On State of The Art in Virtual Machine Security

Qian Chen*, Rajat Mehrotra*, Abhishek Dubey†, Sherif Abdelwahed*, Krisa Rowland‡
*Electrical and Computer Engineering, Mississippi State University, Miss. State, MS
†Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN
‡US Army Engineer Research and Development Center, Vicksburg, MS

***Abstract*—Data centers and computing service providers are striving to improve the utilization of their computing resources. This is primarily due to the need of resources to be more economical and power efficient. Virtualization is one of the concepts that provide flexibility to host multiple operating system stacks on a single hardware. By effectively partitioning the computing resources, it reduces the total number of physical servers and consolidates several services on a single physical rack. Each virtual machine behaves like an independent machine (may be duplicate of the original one) while the scheduling of hardware resources among different virtual machines is performed with the help of a Virtual Machine Monitor (VMM). Proliferation of virtual machines in the enterprise architecture creates need for identification of potential security risks as well as appropriate solutions for the identified risks to ensure the integrity of the underlying applications hosted at the virtual machines. This paper describes available virtualization technologies, corresponding security vulnerabilities, and available solutions.**

I. Introduction

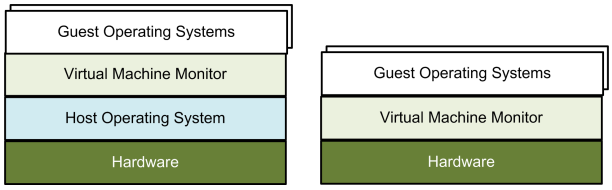
In the virtualized environment, the concept of "Guest OS" and "Host OS" are introduced, the difference between them is where the OSes are located. The former one runs on virtual machine (VM) while the later one runs on physical machine. "Virtual Machine Monitor" (VMM) (Fig. 1) is a virtualization layer that manages guest OS and its interaction with host OS or physical hardware. VMM performs process scheduling, memory management, I/O management, and network management operations as well as provides interface between the guest OS and the host OS or physical hardware. There are multiple advantages of using VMM: simplicity of implementation and debugging, limited service points for virtual machines, ability to manipulate and get the latest state information of real hardware, control and monitor the guest OS and applications running on VMs, and to strengthen OS and applications [1]. The VMs are separated from each other through firewall created by VMM. Also, the confidential code or data can be protected by the VMs [1]. As OS controls both hardware and software, it is always considered as the primary target to attack. However, in virtualized infrastructure, the VMM, which has much higher privilege than an OS, makes it another prime target of the malicious attackers to take control of the system. All of the VMs are hosted on the top of the VMM layer which makes VMs vulnerable to security compromise if VMM is under attack. Therefore, there is a tremendous need of securing VMM and VM both in case of virtualized infrastructure.

There are two primary benefits offered by virtualization technology: resource sharing and isolation.

Resource sharing: In a virtualized environment, the VMs share the physical resources such as memory, disk, and network devices of the underlying host. For instance: Xen hypervisor provides capabilities to manage both memory and hardware resources through the combination of virtual machine monitor and the privileged Xen-modified kernel [2]. The dynamic resource allocation for virtual machines on Xen-based virtualization environment can be achieved by an agent contains Libvirt and JADEX. Libvirt is a toolkit that collects data and control the resource allocation for domains on Xen hypervisor. The policy-based dynamic allocation is based on user-defined policies. The allocation of CPUs, memory and virtual block device based on their usage by domain. If the usage is less than required by user-defined polices, more resources need to be allocated, otherwise allocated resources will be reduced [2].

Data Isolation: Different VMs can communicate with each other but the communication or transfer is maintained under control by shared memory or message passing though the VMM [1]. However, the OSes and applications of the two VMs are isolated and have no shared virtual resources; Additionally, due to isolation, vulnerabilities in one VM neither affects the physical host machine nor the other VMs running on the same physical host [3]. Some techniques such as name space isolation, address isolation, copy-on-write isolation, fault isolation, and performance isolation are procedures to guarantee the security of virtual machines [1].

This paper surveys virtualization technologies, presents the advantages of using virtualization, and introduces its security vulnerabilities. Section II describes various implementation mechanisms for virtualization. Section III enumerates the security vulnerabilities and solutions present in virtual machines and Section IV lists the security threats for VM based infrastructure in web services environment. Section V presents the security aspects of **Xen** hypervisor while Section VI highlights the security issues in **VMware** products. Section VII presents available technologies to improve the virtual machine security. Finally, conclusions of the paper are presented in section VIII.



(a) Architecture of VMware Workstation and VMware Server (b) Architecture of VMware ESX Server and Microsoft Veridian

Fig. 1. Various Types of Virtualization

978-1-4673-1375-9/12/$31.00 ©2012 IEEE

II. Current Virtualization Technologies

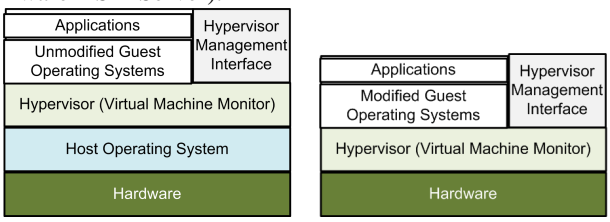
Virtualization can be achieved through diverse technologies, which depend upon the organizational needs and goals. Virtualization methodologies have one common goal that VMs can perform as an independent machine and have their own dedicated hardware resources. The primary difference among available virtualization technologies is the placement of VMM interface with respect to guest OS and physical hardware. More details of the methodologies are presented as follow:

A. Full Virtualization

"Full virtualization" is the most commonly used virtualization technology [4]. In this methodology, the operating systems and applications are designed to execute on the same architecture as the underlying physical hardware (Figure 2(a)). In this approach, I/O ports and DMA channels are considered as virtual resources. The hypervisor translates all OS instructions on the runtime and caches the results for future use while user level instructions run unmodified at native speed. VMware Workstation and VMware Server are examples of the Full Virtualization methodology.

B. Paravirtualization

Another common technique for virtualization is "Paravirtualization" (Figure 2(b)) that supplies a software interface layer between the host hardware and the modified guest OS. To implement in the virtual environment, the guest OS should be modified and be aware that it is running in the virtualized environment. Modifications inside the guest OS ensure high performance, scalability of the virtual environment. However, these changes restrict the usage of closed operating systems (e.g. Microsoft Windows). The guest OS needs to be directly supporting virtualization or open source (e.g. Xen 3.0, VMware ESX Server).



(a) Full Virtualization (b) Para Virtualization

Fig. 2. Virtualization Architectures

C. Application Virtualization

Application virtualization approach allows users to run their applications locally in a virtual execution environment with help of local resources, without installation of the application on local system [5]. Additionally, it provides flexibility through standard APIs for cross platform execution of the application. The most common example of this virtualization is Oracle Java Virtual Machine (JVM) [6]. This approach does not virtualize complete hardware or system to run the application. Instead, it just provides a thin layer between application and the guest OS for an isolated execution environment.

D. Hardware Supported Virtualization

Hardware supported virtualization was introduced by the Intel and AMD independently in two different names IVT and AMD-V respectively to improve the processor performance for common virtualization challenges of address and instruction translation through hypercalls – special virtualization instructions [3]. In this case, hypervisor resides in root mode while guest OS remains in non-root mode. The guest OS is allowed to call out to the hypervisor which is responsible for all of the resource allocations and device interactions by the hypercall (a software trap from a domain to the hypervisor [7]). All of the requests for system resources from the guest OS are served through the hypervisor.

E. Resource Virtualization

In resource virtualization, system specific resources (e.g. storage volumes, name spaces, and network resources) are either combined to create a resource pool or partitioned to assign them to various guest machines. Computer clusters or supercomputers are created by aggregating the processing elements from various computing nodes while a single piece of hard drive can be partitioned in smaller size partitions to provide storage space to multiple network storage machines.

III. Security Vulnerabilities in Virtualization

Security vulnerabilities in the virtual machine environment are usually caused by poor system designs, non-adaptive nature with respect to external attacks and incomplete knowledge of the system. Moreover there are also some security issues that exist with low priority in physical systems but are highlighted in the virtual environment [8]. Authors in [9] shows that security bugs are present in every product with varying level of exploitation opportunity. The major security issues in virtual environment are discussed in following subsections.

A. Problems exacerbated in virtual computing systems

VMs are typically unauthorized machines, which is a major security threats as the VMs' physical addresses are not predefined when they are created. Additionally, network addresses are associated to multiple VMs and VM cannot be associated with a particular physical machine. Moreover, because of quickly created, cloned, and derive, monitoring and managing VMs are even difficult. Off-line VMs that have not updated their security patches are particularly vulnerable.

B. Problems unique to Virtual Environment

New Technologies: The new software layer made by VMM must be protected and sustained as a part of a overall security strategy because compromising of the VMM induces compromising of all guests in the virtual environment. New technologies, such as vSphere [10], vSwitches [11], and centralized management tools vCenter [12] require new exploration of vulnerabilities and hardening techniques. Restricted access to vCenter by using trusted certificates should be made as a standard practice.

Communication Issues: Isolation which prohibits VMs to access applications or resources in other VM or host is the advantage of using VM technique. However, the isolation can also result in a potential security threat. For example, Shared Clipboard functionality provides access for transferring data among VMs and physical host [3]. Two VMs on the same host can communicate with each other through virtual switches (vSwitchs). The flow through vSwitches can be monitored by configuring a VM as an IDS to monitor the traffic and then to set the port group to promiscuous mode [13].

VM Escape: Configuring of VMs flexibly to meet the organizational needs decreased "isolation". In VM Escape, a malicious application running inside the VM can bypass the VMM layer to access the host machine resources. Thus, the malicious application gains root permission and takes undesired decisions. This issue can be avoided by configuring the interaction rules among the VMs and host machine in an orthodox manner or with minimum flexibility [3].

VM Monitoring: In VM implementations, a host system can manipulate the VMs, therefore any security breach in the host machine can result in major compromise of the whole infrastructure. Similarly, if a VM is connected to all other VMs and host machine with a virtualized hub, it makes VM capable of sniffing the network data flowing through all of the VMs. Moreover, the VM can redirect the network packets using ARP poisoning techniques [3] if controlled by a malicious user. Capabilities such as live migration potentially allow for virtual hosts to move from a virtual LAN to another unencrypted and undetected. This could place a VM in an exposed virtual LAN releasing its protection mechanisms.

Resource related Issues: VMs share physical resources with other VMs and host machine. Therefore, in case of denial of service attack on one VM, it will exhaust all useful physical resources. As a result, other VMs, which are functioning correctly, will be denied for the resources as well. Therefore, VMs should be allocated separate physical resources with a maximum cap on them to avoid these situations.

Separation of Duties: In the virtual realm, it becomes much easier for administrators to implement functions collapsing and to become too powerful due to the workload consolidation [14]. This is often seen as a cost benefit for many enterprise implementations. The virtual administrator can access the sensitive data even when the virtual machines are powered off. The method to access the virtual machine disk file (*vmdk*) is different from normal communication channels. For instance: mounting the *vmdk* to another virtual machine or to the host OS or even access the *vmdk* through the third party tools based on VIX API (a program for writing programs and scripts to control VMs and the products that host VMware VMs [15]). To separate the duties, the security code VGate builds an authentication layer includes both the access control and separation of duties which grants the system administrator the privilege that they required to perform their tasks.

C. Issues due to Infrastructure of Virtual Environment

This reflects the risks present outside the virtualized environment. In these security threats, the complete virtualization infrastructure is under attack. For example, the processor hosting the hypervisor layer can be attacked through "Blue Pill", the vulnerability that virtualizes the rest of machine by a thin hypervisor to trap a running instance of the OS [16] that can result in to unavailability of the complete infrastructure [8].

IV. SECURITY ISSUES IN ENTERPRISE WEB SERVICES DUE TO VIRTUALIZATION

HTTP servers, application servers, web service applications, and database management systems are housed in data center which is usually a stand-alone building, are also vulnerable to hardware/software component failures due to potential malicious attacks in virtualized environment. Hence, major threats to enterprise systems with respect to virtual machines must be considered. These threats can be categorized as follows:

A. VM Migration Issues

VMs can be transferred from one physical host to another with help of VM migration features in a halt or a live state. In this case, data inside the VM need to be protected from network. VM migration can suffer over the unsecured network with potential security threats (e.g. ARP/DHCP/DNS spoofing, and IP/route hijacking) [17]. In some cases, malicious users can initiate or redirect the live VM migration towards their own network. Additionally, malicious users from inside the organization can also easily copy the VM on a disk and carry to another place to run on their own physical machine.

B. VM Patching Issues

An enterprise system hosting web services is a collection of large number of VMs located at multiple physical locations. With the increase in number of VMs, its really difficult to manage all of those with security patches related to OS and applications. The condition of patching becomes worse if a VM rolls back to a previous state which was contaminated. In this case, system administrator needs to identify all the patches again to apply on the VMs [18].

C. VM Image Issue

publishers and users are both affected by the security vulnerabilities in VM images stored on the disk. For the VM publisher, the VM image may release sensitive information while the user might run vulnerable images provided by publishers. Both of these are great threats to web services infrastructure due to sensitivity of the stored information. Integrity and safety are required for virtual machine images, therefore an image management system that focuses on access control to ensure the integrity of VM is mentioned in paper [19]. To solve the problem of security violations, the management system supplies filters and scanners to detect and repair the security parameters of the VM image.

D. Transience

Transience [18] is the phenomenon to make large number of VMs appear and disappear suddenly from the network which helps to limit the window for an attacker to get the information related to VMs or make an attempt to hack them. However, this phenomenon is fatal for the situations where a VM becomes infected with a worm, goes off-line before detection, and again infects others vulnerable VMs once it comes on-line in future. Therefore, it also limits the time window for administrator to trace the culprit VMs and fix the issues.

E. Virtualization Network Issues

Virtual environments are composed of several components that include virtual switches, ports and port groups, virtual LANs, network interfaces, trunk ports, and NIC Teams. As mentioned above, vSwitches have many common aspects with physical switches and support greater port density can be set to one of three security policies: 1) promiscuous mode which

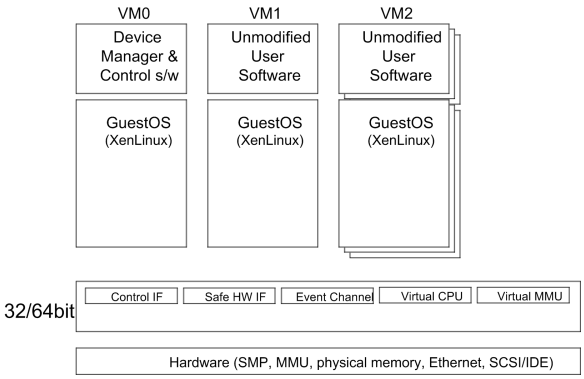


Fig. 3. Xen Architecture

prevents a network interface controller (NIC) from seeing any network traffic passing through the vSwitch for monitoring tools. 2) MAC Address Changes which rejects modified inbound MAC address traffic and 3) Forged Transmits, which rejects modified outbound MAC address traffic.

F. Virtualization Storage, Backup and Recovery related Issues

Virtual Machine File System (VMFS), local storage types, and remote network-enabled virtualization are main storage solutions used in virtualized environment. Remote storage can be implemented in several ways which induces the security issues. One is commonly based on Network-Attached Storage (NAS) using the Network File System (NFS). The second is through a fiber channel Storage Area Network (SAN) [20] which can be sniffed and hijacked. The third storage is iSCSI server [21]. All of these three storages are clear-text protocols. Traffic of NAS needs no authentication except IP-based identification; fiber channel SAN uses Sequence IDS and Sequence Count values to maintain sessions which has a 24-bit routing address mapped to a 64-bit WWN that can be spoofed during a port login (PLOGI) operation thus leads to a potential man-in-the-middle attack or temporary Denial of Service attack by corrupting the stored name on the fiber channel name server [20], [22].

V. Case Study: Xen Hypervisor Security Analysis

In Xen hypervisor architecture [23], guest OS is separated from the physical H/W through a layer of VMM called as hypervisor. OS-Kernel is modified to use the common API supplied by VMM to VMs exposed by the Xen hypervisor. VM0 is considered as Domain0 for this implementation and it defines all the rules and configurations for other domains. All other domains access H/W through Dom0 which contains the control interface for the hypervisor to manage all the elements of the system. Through this control interface, CPU is shared for each virtual machines or their priority can be configured. The kernel of the guest operating system does not need to be modified when Xen provides full virtualization mode [24].

The security challenges of the Xen that affect its performance significantly are mentioned as follow:

A. Isolation and Shared Resources

Xen has the capability to isolate the resources at one VM from the other for better quality of service and resource access control. However, Xen does not have any access control policy to manage the access of the resources in the case of flexible requirements from the user. IBM has developed a secure hypervisor technique for Xen as sHype [25], [26] that mainly manages the access to shared memory and event channel in Xen system. Additionally, [27] develops a multi-level policy for virtual infrastructure that manages resource access by multi-level security control policies and applies a mandatory access control policies in the Xen based environment.

B. Security Bottlenecks inside Xen

As the hypervisor is the most important module, the Dom0 (VM0 in the Figure 3) – more important than other Doms as it provides handle and access model to access the hardware of other VMs should be secured from various of security attacks (e.g. file virus, boot virus, Trojan horse, worm etc.). However, till date there is no extra security policy for Dom0.

C. Life Cycle of a VM in Xen

In Xen, a VM is stored in an image file that can be copied, deleted, and distributed similar to a file system. All the information related to the VM is inside the VM image file. Any public or private keys in the VM can be used again after saving the virtual machine. Malicious users can use the virtual machine image file to create other machines to launch security attacks from their system.

D. General Xen Security Modules (XSM)

XSM modules aim to create general security interfaces, allows custom security functionality as well as remove the particular code of security model from Xen [28]. Existing XSM modules include Flask, access control module (ACM), and dummy. The IBM sHype (also described as ACM) is the first to be brought forward to protect the virtualization environment by adding the mandatory access control (MAC) mechanism into Xen [25]. ACM is one module in NSA XSM since Xen3.0. Flask is a module the contains the flexible MAC which also has undergone Xen nativization [28]. The architecture of Flask is similar to Security-Enhanced Linux. The XSM security check functions also named hooks is the core part of XSM, it is responsible to check whether a virtual machine has the privilege to do a certain operations on another virtual machine. To guarantee only authorized operations can be executed, the hooks must be added before each security sensitive operation [29].

VI. Case Study: VMWare Security Analysis

VMware ESX Server runs on physical hardware and hosts virtual machines through its virtualization layer. For the security of ESX server, its firewall is considered in the level of high security which cannot install any third party firewall on it except for IP tables. VMware tools are strongly recommended to be installed in the guest OS as they optimize the overall performance, usability, and manageability of the VM.

Products of VMware (vCenter, ESX server etc) can manage large enterprise infrastructures with less operating cost and with greater flexibility in utilization of available resources. However, they also suffer from typical security issues of virtual environments. These issues include communication issues, VM escape, VM monitoring, denial of service, VM

migration and compliance issues. These issues have been already described in previous section in detail. In current section, some issues related to VM security in Xen also trouble the VMs in VMware. Additionally another major issue called "VMDetect" in VMware products (Work Station and ESX Servers) and for Microsoft Virtual PC 2007 is introduced.

A. Isolation and Shared Resources

Even each virtual machine is independent, as they can communicate with each other by the connected shared network, a virtual machine on ESX server still can be attacked from other virtual machines on the same networ. For example, in the case of DoS attack by Address Resolution Protocol (ARP) addressing spoofing, network segmentation lists the virtual machines and their linked network can mitigate the risk of DoS attacks as well as other attacks which try to ruin the virtual network [2].

B. Security Bottlenecks inside VMware

The vCenter which manages ESX servers and VMs is the prime target to the attackers. Products and security hardening documents have been developed to protect the VMs. For Example: vCenter Protect Essentials Plus [20] provides centralized patch management, asset inventory, antivirus, power management, and configuration management for protecting both virtual and physical systems. The administrator follows Security Hardening [30], [31] can also protect the ESX host as well as the VMs on it.

C. Life Cycle of a VM in VMware

VM images of VMware products can be launch security attacks similar to the VM images in Xen. "HighCloud VM-Centric Security" is the module for protecting and encrypting data in the VM and the VM images [20]. This module secures OS, memory files, copies, snapshots, templates, and the sensitive data they contain by ways of automatic key rotation, secure and centralized key management, and encryption throughout the VM life cycle.

VII. Technologies to Improve Virtual Machine Security

There are several exiting solutions to improve the security level in virtual machine based infrastructure. We categorize them similar to the categories in previous section.

A. Communication Issues related technologies

Issues related to communication of data in a virtual environment are addressed through various isolation techniques. These isolation techniques are used to group the VMs or resources to various categories. These categories restrict access of data for the VMs lying outside the category.

Name space Based: Name space isolation indicates the access privilege of one VM towards the resources in another VM. These resources include files, memory addresses, process Ids, and ports. A VMM that supports strong name space isolation does not allow a VM to access the resources at another VM directly. However, a VMM that supports weak name space allows a VM to access some resources (e.g. global file system) of other VMs [32].

Performance space based: Performance isolation indicates the VMs capability for resource usage that corresponds to another VM. Performance isolation controls the access of physical resources (e.g. CPU, link, disk space, and memory buffers etc). A VMM that supports strong performance isolation will restrict the amount of resource (e.g.1.5 GHz CPU, 1Mbps Link, 20 GB disk space) used by a VM, while weak performance isolation will allow sharing of these resources on demand basis. Additionally, the hybrid approach (combination of both strong and weak performance isolation) creates a VM cluster can be used where an intra class VMs share weak isolation while inter class VMs have strong isolation [32].

B. Trustworthy Virtual Machines

To protect the data in VM, Trusted Platform Module (TPM) is brought forward to implement the authentication that only trusted entities can access critical operation. A trusted VM can be created either through virtual TPM or through trusted VMM [18] with different protection levels. Some systems protect private data and passwords, some protect kernel and application codes, while others provide authentication to the kernels.

Virtual Machine Introspection (VMI) is a host IDS which installs security tools outside the VM at a safe place" [33]. These tools monitor virtual machines and weigh the security risks. The VMI system can be classified into threat monitoring and interfering as the following list:

Livewire [34]: This system is implemented by user lie detector. A program integrity detector checks integrity of user level programs. A signature detector scans malicious file system programs as well as using raw socket detector to catch malicious applications. Besides memory access enforcer and NIC access enforcer, control access of fault code which also prevents the misuses of NIC.

Xenon [35]: Xenon is a derivation of Xen, which provides a high-assurance separation of hypervisor. The flow control provided by Xenon is more robust than Xen. Additionally, it has some specific functionalities of self-protection for various users on the same hardware and tamper-resistance. This project utilizes concepts of re-engineering Xen internals to achieve the main target of developing Xenon, which is higher assurance open source software.

XenRIM [36] and **XenFIT** [37]: Daemon in Dom0 accesses the kernel of DomUs to monitor I/O system calls and reports intrusions bases on separate security policies of DomUs in real-time manner.

sHype [25]: It provides secure services in VM to control resource usage and sharing virtual resource between VMs. Also, sHype makes integrity measurement on Xen and VMs.

C. Reconfiguring Security Policy Based on Role of Server

It is difficult to design a universal access policy for the entire enterprise systems infrastructure based upon virtualization due to the complex configuration and variable requirements of the infrastructure. However, the abstraction layer defined in hypervisor or VMM can be utilized for reducing the complexity of access control policies. A layered approach describes in

[38] for defining access policies for systems composed of host machines and VMs. In this approach simple security policy for each layer is created and enforced that builds an overall system security policy with negligible overhead. Different layers of abstraction can be proxy server, OS kernel, and hypervisor. Each layer can have its own policy and is protected from the higher layers. The higher layers will have more abstract policies compared to lower layers.

VIII. Conclusion

In this paper, a survey of virtualization infrastructure is presented with overview of existing virtualization technologies and their security vulnerabilities. Case studies related to Xen hypervisor and VMware are presented with potential security issues in web service environment. Virtualization is not inherently unsecured, but the latest added features have created the scope for potential security threats. One of the major challenges is to manage the large number of VMs in an enterprise environment and maintain the patching level for all of them while they remain on-line or off-line. Additionally, hypervisor should not be a security bottleneck but it should take preventive and corrective measure to maintain the integrity of the VMs while taking management decisions.

IX. Acknowledgement

This work is supported in part by the NSF I/UCRC CGI Program grant number IIP-1034897 and The Engineer Research and Development Center (ERDC) at Vicksburg, MS.

References

[1] Siqin Zhao, Kang Chen, and Weimin Zheng. The application of virtual machines on system security. In *CHINAGRID '09: Proceedings of the 2009 Fourth ChinaGrid Annual Conference*, pages 222–229, Washington, DC, USA, 2009. IEEE Computer Society.

[2] D Gracanin and T Williams. *A virtual reality based interface to a dynamic resource allocation scheduler*, pages 254–258. 1995.

[3] J. Kirch. Virtual machine security guidelines. September 2007.

[4] Matias Zabaljauregui. Grid operating systems/middlewares and new virtualization techniques. Technical report, New Technologies Research Laboratory, 2009.

[5] Jenni Susan Reuben. A survey on virtual machine security. 2007.

[6] Java se at a glance. http://www.oracle.com/technetwork/java/javase/overview/index.html[Nov2011].

[7] Hypercall xen wiki. http://wiki.xen.org/xenwiki/hypercall[Nov2011].

[8] Michael T. Hoesing. Virtualization security assessment. *Information Security Journal: A Global Perspective*, 18(3):124–130, 2009.

[9] Tavis Ormandy. An empirical study into the security exposure to hosts of hostile virtualized environments. *Test*, pages 1–10, 2007.

[10] Vmware vsphere for enterprise. http://www.vmware.com/products/vsphere/overview.html[Nov2011].

[11] Using vmware: Understanding the virtual switch. http://www.virtualizationadmin.com/articles-tutorials/vmware-esx-and-vsphere-articles/installation-deployment/vmware-understanding-virtual-switch.html[Jan2012].

[12] Vmware vcenter server virtualization management. http://www.vmware.com/products/vcenter-server/overview.html[Nov2011].

[13] S. Zhou. Virtual networking. SIGOPS Oper. Syst. Rev:44:80–85, December 2010.

[14] Separation of duties in virtual environments. 2011.

[15] Vmware: Vix api blog: What is vix and why does it matter? http://blogs.vmware.com/vix/2008/07/what-is-vix-and.html[Nov2011].

[16] Blue pill (software). http://en.wikipedia.org/wiki/Blue_Pill_(malware)[Nov2011].

[17] Jon Oberheide, Evan Cooke, and Farnam Jahanian. Exploiting Live Virtual Machine Migration. In *BlackHat DC Briefings*, Washington DC, February 2008.

[18] Tal Garfinkel and Mendel Rosenblum. When virtual is harder than real: Security challenges in virtual machine based computing environments. In *In 10th Workshop on Hot Topics in Operating Systems*, 2005.

[19] Jinpeng Wei, Xiaolan Zhang, Glenn Ammons, Vasanth Bala, and Peng Ning. Managing security of virtual machine images in a cloud environment. In *Proceedings of the 2009 ACM workshop on Cloud computing security*, CCSW '09, pages 91–96, New York, NY, USA, 2009. ACM.

[20] Introduction to storage area networks. http://www.redbooks.ibm.com/redbooks/pdfs/sg245470.pdf[Jan2012].

[21] iscsi host platforms. http://www.netapp.com/us/products/protocols/ip-san/[Nov2011].

[22] Zhang Xiao and Li Zhanhuai. Research on security of storage area network. In *Proceedings of the 3rd international conference on Information security*, InfoSecu '04, pages 238–239, New York, NY, USA, 2004. ACM.

[23] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 164–177, New York, NY, USA, 2003. ACM.

[24] Geoffrey Strongin. Trusted computing using amd "pacifica" and "presidio" secure virtual machine technology. *Inf. Secur. Tech. Rep.*, 10:120–132, January 2005.

[25] Reiner Sailer, Trent Jaeger, Enriquillo Valdez, Ramon Caceres, Ronald Perez, Stefan Berger, John Linwood Griffin, and Leendert van Doorn. Building a mac-based security architecture for the xen open-source hypervisor. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 276–285, Washington, DC, USA, 2005. IEEE Computer Society.

[26] Trent Jaeger Ronald Perez Leendert van Doorn John Linwood Griffin Stefan Berger Reiner Sailer, Enriquillo Valdez. shype: Secure hypervisor approach to trusted virtualized systems. Technical report, IBM, 2005.

[27] Paul A. Karger. Multi-level security requirements for hypervisors. In *ACSAC '05: Proceedings of the 21st Annual Computer Security Applications Conference*, pages 267–275, Washington, DC, USA, 2005. IEEE Computer Society.

[28] George Coker. Xen security modules (xsm). *Xen Summit*, pages 1–33, 2006.

[29] Wei Han, Yeping He, and Liping Ding. Verifying the safety of xen security modules. *Secure Software Integration and Reliability Improvement Companion, IEEE International Conference on*, 0:30–34, 2011.

[30] C. Chaubal. Security hardening vmware infrastructure 3 (vmware esx 3.5 and vmware virtualcenter 2.5). 2008.

[31] Vmware vsphere 4.0 security hardening guide. 2010.

[32] Steve Muir, Larry Peterson, Marc Fiuczynski, Justin Cappos, and John Hartman. Proper: Privileged operations in a virtualised system environment. pages 50–56, 2005.

[33] Kara Nance, Matt Bishop, and Brian Hay. Virtual machine introspection: Observation or interference. *IEEE Security and Privacy*, 6:32–37, September 2008.

[34] Tal Garfinkel and Mendel Rosenblum. A virtual machine introspection based architecture for intrusion detection. In *In Proc. Network and Distributed Systems Security Symposium*, pages 191–206, 2003.

[35] John McDermott and Leo Freitas. A formal security policy for xenon. In *Proceedings of the 6th ACM workshop on Formal methods in security engineering*, FMSE '08, pages 43–52, New York, NY, USA, 2008. ACM.

[36] Nguyen Anh Quynh and Yoshiyasu Takefuji. A real-time integrity monitor for xen virtual machine. *Networking and Services, International conference on*, 0:90, 2006.

[37] Nguyen Anh Quynh and Yoshiyasui Takefuji. A novel approach for a file-system integrity monitor tool of xen virtual machine. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security*, ASIACCS '07, pages 194–202, New York, NY, USA, 2007. ACM.

[38] Bryan D. Payne, Ron Perez, Reiner Sailer, Wenke Lee, and Ramon Caceres. A layered approach to simplified access control in virtualized systems. *Operating Systems Review*, 41, 2007.