2009 International Conference on Computational Science and Engineering

Security Implications of Virtualization: A Literature Study

André van Cleeff, Wolter Pieters, Roel Wieringa
Information Systems Group
University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands
{a.vancleeff,w.pieters,r.j.wieringa}@utwente.nl

***Abstract*— Server virtualization is a key technology for today's data centers, allowing dedicated hardware to be turned into resources that can be used on demand. However, in spite of its important role, the overall security impact of virtualization is not well understood. To remedy this situation, we have performed a systematic literature review on the security effects of virtualization. Our study shows that, given adequate management, the core virtualization technology has a clear positive effect on availability, but that the effect on confidentiality and integrity is less positive. Virtualized systems tend to lose the properties of location-boundedness, uniqueness and monotonicity. In order to ensure corporate and private data security, we propose to either remove or tightly manage non-essential features such as introspection, rollback and transfer.**

I. Introduction

Server virtualization, running many virtual servers on one physical machine, has become widespread in recent years. With respect to security, claimed improvements are the increased availability of applications and the isolation of processes. Virtualization also has security drawbacks, such as exploitable weaknesses in virtualization software, the existence of covert channels and the possibility of new types of malware. However, apart from these distinct threats, not so much is known about the overall security effect of virtualization. Under which circumstances does virtualization improve security, and under which does it pose a threat? In order to answer these questions, we have to distinguish between different features of virtualization and show their interactions. Based on a systematic literature review, we aggregate the literature on the security impact of virtualization for each feature and show how these features fit together. We conclude with an overview of how to maximize the security benefits, from both a technical and a management point of view.

II. Background on virtualization

A. Types of virtualization

Generally, virtualization is a software layer that implements a (hardware) architecture. Such an approach can be useful for resources such as harddisks, networks and complete machines, providing a consistent interface to decouple software systems from the hardware on which they are running, making them more portable and providing easier management. In this paper we focus on machine virtualization, where the instruction set of a CPU architecture is emulated on a real physical machine. [1] With respect to virtual machines (VMs), the abstraction layer is called virtual machine monitor (VMM). The VMM controls the VMs running on top of it. VMMs can also be joined together, resulting in a virtualized infrastructure. In this infrastructure, capabilities such as load balancing are managed from a central location. We call this infrastructure and its management the "VMMM", an acronym for *virtual machine monitors' management*.

B. Usage of virtualization

Virtualization can be used for many purposes, for example software testing or providing uniform desktops to end-users. Obviously, the security effects differ for each type of usage. In the remainder of the paper, we concentrate on the security effects for running production applications, where a business's applications are placed inside VMs. The reason for this choice is that here the security concerns are the greatest.

III. Research approach

A. Research design

The literature study is based on the method described by Webster and Watson. [2] Here, literature is retrieved from well-known sources such as leading journals and additional literature is found by tracing back the cited papers and forward towards conferences papers. The findings are presented *concept-centric*, meaning all literature on a certain concept is discussed in one section. For this study, we began with a literature search on Scopus[2] yielding a total of 151 papers of which 46 were relevant. Literature from other sources was also included, such as datasheets from virtualization product vendors such as VMware.

Here, we choose to present the results centered around specific *features* of virtualization. Obviously, a complex technology such as virtualization can be split up in many ways, and each choice is to some extent arbitrary. For this model, we used the following guidelines:

1) A feature represents a distinct piece of functionality, characteristic or architecture, having a unique impact on security.

[1] For a taxonomy on virtualization we refer to Smith and Nair. [1]

[2] www.scopus.com

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.267

2) Existing decompositions found in the literature should be used as much as possible.
3) A feature should be widely used, not just being implemented by one vendor in one product.

For each feature, we present the security effects, together with a key reference. When no direct security effects were found, we added analytical claims based on the literature studied.

B. Conceptual model

Virtualization technology consists of features, which are divided into five groups:

1) features of virtualization capable hardware
2) features of VMs
3) features of individual VMMs
4) features of VMMMs
5) features arising from unintended interactions between features

Figure 1 shows a graphical representation of the groups. The hardware (1) enables virtualization, several VMs (2) run on top of a group of VMMs (3) and the VMMs are managed by a VMMM(4), leading to emergent features (5).

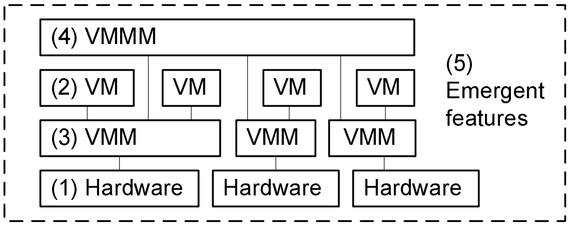

Fig. 1. Groups of virtualization features

The threat model used in this paper is essentially a white box, assuming no trusted components of the virtualization technology. As such, we depart from earlier threat models created by Vaarela [3] and VMware [4]. Since we are considering production applications, the ultimate security objective is to protect the application running inside a VM. In our model, threats to this application can originate from five different components: (i) hardware, (ii) other VMs, (iii) VMMs, (iv) VMMMs and (v) network.

Hardware threats are not discussed in this paper, because these are mostly generic threats such as theft that are not specifically relevant virtualization. More technical information on virtualization hardware is available from Perez et al. [5]

Combined, this leads to the model of threats depicted in figure 2.

IV. Security impact per feature group

We now list what is currently known about the security impact of all the different features per group. Security claims in the literature are split over three categories: (i) analytical claims, based on logical arguments, (ii) empirical claims, based on experiments and (iii) claims based on mathematical models. The first category was dominant. Only one paper fitted into the third category, calculating the reliability of different virtualization architectures. [6])

Threat source	Explanation
Network ► VMMM	An outsider attacks the VMMM
Network ► VMM	An outsider attacks the VMM
Network ► VM	An outsider attacks the VM
VMMM ► VMM	A VMMM attacks a VMM
VMM ► VM	A VMM attacks a VM
VM ► VM	A VM attacks another VM

Fig. 2. Virtualization threats

A. Features of hardware

We list two important hardware features below. For more detailed information on other virtualization hardware issues, we refer to Uhlig et al. [7] and Van Doorn [8].

1) Trap program execution: The essential hardware feature enabling virtualization (present in all modern x86 hardware) is the ability to trap the execution of a running process and hand over control of the CPU to the VMM. [9] This allows the VMM to intervene in the execution of the process. In such a way, the VMM can perform two critical tasks: (i) emulate certain hardware and (ii) isolate the virtual machine from other running processes.

2) Trusted Platform Module: An optional hardware feature is the Trusted Platform Module (TPM) chip, which can be used to verify that the proper VMM is indeed running, as opposed to an insecure version installed by an attacker. [5]

B. Features of VMs

1) Store VM as image: VMs consist of files, holding the machine's own data, as well as some metadata (amount of memory used, number of CPUs)[3]. This approach allows easy copying of the VM by a VMM, at the cost of possible confidentiality and integrity breaches.

2) Modified VM software: The software running inside VMs can be equipped with so-called hooks that can be used to contact the VMM in order to execute security checks. [12]

C. Features of individual VMMs

1) Small footprint: Generally, the amount of exploitable vulnerabilities is proportional to the amount of code. VMMs are notably smaller than the previous hardware interface layer, the operating system (OS) and are therefore deemed to be more secure. [13]

2) Hierarchical control: The VMM layer is designed to control the VMs using the underlying hardware. Therefore it should not be possible for code running inside the VM to "escape" to the VMM and gain control over it. However, such escapes are possible, both in laboratory experiments [15] as well as in production environments [16].

3) Isolation between processes: VMMs provide better isolation between virtual machines than an operating system, in which applications can normally interact. [17] However, covert channels are still an issue in existing VMM implementations. [13]

[3] An overview of common VM data elements was created by the Distributed Management Task Force (DMTF), an IT industry consortium. [10]

4) Logging: Virtualization can help to implement secure logging: during the execution of the VM, the VMM collects data and stores it in a place outside of the VM. Therefore it cannot be altered by an exploit that is contained inside the VM. Such a feature is implemented in ReVirt. [18]

5) Load balancing: VMMs can determine (and limit) the CPU cycles and disk space that a VM uses. This prevents a VM from starving the other VMs of resources.

6) Copy and backup VMs: Making backups and copies of VMs is easier than making copies of data on physical machines. Therefore, a defective VM can be easily replaced by a working version.

7) Introspection: Since the VMM is placed at a lower level than the VM, it has the ability to "look inside the VM", to see its data and monitor its execution. This process is called virtual machine introspection. [19] The functionality can be used to run security applications such as virus scanners, intrusion detection systems such as Livewire [20], and policy checkers. Because these are outside of the VM, the VM cannot interfere with them. However, this feature also provides a new location for installing malware.

8) Attestation: If an inspection of a VM is performed, the VMM can send the results (the attestation) to another party. Based on the results, the receiving party can authenticate the VM, and decide whether to trust it. For optimal confidence, this function is best used in combination with trusted hardware (as in the Terra VMM [21]).

9) Interference: In some cases intrusions cannot just be detected, but intervention may be possible: Once the VMM detects a malicious program running inside a VM, its memory can be altered, preventing its execution. [19]

10) Power functions: VMMs control the execution of a VM, as if it would have a virtual power button. Commonly, several functions are provided: start/power on, stop/power off, pause/suspend and reboot/reset. These functions are useful to increase availability, for example if a VM crashes, it can be more easily rebooted than if it were a physical machine, and it is also easier to limit resource usage by simply pausing VMs that are not needed.

11) Networking: VMMs can configure the network, so that a VM only communicates with predefined VMs, forming a virtual network that does not require hardware changes. The networking functions can also be used to monitor traffic, for example running an intrusion detection system.

12) Rollback: Another feature that can be provided by a VMM is to rollback actions of a VM. [22] For example Microsoft Hyper-V has a feature to create a snapshot. [23] If a problem is detected, the VM can be restored to an earlier state.

13) VM Management: In order to manage the previously described features, a user interface has to be provided to control the VMM and the VMs that it runs. The management can be done from another machine (virtual or physical). For example Xen is designed to use a specific VM ("Dom0") for its management. [13] Obviously, this enlarges the small footprint that was supposed to be a strong asset of the VMM.

D. Features of VMMMs

1) Transfer: VMMM can transfer (”migrate") running virtual machines between physical servers. [24] [25] This feature can be useful if a physical machine has to undergo maintenance. Unfortunately, the feature also creates an identity problem, for example when the network ID (often the MAC address) is reassigned randomly after the transfer [22]. Some VMMs can retain the MAC address during a live transfer. [24] If the transfer channel is not secure, the VM can be modified while in transit. [26] Improper management can also lead to the transfer of VMs to unsecure hosts, to breaches of confidentiality and to denial-of-service (DOS) attacks if too many virtual machines are migrated to one VMM host.

2) Replication: Apart from being transferred, VMs can also be replicated on different physical servers. [27] This is useful to ward off a DOS attack, to distribute workload and to cope with hardware failures. At the downside, replication can be used to have VMs run on less secure locations, as was the case for the transfer feature. Obviously, the security of a VM is determined by all the physical hosts it has run on.

3) Load balancing: Features such as transfer and replication can be used for load balancing across different physical machines, increasing the availability of applications.

4) Patching: A benefit of VMMs is that they can contain software to ease the process of patching, such as VMware's vCenter Update Manager. [28] This software inspects VMs to check for missing patches. Before patching, a snapshot is made of the system. If the patching fails, the VM can revert to the snapshot. Thus VMMs make it extremely easy to rollback patches, making patching a non-monotonic process [22].

An additional issue is that VMMs themselves also have to be patched [29], and this is more critical - if a VMM is breached, the breach might be impossible to detect because the VMM is the lowest layer of the IT stack.

5) VMM management: If several VMMs are linked together, their work needs to be coordinated from a special server, such as vCenter from VMware [28]. This gives the administrators enormous power, since they can control any VMM and any VM running on it from a single point. Another side effect is that the trusted code base is increased (similar to the management domain for a single VMM).

E. Features emerging from interactions

In the past sections, we have discussed many virtualization features, some of which have a positive effect on security, whereas others have a negative impact. In this section, we aggregate these into three emerging features that were not explicitly designed, but rather evolved from the interaction between existing features. This high-level clustering helps to understand the effects.

1) Loss of uniqueness of machines and data: In a non-virtualized server environment, applications, servers and data are to a great extent unique. However, the replication and copy/backup features reduce the uniqueness of these.

2) Loss of location-boundedness of data: It is difficult to ascertain the location of a certain VM [30], since it can move between different physical servers, due to features such as transfer, replication and backup. In fact this also holds for the *virtual* location of a VM: if the networks are also virtualized, a VM can be accidentally dragged-and-dropped outside of a DMZ.

3) Loss of monotonicity of program execution: Virtualization technology causes a server's history to stop being a straight line. [22] Instead it becomes a graph, where branches are made on replication and copy operations, and a previous state can be reached when a restore is performed. Data is hard to delete as there are potentially many copies and the VM can be restored to an earlier version. [22]

V. Overall impact of virtualization

The overall effects of virtualization technology depend not just on the technology itself, but also on the environment in which it is used. Conceptually we therefore split the causes into three groups:

1) the technical capabilities of features and their security effects
2) the selection of features that can be made in practice
3) the management of the selected features in practice

The first aspect is illustrated in figure 4 where a summary of all features and their effect on security is provided. For each feature, we show its known effect on all five security properties. A + sign indicates that the functionality provided by the feature has a positive effect on security (compared to a non-virtualized situation), without placing high demands on the environment regarding management, whereas a − indicates the opposite: the feature is technically vulnerable or can be easily misused. Furthermore a ± indicates that the security effect depends on the particular implementation or the circumstances, or that the literature indicates both positive and negative effects. Finally an empty cell indicates that we could not find or deduce an effect on a security property. In the threats column of figure 4 we can find which feature requires strict management.

The dependencies shown in figure 3 shed light on the second group, showing which features are optional. As for the precise meaning of feature dependencies, we take the practical approach that given two features A and B, A depends on B when logically it cannot function independently from B without duplicating part of B's functionality. To keep the diagram clear, the dependencies of the management features are not shown, but obviously, these depend on the other features in the VMM and VMMM layer respectively.

A technological weakness can be mitigated by strong procedural security, and vice versa can a secure technology be misused in practice. Taking this into account, we have attempted to present the overall security impact of virtualization, as can be expected in practice for each of the five security properties in the next sections.

A. Confidentiality

Virtualization threatens confidentiality in several ways. First, the introspection feature gives the VMM the ability to look inside the VM. This feature can be misused or attacked. The problem is aggravated by the fact that VMs can be transferred between different physical servers. Obviously, replication features only increase the problem.

B. Integrity

Together with the features of intervention and rollback comes the ability to manipulate the state of a VM, which threatens the integrity of the transactions done on a VM.

C. Availability

The availability of applications running on VMs will most likely improve: VMs can be transferred for maintenance purposes, restored when a failure occurs and replicated if the workload requires it. (Cf. Jansen et al. [6]) However it must be noted that virtualization is not without availability risks, as was demonstrated by an attack on the virtualization infrastructure of the web hosting company VAServe, where 100.000 sites were deleted [31].

D. Authenticity

Virtualization creates identification problems. If a VM is duplicated there no longer is one original machine. Also, the identity of the machine used for communications (such as the MAC address) might change during a transfer.

E. Non-repudiation

Since virtual machines can be duplicated, rolled back and restored, there seems to be a fundamental problem regarding non-repudiation. If evidence of transactions is stored in a VM in the form of a transaction log, this can be lost if the state of a VM is restored. If transactions are signed, the key with which this is done is also stored on the VM, and can thus be copied.

VI. Conclusions

This paper presents the results of a literature study on the effects of virtualization technology. As a whole, virtualization has a positive effect on availability but threatens confidentiality, integrity, authenticity and non-repudiation, even though many features are designed with these goals in mind.

Some core features of virtualization (isolation between processes and small footprint) have clear positive effects on most security properties, but it is clear that these effects are mitigated by newer features built on top of these. They make the VMM layer bloated [9], increasing the likelihood of bugs, giving much more control to the administrators and lead to the emergent properties of loss of uniqueness, location boundedness and monotonicity.

Especially because the technology is widely deployed, these security problems are surprising, which might underscore the need for further research, especially in the form of case studies regarding current practices in enterprises. However,

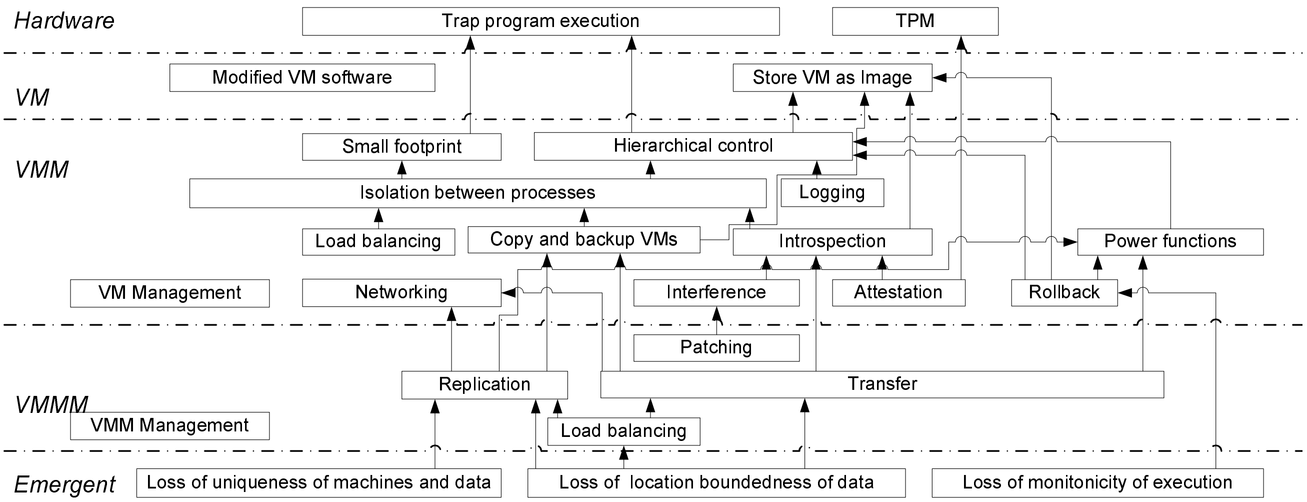

Fig. 3. Feature dependencies (Arrow from A to B indicates that A depends on B)

whatever the outcome of these case studies might be, we believe that there is a fundamental limit as to what safely can be virtualized. A network where the audit trails of the VMMs, the identity of the administrators, and the DNS servers are all virtualized, lacks a single system from which to build trust as all systems becomes fluid.

A. Recommendations

In this final section, several recommendations are discussed, concerning how to develop and use virtualization technology securely.

1) Virtualization technology design: We propose to limit the possibilities for introspection and intervention of VMMs, such that they cannot affect the application (threatening integrity) and cannot steal data from them (threatening confidentiality). If virtualization really is just an infrastructure layer, its functionality should simply focus on availability.

2) Application design: Considering the problems of bookkeeping in a virtualized infrastructure, the question can be raised whether applications running inside VMs should be redesigned to facilitate batch processing and checkpointing. A single run of a VM (from begin to end of a job) can then be validated as a whole, and security issues with restores and rollback operations do not apply any more.

3) Virtualization deployment and management: If anything, the management of this infrastructure should be more strictly organized than a physical one because it is much more powerful. In fact, the virtual infrastructure should be considered to be a complete machine in itself, which should be put in one physical location, with tight security controls.

ACKNOWLEDGMENT

This research is/was supported by the research program Sentinels (www.sentinels.nl). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs under project number TIT.7628.

REFERENCES

[1] J. Smith and R. Nair, “The architecture of virtual machines,” *Computer*, vol. 38, no. 5, pp. 32–38, 2005.

[2] J. Webster and R. T. Watson, “Analyzing the past to prepare for the future: Writing a literature review,” *MIS Quarterly*, vol. 26, pp. 13–23, 2002.

[3] S. Vaarala, “Security considerations of commodity x86 virtualization,” May 2006, Helsinki University of Technology, Licentiate Thesis.

[4] Corsec Security, “VMware Inc ESX Server 3.0.2 and VirtualCenter 2.0.2,” www.commoncriteriaportal.org/files/epfiles/vmware-sec-e.pdf, 2008, common Criteria Security Target, EAL4+, Retrieved 2009-04-20.

[5] R. Perez, L. van Doorn, and R. Sailer, “Virtualization and Hardware-Based Security,” *IEEE Security & Privacy*, vol. 6, no. 5, pp. 24–31, 2008.

[6] B. Jansen, H. Ramasamy, M. Schunter, and A. Tanner, “Architecting Dependable and Secure Systems Using Virtualization,” *Lecture Notes In Computer Science*, pp. 124–149, 2008.

[7] R. Uhlig, G. Neiger, D. Rodgers, A. Santoni, F. Martins, A. Anderson, S. Bennett, A. Kagi, F. Leung, and L. Smith, “Intel virtualization technology,” *Computer*, vol. 38, no. 5, pp. 48–56, 2005.

[8] L. van Doorn, “Hardware virtualization trends,” in *Proceedings of the 2nd international conference on Virtual execution environments*. ACM New York, NY, USA, 2006, pp. 45–45.

[9] S. Bratus, M. Locasto, A. Ramaswamy, and S. Smith, “Traps, events, emulation, and enforcement: managing the yin and yang of virtualization-based security,” in *VMSec ’08: Proceedings of the 1st ACM workshop on Virtual machine security*. New York, NY, USA: ACM, 2008, pp. 49–58.

[10] Distributed Management Task Force, Inc., “Open virtualization format specification,” www.dmtf.org/standards/published_documents/DSP0243_1.0.0.pdf, February 2009, version 1.0.0, Retrieved 2009-04-20.

[11] S. Crosby and D. Brown, “The virtualization reality,” *Queue*, vol. 4, no. 10, pp. 34–41, 2007.

[12] B. D. Payne, M. Carbone, M. Sharif, and W. Lee, “Lares: An architecture for secure active monitoring using virtualization,” *Security and Privacy, IEEE Symposium on*, pp. 233–247, 2008.

[13] P. Karger and D. Safford, “I/O for Virtual Machine Monitors: Security and Performance Issues,” *IEEE Security & Privacy*, vol. 6, no. 5, pp. 16–23, 2008.

[14] VMware, “VMware ESXi 3.5 datasheet,” www.vmware.com/files/pdf/vmware_esxi_datasheet.pdf, 2008, retrieved 2009-03-18.

[15] T. Ormandy, “An empirical study into the security exposure to host of hostile virtualized environments,” in *CanSecWest 2007*, Vancouver BC, April 2007.

Group	Feature	Benefits	Threats	Vulnerabilities	Attacks	Confidentiality	Integrity	Availability	Non-repudiation	Authenticity
Hardware	trap program execution	facilitate virtualization	N/A	N/A	N/A					
	TPM	verify VMM	N/A	N/A	N/A	+	+	+	+	+
VM	store VM as image	backup VM	VM image modification	software	VMM ► VM	-	-	+		
	modified VM software	security checks	attack VMM	software	VM ► VMM	+	+	+	+	+
VMM	small footprint	fewer vulnerabilities	VMM rootkit	software	VMM ► VM	+	+	+	+	+
	hierarchical control	control untrusted VM	enlarged footprint, VM escape	software	VM ► VMM	+	+	+	+	+
	isolation between processes	isolate untrusted VM	N/A	covert channels	VM ► VM	+	+	+	+	+
	logging	store log securely	N/A	N/A	N/A		+		+	
	load balancing	prevent DOS	N/A	software	VM ► VM			+		
	copy and backup VMs	facilitate backup	VM branching	management	N/A	-		+		
	introspection	virus scan, attestation	misuse	software	VMM ► VM	±	±	+		
	attestation	authenticate VM	N/A	N/A	N/A	+	+		+	+
	interference	prevent and stop attacks	misuse	software	VMM ► VM	±	±	±	±	±
	power functions	recover from errors	start unpatched VM	management	VMM ► VM			+		
	networking	isolation	network traffic snoop	software	VMM ► VM, network ► VM	±	±	±	±	±
	rollback	recover from errors	rollback patch	management	VMM ► VM	-	-	+	-	-
	VM management	facilitate control	abuse	management		±	±	+	±	±
VMMM	transfer	migrate if error	DOS, in-transfer modification, transfer off-site	management	Network ► VM, VMM ► VM, VMMM ► VMM	-	-	+	-	-
	replication	anti DOS	clone, replicate	management	VMMM ► VMM	-	-	+	-	-
	load balancing	anti DOS	abuse	management	VMMM ► VMM	-	-	+	-	-
	patching	facilitate patching	N/A	N/A	N/A	±	±	±		
	VMM management	facilitate control	abuse	abuse	VMMM ► VMM	±	±	+	±	±
emergent	loss of uniqueness	N/A	exploits	management	N/A	-		+	-	-
	loss of location-boundedness	N/A	exploits	management	N/A	-	-	+	-	-
	loss of monotonicity	N/A	exploits	management	N/A		-		-	-

Fig. 4. Virtualization threats

[16] NIST, "CVE-2009-1244," http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-1244, April 2009, retrieved 2009-04-20.

[17] T. Garfinkel and A. Warfield, "What virtualization can do for security," *;LOGIN:*, vol. 32, no. 6, pp. 28–34, December 2007.

[18] G. Dunlap, S. T. King, S. Cinar, M. Basrai, and P. Chen, "Revirt: Enabling intrusion analysis through virtual-machine logging and replay," in *In Proceedings of the 2002 Symposium on Operating Systems Design and Implementation (OSDI)*. New York, NY, USA: ACM, 2002, pp. 211–224.

[19] K. Nance, M. Bishop, and B. Hay, "Virtual Machine Introspection: Observation or Interference?" *IEEE Security & Privacy*, vol. 6, no. 5, pp. 32–37, 2008.

[20] T. Garfinkel and M. Rosenblum, "A virtual machine introspection based architecture for intrusion detection," in *Proceedings of Network and Distributed Systems Security Symposium*, 2003, pp. 191–206.

[21] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra: a virtual machine-based platform for trusted computing," *SIGOPS Oper. Syst. Rev.*, vol. 37, no. 5, pp. 193–206, 2003.

[22] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: Security challenges in virtual machine based computing environments," in *10th Workshop on Hot Topics in Operating Systems*, 2005.

[23] Microsoft Corporation, "Windows Server 2008 Hyper-V Technical Overview," http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx, 2008, retrieved 2009-04-20.

[24] VMware, "VMware VMotion datasheet," www.vmware.com/pdf/vmotion_datasheet.pdf, 2007, retrieved 2009-03-18.

[25] Citrix, "Xenserver administrator's guide," support.citrix.com/servlet/KbServlet/download/18051-102-19048/reference.pdf, September 2008, retrieved 2009-03-18.

[26] J. Oberheide, E. Cooke, and F. Jahanian, "Empirical exploitation of live virtual machine migration," in *Proceedings of BlackHat DC convention*, 2008.

[27] M. Rosenblum and T. Garfinkel, "Virtual machine monitors: Current technology and future trends," *Computer*, vol. 38, no. 5, pp. 39–47, 2005.

[28] VMware, "VMware vCenter Update Manager," www.vmware.com/files/pdf/update_manager_datasheet.pdf, 2008, product datasheet, Retrieved 2009-04-20.

[29] C. Gebhardt and A. Tomlinson, "Security consideration for virtualization," Royal Holloway, University of London, Egham, Surrey TW20 0EX, England, Tech. Rep. RHUL-MA-2008-16, 2008.

[30] S. Vaughan-Nichols, "Virtualization Sparks Security Concerns," *Computer*, vol. 41, no. 8, pp. 13–15, 2008.

[31] The Register, "Webhost hack wipes out data for 100,000 sites," http://www.theregister.co.uk/2009/06/08/webhost_attack/, 2009, retrieved 2009-06-25.