Sevigator

virtualization-based tool for preventing sensitive data leaks from a computer running untrusted commodity operating system

Denis Efremov (efremov@ispras.ru)
The Institute for System Programming of the Russian Academy of Sciences

The Challenge

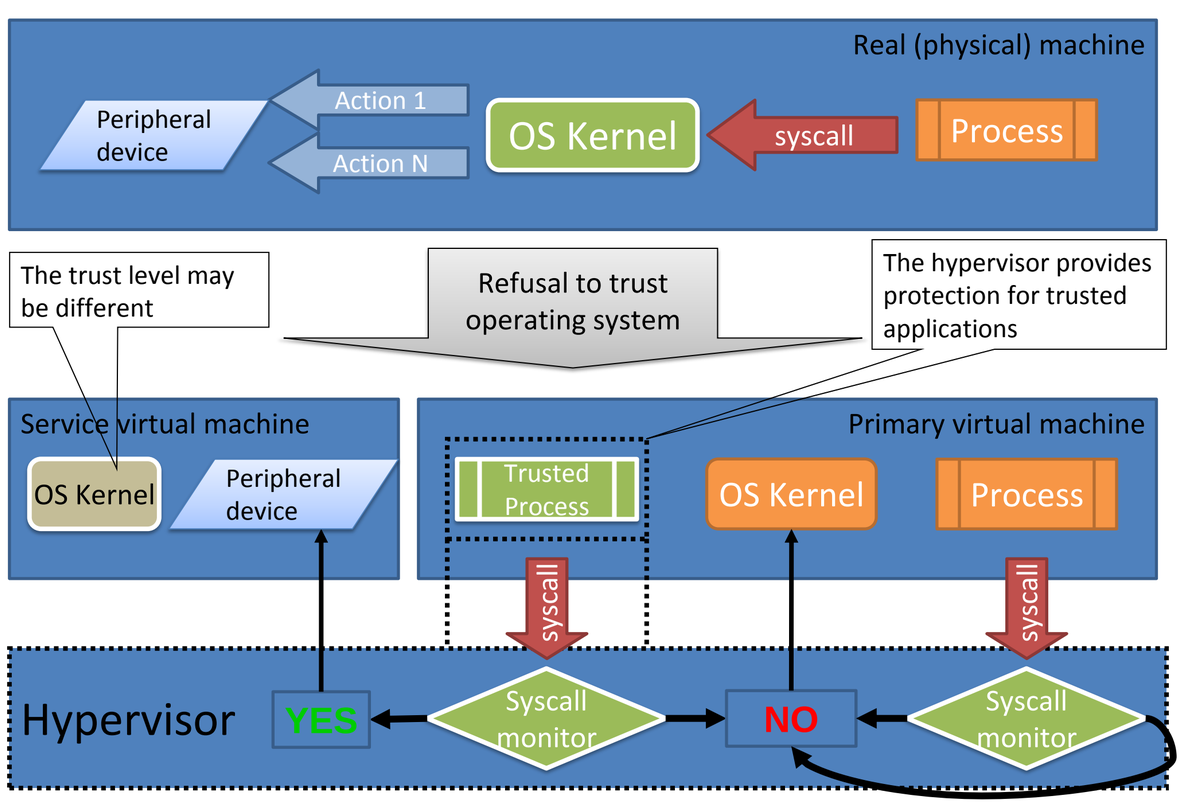
The development of the x86 architecture and operating systems based on it was originally focused on achieving maximum performance. Proper attention to the problem of sensitive data confidentiality was not given, because at the early beginning there were no serious threats. Reduction in component's base cost and unification of the computing machines units has led to a rapid distribution of computers, based on x86 architecture.

However, with the development and widespread deployment of networks, the whole area of data confidentiality protection was problematized. Contemporary commodity operating systems are rather big and not sufficiently reliable and secure due to the monolithic kernel architecture meaning that exploiting just one kernel vulnerability may provide malware with unrestricted access to the system resources.

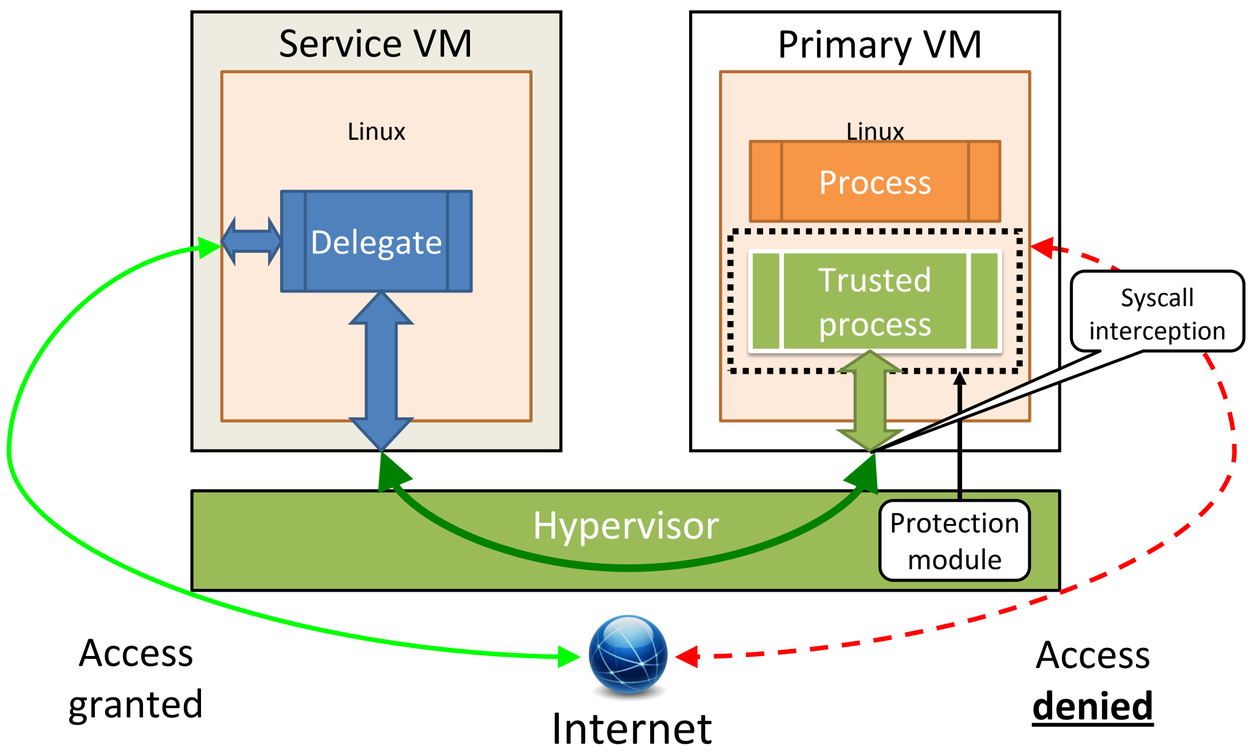
Burden of legacy software (drivers, applications) does not allow us to easily and painless abandon existing developments and switch to a brand new computer and operating system architectures.

All of these conditions suggests in favor of creation of security systems based on the principles that are orthogonal to those used in commodity operating systems. Hardware virtualization provides an opportunity to create such security systems without modification of existing software.

The Idea



System Architecture



Hypervisor ensures simultaneous execution of two virtual machines(VM): primary and service, completely isolated from each other. User works in the primary VM which controls all hardware devices except network adapter. Hypervisor runs primary VM without network adapter since OS in the primary VM is not trusted and may use network connection to leak sensitive data from the VM. The primary VM contains software(both trusted and untrusted) handling sensitive data. Protection system does not limit access of processes to data making its processing possible by any programs including untrusted ones. Authenticity of trusted processes is validated by means of secure control sums of the memory-mapped file pages of the application or a library. Service VM may run in the background since the only purpose of this VM is to serve socket-related system calls executed by the trusted processes in the primary VM. System call requests are delivered by the hypervisor to the service VM through the tamper-proof inter-VM channel. It should be noted that the only device that is controlled by the service VM is network adapter and virtual machines do not share any peripheral devices.

Current Status

Currently, the protection system is implemented as extension of KVM hypervisor. KVM runs in a host operating systems and uses QEMU for device emulation. We assume that computer has CPU supporting AMD Secure Virtual Machine(SVM) technology including memory virtualization (Rapid Virtualization Indexing(RVI)). SVM extension is used for controlled execution of unmodified operating systems. RVI extension (formerly known as Nested Page Tables) is used for the purpose of reducing trusted code base and by the hypervisor's subsystem of trusted processes integrity controlling. And Device Exclusion Vector extension is used for hypervisor's self-protection against attacks by means of Direct Memory Access(DMA).
Current limitation for operating system of virtual machines is – 32-bit GNU\Linux.

Ongoing & Future Work

- Protection of IPC mechanisms
- Utilization of input/output memory management unit(IOMMU) (more reliable isolation of virtual machines from one another and protection of hypervisor against DMA attacks)
- Bare-metal implementation of hypervisor
- Intel virtualization platform support
- Windows support
- Trusted boot of the security system and its on-the-fly activation