IMPROVE SECURITY IN THE VIRTUAL MACHINE HYPERVISOR

SR - SEMINAR

Luca Rinaldi

AGENDA

- Intro to VM hypervisors
- NoHype: removing the hypervisor
- NOVA: minimizing the hypervisor
- Conclusions

HYPERVISOR

It **emulates** a given hardware and permitting to the guest OS to run directly on it.

It usually needs a large and complex piece of code and frequent interactions with the guest OS.

A **VM exits** are generated by the vCPU when any operation performed by the guest OS needs the intervention of the hypervisor.

SECURITY THREAD IN VIRTUALISATION

Hypervisors should maintain VM-VM and VM-host isolation.

There are cases in which, through hypervisor vulnerabilities, an attacker can exit the virtual machine.

Usually hypervisors have a large TCB.

NOHYPE

It eliminates the whole hypervisor and its attack surface.

It removes the need for VMs to constantly interact with the hypervisor during their lifetime.

MAIN DESIGN

- Pre-allocating Memory and Cores
- Using only Virtualised I/O Devices
- Short-Circuiting the System Discovery
- Avoiding Indirection

PRE-ALLOCATING MEMORY AND CORES

Without hypervisor it is not possible to dynamically allocate memory and computational resources.

It fix the memory allocation and the cores before the VM starts.

USING ONLY VIRTUALISED I/O DEVICES

Without hypervisor it is not possible to emulate I/O devices.

It can use dedicated I/O devices or devices with virtualisation technology.

The only true indispensable I/O device in the cloud is the NIC card because the storage device can be accessed through the network.

SHORT-CIRCUITING SYSTEM DISCOVERY

The guest OS do not run discovery calls on the hardware at runtime, because there is no hypervisor to answer them.

Gust OS kernel is modified, so that those operations are executed only at boot time and the answer is cached.

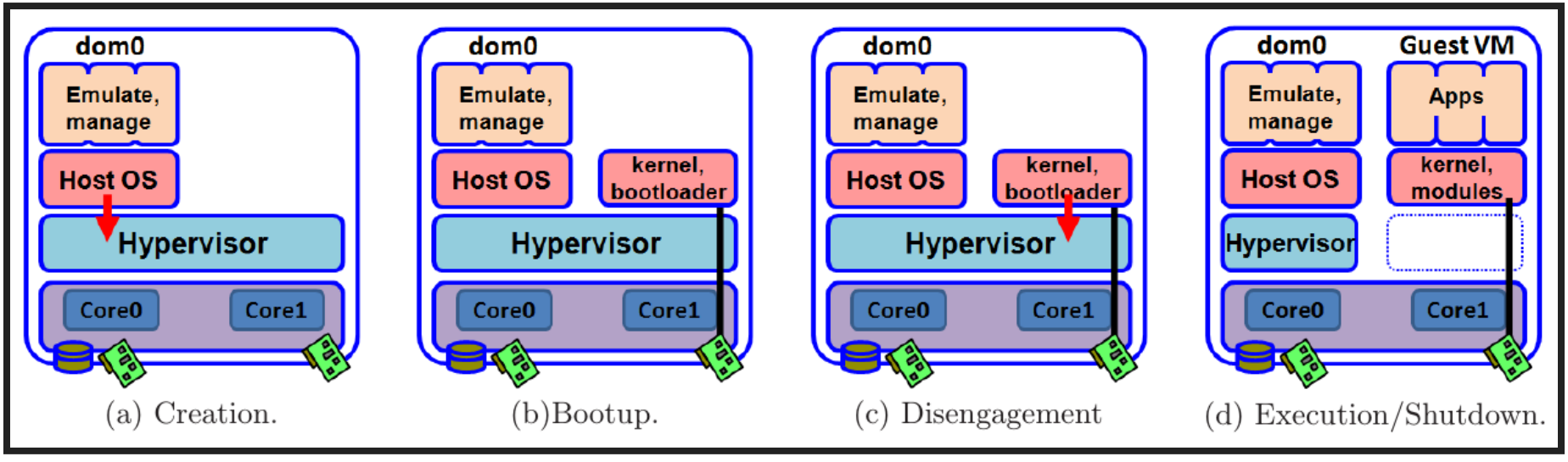
AVOIDING INDIRECTION

The hypervisors use indirection to modify the process IDs inside the guest OS so that they start counting from 0.

With noHype and static allocation of VMs to core, each VM can access the correct process and avoids indirection.

For the same reason re-route of the interrupts is removed.

PROTOTYPE DESIGN



Hypervisor: *Xen 4.0*, Guest OS: *Linux 2.6.35.4*, CPU: *Intel XEON W5580*, NIC card controller: *Intel 82576*

VM CREATION

The Xen hypervisor manages VM initialisation:

- it sets core affinity
- it sets the NIC by a PCI pass through
- it sets memory allocation using VT-d

VM BOOTUP

These are the main phases:

- load the kernel and the initial RAM disk using the network
- run the modified Linux kernel
- disengage the hypervisor
- mount the iSCSI HD drive
- exec user code

VM SHUTDOWN

VM controller structure (VMCS) of the CPU is configured so to exit the VM mode when NMI interruptions are received.

This gives the possibility to the VM to handle the exit signal, so that the VM can shutdown correctly.

PROBLEMS OF THIS APPROACH

- Static allocation of the resources.
- The virtualisation layer could no longer be used for VMs interposition.
- Trust is still in the hypervisor.
- Changes of existing hardware and virtualisation stack.
- It only considers software attacks and cannot guard against wore complex attack.
- Modified Kernel on the Guest can introduce incompatibility with user software or new vulnerabilities.
- Without indirection guest can easily know to be inside a VM.

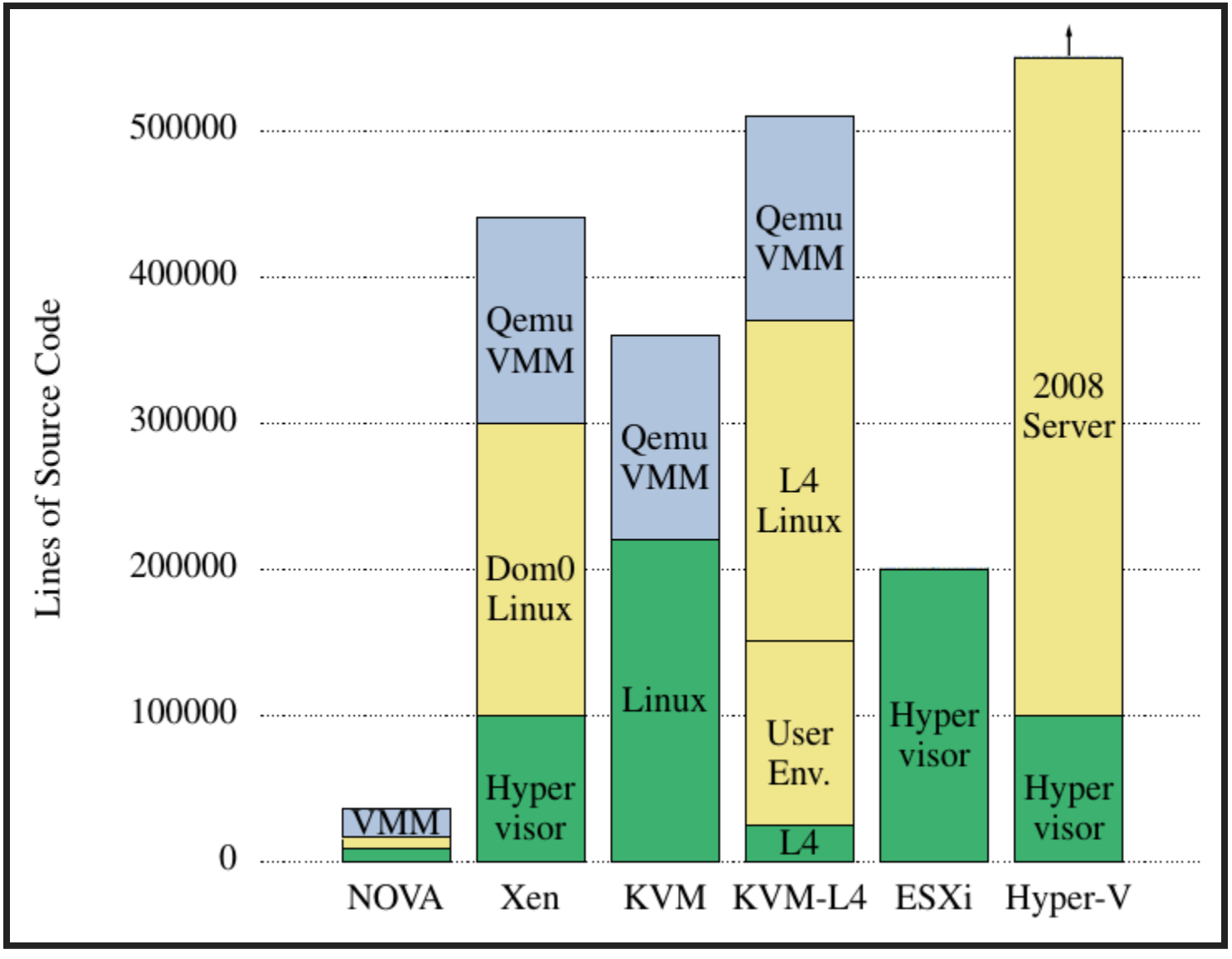
NOVA

It is a micro-hypervisor that minimizes the amount of code in the privileged hypervisor.

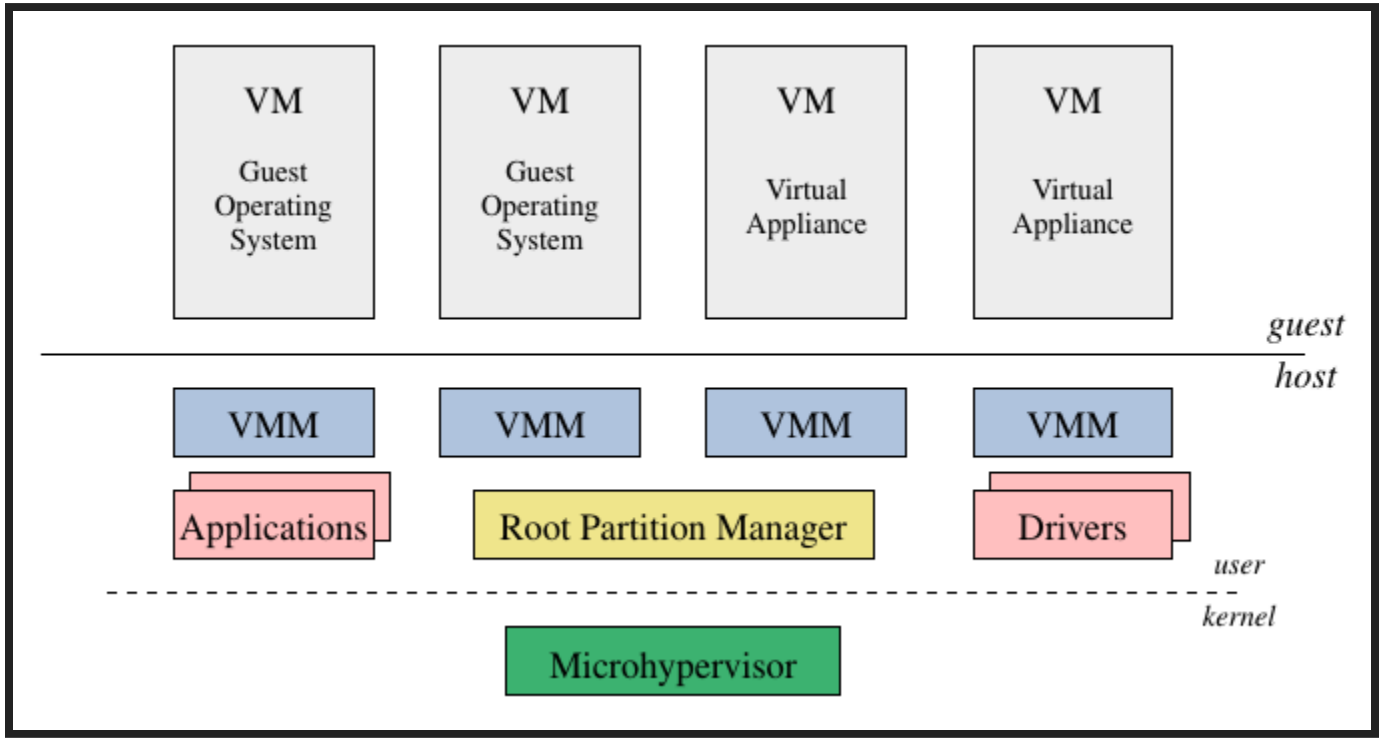
The main features are:

- virtualisation at user level.
- VM's TCB reduction by one order of magnitude.

LINES OF CODE



DESIGN



Least privilege among the components.

MICRO-HYPERVISOR

It implements a capability-based hypercall interface

Capabilities are opaque and immutable to the user, they cannot be inspected, modified, or addressed directly.

It is possible to **delegate** copies of a capability with the same or reduced permissions to other domains that require access to the object.

MICRO-HYPERVISOR OBJECTS

The main features of the kernel are implemented by means of:

- Protection domains
- Execution contexts
- Scheduling contexts
- Portals
- Semaphores

ROOT PARTITION MANAGER

It manages all the memory regions, I/O ports, and interrupts of the system.

It is the first protected domain created by the micro-hypervisor after boot and it can create and delegate new protected domains to assign resources.

VIRTUAL MACHINE MONITOR

It supports the execution of an unmodified guest OS in the VM, emulating sensitive instructions and providing virtual devices.

It runs as a user-level application and there is one for each VM.

PROBLEM OF THE NOVA APPROACH

- No rich set of features for deployment in commercial hosting environments.
- They had to remove hardware supports to achieve the reduction of the TCB.
- Assumption that an attacker cannot violate the booting sequence.
- The micro-hypervisor is still in charge of complex management duties. (like address space management, interrupt and exception handling, and communication between the running workloads).
- Reduction of the hypervisor LOCs does not imply that formal verification methods can apply.
- The use of a VMM for each VM reduces the risk of damaging other VMs but does not reduce the possibility to attack the hypervisor kernel.

CONCLUSIONS

VMs are fundamental in the Cloud scenario and guaranteeing their security is very important.

Hypervisors have to improve their security and reliability.

Being too drastic and requiring considerable "re-design" of the infrastructure nor NOVA NoHype approaches can be implemented in real life.

REFERENCES

- Szefer, J., Keller, E., Lee, R. B., & Rexford, J. (2011, October). **Eliminating the hypervisor attack surface for a more secure cloud.** In Proceedings of the 18th ACM conference on Computer and communications security (pp. 401-412). ACM.
- Steinberg, U., & Kauer, B. (2010, April). **NOVA: a microhypervisor-based secure virtualization architecture.** In Proceedings of the 5th European conference on Computer systems (pp. 209-222). ACM.
- Zhang, Y., Pan, W., Wang, Q., Bai, K., & Yu, M. (2012). **HypeBIOS: Enforcing VM Isolation with Minimized and Decomposed Cloud TCB.** Virginia Commonwealth University, Technical report.
- Zhang, F., Chen, J., Chen, H., & Zang, B. (2011, October). **CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization.** In Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles (pp. 203-216). ACM.