IaaS Clouds: Which Security for VMs and Hypervisors?

Marc Lacoste
Orange Labs

ICAR (Intergiciel et Construction d'Applications Réparties) Summer School.
Grenoble, August 28th, 2013.
The Two Facets of Cloud Computing

Many benefits
- Cost reduction.
- Flexibility.
- Scalability.
- Pay-per-use...

Many forms
- Private, public, hybrid, community.
- IaaS, PaaS, SaaS.
- Data center, mobile, personal, on chip...

Virtualized resources for multiple services

Many threats

Virtualization layer
- VM-to-VM.
- Rootkits: Bluepill, CloudBurst, Virtunoid...

Amazon's cloud nightmare

The exact feature that was supposed to be Amazon EC2's strength -- reliability -- is what failed and brought the cloud low yesterday. Still, cloud computing isn't going anywhere.

By Dana Mitchell, contributor

FORTUNE -- The snafu at Amazon's EC2 hosting service on Thursday, which knocked several big web sites out of service, is being called a "black eye" for the cloud-computing business -- a "we told you so" moment, according to cloud critics. But it could simply be a black eye for EC2.
Outline

- Part I: IaaS Threats and Security Challenges.
- Part II: VM Security.
- Part IV: What does the Future Holds for IaaS Security?
Part I: IaaS Threats and Security Challenges
Threats
A Typical IaaS Infrastructure

Examples:

- **Hey You! Get Off My Cloud!** on Amazon VMs [Ristenpart et al., CCS’09].
- **Cross-VM Side Channels and Their Use to Extract Private Keys** [Zhang et al. CCS’12].

VM-to-VM threats

- Fool VM placement strategy to become co-located with VM attack target.
- Launch side-channel attack to steal/corrupt information from target VM.
Threats in a IaaS Infrastructure

Examples:
- **Virtunoid**: KVM isolation breakout [Elhage, DEFCON’11].
- **CloudBurst**: VMware guest VM escape [Kortchinsky et al. BLACKHAT'09].
- **Bluepill**: rogue hypervisor beneath VMs [Rutkowska et al., BLACKHAT’06].
- **SubVirt**: VM-based rootkit [King et al., Security&Privacy’06].

Hypervisor subversion
- Compromise VMM from malicious VM.
- Misconfigurations, device drivers.
- Threaten hypervisor integrity, CIA attacks against VMs.
Network threats

- Traffic snooping.
- Address spoofing.
- VLAN hopping.

Example:
- Critical vulnerability in Eucalyptus open source cloud (2011).
Threats in a IaaS Infrastructure

Availability threats

- Resource starvation due to faulty or malicious VM behavior.
- Crimeware-as-a-Service.

Examples:

- Major outage on Amazon EC2 storage (2011).
- DDoS attack on AWS brings Bitbucket services to a halt (2009).
- EC2 cloud used against Sony’s PlayStation Network (2011).
BotClouds Threats

BotCloud attacks - are cloud providers prepared?

botCloud – an emerging platform for cyber-attacks

Hosting network services on Cloud platforms is getting more and more popular. It is not in the scope of this article to elaborate the advantage of using Cloud computing, instead, as the title of might have already inspired you, here we discuss the potential benefits available to malicious entities in using a Cloud platform (CP). In particular, we are going to see:

- What benefits do attackers get by using CP for their nefarious purposes?
- Can a CP be programmed to launch security attacks, propagate malware, or perform denial-of-service attacks?
- Are the current security features of CP providers robust in their detection and prevention of malicious usage?

The questions above were based a research study conducted at the Stratesec IT Security Winter School 2012[1]. The objective of this research was to investigate the security posture of Cloud providers in protecting against malicious usage (the security point of view), as well as assessing the effectiveness of such CPs for launching malicious activities (the attacker point of view). We define “botCloud” as a group of Cloud instances that are commanded and controlled by a malicious entity to initiate cyber-attacks.

BotCloud hosting is not something that’s been spoken about much but it’s something that could spell trouble for cloud providers

When talking about cloud security, most people quite rightly think in terms of securing the data that they themselves are storing or accessing through the cloud.

Quite often this is limited pretty much to the areas of authentication, encryption and regulatory compliance. Those with a little more understanding of the field may broaden the definition to include predictable availability and data recovery, application security and privacy issues surrounding e-discovery and physical access to datasets with an emphasis on logging and audit trails.

The reason I have gone over old ground and stated the obvious is simple, there’s one area of cloud security that is all too often overlooked with potential dire consequences: botCloud hosting.
Security Challenges
Endpoint Security

Guarantee security when computing resources are virtualized.

Virtualization brings many threats.

Hyperjacking, misconfigurations, malicious device drivers, backdoors between VM and hardware...

VMs also have their vulnerabilities.

Such threats may be mitigated by:

- Hardened images.
- Strict VM security life-cycle management.

⇒ Apply security-by-default configurations.
Network Security

Guarantee security when network resources are virtualized.

Isolation is no longer physical but logical.

- Isolation is less precise.
- Security guarantees are weaker.

Challenge: mapping existing network security components to new cloud architectures.

Is traditional security still effective?

- Risks are similar to known networks.
- Many mechanisms are still applicable.

VPNs, VLANs, firewalls, IDS/IPS, encryption, signature…
Network Security

Guarantee security when network resources are virtualized.

Fully automated security management is still lacking.

- **Research:** autonomic (self-protecting) security architectures.
- **Operational:** early warning, proactive security systems.

**Barrier #3: Elastic Security**

Flexible security provisioning to match fast evolving risks.
- **First solutions:**
  - Flexible management of VPNs.
  - Overlays in full network virtualization.

**Fully automated security management is still lacking.**
Data Protection

Guarantee data security in a shared multi-tenant environment.

Barrier #4: Identity

Lack of end-to-end identity management

- Issues: scalability, heterogeneity.
- Authentication: should be overcome.
- Authorization: in its infancy.
- Security-as-a-Service opportunities.

Barrier #5: Privacy

Strong isolation during personal information life-cycle.

- Many tough questions:
  - Secure data storage, data retention and destruction, legal implications...
- Today’s PETs are not enough!
Data Protection

Guarantee data security in a shared multi-tenant environment.

Barrier #6: Traceability

How to locate the data and its path?

Legal, political, and trust issues:

- Compliance.
- Data hosted abroad exposed to foreign governments.
- Proving data comes from a trusted source?

Barrier #7: Legal issues

Multiple conflicting jurisdictions for cloud data flows.

- **Providers**: how to provide assurance of regulation compliance?
- **Customers**: what are the rights and obligations of each party?

⇒ Importance of security SLAs.

Marc Lacoste, Orange Labs, ICAR 2013
Trust Enablers

*Prove to third parties that the cloud infrastructure is trustworthy.*

**Barrier #9: Openness**

*Avoid vendor lock-in.*

- **Main issue:** API portability across providers.
- **Basis of inter-cloud infrastructures.**
- **Flexibility and security benefits of open source cloud architectures.**

**Barrier #8: Transparency**

*Prove security hygiene of provider infrastructure to third parties.*

- **Auditability,** certification process, risk analysis methodologies, **compliance.**
- **Trusted cloud computing** technologies provide cryptographic evidence.
- **Clear-cut SLAs** to clarify responsibilities.

**Barrier #10: End-to-End Security**

*Orchestrating security mechanisms.*

- A **cloud reference security architecture** is needed for overall view of cloud security.
- **Importance of standardization.**

Part II: Protecting Virtual Machines
Virtual Security Appliances

Security objective: 360° confidentiality, integrity, and availability of VMs.

Key properties
- **Isolation**: control distributed information flow between VMs.
  ⇒ **Zoning**.
- **Oversight**: observe and intervene in VM state / behavior.
  ⇒ **Introspection**.

Design issues
- **Horizontal**: which network security architecture?
  ⇒ Physical, virtual, hybrid…
- **Vertical**: which software layer?
  ⇒ vSwitch, hypervisor, VM, multi-layer…

Zoning Architectures
1. Physical
2. Virtual
3. Hybrid
VM Introspection

Limitations of pure network-based and host-based monitoring for cloud infrastructures.
VM Introspection

- **Monitored VM**
  1. Monitoring agent

- **Security VM (Virtual Appliance)**
  2. Monitoring agent

- **Hypervisor**
  3. Monitoring agent

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

**Some Systems**

<table>
<thead>
<tr>
<th></th>
<th>In-VM monitoring:</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SIM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>With no hooks in VM:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2, 3</td>
<td>CloudSec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>With hooks in VM:</th>
</tr>
</thead>
<tbody>
<tr>
<td>2,3</td>
<td>Lares, XenAccess, KVMSec</td>
</tr>
</tbody>
</table>

**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

Check out paper « Engineering Intrusions Services for IaaS Clouds : The Way of The Hypervisor » at IEEE SOSE 2013 for more information!!

Marc Lacoste, Orange Labs, ICAR 2013
An extensive number of generic techniques for intrusion and malware detection:
increasingly use virtualization to mitigate both known and unknown threats.

**Policies:** some flexibility.

**Cross-layering:** some attempts using VMI and semantic-view reconstruction.

**Openness:** to enable selection and composition of multiple detection / reaction algorithms.

---


---

Marc Lacoste, Orange Labs, ICAR 2013
vShield EndPoint

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:** EPsSec API.

Marc Lacoste, Orange Labs, ICAR 2013

Source: VMware.
Part III: Protecting the Hypervisor
Virtualization Revisited
From System Virtualization to the Hypervisor

System virtualization is the use of “an encapsulating software layer that surrounds or underlies an operating system, providing the same inputs, outputs, and behavior that would be expected from physical hardware”. M. Pearce et al. Virtualization: Issues, Security Threats, and Solutions. ACM Computing Surveys, 45(2), 2013.

This task is performed by the hypervisor or Virtual Machine Monitor (VMM):
- Allocation of physical resources (e.g., CPU, storage, network) to Virtual Machines (VMs).
- VM isolation.

**Type I: Bare Metal**
- Hypervisor provides its own drivers

**Type II: Hosted**
- Drivers may be shared with the host OS

---

Marc Lacoste, Orange Labs, ICAR 2013
Properties of a Virtualized Architecture

- **Theoretical foundations [Popek and Goldberg74]:** Analysis of requirements for a physical architecture to be virtualizable.

- **VMM requirements:**

  1. **Efficiency**  
     The major part of instructions must be run directly on the CPU, without VMM intervention.

  2. **Resource Control**  
     The VMM must be in complete control over physical resources, e.g., for multiplexing, isolation, complete mediation.

  3. **Equivalence**  
     Programs running on the VMM must have the same behavior as if running directly on an equivalent physical machine.
Instructions

Sensitivity

- **Sensitive**: may interfere with a factor under VMM control.
  - Control sensitive.
  - Behavior sensitive.
- **Innocuous** otherwise.

Privilege level

- **Privileged**: requires process to be highly privileged to be called.
  - Traps if CPU in user mode
  - Not if CPU in supervisor mode.
- **Non-privileged** otherwise.

**Result**: The architecture is fully virtualizable if:

*Sensitive Instructions* \( \subseteq \) *Privileged Instructions.*

- « Trap-and-emulate » approach to virtualization.
- Unfortunately, x86 architecture is not fully virtualizable!
**Virtualization and x86 Privilege Rings**

- **x86 architecture defines 4 protection rings.**

**Software-approach to virtualization:**
- VMM runs in ring 0, guest OS in ring 1.
- Some sensitive instructions may not work properly due to insufficient privileges.

**Hardware-assisted virtualization:**
- VMM runs in hardware-enforced “ring -1”.
- The OS can run transparently in ring 0 as in non-virtualized systems.
Some Methods for Virtualizing the x86 Architecture

- **Paravirtualisation:** The guest OS is modified to better cooperate with the hypervisor. Sensitive non-privileged instructions are replaced by hypercalls.
  - Only a limited number of paravirtualized drivers are needed.
  - Not compatible with proprietary kernels.

- **Binary translation:** The VMM converts “problem” instructions in smoother binary code
  - Compatible with most guest OSes. Does not require specific hardware support.
  - Requires many optimizations to be efficient.

- **Hardware-assisted virtualization:** The hardware facilitates virtualization with specific instructions (e.g., Intel VT-x). The guest OS runs transparently without modifications.
  - Allows to run OS which cannot be paravirtualized. Security is also enhanced.
  - Hardware context switching might be costly. Implementation may also be difficult.
I/O Management

Device drivers implementation:

- **Virtualized**: split (back-end/front-end), emulated (HVM), or hypervisor direct.
- **Passthrough**: from guest OS driver to device without hypervisor intervention.
I/O Management

Device drivers implementation:

- **Virtualized**: split (back-end/front-end), emulated (HVM), or hypervisor direct.
- **Passthrough**: from guest OS driver to device without hypervisor intervention.

Maps device addresses to physical addresses in main memory. Very useful to mitigate DMA attacks.

Marc Lacoste, Orange Labs, ICAR 2013
Network Management

The VM virtual network interface (VNIC) can be:

- **Bridged** to a physical network interface (PNIC).
- **Part of a Virtual Network (VN):** devices are connected to (virtual) hubs, connected to other hubs or to physical networks via virtual routers.

VNs can be **isolated**, **routed**, or **NAT-routed**.

Marc Lacoste, Orange Labs, ICAR 2013
Hypervisor Security Mechanisms
Hypervisor Integrity and Authenticity Threats

- The VMM is implicitly trusted. Is this really true?
- **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>
<th>Integrity checking</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TCG IMA, Hyperguard, HyperCheck, HyperSentry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Systems</th>
<th>Control flow integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HyperSafe</td>
</tr>
</tbody>
</table>

Marc Lacoste, Orange Labs, ICAR 2013
Hypervisor Integrity and Authenticity Threats

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

**Host OS drivers ??**

Marc Lacoste, Orange Labs, ICAR 2013
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?
## DoS Threats

Threats against *availability*.

<table>
<thead>
<tr>
<th>Local Threats</th>
<th>Network Threats</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>VMM bugs</strong>: privilege escalation, VMM crash, disable access to administration channels.</td>
<td><strong>Attack vectors</strong>: network channels, e.g., network flooding.</td>
</tr>
<tr>
<td><strong>Resource starvation</strong>: allocation of too many resources causing failure of other components.</td>
<td><strong>Targets</strong>: VMM, host OS, guest OSes.</td>
</tr>
<tr>
<td>- From host or VMs.</td>
<td><strong>Mitigation</strong>: network security countermeasures (NIDS, firewalls, sinkholes).</td>
</tr>
<tr>
<td>- <strong>Mitigation</strong>: resource allocation limits.</td>
<td>- <em>Level 2</em>: ebtables (Xen/KVM)…</td>
</tr>
<tr>
<td></td>
<td>- <em>Level 3</em>: Open Vswitch (VMware), CISCO Nexus 1000v (Xen/KVM).</td>
</tr>
</tbody>
</table>
Information Leakage and Privilege Escalation Threats

Threats against **confidentiality** and **integrity**.

<table>
<thead>
<tr>
<th>VM information leakage</th>
<th>Privilege escalation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exfiltrating information out of VMs</strong> using covert channels (hardware and software).</td>
<td><strong>VM escape</strong>: a VM breaks the hypervisor isolation code to become over-privileged.</td>
</tr>
<tr>
<td>Such channels may also serve to <strong>corrupt VMs</strong>.</td>
<td><strong>Escape to host</strong>:</td>
</tr>
<tr>
<td><strong>Leaked information</strong>: customer data, resource usage, location, host or network information.</td>
<td></td>
</tr>
<tr>
<td>Two representative classes of attacks:</td>
<td><strong>Mitigation</strong>: VMM sandboxing, host security.</td>
</tr>
<tr>
<td>▪ <strong>Cache-based attacks</strong>.</td>
<td></td>
</tr>
<tr>
<td>▪ <strong>Timing-based attacks</strong>.</td>
<td><strong>Mitigation</strong>: +Trusted Virtual Domains.</td>
</tr>
<tr>
<td>Very few protections against such threats.</td>
<td><strong>Escape to Virtual Networks</strong>.</td>
</tr>
<tr>
<td></td>
<td><strong>Mitigation</strong>: + network security.</td>
</tr>
</tbody>
</table>

Marc Lacoste, Orange Labs, ICAR 2013
The « Hey You! Get Off My Cloud » Attack

1. **Map the Cloud**
   Identify potential targets

2. **Determine co-residence**
   Check if two VMs are co-located on same physical server

3. **Send probe VM**
   Co-locate attacker VM with target

4. **Use VM side-channel**
   Extract information, perform DoS

Example: infer number of web site visitors from traffic load.
Sandboxing Device Driver Threats

**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**

### Limitations

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

---

Marc Lacoste, Orange Labs, ICAR 2013
Part IV: What does the Future Holds for IaaS Security?
Major Evolutions in IaaS Architecture Ahead

- Architecture is fundamental for IaaS security...
- ... But hypervisor architecture is changing rapidly!
  - New hypervisor architecture are defined to mitigate new threats.
  - Virtualization is expanding outside the data center.

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

- A Big Picture...

Three main trends
1. Virtualization goes embedded.
2. Security moves towards the hardware.
3. The cloud becomes user-centric.
Major Evolutions in IaaS Architecture Ahead!

Trend #1: Extension to Embedded Systems
- Hypervisor for "cloud-on-chip"
- Embedded hypervisor
- Micro-hypervisor
- Virtualized hypervisor
- Hypervisor in hardware

Trend #2: Evolution Towards Hardware

Trend #3: Evolution Towards Multi-Clouds
- Distributed hypervisor

Abstraction Level

Highly Embedded  Scale  Highly Distributed

Marc Lacoste, Orange Labs, ICAR 2013
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

Cloud Generation IV
Cloud Generation III
Mobile Cloud
Cloud Generation II
Virtualized Desktop
Cloud Generation I
Virtualized Data Center

Marc Lacoste, Orange Labs, ICAR 2013
Embedded Hypervisors

Hypervisors for mobile phones, sensors, automotive systems, avionics…

Applications
- BYOD.
- Segment professional / personal environments…

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

Which Architecture?
- Hypervisors have strong limitations.
- Micro-kernels seem better suited.
- Micro-visors are perhaps even better.
- One of the most advanced designs is OKL4 (Open Kernel Labs).

Marc Lacoste, Orange Labs, ICAR 2013
Microvisor Architectures

Microvisor = convergence of hypervisors and micro-kernels:

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Hypervisor</th>
<th>Micro-kernel</th>
<th>Micro-visor</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>


OKL4 architecture:

Source: Marc Lacoste, Orange Labs, ICAR 2013
Towards the Cloud-on-Chip

Hypervisors for multi-core architectures

Key properties

- Strong resource sharing limitation.
- Massive scalability.

Key points

- Multiple hypervisors on the 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.

Source: Intel.

Disruption #2: Security Moves Towards the Hardware
**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.

Marc Lacoste, Orange Labs, ICAR 2013
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

DC Hypervisor
Micro-hypervisors
Virtualized hypervisors

NOVA Architecture


XOAR Architecture

Some hard problems

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

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

- **Decision**
  - Decide to reconfigure security mechanisms
  - Derive security context by attribute aggregation
- **Detection**
  - Define security adaptation strategy
  - Monitor security attributes
- **Reaction**
  - Tune counter-measures to security context

Marc Lacoste, Orange Labs, ICAR 2013
Some hard problems

security component heterogeneity between layers and domains.
infrastructure complexity ⇒ 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

**Principle #1:** Policy-Based Self-Protection

**Principle #2:** Cross-Layer Defense

**Principle #3:** Multiple Self-Protection Loops

**Principle #4:** Open Architecture

**VESPA** = Virtual Environments Self-Protecting Architecture

An autonomic security framework for regulating protection of IaaS resources.

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

**Application to hypervisor self-protection:** in progress.
Flexible confinement of VMs according to risk level
Flexible confinement of VMs according to risk level

Illustration
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: The Cloud Becomes User-Centric

Marc Lacoste, Orange Labs, ICAR 2013
Provider-centric cloud deficiencies

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

The user-centric cloud (a.k.a super-cloud)

- Cloud resource distribution plane separating production from consumption.
- Benefits:
  - Independence from provider.
  - Increased customizability.
  - New business opportunities.

Marc Lacoste, Orange Labs, ICAR 2013
Provider-centric cloud deficiencies

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

The user-centric cloud (a.k.a super-cloud)

- Cloud resource distribution plane separating production from consumption.

Benefits:

- Independence from provider.
- Increased customizability.
- New business opportunities.

Toward fully distributed hypervisors….
Exploitation of virtualization vulnerabilities are some of the most serious cloud threats, making the hypervisor a keystone component of cloud security.

Some key points:

- 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 vibrant 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