Security consideration for virtualization

Carl Gebhardt and Allan Tomlinson

Technical Report
RHUL–MA–2008–16

Royal Holloway
University of London

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

Abstract

Virtualization is not a new technology, but has recently experienced a resurgence of interest among industry and research. New products and technologies are emerging quickly, and are being deployed with little considerations to security concerns. It is vital to understand that virtualization does not improve security by default. Hence, any aspect of virtualization needs to undergo constant security analysis and audit. Virtualization is a changeable and very dynamic field with an uncertain outcome. In this paper we outline the security model of hypervisors and illustrate the significance of ongoing security analysis by describing different state of the art threat models. Finally, we provide recommendations and design considerations for a more secure virtual infrastructure.

1 Introduction

Virtual machines (VMs) offer many benefits for data-centers, developers as well as consumers. They provide a vast amount of flexibility, as deployment and handling is similar to that of handling a single file. VMs promise to increase the utilization of servers, whilst at the same time cutting down administration and operational cost. However, with the ever increasing popularity of modern virtualization techniques, there is a corresponding increase in the threat level.

Many vendors are aggressively pushing their products into the market trying to get their share of the business. Oracle announced their Xen based hypervisor (a synonym for Virtual Machine Monitor or VMM) in November 2007 [1]. Microsoft will be giving away their own hypervisor in 2008 for less than $30 [2]. Sun made their Xvm products, based on the Xen hypervisor, available in late 2007 [3]. At the same time, many organizations are already migrating their physical servers to virtual ones. Without careful consideration, this could lead to unforeseen behavior, and undermine security and availability.

However, virtualization has the potential to fundamentally change the way computing resources are consumed. New features are often implemented and new products rolled out before security issues are properly considered or even understood. This article therefore seeks to provide an updated insight into some of the current security issues, and to identify new threat scenarios.

The remainder of this paper is structured as follows: First we review and discuss related work in section 2. In section 3 we outline the security base of modern hypervisors and investigate how the use of a hypervisor impacts the overall system security. In section 4 we explain the importance of virtual machine detection for security. We provide an updated view of potential threats in section 5. In section 6 we conclude and provide recommendations to strengthen virtual security in section 7. Finally, we discuss future work in section 8.

2 Related Work

Security aspects of virtualization has been an ongoing field of research ever since the technology emerged in the 1960's. One of the first security analyses was carried out by Madnick and Donovan in the early 1970's [4]. Since then researchers have been investigating the various security benefits and threats of virtualization. Some papers have documented the use of Virtual Machines for malware analysis [5],[6]. This has also generated new thinking amongst malware designers, who are now trying to hide their malicious behavior while they suspect being inside a virtual machine. Consequently, the ability to detect the presence of a virtual machine is constantly undergoing research [7],[8]. The combination of techniques such as trusted computing and the isolation provided by different virtualization technologies are another popular field of research. Applications range from enhancing Grid security [9],[10] to embedded car to car communication [11].

A security analysis of virtual machines was published in 2005 [12], and more recently the vulnerabilities of virtualized environments have been stressed by [13]. This has provided a sophisticated introduction to the security aspects of virtualized environments. However, past work on virtualization security has often focused on detection; or exploiting a specific virtualization product, implementation or design; rather then identifying the more general threats to virtualization itself. This is what we seek to do in the following.

3 Security Anatomy

The reasons to use virtualization are diverse. The main security goal, however, is the provision of a strong separation of resources being used by virtualized entities. Thus, a consumer of a resource does not know – and does not need to know – where or how this resource is implemented. In this paper our focus is on security, therefore we begin differentiating between security provided by virtualization technologies and security for the hypervisor itself.

Security provided by hypervisors is based on their ability to strongly isolate processes from each other. In an x86 architecture, once a malicious party has access to ring 0, the most privileged mode of operation, there is no limitation to what it is authorized to do. It can read, write and modify any data in memory or harddisk. Numerous approaches to process isolation have been developed, such as Jails, Chroot, JAVA or SELinux. However, some difficulties – such as the in-depth knowledge required to handle the complex configuration process of SELinux, or performance issues of JAVA – have prevented these techniques from being widely used.

The problem of weak process isolation is further exacerbated by legacy CPU architectures which fail to provide fine grained access restrictions to memory or busses. Features, mostly added for performance reasons, allow any device to bypass the CPU and thus any kind of control mechanism. New technologies for the x86 architecture promise to mitigate this problem. Intel's VT and AMD Pacifica offer a new set of security and virtualization based features. Even though they are not compatible with each other, their principles are the same: adding support for virtualization and providing a more fine grained access control to hardware resources.

Security for hypervisors has different aspects depending on the application. The work mentioned in section 2 has already discussed the specific security concerns of different hypervisors. The hypervisor is different to past software sandboxing techniques, as it addresses isolation on a more rudimentary level. Rather than isolate processes, the hypervisor isolates whole operating systems. Each operating system is then responsible for its own security mechanisms. The hypervisor handles access to the privileged part of the hardware and provides each isolated guest the illusion of running directly on dedicated hardware. If a malicious party manages to compromise one virtual system, it still can't access resources used by different virtual guests or the host system.

While the code base of a Virtual Machine Monitor such as Xen is relatively small (i.e. tens of thousand lines of code) incompatible hardware virtualization support of AMD and Intel will result in some extra code being added into Xen. Consequently, this will result in a different set of binaries. In general, less code means fewer bugs, thus fewer potential security flaws. Furthermore, the smaller codebase also allows a much higher assurance level. The team of J. McDermott et. al. are working on a high robustness hypervisor called xenon [14, 15] to fulfill the requirements of a strong EAL 5/6 criteria. They are intending to achieve this by, sacrificing features and simplifying code as well as separating policy-enforcing code.

An alternative approach towards strong isolation and untrusted kernel code can be achieved through a microkernel or an exokernel architecture

[16]. A microkernel itself only implements basic methods necessary to provide inter-process-communication, thread and address space management. Recently, the L4 microkernel has been heavily discussed as a potential hypervisor [17]. From a security perspective, a microkernel based hypervisor is an appealing approach as it promises to increase both reliability and isolation. However, microkernels have to step back in terms of performance and thus are mostly used for proof-of-concept designs and research projects.

Moreover, the granularity of isolation with a microkernel is also dependent on the requirements and technique used. High isolation virtual systems, for instance IBMs logical partitions (LPARs) separate virtual machines on the hardware level. LPARs are Common Criteria EAL5 certifiable, as they provide the same level of isolation offered by dedicated hardware. Xen, VMware, Microkernels and any other software product, rely on software partitioning to ensure isolation. Additionally, the level of isolation for software partitioning is very similar and mostly differs in implementation detail and performance. A software based approach is a good compromise and offers a high amount of flexibility, but also inflicts new security concerns.

4 Virtual Machine Detection

An important aspect of security for virtualization is the ability to detect the presence of a virtual machine or hypervisor. From a security perspective this information will provide valuable knowledge for further attacks. Consequently, virtual machine detection has been discussed by numerous research papers [7],[8]. Some approaches to hypervisor detection, such as localization of the Interrupt Descriptor Table (IDT) and Global Descriptor Table (GDT) may not work with future virtualization hardware support of the x86 architecture. However, identifying a virtual machine by registry string search, vendor specific MAC addresses or looking for a virtualization support application (i.e application responsible for clipboard management etc.) is still likely to be reliable in the future. In particular, a VMware specific implementation of a communication channel allows VM detection with high probability. This communication channel may then be triggered by the guest operating system by filling the processor register with a particular value (VMXh).

Listen et al. [7] examined different mechanisms to change VMware's behavior, but results are not applicable to live systems. However, modifying the guest operating system to use certain rootkit-like techniques to prevent VM detection, might be useful in a hostile environment, such as honeypots or malicious code analysis. For example, a patch provided by Kortchinsky [18] allows honeypot specific VMware modification, whereas Kirch [19] provides configuration advice to defer detection.

In 2006 the hypervisor based rootkit developed by security analyst Joanna Rutkowska [20] created a lot of attention. This was mainly because it was targeting new software and hardware techniques such as Windows Vista and AMDs Pacificia, however, it also targeted a long feared threat – undetectable malware. This stealth property was based on the fact, that the malware could be executed as a hypervisor and thus run underneath the operating system. This certainly makes detection even for experienced users more difficult, but not impossible. First of all, there will be a mapping between physical hardware and the virtual interfaces, which the operating system will see. Secondly, the consumption of resources such as CPU time and memory will create an anomaly on the system. Those anomalies will also be reflected in latency and timing characteristics. More importantly this technique allows the detection of all hypervisors, and not only the proposed malware.

Preventing hypervisor detection is a double edged sword. On the one hand it could prevent malware from detecting that it is running inside a virtual machine, on the other hand it could prevent an operating system from detecting that it is running in a hostile environment.

5 Threat Analysis

Virtualization is a very dynamic field, requiring ongoing research and security analysis to identify possible threats. In this section we provide an updated review of past security analyses [12, 21], and outline new threats which we consider to be important.

Once an entity successfully detects being in a virtual environment, a new scope of attack vectors apply. The type of attack coming from within the guest operating system, is mostly dependent on the virtualization product and version being used. However, threats from the virtual machine can be denial of service, such as fork bombing, crashing the virtual machine, or in the worst case VM escaping [9, 13]. A VM escape can be any type of attack, which compromises the isolation and interacts directly with the hypervisor or host system.

5.1 Indirect Threats

We consider indirect threats to be those that could enable, or lead to, the direct threats discussed in the following. For instance, an early migration to virtualization and a specific product could led to unforeseen behavior and undermine existing security measures. The current trend in many organizations is to hop on the virtualization bandwagon without knowing or calculating the potential risks properly. An operator of a virtual data center must not assume that running a virtual machine increases security by default. Furthermore, patches and security updates have to be applied to the guest operating system, host operating system as well as the hypervisor itself. This could inflict hidden costs as the use of virtualization does not decrease the operational costs by default either. Only if the virtual center has been carefully designed and potential pitfalls mitigated, can the costs can be reduced and security increased.

Moreover, the respective virtualization management tools have to be considered and configured carefully. As most research and programming efforts target a secure hypervisor or virtual machine, most overlook the relevance of the management interface. Malicious entities will not attack well known or proven safeguards, but will search for weaknesses in the rest of the security chain.

Future attacks or design flaws are more likely to affect management applications and tools. The difficult task of managing a virtualized infrastructure requires a variety of different management tools. Those tools will likely be web interfaces, for convenience, or special management consoles, for performance reasons. Virtualization products themselves and the management of their infrastructure is becoming an increasingly complex task, which can easily overburden administrators and executives. Thus a sophisticated security concept could easily be undermined by a simple cross site script vulnerability in a management interface.

Furthermore, patch management for a large numbers of guest systems will become crucial. Applying security critical patches as soon as possible, reduces the potential time window for an attack; at the same time it reduces the time to thoroughly test new patches. New ideas and products [22] enable administrators to apply guest operating patches to the hypervisor. This allows them to apply one patch to many different machines at the same time, as well as offering the possibility to easily rollback a patch. Unfortunately, the same patching interface of the hypervisor could be used to transparently apply harmful code to all guest operating systems. As a result, security software inside a guest system might not properly detect that it has been compromised.

5.2 Information Leakage

We now consider the general threat of information leaking from an isolated virtual machine. In virtualization terminology, the word "paravirtualization" is used for guest operating systems, which are aware that they are running on top of a hypervisor. Such "enlightened" guests pose a new threat since they weaken one of the virtualization security goals namely, strong isolation. Isolation may also be deliberately weakened to allow inter-OS communication or for performance reasons. Both these threats may result in information leakage through covert channels. The possibility or the presence of a communication channel between guests or host system require new mechanisms to monitor and manage them. IBMs sHype, for instance, is a hypervisor security architecture for several hypervisors [23]. sHypes goal is to provide strong isolation, policy-based sharing and communication between Virtual Machines. It also relies on the Trusted Platform module to guarantee integrity of the hypervisor itself. However, adding complex control mechanisms into the hypervisor could lead to a more secure system design; this is, however, where Microkernels are today.

Despite the fact that information leakage through covert channels is not currently considered as a serious threat, it demonstrates how guest isolation can be weakened. Liston et. al. presented a proof-of-concept application which utilizes VMware's communication channel to establish a channel between two virtual machines running on the same host [7]. In general, it is difficult to prevent covert channels, but the possibility of hidden communication has a direct impact on how virtual machines with different security requirements can be run on the same host. This is true not only for deployment, but also for operation and fault tolerance. For example, security policy might prohibit applications classified as "top secret", from running together with "confidential" applications on the same host. Where applications from the same security level run on one host, all data must be kept confidential on a virtual machine unless it wishes to communicate over the network. Jaeger et. al. [24] investigated the risk management of covert information flow in virtual machine systems.

5.3 Integrity Violation

In addition to confidentiality, many systems require that data or applications are not modified either deliberately or by accident. Detecting such modifications and ensuring the integrity of a virtual image is difficult to achieve. A virtual disk could be copied, modified and replaced within minutes. Thus a malicious party could inject code into a disk image without the knowledge of the legitimate owner or user. Even if the disk image is encrypted the attacker could successfully implement a denial of service attack by randomly modifying an image.

5.4 Unauthorized Access

One much extolled feature of virtualization is the ability to create snapshots and rollback to a previous system state. This allows the user, in case of misconfiguration, crash or data loss, to return to an earlier system state. The snapshot includes CPU state and memory content – it is an exact copy of the earlier system. This image can contain security information such as login states, server tickets and credentials. This enables a new threat, unauthorized access to restricted resources. The snapshot could be leaked, giving an attacker the possibility to analyze an image and prepare further attacks. A virtual machine image, including security credentials, certificates or similar could be stolen, run and compromised from a different location. This potentially undermines the complete company's security concept. i.e. allowing the thief to access the same resources, the copied system would be granted access to. Additionally, the important randomness used for security applications becomes worthless as a snapshot turns any operating system into a predictable machine. This makes the system vulnerable to replay attacks. A compromised image might also be used to fake a person's or a company's identity, causing serious damage such as fraud.

A second threat of unauthorized access arises from "virtual appliances". These are virtual machine images that contain operating system, middleware and application. Virtual appliances therefore offer easy deployability and make it easy to try-out new applications. However such a virtual appliance may contain vulnerabilities, for example default configurations and passwords, giving an attacker unauthorized access to system resources. In a worst case scenario the newly installed appliance could already contain malicious code.

5.5 Denial of Service

We have seen previously how the threat to the integrity of virtual images also enables a Denial of Service (DoS) threat. DoS attacks themselves can lead to a number of other threats. These threats arise from the ability of virtualization to allocate resources and even migrate running operating systems between hosts. This is beneficial for the management and availability of an IT-infrastructure, but also adds a new level of complexity. A malicious entity could therefore force the migration of a running virtual machine on to a compromised host by performing a DoS attack on the target machine.

Moreover, as pointed out by Ramasamy [25], the reliability of a virtualized system depends on the amount of concurrent virtual systems running on the same host. However, a hardware failure or successful DoS attack on the a host will take down all virtual machines on the platform.

5.6 Exploits

Even although VM escape has been a concern since virtualization has emerged, no VM escape code has yet been seen in the wild. A couple of security issues with VMware's Workstation have however been discovered recently [26],[27],[28],[29]. The type of attacks depend on specific software product and version. These exploits can be used under certain circumstances to cause the host environment to execute arbitrary code, but they require a weak configuration or a specific user interaction (i.e. Drag & Drop). Additionally, they target a specific VMware product and version, some of the vulnerabilities have already been patched and it is most likely that the rest will be patched in the near future.

Still, even if a problem with the hypervisor is not exploitable, it might be used to crash a virtual machine. As stressed by Ormandy [13] none of the virtual machines tested in this work was found to be robust enough to withstand simple I/O fuzzy testing. Further, Ormandy presents multiple exploitable flaws in some popular closed and open-source virtualization products.

The open source hypervisor, Xen, uses native xeno-aware operating system drivers in favor of portability and flexibility, VMware provides guest operating system drivers for their standardized interfaces. Drivers have direct access to the hardware and thus contain a considerable amount of untrusted and privileged code. Therefore poorly written drivers pose a threat for every computer system. For instance, a heap overflow in Xen's NE2000 driver has been discovered by Ormandy [13], which allows a malicious entity within a guest operating system to execute arbitrary code on the hosting system [30].

6 Conclusion

Security aspects of modern virtualization are still in an early stage of evolution. Moreover, this is a highly dynamic and changeable field, which requires continuous research and evaluation. New products and technologies are constantly emerging offering security solutions, as well as creating new security problems. As a consequence this review provides an updated threat assessment for virtual systems, identifying and classified several threats as well as outlining the practical implications.

We are aware that some of these threats may seem difficult to exploit today, however they may well be seen in real world applications in the future. We are also aware that some threats, dealing with software issues, are based on inherently insecure architectures such as x86. Current development of hardware support for virtualization and access control on this architecture promises to mitigate this problem.

But complex software as it is used in virtualization will always have security issues, and care has to be taken to ensure that these issues are not easy exploitable. Adding features for convenience and performance reasons can easily weaken isolation or completely compromise security.

The ability to manage and monitor a virtual data center or even a single virtual machine will become a centerpiece of future virtual infrastructures. This will result in a complex combination of different tools from different sources, where each tool will have its unique vulnerabilities. The hypervisor itself is here only a small but very important piece.

7 Recommendations

Virtualization is not naturally insecure when compared to the security flaws contained in a modern operating system. But it does not increase security by default. It is a powerful tool to optimize existing and future IT infrastructures. However, hypervisors are as vulnerable as any other piece of software as they become more sophisticated and more complex. Therefore it is vital to carefully plan the virtual infrastructure. The preceding discussion allows us to present some design considerations and recommendations:

7.1 planning

The security model should be well thought out, considering the possibility of a potential VM escape. Thus only systems with similar security requirements should be grouped together on one host. Further, the existing security policies should be adopted and transformed, to satisfy the need of a virtualized environment. This includes a suitable deployment strategy. Also it is essential that the appropriate tools and technologies are chosen right from the process of migration. An access control model can be enforced with a mandatory access control such as [23] or Xen Security Modules (XSM).

7.2 configuring

The whole infrastructure should be hardened against possible attacks, beginning with the host operating system, the hypervisor; and including guest systems and the infrastructure management tools. As in every secure configuration, the attack surface should be reduced to a minimum by disabling unnecessary features and functions (i.e. disable unused virtualized hardware). Virtualization offers many benefits, but the complex interaction between the required tools makes maintaing security a challenging and sophisticated task. Secure VMware specific configuration recommendations have been made by Kirch [19], Liston [7] and Ormandy [13] section V. Best practices and security design considerations for Xen can be found at the Xen wiki [31].

7.3 managing

Exploiting the hypervisor is currently the biggest fear of the virtualization industry, but simpler threats i.e. cross site scripting or hijacking a management session, pose much bigger security risks. Managing access credentials and monitoring virtual machine behavior to track anomalies as soon as they appear is a key benefit to secure a virtual IT infrastructure. Of course, it is vital to keep the system on a up-to-date patch level – this includes the host system, the guest system, the hypervisor itself as well as any other management tools involved.

8 Future Work

An active area of research is the combination of ideas and concepts from trusted computing, with the strong isolation provided by virtualization [9], [10],[11]. Trusted computing has the potential to improve hypervisor security by ensuring the correct launch and integrity of any hypervisor. However, this inflicts the question of where to start putting trust in a virtualized platform? For instance, Xen dom0 is running in privileged mode, assuming that it is a correct and trustful base for all guests. Designing a new trust concept for virtualized platforms provides a challenging task for future work. Additionally, there are still unaddressed issues with the current Trusted Platform Module design such as performance consideration and the seamless integration into virtualized systems. Thus, IBM is currently investigating detailed solutions for virtual TPMs [32]. Our current work incorporates a TPM based approach to hash whole virtual disk images in a fast and secure manner.

In the future there might be dedicated hardware to securely manage and monitor hardware resources or virtual environments. Phoenix, one of the biggest manufacturers of PC Bioses, recently publicized their vision of a new generation of PCs which integrates a hypervisor directly into the BIOS [33].

Furthermore, antivirus or intrusion detection will likely be installed on the host systems to monitor concurrent virtual machines simultaneously. Access monitoring and policy enforcing such as discussed in [23] and [34] are other current areas of research. In Xen version 3.2 a new standardized security framework, Xen Security Module, has been introduced. This technique provides a basis for developing new security modules, such as access control or intrusion detection mechanism.

Additionally, AMD's and Intel's implementation of hardware based virtual machines have to prove their robustness under rigorous security analysis. Due to its dynamic nature, future research in virtualization hardware and software will require an ongoing review of vulnerabilities and a continuous threat monitoring.

References

[1] "Oracle Unveils Oracle® VM." [Online]. Available: http://www.oracle.com/corporate/press/2007_nov/ovm-ga-111107.html

[2] "Microsoft Outlines Pricing, Packaging and Licensing for Windows Server 2008, Including the New Microsoft Hyper-V Server Product." [Online]. Available: http://www.microsoft.com/presspass/press/2007/nov07/11-12HyperVPR.mspx

[3] "Sun xvm." [Online]. Available: http://www.sun.com/software/products/xvm/

[4] S. E. Madnick and J. J. Donovan, "Application and analysis of the virtual machine approach to information system security and isolation," *Proceedings of the workshop on virtual computer systems*, pp. 210 – 224, 1973.

[5] P. Ferrie, "Attacks on Virtual Machine Emulators," Symantec Security Response, Tech. Rep., 2006.

[6] J. R. Crandall, G. Wassermann, D. A. de Oliveira, Z. Su, S. Wu, and F. T. Chong, "Temporal search: detecting hidden malware timebombs with virtual machines," *SIGARCH Computer Architecture News*, no. 25-36, 2006.

[7] T. Liston and E. Skoudis, "On the Cutting Edge: Thwarting Virtual Machine Detection," SANS Internet Storm Center, 2006.

[8] A. A. Omella, "Methods for virtual machine detection," Grupo S21sec Gestión S.A., June 2006.

[9] H. Lohr, H. V. Ramasamy, A.-R. Sadeghi, S. Schulz, M. Schunter, and C. Stuble, "Enhancing grid security using trusted virtualization." in *ATC*, ser. Lecture Notes in Computer Science, B. Xiao, L. T. Yang, J. Ma, C. Muller-Schloer, and Y. Hua, Eds., vol. 4610. Springer, 2007, pp. 372–384. [Online]. Available: http://dblp.uni-trier.de/db/conf/atc/atc2007.html#LohrRSSSS07

[10] H. Chen, F. Zhang, C. Chen, Z. Yang, R. Chen, B. Zang, and W. Mao, "Preserving Software Privacy from Hostile OSes Using Virtualization."

[11] F. Stumpf, M. Benz, M. Hermanowski, and C. Eckert, "An approach to a trustworthy system architecture using virtualization," in *Proceedings of the 4th International Conference on Autonomic and Trusted Computing (ATC-2007)*, ser. Lecture Notes in Computer Science, vol. 4158. Hong Kong, China: Springer-Verlag, July 2007, pp. 191–202.

[12] T. Garfinkel and M. Rosenblum, "When virtual is harder than real: security challenges in virtual machine based computing environments," in *HOTOS'05: Proceedings of the 10th conference on Hot Topics in Operating Systems*. Berkeley, CA, USA: USENIX Association, 2005, pp. 20–20.

[13] T. Ormandy, "An Emperical Study into the Security Exposure to Hosts of Hostile Virtualized Environments," Google, Inc., Tech. Rep., 2007.

[14] *An Open-Source High-Robustness Virtual Machine Monitor*. The 22st Annual Computer Security Applications Conference, Dec 2006.

[15] J. McDermott, "Xenon: High-assurance xen." [Online]. Available: http://www.xensource.com/files/xensummit_4/XenSummitSpring07_McDermott.pdf

[16] G. R. Ganger, D. R. Engler, M. F. Kaashoek, H. M. Briceno, R. Hunt, and T. Pinckney, “Fast and flexible application-level networking on exokernel systems,” *ACM Transactions on Computer Systems*, vol. 20, no. 1, pp. 49–83, February 2002.

[17] S. Biemüller, “Hardware-supported virtualization for the l4 microkernel,” 2006.

[18] K. Kortchinsky, “Honey-vmware patch.” [Online]. Available: http://honeynet.rstack.org/tools/vmpatch.c

[19] J. Kirch, “Virtual machine security guidelines,” The Center for Internet Security, Tech. Rep., 2007.

[20] J. Rutkowska, “Subverting vista kernel for fun and profit.” [Online]. Available: http://www.invisiblethings.org/papers/joanna%20rutkowska%20-%20subverting%20vista%20kernel.ppt

[21] I. Arce, “Ghost in the virtual machine,” *IEEE Security and Privacy*, vol. 5, no. 4, pp. 68–71, 2007.

[22] BlueLane Technology, “Servershield.” [Online]. Available: http://www.bluelane.com/products/servershield/

[23] IBM, “shype - secure hypervisor.” [Online]. Available: http://www.research.ibm.com/secure_systems_department/projects/hypervisor/

[24] T. Jaeger, R. Sailer, and Y. Sreenivasan, “Managing the risk of covert information flows in virtual machine systems.” in *SACMAT*, V. Lotz and B. M. Thuraisingham, Eds. ACM, 2007, pp. 81–90. [Online]. Available: http://dblp.uni-trier.de/db/conf/sacmat/sacmat2007.html#JaegerSS07

[25] H. V. Ramasamy and M. Schunter, “Architecting dependable systems using virtualization,” IBM Zurich Research Laboratory, Tech. Rep., 2007.

[26] “Vmware workstation shared folders directory traversal vulnerability.” [Online]. Available: http://labs.idefense.com/intelligence/vulnerabilities/display.php?id=521

[27] GOODFELLAS Security Research TEAM, “VmWare Inc version 6.0.0 CreateProcess & CreateProcessEx Remode Code Execution Exploit.” [Online]. Available: http://www.milw0rm.com/exploits/4245

[28] ——, “vielib.dll 2.2.5.42958 VmWare Inc version 6.0.0 Remode Code Execution Exploit.” [Online]. Available: http://www.milw0rm.com/exploits/4244

[29] ——, “IntraProcessLogging.dll 5.5.3.42958 VmWare Inc Arbitrary Data Write Exploit.” [Online]. Available: http://www.milw0rm.com/exploits/4240

[30] Redhat, “xen security update.” [Online]. Available: http://rhn.redhat.com/errata/RHSA-2007-0323.html

[31] Xen, “Users’ manual. xen v3.0,” Xen, Tech. Rep.

[32] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, “vtpm: virtualizing the trusted platform module,” in *USENIX-SS’06: Proceedings of the 15th conference on USENIX Security Symposium.* Berkeley, CA, USA: USENIX Association, 2006, pp. 21–21.

[33] Phoenix, “Phoenix technologies ltd. to present pc 3.0$^{\text{TM}}$ vision and financial results at upcoming investor conferences.” [Online]. Available: http://www.phoenix.com/en/About+Phoenix/Investors/News+Releases/

[34] B. D. Payne, M. Carbone, and W. Lee, “Secure and flexible monitoring of virtual machines,” in *Proceedings of the 23rd Annual Computer Security Applications Conference (ACSAC 2007)*, December 2007.