the whole system
  - Misbehavior outside the TCB cannot affect the enforced security policy
- Quantify the software TCB size by lines of code (LOC)
- Hardware is commonly included in the TCB
  - What if the hardware is buggy?

Trusted Computing Base (TCB)

- Examples: the TCB of computer software systems
  - The OS kernel: enforces process isolation
  - The hypervisor: enforces VM isolation
  - Java: the runtime and standard library that provide security features
- Software systems are buggy
  - Result in poor security if the software TCB is large!

Challenges: Virtualization-based Secure Systems

- In a non-virtualized setting, the OS trusts the hardware
- In a virtualized setting, VMs trust the hypervisor
  - Are the hypervisors as trustworthy as the hardware?
  - Hypervisors become more complex — resulting in a large TCB
- Case study: random number generation in VMs
  - Challenging for the VMs to generate fresh random numbers
  - VMMs to virtualize random number generators



Improve the Security of Existing Software Systems

- Makes the system harder to exploit
  - Reduce the software TCB
    - Software: introduce new design for existing software
    - Hardware: offload responsibility the hardware to offer protection
- What else?

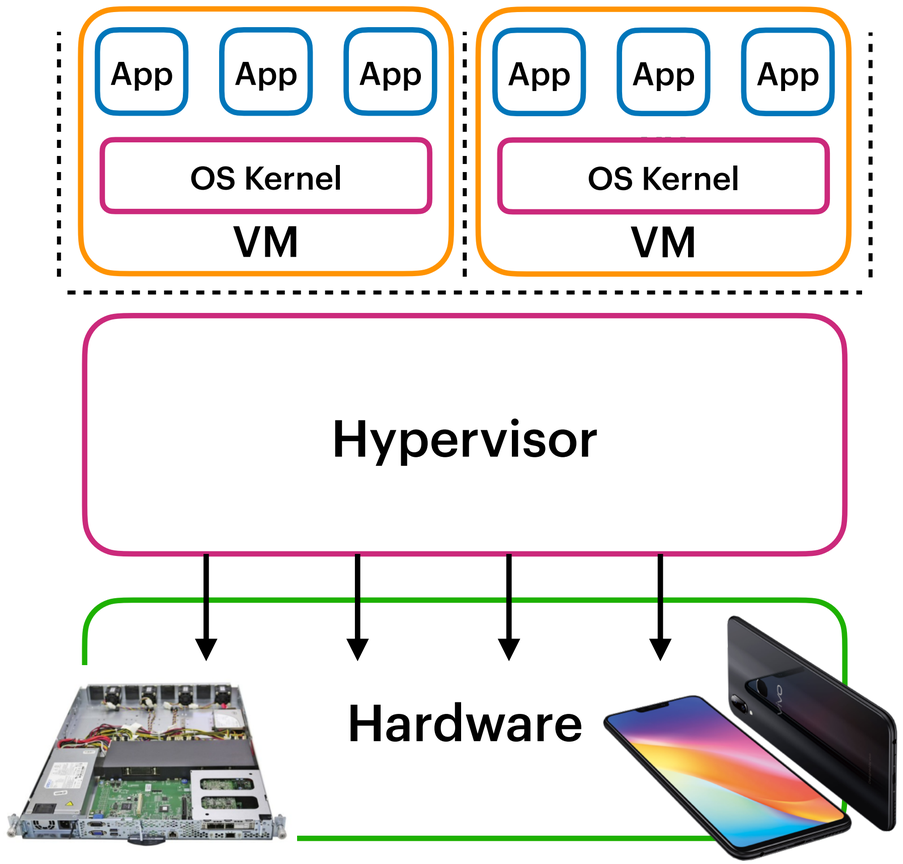
Rewind: Countermeasures

- Limitations: TPM cannot prevent an attacker from exploiting existing hypervisor vulnerabilities
  - What if the hypervisor code contains zero-day vulnerabilities?
  - VM data is at risk if the hypervisor gets compromised
- How can we address the limitations here?

Agenda

- Virtual machine based rootkits
- Intro to virtualization-based secure systems
- **Improving hypervisor security**
  - **Problems**
  - New Hypervisor Design

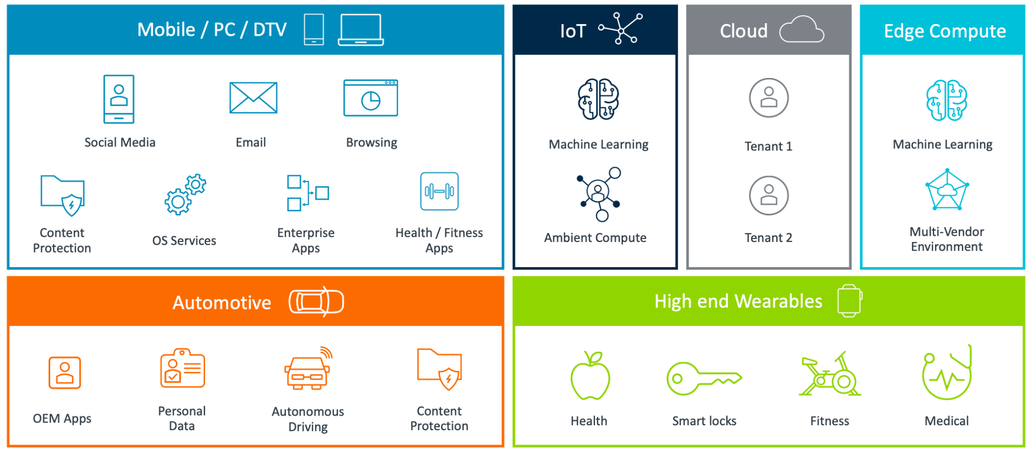
Review: Modern Hypervisors



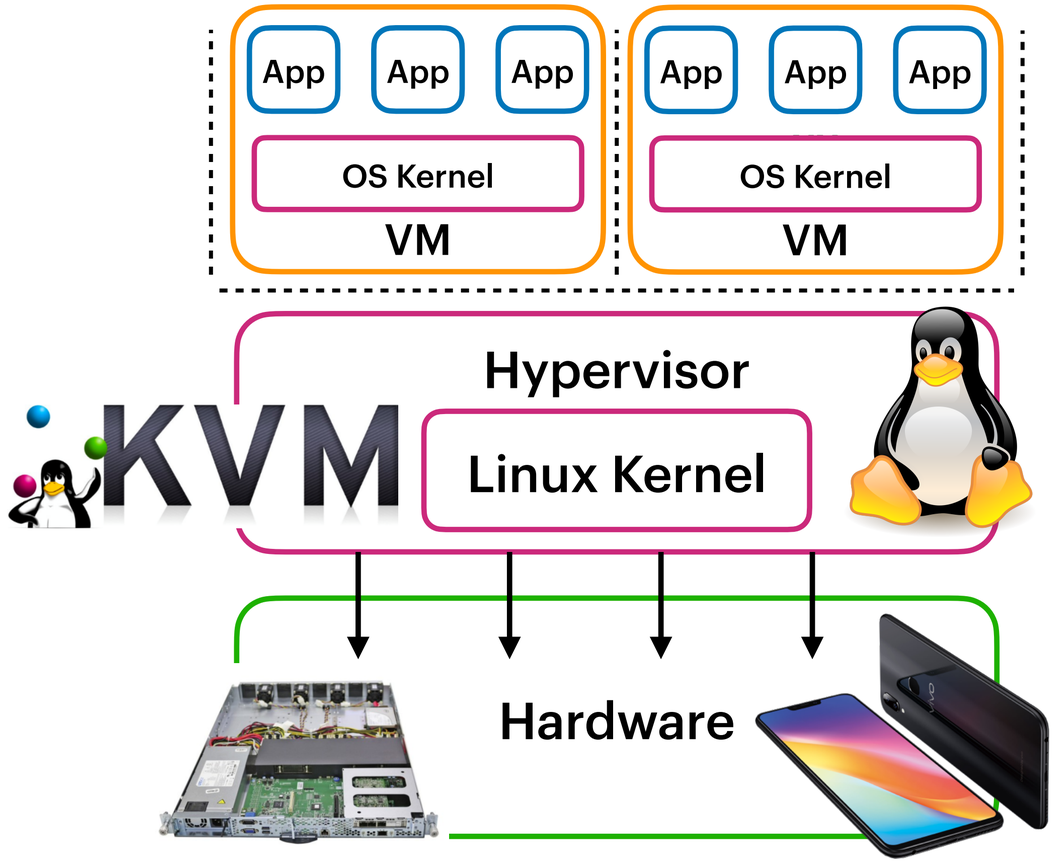
KVM

Microsoft Hyper-v

Xen Project



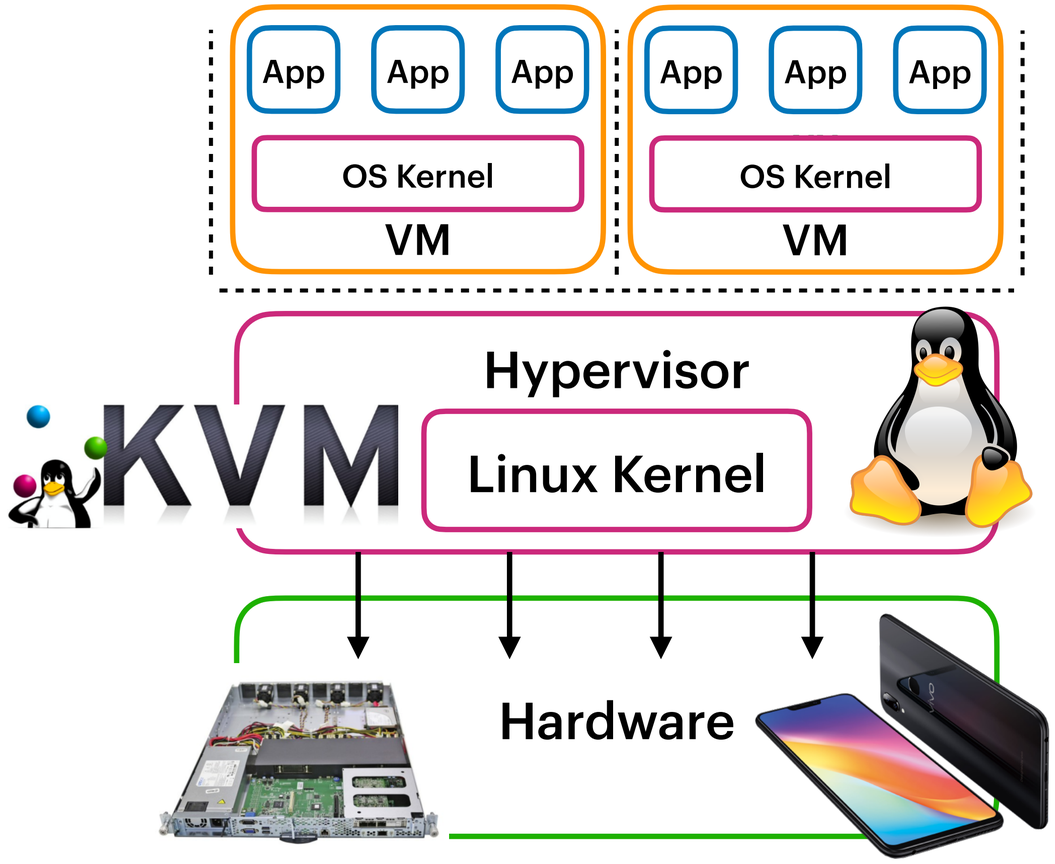
Security Issues in Modern Hypervisors



SIGN IN The Register®

Linux in 2020: 27.8 million lines of code in the kernel

Security Issues in Modern Hypervisors

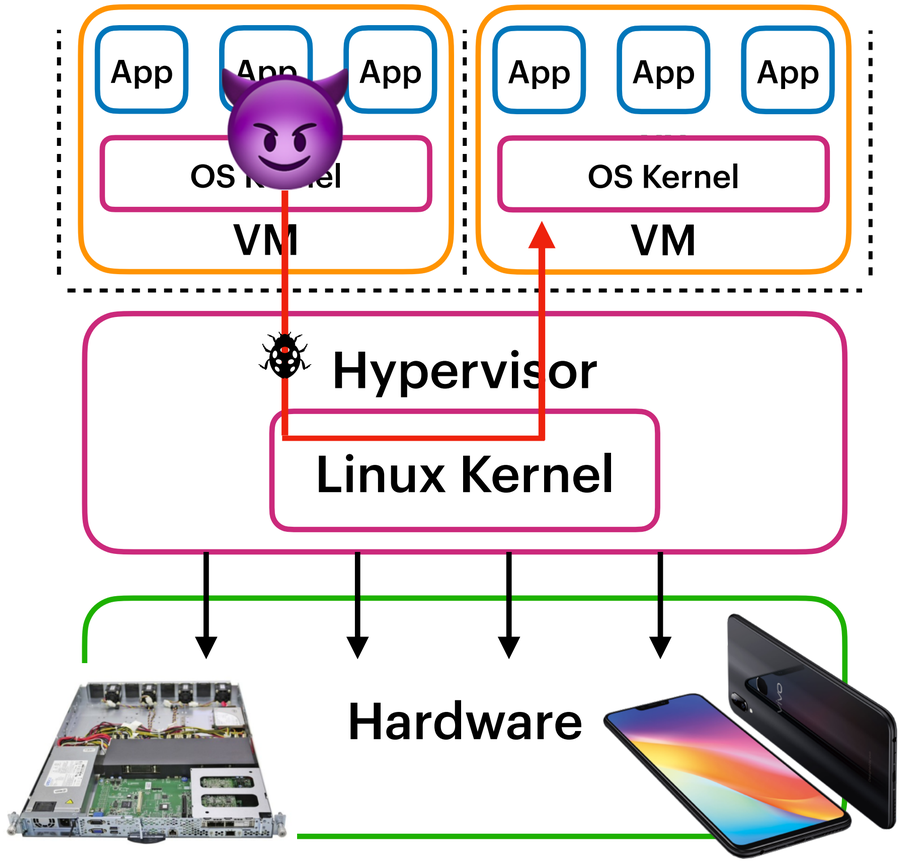


Roughly 2000 bugs (CVEs) were reported in the Linux kernel in the past 10 years!

SIGN IN The Register®

Linux in 2020: 27.8 million lines of code in the kernel

Security Issues in Modern Hypervisors

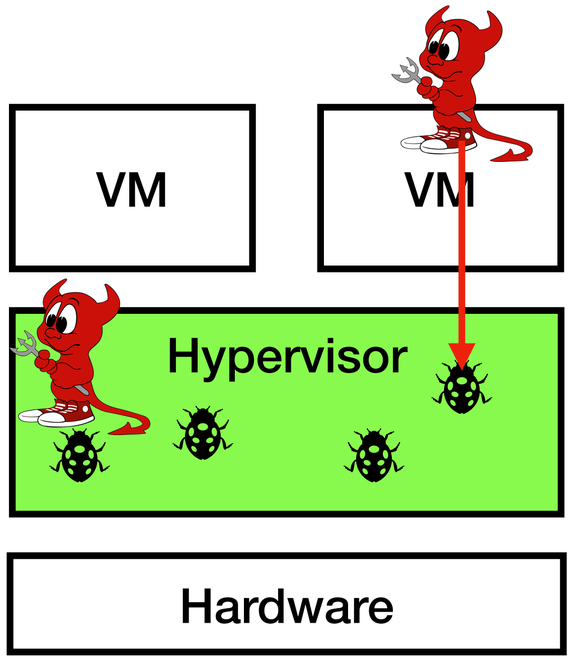


Attackers can exploit bugs in the OS/hypervisor code to gain supervisor privileges!

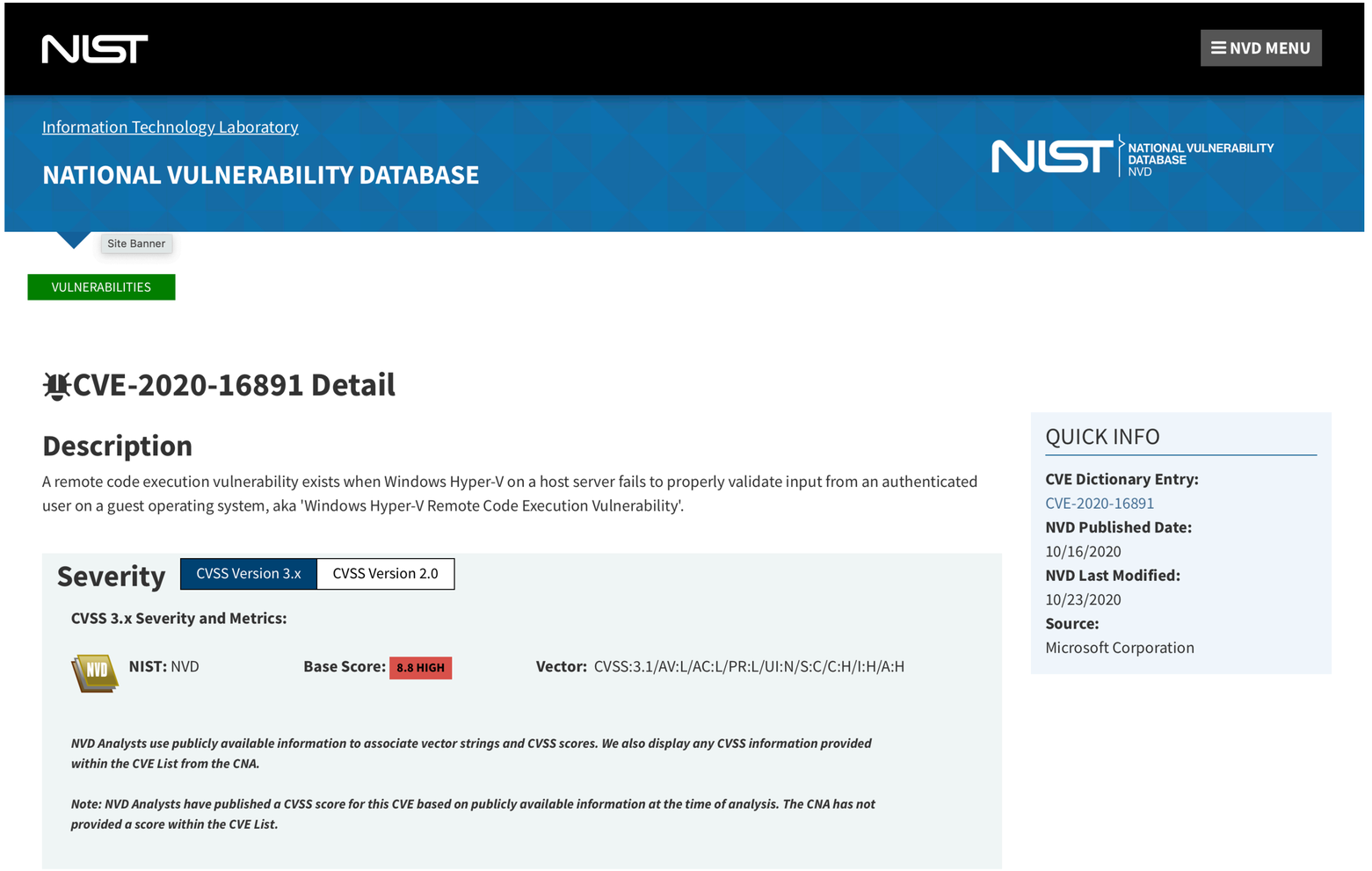
Question: what can the attackers do?

Hypervisor Vulnerabilities

- A malicious host or VM that exploits hypervisor vulnerabilities results in:
  - Code execution:
    - CVE-2020-16891, CVE-2019-0721
  - Privileged escalation:
    - CVE-2020-1047, CVE-2018-18021
  - Information leakage:
    - CVE-2019-1230, CVE-2019-7222



Hypervisor Vulnerabilities



Hypervisor Vulnerabilities

VMware urges emergency action to blunt hypervisor flaws

Critical vulns in USB under ESXi and desktop hypervisors found by Chinese researchers at cracking contest

Simon Sharwood — Thu 7 Mar 2024 // 07:30 UTC

CVE-2024-22252 Detail

AWAITING ANALYSIS

This vulnerability is currently awaiting analysis.

Description

VMware ESXi, Workstation, and Fusion contain a use-after-free vulnerability in the XHCI USB controller. A malicious actor with local administrative privileges on a virtual machine may exploit this issue to execute code as the virtual machine's VMX process running on the host. On ESXi, the exploitation is contained within the VMX sandbox whereas, on Workstation and Fusion, this may lead to code execution on the machine where Workstation or Fusion is installed.

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

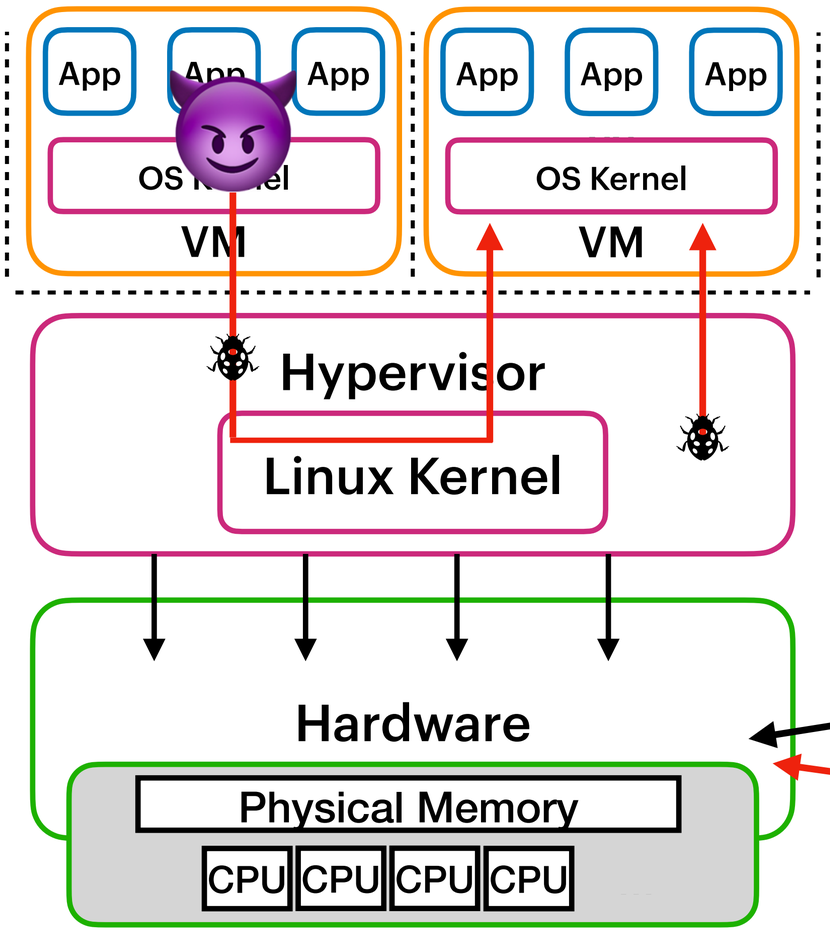
NIST: NVD — **Base Score:** N/A — NVD score not yet provided.

CNA: VMware — **Base Score:** 9.3 CRITICAL — **Vector:** CVSS:3.1/AV:L/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

QUICK INFO

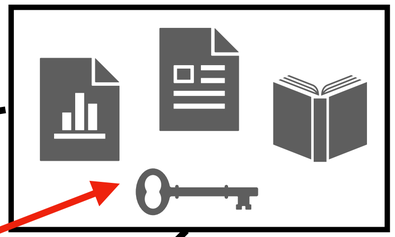
CVE Dictionary Entry: CVE-2024-22252
NVD Published Date: 03/05/2024
NVD Last Modified: 03/05/2024
Source: VMware

Security Issues: Compromising User Data Privacy



A compromised hypervisor has full access to user ***data in use*** from CPU and memory!

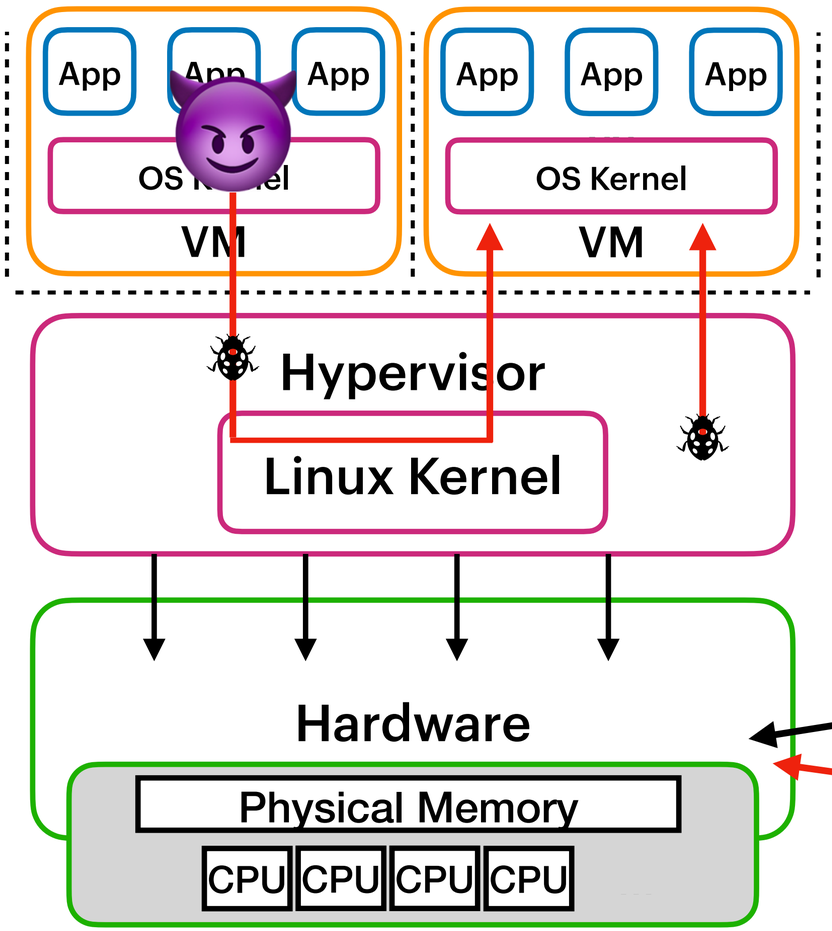
aws



unencrypted

encrypted

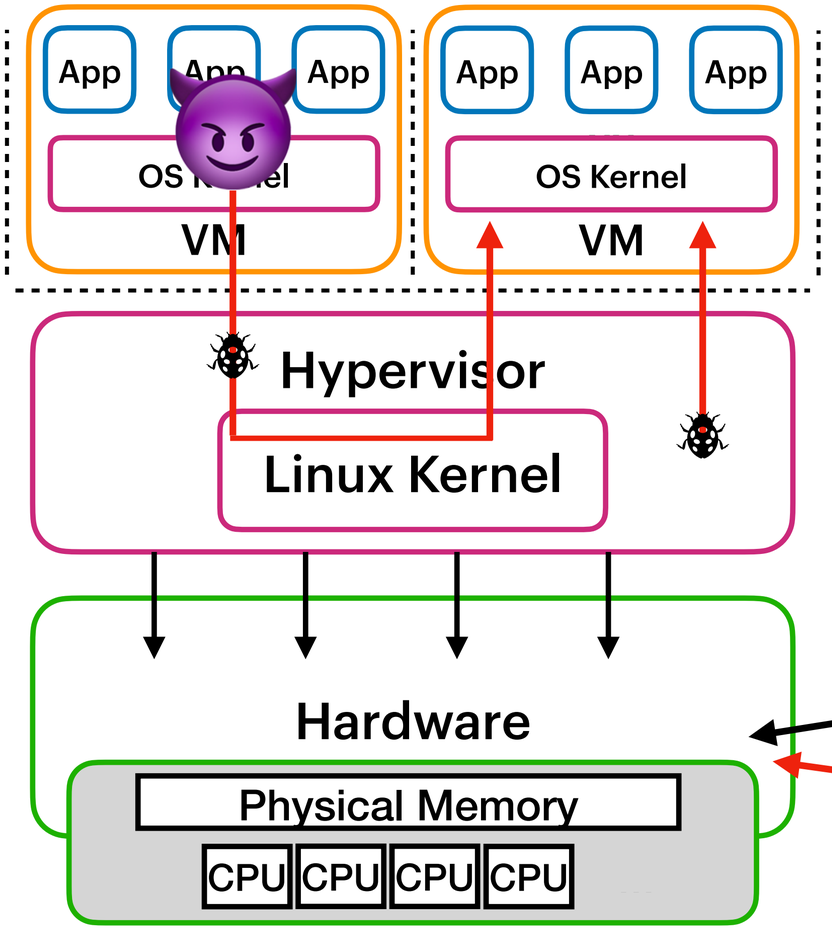
Security Issues: Compromising Program Execution



A compromised hypervisor has full access to user ***data in use*** from CPU and memory!

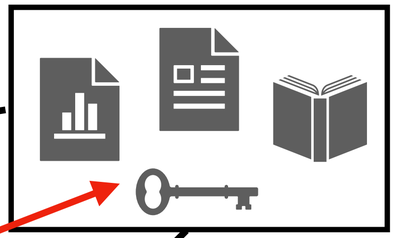
aws

Security Issues: Cloud Admin with Bad Intention



The admin who gains the hypervisor privileges can fully access or modify user data

unencrypted



encrypted

Security Issues: Cloud Admin with Bad Intention

- What can a privileged cloud administrator do?

Security Issues: Cloud Admin with Bad Intention

- What can a privileged cloud administrator do?
  - Dump VM memory to a file
  - Overwrite VM disk file
  - Intercept Network I/O traffic

Problem: potentially buggy systems software with supervisor privileges

Agenda

- Virtual machine based rootkits
- Intro to virtualization-based secure systems
- Improving hypervisor security
  - Problems
  - **New Hypervisor Design**

Virtualization Research?

- Virtualization is an area of focus in the Operating Systems research

Call for Papers

The 28th ACM Symposium on Operating Systems Principles (SOSP) seeks to present exciting, innovative research related to the design, implementation, analysis, evaluation, and deployment of computer systems software.

Topics

SOSP takes a broad view of the systems area and solicits contributions from many fields of systems practice, including, but not limited to, operating systems, file and storage systems, distributed systems, cloud computing, mobile systems, secure and reliable systems, systems aspects of big data and machine learning, embedded systems, virtualization, and management and troubleshooting of complex systems. We also welcome work that explores the interaction with related areas such as computer architecture, networking, programming languages, verification, and databases. In keeping with SOSP tradition, we will favor work that explores new territory, continues a significant research dialogue, or reflects on experience with or measurements of state of the art implementations.

Harden Hypervisors: Xoar

- From the SOSP 11 paper by researchers at Microsoft research: "*Breaking Up is Hard to Do: Security and Functionality in a Commodity Hypervisor*"
- **Goal:** reduce the *attack surface* of the Xen hypervisor
- **Threat Model:** assume the attackers from user VM; Assume the hypervisor is initially benign but could contain bugs; Assume attackers have no physical machine access
- **Mechanism and Policy:** split the privileged Dom0 in Xen into isolated service VMs; Xen enforces modularity and isolation among service VMs

Harden Hypervisors: Xoar

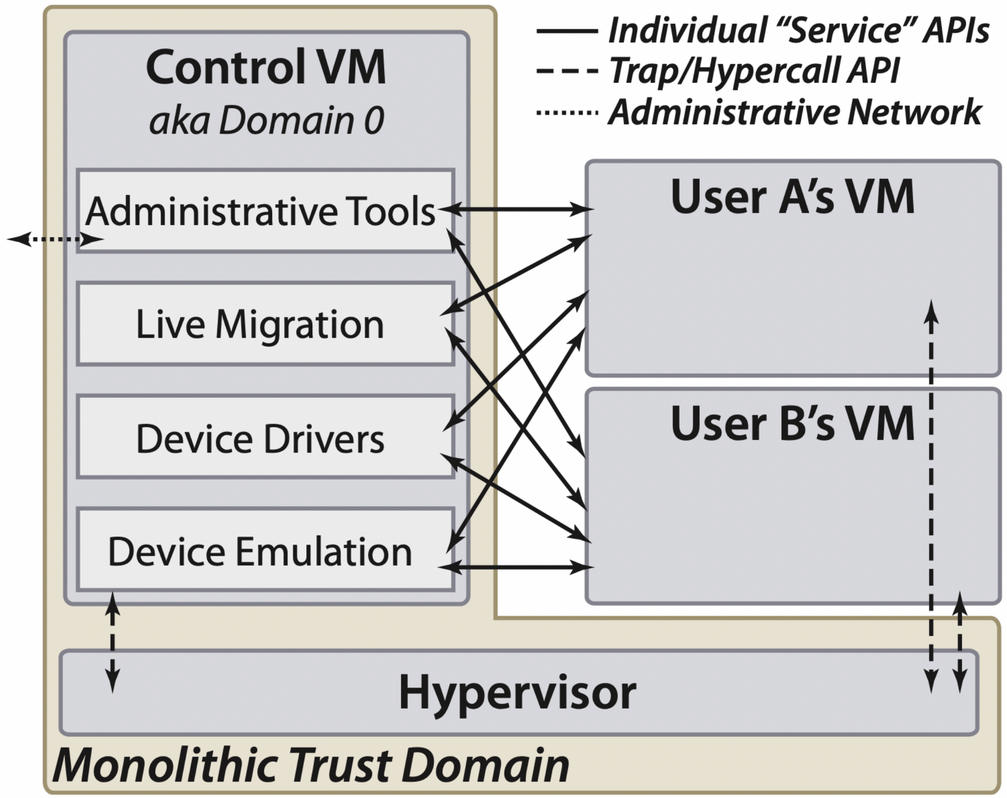
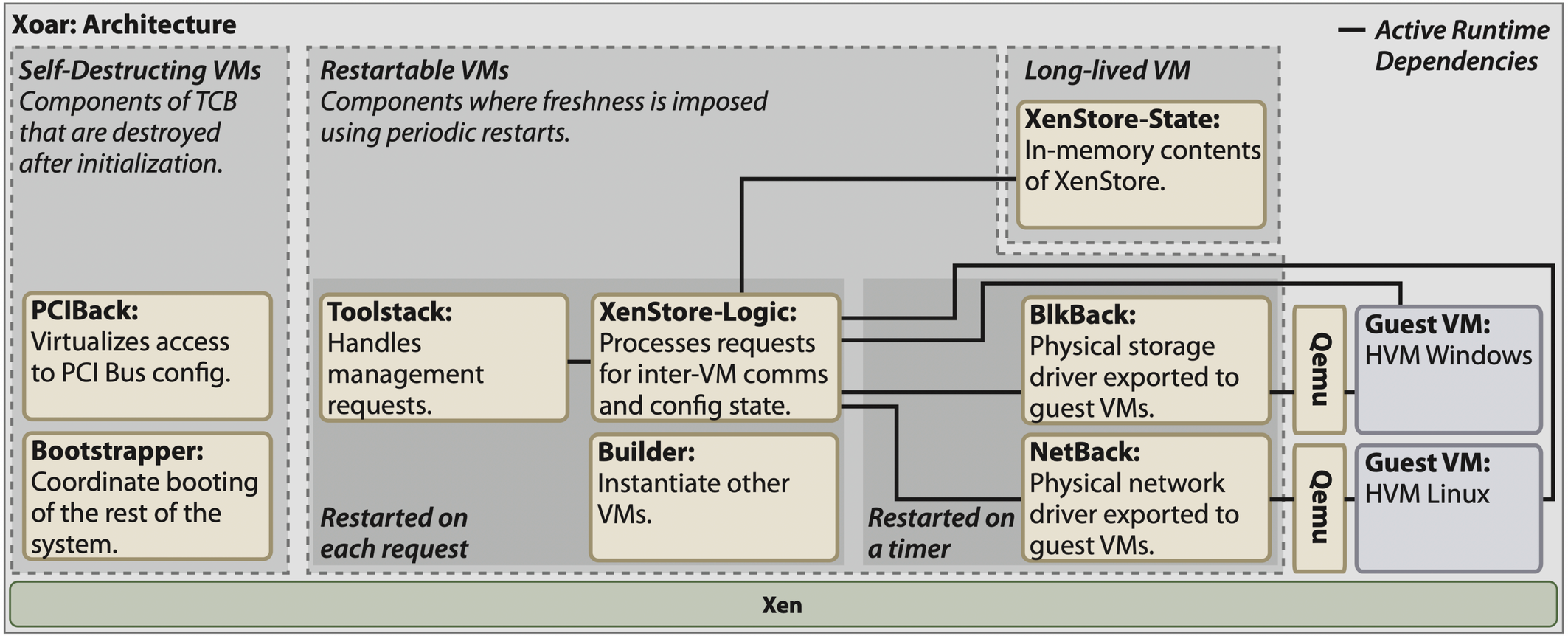


Figure 1: The control VM is often a full operating system install, has privilege similar to the hypervisor, and offers multiple services over numerous interfaces to guest VMs.

Harden Hypervisors: Xoar



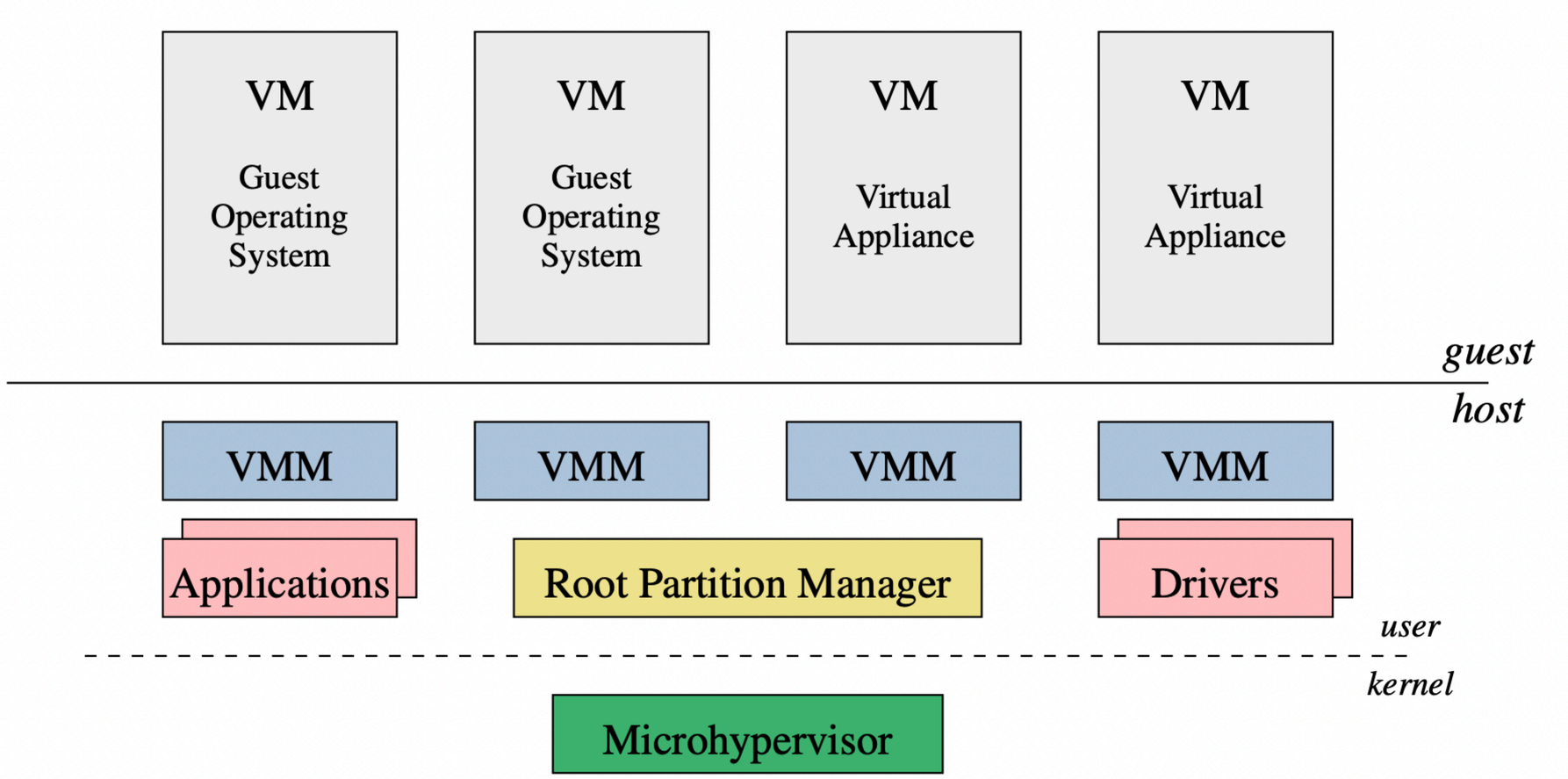
Harden Hypervisors: Limitations

- HyperSafe is applied to a simple hypervisor but not a commodity hypervisor like Xen or KVM - what are the challenges here?
- Cannot offer VM protection if new exploits can bypass the hardening mechanism
- What else can we do to harden hypervisors?

Reducing Hypervisor TCB: NOVA

- **Goal:** reduce the hypervisor TCB
- **Threat model:** assume attackers can exploit software bugs in user-level services and VMM; assume attacker cannot physically steal the machine
- **Mechanism and Policy:** employ a microkernel approach to construct the NOVA hypervisor framework from scratch
  - Include a privileged microhypervisor (TCB), shared user-level services, and per-VM user-level VMM to support virtualization

Reducing Hypervisor TCB: NOVA



Reducing Hypervisor TCB: NoHype

- **Goal:** reduce the hypervisor TCB
- **Threat model:** assume attackers can control the VM; assume attacker cannot physically steal the machine
- **Mechanism and Policy:** employ a thin hypervisor to bootstrap the VM
  - The hypervisor statically partitions CPU and memory to VMs

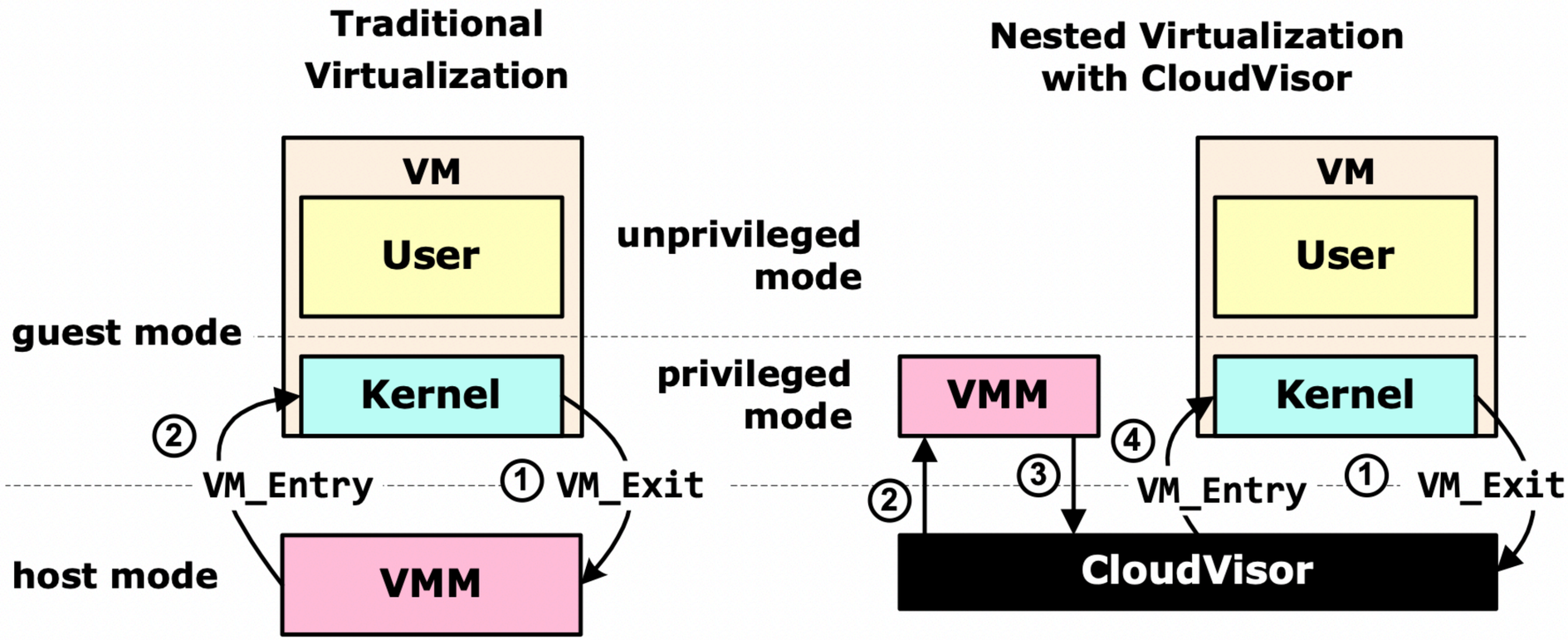
Reducing Hypervisor TCB: Limitations

- Features are lacking:
  - Microkernel based systems like NOVA requires substantial porting effort
  - NoHype cannot dynamically manage VM resources
- NOVA suffers from the performance overhead inherited from the microkernel design

Deprivileging Hypervisors: CloudVisor

- **Goal:** protect VM confidentiality and integrity against a malicious Xen hypervisor
- **Threat model:** assume attackers fully control the hypervisor; assume attackers cannot physically steal the machine
- **Mechanism and Policy:** employ nested virtualization to deprivilege Xen

Deprivileging Hypervisors: CloudVisor



Any Questions?