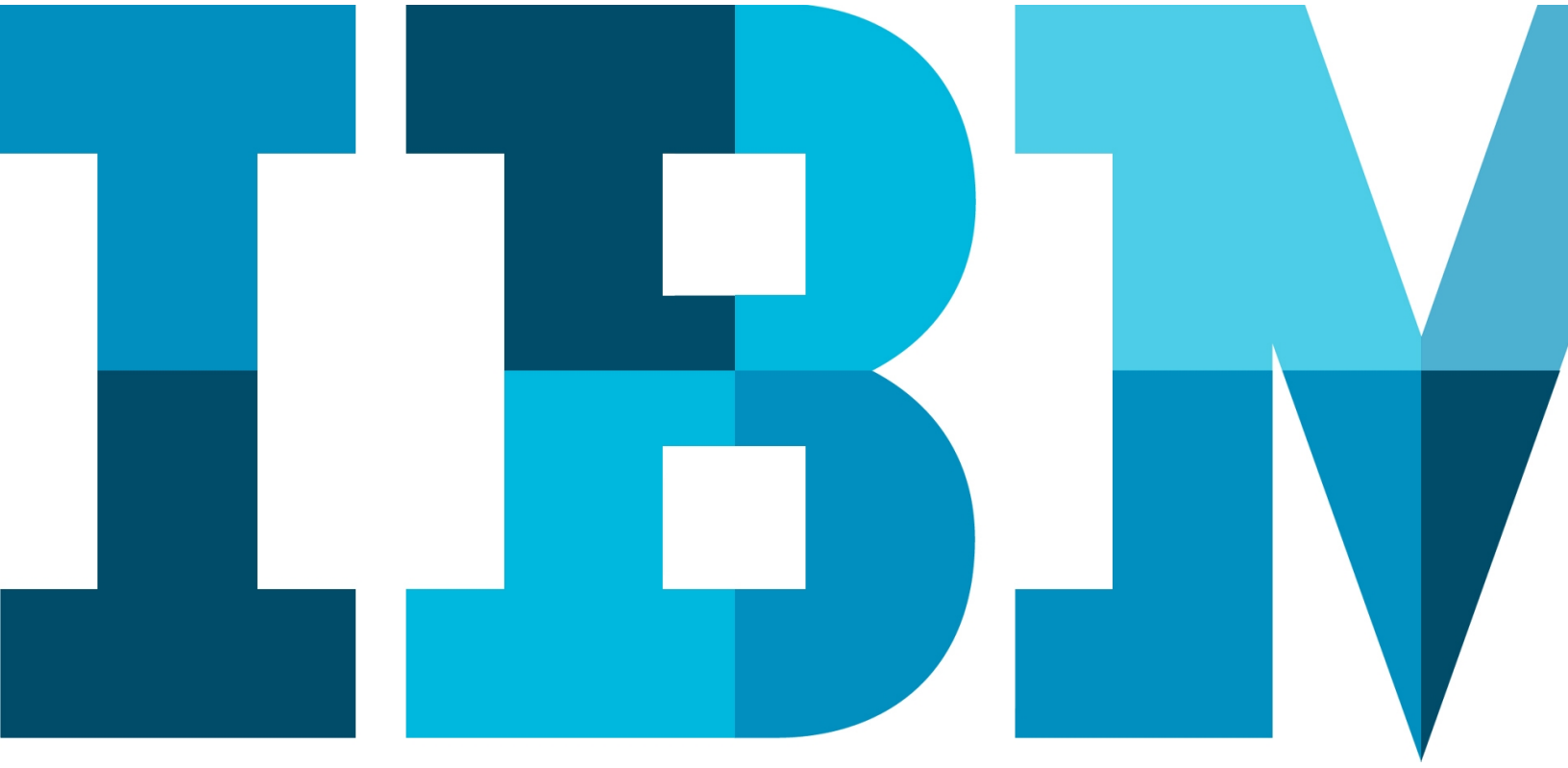
KVM: Hypervisor Security You Can Depend On



George Wilson, IBM Linux Security Architect gcwilson@us.ibm.com
Michael Day, IBM Open Systems Chief Virtualization Architect mdday@us.ibm.com
Beth Taylor, IBM Linux Information Development jetaylor@us.ibm.com

Contents

What virtualization technology can you trust for the security of your cloud?

Executive summary

You've probably been reading about the economics of cloud computing. The promises of efficient, virtualized computing platforms are attractive: low entry cost, dynamic sizing to accommodate varying workloads, automated management, and more. The value proposition looks equally compelling for both emerging and well-established organizations. Moving your mission-critical workloads to a cloud could save your organization a substantial fraction of its current IT expense. However, there is an obstacle significant enough to prevent you from ever taking advantage of the benefits cloud computing offers. That obstacle is a vital question of security. What virtualization technology can you trust for the security of your cloud? Who can provide it? The answer: You can trust the company that has the most virtualization experience. You can trust the open source technology that powers its clouds. That company is IBM®, and that technology is KVM.

Key benefits of KVM

The kernel-based virtual machine (KVM) hypervisor provides a full virtualization solution based on the Linux operating system. The following key benefits of KVM are described in more detail later in this paper.

- KVM has strong guest isolation with an extra layer of protection against guest breakouts. Mandatory access control adds a level of isolation beyond basic process separation.
- KVM's bare metal design (Type 1 design) is similar to other x86 hypervisors.
- KVM is rigorously implemented and tested. With open source, developers are continuously inspecting KVM for flaws.
- KVM has the advantage over other x86 hypervisors in terms of lower total cost of ownership and greater flexibility than competing hypervisors.

Features and functions

The following features exist in various hypervisors. KVM is competitive in every area.

	KVM	VMWare ESX	Microsoft HyperV	Critix XenServer
Process isolation	✔	✔	✔	✔
MAC isolation by default	✔	✘	✘	✘
RBAC	✔	✔	✔	✔
Bare metal type 1 hypervisor	✔	✔	✔	✔
Flexible authentication	✔	✔	✔	✔
Audit trail	✔	✔	✔	✔
Common criteria certified	✔ EAL4+ *	✔ EAL4+	✔ EAL4+	✔ EAL2+
Common criteria test suite freely available	✔	✘	✘	✘
FIPS 140-2 validated cryptographic modules	✔	✔	✔	✘
Source code available	✔	✘	✘	✔
Resource control	✔	✔	✔	✔
Disk encryption	✔	✔	✔	✔

* Note: It is anticipated that the common criteria certification of KVM in Red Hat Enterprise Linux 5.6 will complete in late 2011. See the BSI new certificates page for the latest status: https://www.bsi.bund.de/ContentBSI/EN/Topics/Certification/newcertificates.html

One company holds the distinction of inventing the hypervisor, and has more experience with virtualization than any other: IBM.

Virtualization and security

Virtualization, the ability to emulate hardware to run multiple operating system instances on a single computer, has been an accelerating trend in the last several years. Low-end computing devices have increased in sophistication; they now have virtualization capabilities that until fairly recently were associated only with mainframe and some high-end midrange systems. Virtualization brings many advantages to the space, including higher efficiency due to increased utilization, energy savings per computation unit, and the flexibility to create and destroy machines on demand to meet the needs of continuously transforming organizations.

Executing multiple workloads per physical machine brings risks as well as benefits. Instances must be separated to prevent their interfering with one another, either intentionally or unintentionally. The hypervisor, or virtual machine monitor (VMM), is the software that virtualizes the hardware and provides isolation, or separation, between guests. Given the relative newness of non-mainframe virtualization and the need to handle sensitive workloads, hypervisor security is a great and well-placed concern.

Much is written in the press today regarding virtualization, with articles commenting on various aspects of it. A substantial portion of the content deals with the topic of hypervisor security. Many confusing and sometimes contradictory assessments are made regarding hypervisor security characteristics. Clearly, the discussion casts enough doubt on the strength of security controls to make organizations reluctant to move workloads to virtualized infrastructure such as clouds.

Important details are often lost in the din of the hypervisor security debate, sometimes causing flawed conclusions to be drawn. It's hard to know who to trust when it comes to the security of your business-critical virtualized workloads. But one company holds the distinction of inventing the hypervisor, and has more experience with virtualization than any other: IBM.

No other general-purpose x86 hypervisor implements MAC by default, providing KVM with a layer of defense beyond that of other hypervisors.

Strong guest isolation

One of the first things that comes to mind regarding hypervisor security, particularly in a cloud environment where multiple clients are served by one software instance, is guest isolation. In the cloud, clients place their trust in the hypervisor. Unquestionably, the hypervisor must be protected against security breaches involving guests operating on top of the hypervisor. These security issues include:

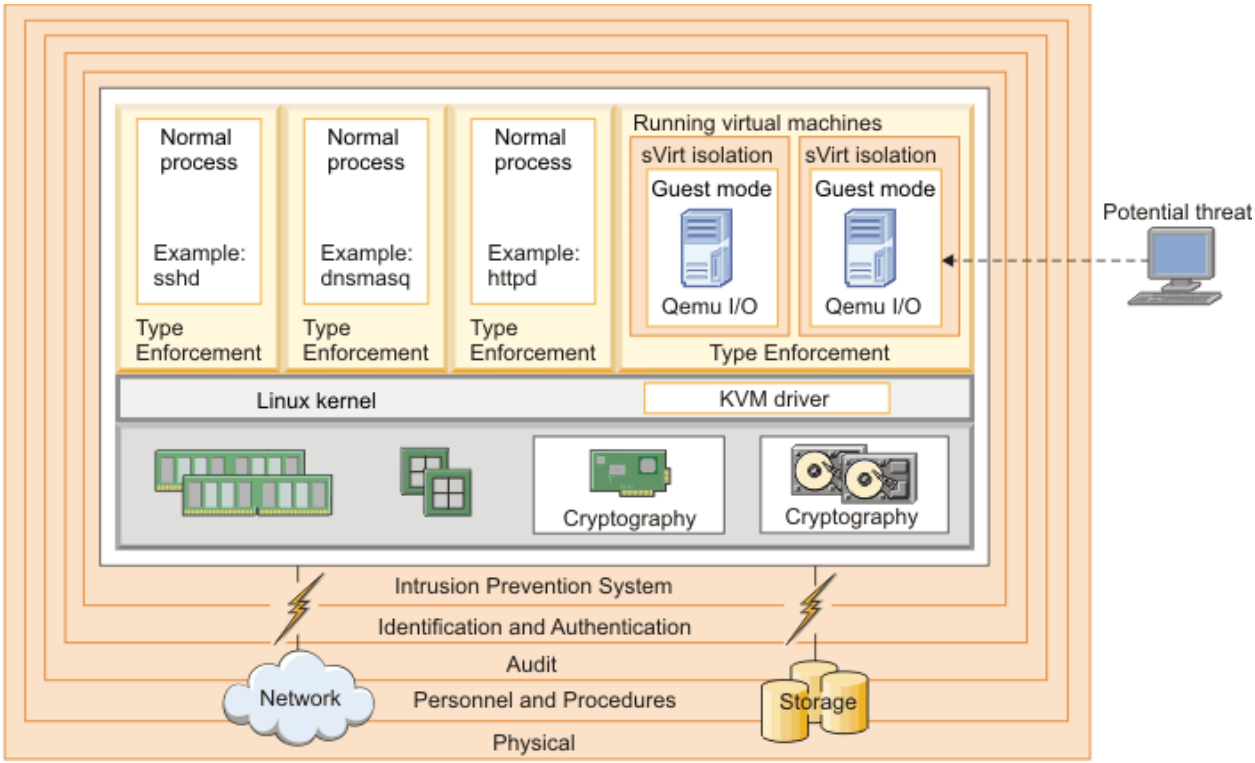
- Guests bypassing security controls to access either the host or other guests in ways that violate the host security policy
- Guests intercepting client data or host resources to which they are not authorized
- Guests attempting or becoming the victim of security attacks, which could possibly take down the entire cloud.

In addition, client data must be protected from unnecessary access from the hypervisor itself. Finally, guests need the capability to create controlled shared storage for collaboration purposes.

Because KVM is built into Linux, KVM guest processes are subject to all the usual user space process separation that is integral to the Linux kernel's operation. Linux process separation continues to evolve over time. However, the most basic protection mechanisms have existed since early in the development of the Linux kernel, and are well tested and certified. On x86 systems, the kernel, at the lowest level, uses the central processing unit (CPU) chip set hardware to achieve separation between user space mode and kernel (privileged or supervisor) mode. Inside the kernel, discretionary access control (DAC) prevents user space processes from unauthorized access of resources or other processes. DAC is the traditional set of access controls in which users own their own resources and can manage access to those resources at their discretion.

Mandatory access control

KVM goes even further than basic DAC separation by incorporating mandatory access control (MAC) through Security-Enhanced Linux (SELinux). With MAC, it is the administrator, not the process owner, who determines the access a process has to resources. MAC implements strong guest isolation and controls resources available to guests. The sVirt API, which integrates MAC and Linux virtualization within SELinux, is enabled by default in RHEL 6. As of the writing of this document, no other general-purpose x86 hypervisor implements MAC by default, providing KVM with a layer of defense beyond that of other hypervisors.



Network controls allow separation to be extended to communications. Similar principles apply to the networks as to other resources. Because network packets travel between machines, their flow must be carefully governed. Failure to do so can result in data leaks, spoofing, and denial of service attacks. Fortunately, KVM uses network filtering built into Linux. Bridge filtering controls guest traffic at the Data Link Layer (Layer 2). Additionally, virtual LANs (vLANs) can be created to segregate traffic associated with different security domains. vLANs also permit management network traffic to be separated from guest traffic. Network filtering governs traffic at the Network Layer (Layer 3) to provide a dense firewall. New Linux features allow virtual networking to work well with hardware switches; actual enforcement can occur on physical switches in accordance with host security policy, and packets can be detoured to deep packet inspection (DPI) engines.

Hardware-based isolation

In addition to the access control provided through Linux DAC and sVirt MAC, KVM uses virtualization-specific processor instructions to ensure isolation of guests from the hypervisor and from each other. Intel's virtual machine extensions (VMX) and AMD Secure Virtual Machine (SVM) instructions add a third level of isolation and protection by running guests in a restricted (guest) mode.

Any attempt by a guest to execute a processor instruction that might change the isolation parameters of the host results in an immediate transfer of execution to the hypervisor. The hypervisor has the opportunity to validate or deny any attempt by a guest to execute instructions that have the potential for breaking the isolation properties of the host. A guest that manages to overcome the DAC and MAC mechanisms must still breach the hardware isolation protections before it can completely control the host computer.

Hypervisor type is a diversion in the hypervisor security discussion. KVM's direct access to hardware puts it in the same class as other Type 1 hypervisors.

Bare metal design

In 1973, Robert Goldberg classified hypervisors according to their proximity to hardware instructions.[1] Goldberg 's "Type 1" hypervisor was defined as one that translated physical to virtual resources once; a "Type 2" hypervisor was one that made the resource translation twice. More recently, these definitions were extended to "bare metal" Type 1 hypervisors (running directly on the hardware), versus "hosted" Type 2 hypervisors (running within the operating system). The industry press has confused the discussion further, debating whether a traditional operating system (OS) environment is part of the hypervisor's resource management code, or even whether a traditional OS is visible to or hidden from the administrator.

KVM meets all the criteria Goldberg defined for a Type 1 hypervisor. First, the virtual machine monitor (VMM) runs in privileged mode and directly uses hardware instructions to virtualize the guest. Guest code executes most of the time directly on hardware at full speed. Most importantly, the virtual-to-physical resource translation occurs just once. In meeting these criteria, KVM is equal to VMWare, Xen, z/VM, and other bare metal hypervisors. The fact that KVM can co-reside with an enterprise Linux OS does not change any of its Type 1 characteristics.

In fact, KVM is packaged today both with and without a full Linux environment. Red Hat offers a locked-down, hypervisor-only KVM product that omits the Enterprise Linux OS and restricts administrator access to a small set of controlled interfaces. This implementation demonstrates the flexibility of KVM's bare-metal design.

Regardless, the plain truth is that the hypervisor type is a false indicator of security. While design and implementation are important considerations to hypervisor security, hypervisor structure is not. A badly designed Type 1 hypervisor can be much less secure than a well-written Type 2 hypervisor, and the reverse is also true. KVM's hypervisor design provides isolation properties that are similar to VMware ESX. The trusted code base of KVM is generally the same as for other x86 hypervisors.

Simply put, hypervisor type is a diversion in the hypervisor security discussion. KVM's direct access to hardware puts it in the same class with other Type 1 hypervisors.

Common Criteria certification demonstrates that KVM lives up to its security claims and exceeds other comparable hypervisors in many respects.

Rigorous implementation and testing

Open source is a method of engineering that distributes design and development effort globally. Participants contribute labor while benefiting from the work of others to solve different problems. Almost all work takes place on Internet mailing lists in the form of patch submissions to open source communities. Anyone can read, comment on, and contribute to the mailing lists. Communities collectively judge individual submissions, and meritocracies form organically. Maintainers bubble up from the communities who are specialists in their fields and lead the communities. Open source communities attract experts worldwide in specific problem domains that would otherwise be difficult or impossible to assemble.

---

[1]Goldberg, R. Architectural Principles for Virtual Computer Systems. PhD thesis, National Technical Information Service, February 1973.

All KVM development takes place in open source communities. The development methodology brings great benefits to KVM security. Maintainers and community members perform continuous inspection and testing to find bugs. Weaknesses are identified and patched quickly. Relentless analysis of the source code by multiple experts is particularly important to minimize the possibility of unknown vulnerabilities getting into the code base and leading to zero-day exploits. This development approach is a particular advantage that open source has over proprietary development. Proprietary development is opaque; it is difficult or impossible to obtain information about proprietary hypervisor internals. Are guests really separated? Are communications paths adequately controlled? Are the privileged management APIs coded correctly? Without security certification results available, you have little choice but to trust proprietary vendor security claims. However, there is zero mystery regarding the contents of KVM and its broader ecosystem; all its source code is available for viewing.

One need not simply believe that KVM has strong security; KVM is in the process of achieving the Common Criteria for Information Technology Security Evaluation certification[2]. Common Criteria certifications scrutinize all security aspects of a product: design, source code, source code control, development process, and flaw remediation processes. Through this certification, a Common Criteria accredited lab verifies that KVM is strong enough for military applications. A Security Target document provides details of the evaluation for an objective comparison with other hypervisor security targets. Common Criteria certification demonstrates that KVM lives up to its security claims and exceeds other comparable hypervisors in many respects, particularly strength of guest separation.

By embracing KVM, IBM is continuing its tradition of hypervisor excellence.

The KVM advantage

KVM is IBM's strategic hypervisor for Intel- and AMD-based systems. Selecting a primary hypervisor for cloud platforms was not an easy task for IBM. Undoubtedly, several proprietary and open source hypervisors provide adequate virtualization capabilities. However, few have the qualities called for to become the stand-out choice for IBM's cloud offerings. Once hardware acceleration was introduced into the Linux kernel in 2007, KVM quickly rose to the top. A number of factors solidify the preference for KVM.

Cost: Given its open source nature, KVM has a lower total cost of ownership.

Rapid progress to maturity: A community of experts continuously enhances KVM.

Exploitation of advances in Linux: KVM is built into Linux and benefits from the entire Linux community.

***Efficiency*:** KVM takes advantage of modern hardware design to securely execute directly on the host CPU, and is engineered to perform well even in memory- and CPU-constrained environments.

Active and responsive community: Customer feature requirements and security vulnerabilities are quickly addressed.

Truly open source: The code and its repository data are available, continuously inspected, and transparent in modification rationale throughout the product life cycle.

Control: Because IBM is a primary KVM community member, it is influential in setting KVM development priorities.

[2]It is anticipated that the common criteria certification of KVM in Red Hat Enterprise Linux 5.6 will complete in late 2011.

By embracing KVM, IBM is continuing its tradition of hypervisor excellence. IBM helps to lead KVM development, while implementing the features that our clients demand. In addition, IBM uses KVM for workload consolidation as well as our cloud offerings.

KVM is the clear choice for your virtualization technology.

Conclusion

KVM is a trusted solution for implementing virtualized environments, such as clouds that contain multiple tenants. KVM security stacks up well against other general-purpose x86 hypervisors. It implements layers of controls, including mandatory access control and hardware-based isolation, to achieve deep defense against attacks. KVM's direct access to hardware provides the same level of protection as other bare metal hypervisors.

Based on Linux, KVM benefits from the open source development community, including constant inspection for potential security flaws. Furthermore, KVM will soon achieve Common Criteria certification at an EAL4+ level[3].

The decision to make KVM the foundation of IBM's cloud offerings is not arbitrary. IBM has more experience with virtualization than any other company; IBM has demonstrated confidence in KVM as a superior hypervisor. KVM is the clear choice for your virtualization technology.

References

Special report: Government in cyber fight but can't keep up, Phil Stewart et al., Reuters, June 16, 2011, http://www.reuters.com/article/2011/06/16/us-usa-cybersecurity-idUSTRE75F4YG20110616

BM Cloud Computing - Cloud Security, website, IBM, http://www.ibm.com/cloud-computing/us/en/#!cloud-security

Common Criteria Portal, website, http://www.commoncriteriaportal.org/products/#OS

Strategies for assessing cloud security, IBM GTS, Nov. 2010, http://public.dhe.ibm.com/common/ssi/ecm/en/sew03022usen/SEW03022USEN.PDF

KVM Security, IBM Linux Information Center, IBM, Nov. 2010, http://publib.boulder.ibm.com/infocenter/lnxinfo/v3r0m0/topic/liaat/liaatseckickoff.htm

Harmonizing the Twin Trends of Open Source and Virtualization: How Kernel Based Virtual Machine (KVM) Drives Enterprise Business Value, White paper, Srini Chari, Cabot Partners with IBM sponsorship, Nov. 2010, ftp://public.dhe.ibm.com/linux/pdfs/Cabot_Partners_-_Harmonizing_the_Twin_Trends_of_Open_Source_and_Virtualization.pdf

Virtualization System Security, Bryan Williams and Tom Cross, IBM ISS, 2010, http://blogs.iss.net/archive/papers/VirtualizationSecurity.pdf

KVM Security Comparison, Stephan Mueller, atsec, Nov. 2009, http://www.atsec.com/downloads/white-papers/kvm_security_comparison.pdf

[3]It is anticipated that the common criteria certification of KVM in Red Hat Enterprise Linux 5.6 will complete in late 2011.

IBM home page: http://www.ibm.com

IBM Corporation
Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
November 2011

LXW03004-USEN-00