What Does The Future Hold for Hypervisor Security?

Marc Lacoste
Orange Labs

Workshop on Trustworthy Clouds, ESORICS 2013.
Royal Holloway, University of London, UK, September 12th, 2013.
Major Evolutions in IaaS Architecture Ahead!

Virtualization:
- Fuels growth of cloud computing…
- …but raises many security concerns.

Architecture is fundamental for IaaS security…

… But hypervisor architecture is changing rapidly!
- New hypervisor architectures are defined to mitigate new threats.
- Virtualization is expanding outside the data center.
Major Evolutions in IaaS Architecture Ahead!

- **Virtualization:**
  - Fuels growth of cloud computing…
  - …but raises many security concerns.

- **Architecture is fundamental for IaaS security…**

- **… But hypervisor architecture is changing rapidly!**
  - New architectures are defined to mitigate new threats.
  - Virtualization is expanding outside the data center.

Are current architectures addressing upcoming threats? What is the overall view of such evolutions?
Major Evolutions in IaaS Architecture Ahead!

- **Virtualization:**
  - Fuels growth of cloud computing…
  - …but raises many security concerns.

- **Architecture is fundamental for IaaS security…**

- **… But hypervisor architecture is changing rapidly!**
  - New hypervisor architectures are defined to mitigate new threats.
  - Virtualization is expanding outside the data center.

- **Contributions:**
  1. Identify some major disruptions shaping up the future of hypervisor security.
  2. Abstract hypervisor evolution into a consistent roadmap.
  3. Give an overview of challenges, benefits, limitations of each architectural approach.
Outline

- A Big Picture.
- Trend #1: Extension to Embedded Systems.
- Trend #2: Migration of Security Towards the Hardware.
- Trend #3: Evolution towards Multi-Clouds.
- Conclusion.
A Big Picture
Changes in Hypervisor Security Architecture

Some bottom-line technological trends:

- Availability of increasingly small-scale devices.
- Rising software complexity, commoditization of hardware for dedicated processing.
- Fall of barriers between virtualized systems, increasingly distributed.

Two dimensions in change:

- Scale.
- Abstraction-level.
Changes in Hypervisor Security Architecture

- **Some bottom-line technological trends:**
  - Availability of increasingly small-scale devices.
  - Rising software complexity, commoditization of hardware for dedicated processing.
  - Fall of barriers between virtualized systems, increasingly distributed.

- **Two dimensions in change:**
  - Scale.
  - Abstraction-level.

**Three main trends**

1. Virtualization goes embedded.
2. Security moves towards the hardware.
3. The cloud becomes user-centric.
A Big Picture

Minimalism
Abstraction
Performance
Embedded Constraints
Security

Abstraction
Interoperability
Flexibility
Security

Minimalism
Reduce complexity
Flexibility
Performance
Security
Disruption #1: Virtualization Goes Embedded
Disruption #1: Virtualization Goes Embedded

Virtualization on Chips (multi-core processors)
- Hypervisors on multi-core processors, multi-kernels

Virtualization in Embedded Mobile Devices (phones, tablets, ...)
- Embedded hypervisors (microvisors), micro-kernels

Virtualization in Client Desktop Computers
- Mainstream hypervisors, container-based virtualization

Virtualization in Enterprise Servers
- Mainstream hypervisors
Embedded Hypervisors

Embedded systems features

- Rising complexity
- Expanding code size
- Heterogeneous sub-systems
- Hardware diversity
- Open architectures
- Feature-rich platforms

Security issues

Key design challenges

- Resource abstraction: overcome resource heterogeneity (multicore support, multiple OSes on same platform…).
- Isolation: contain faults/attacks between sub-systems.
- Performance: efficient inter-sub-system communication.
- Minimal TCB: reduce attack surface, strong assurance.
- Real-time guarantees: unique scheduling control point.
- Modularity: facilitate code reuse in open ecosystems.
- Fine-grained resource control: unique control point of security policy enforcement

Source: GreenHills software, Integrity multivisor.


Source: OpenSynergy, COQOS platform.
Embedded Hypervisors

Which Architecture?

- **Hypervisors** have strong limitations.
- **Micro-kernels** seem better suited.
- **Micro-visors** might be even better…

### Traditional hypervisors

- VM multiplexing, isolation
- May be improved (vSwitch)
- Huge TCB
- 2-level scheduling
- Complexity of driver sharing
- Heavyweight VMs

### Key properties

- Resource abstraction
- Isolation
- Performance
- Minimal TCB
- Real-time guarantees
- Modularity
- Fine-grained control

### Micro-kernels

- Increasing virtualization support
- Strong isolation
- Efficient IPCs
- Extremely minimal kernel
- Well-established RTOS approach
- Flexible driver sharing patterns
- Lightweight threads

---

Orange Labs
Microvisor Architectures

- **Microvisor** = convergence of hypervisors and micro-kernels:

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Hypervisor</th>
<th>Micro-kernel</th>
<th>Micro-visior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execution model</td>
<td>VM, vCPU.</td>
<td>Threads.</td>
<td>vCPU.</td>
</tr>
<tr>
<td>Memory</td>
<td>vMMU.</td>
<td>Address space</td>
<td>vMMU.</td>
</tr>
<tr>
<td>I/O</td>
<td>Virtual device drivers in VM or hypervisor</td>
<td>User mode drivers.</td>
<td>User mode drivers.</td>
</tr>
<tr>
<td>Communication</td>
<td>Virtual networks.</td>
<td>IPCs.</td>
<td>Virtualized interrupts.</td>
</tr>
</tbody>
</table>

Abstraction: TCB minimization

- **OKL4 architecture:**

Towards the Cloud-on-Chip

Hypervisors for multi-core architectures

Key challenges

- Resource sharing limitation.
  - Poor physical isolation (memory, storage, CPU, I/O).
  - Failure/attack propagation.

- Massive scalability.
  - Hyperscale server consolidation.
  - Synchronization.
  - Fair resource allocation.

Single hypervisor for multi-cores

- Multi-core management in guest OS: strong scalability restrictions.
- Multi-core management in hypervisor: scalability and security limitations, e.g.,
  - Risk of resource starvation.
  - System-wide hypervisor state sharing.
  - Hypervisor = single point of failure.
  - Hypervisor vulnerabilities poorly confined.
Towards the Cloud-on-Chip

Multiple hypervisors on same chip

- Independent security realms
  - per hypervisor,
  - with dedicated cores and memory.
- Two-level resource management:
  - Intra-hypervisor for VMs.
  - Inter-hypervisor using multiplexing HAL.

Benefits

- Increased resilience:
  - Avoid platform-wide bug/attack propagation through realm confinement.
- Better scalability:
  - Hardware platform = distributed system.
  - Decentralize VMM functionalities for finer-grained control.

Disruption #2: Security Moves Towards the Hardware
Disruption #2: Security Moves Towards the Hardware
VM Introspection

Compute, network, storage introspection…
Fast path, slow path, hybrid path architectures…

In-VM Placement
Detection accuracy: proximity to target
Stealth: protecting the monitoring component

Security Appliance
Security, performance improvements
Less reactive?

Hypervisor-Based
Transparent VM access
Security of monitoring component
Semantic gap
Little remediation actions

VM Introspection Idea: use the capabilities of the hypervisor to supervise VM behaviors

Some Systems

1. In-VM monitoring: SIM
2, 3. With no hooks in VM: CloudSec
2, 3. With hooks in VM: Lares, XenAccess, KVMSec
An Example

vShield = VMware’s IaaS security suite

- **vShield App/Zones**
  Hypervisor-level firewall for VM network security.

- **vShield Manager**
  Centralized administration.

- **vShield Edge**
  Virtual appliance firewall for perimetric security.

- **vShield Endpoint**
  Anti-malware virtual appliance for intra-VM security.

**vShield Endpoint**

- **Security features:** anti-malware, integrity monitoring, firewall, Deep Packet Inspection (DPI), log inspection.
- **Policy-based management.**
- **Cross-layering:** module in hypervisor + security appliance.
- **Openness:** EPSec API.

Source: VMware.
**Micro-Hypervisors**

**The problem**
- Hypervisors are **too big, too complex**.
- Source of vulnerabilities: **bounce attacks**.

**Solutions**
- **TCB hardening**: mechanisms
  - *Protect « by hand » hypervisor from subversion.*
  - Trusted computing, language techniques, sandboxing…
TCB Hardening: Trusted Computing Architectures

- **Security objective:** trustworthy VMM, with high assurance for authenticity and integrity.

  **Trusted computing technologies.**

  Provide attestation of integrity of software/hardware components relying on chain of trust.

For the Hypervisor

<table>
<thead>
<tr>
<th>Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Integrity checking</strong></td>
</tr>
<tr>
<td>TCG IMA, Hyperguard, HyperCheck, HyperSentry</td>
</tr>
<tr>
<td><strong>Control flow integrity</strong></td>
</tr>
<tr>
<td>HyperSafe</td>
</tr>
</tbody>
</table>
TCB Hardening: Trusted Computing Architectures

For VMs

- Monitored VM e.g., for integrity
- Management VM
  - 1. Monitoring agent
  - 2. Monitoring agent

Systems

- Trusted VMM
  - Terra + TPM
- In management VM
  - vTPM

Hypervisor

Host OS drivers ??
Benefits and Limitations

**Strong security:** atestation capabilities.

**Vulnerable if software-only. Stealth? SMM vulnerabilities?**

**Flexibility:** different security policies

**Limited to integrity measurement. No remediation.**

**Easy to perform statically**

**In-context measurement is hard: hypervisor or processor context?**
TCB Hardening: Driver Sandboxing

Idea: confine malicious code by controlling communications between driver, and device, kernel, and VM space.

Example of Systems

1. **Reference Monitor (RM) between driver / VM space:** MicroDrivers, Proxos
2. **RM between driver and hypervisor:** Software Fault Isolation (SFI) techniques
3. **RM between driver and device:** Nooks

### Benefits

- **Strong security**
- **Good performance**
- **Reduced code size**
- **Some isolation flexibility**

### Challenges

- **RM difficult to protect without hardware mechanism**
- **No remediation, only containment**
- **Hypervisor is modified**
- **Policies difficult to configure**
Micro-Hypervisors

The problem
- Hypervisors are too big, too complex.
- Source of vulnerabilities: bounce attacks.

Solutions
- **TCB hardening**: mechanisms
  - Protect « by hand » hypervisor from subversion.
  - Trusted computing, language techniques, sandboxing…
- **TCB reduction**: architectures
  - Reduce code size and complexity and increase modularity.
  - For the core hypervisor: Micro-hypervisors.
  - For the management VM: Disaggregated hypervisors.

Reducing the TCB

Core hypervisor: virtualization
iKernel (for drivers), NOVA, NoHype

Expel as much code as possible from TCB
- Strong security
- Flexibility with open architecture.
- Extensive code rewriting
- Limited operational services
- Hard to apply to legacy hypervisors.

Orange Labs
Micro-Hypervisors

The problem

- Hypervisors are too big, too complex.
- Source of vulnerabilities: bounce attacks.

Solutions

- **TCB hardening:** mechanisms
  - Protect « by hand » hypervisor from subversion.
    - Trusted computing, language techniques, sandboxing...
- **TCB reduction:** architectures
  - Reduce code size and complexity and increase modularity.
    - For the core hypervisor: Micro-hypervisors.
    - For the management VM: Disaggregated hypervisors.

Reducing the TCB

Management VM: componentization

XOAR, MinV, Disaggregated Xen

Transform Dom0 into a set of service VMs, limiting resource sharing, reducing privileges.

- Improved security, flexibility, and control.
- Does not limit operational services.
- More ready to apply to legacy hypervisors.
Some Examples

NOVA Architecture


XOAR Architecture

security component heterogeneity between layers and domains. infrastructure complexity $\Rightarrow$ impossibility of manual administration.

**Autonomic security approach**: clouds with self-defense capabilities

- Lighter administration.
- Increased reactivity.
- Lower operational costs.
- Graduated response.
- Security supervision enabler.
VESPA: Multi-Layer IaaS Self-Protection

An autonomic security framework for regulating protection of IaaS resources.

**Implementation:** KVM-based IaaS infrastructure.

**Application to hypervisor self-protection:** in progress.
Example: The VESPA Framework

Key points

- VESPA: architecture for effective and flexible IaaS self-protection.
- Two-level tuning of security policies, within and across layers.
- Coordination of multiple loops for rich spectrum of defense strategy.
- Multi-plane open design for easy integration of detection/reaction COTS.
Flexible confinement of VMs according to risk level
Virtualized Hypervisors

The problem

IaaS infrastructures lack:

Vertically: security
  - Untrustworthy, vulnerable layers.

Horizontally: flexibility, interoperability
  - (Security) features not deployed.
  - Too monolithic for customization.
Virtualized Hypervisors

Idea: Virtualize the hypervisor

Hypervisor-Secure Virtualization (HSV):
- The hypervisor is no longer part of the TCB.
- Protection by a security layer underneath.
- Separation of resource management from security.

Software HSV approach: nested virtualization.

Source: IBM, Turtles project, OSDI’10.
Virtualized Hypervisors

Benefits

Vertically: more security
- Trustworthy security layer.

Horizontally: more flexibility, interoperability
- Distributed security abstraction layer.
- Enabler for cross-provider security services.

Source: Zhang et al., CloudVisor, SOSP’11.
The Hypervisor in Hardware

**Hardware HSV**

A hardware controller as only security manager.
- Dedicated Page Ownership Tables for checking memory mapping permissions.

The VMM performs transparently VM scheduling and resource allocation.

**Benefits**

Stronger security and better performance than software solutions

Cost might no longer be a barrier:
- Changes in micro-architecture are fairly small.
- Providers might pay for extra assurance level.

Disruption #3: Evolution Towards Multi-Clouds

Trend #1: Extension to Embedded Systems

- Hypervisor for “cloud-on-chip”
- Embedded hypervisor

Trend #2: Evolution Towards Hardware

- Hypervisor in hardware
- Virtualized hypervisor
- Micro-hypervisor

Trend #3: Evolution Towards Multi-Clouds

- Distributed hypervisor
- Data center hypervisor
Towards User-Centric Clouds

Provider-centric cloud deficiencies

- Lack of unified control: vendor lock-in, monolithic infrastructures
- Lack of interoperability: for infrastructure services
Towards User-Centric Clouds

User-centric clouds (super-clouds)

- **Cloud resource distribution plane.**
- **Benefits:**
  - Independence from provider.
  - Increased customizability.
  - New business opportunities.
Towards User-Centric Clouds

Towards fully distributed hypervisors...

- Split infrastructure into provider- / user-controlled domains/modules.
- Some design alternatives:
  - Extensible hypervisors [« Unshackle the Cloud! », HotCloud’11].
  - Modular management interface [« Towards Self-Service Clouds », CCS’12].
  - Nested virtualization [XenBlanket, EUROSYS’12;Inception, USENIX ATC’13].

A research domain in full expansion...
Exploitation of virtualization vulnerabilities are some of the most serious cloud threats, making the hypervisor a keystone component of cloud security.

Looking back...
Exploitation of virtualization vulnerabilities are some of the most serious cloud threats, making the hypervisor a keystone component of cloud security.

Looking back…

- The main challenges are rising infrastructure complexity and rapid threat evolution.
- Mechanisms are not well integrated. New architectures are promising but far from mature.
- Two ultimate goals are cross-layer protection and end-to-end security.
Exploitation of virtualization vulnerabilities are some of the most serious cloud threats, making the hypervisor a keystone component of cloud security.

Looking back…

- The main challenges are rising infrastructure complexity and rapid threat evolution.
- Mechanisms are not well integrated. New architectures are promising but far from mature.
- Two ultimate goals are cross-layer protection and end-to-end security.
- As virtualization expands, not one but multiple « good » security architectures.

⇒ A fast moving research domain…
⇒ …critical to monitor to protect future cloud systems.
Thanks!

Contact: Marc Lacoste
Orange Labs
Senior Research Scientist
38-40 rue du Général Leclerc
92794 Issy-Les-Moulineaux, France
marc.lacoste@orange.com