Hindawi
Scientific Programming
Volume 2022, Article ID 2419291, 10 pages
https://doi.org/10.1155/2022/2419291

Review Article

Next-Generation Optimization Models and Algorithms in Cloud and Fog Computing Virtualization Security: The Present State and Future

Rohit Verma,[1] Dheeraj Rane,[2] Ravi Shankar Jha,[3] and Wubshet Ibrahim[4]

[1]*Department of CSE, Manipal Institute of Technology, MAHE, Manipal 576104, India*
[2]*Department of CSE, Indore Institute of Science and Technology, Indore 453331, India*
[3]*Department of CSE, Sharda University, Greater Noida 201310, India*
[4]*Department of Mathematics, Ambo University, Ambo, Ethiopia*

Correspondence should be addressed to Wubshet Ibrahim; wubshet.ibrahim@ambou.edu.et

Received 28 January 2022; Revised 19 April 2022; Accepted 26 May 2022; Published 30 July 2022

Academic Editor: Debo Cheng

Virtualization is becoming popular and is gaining widespread attention in recent years. However, with its popularity comes the challenge of securing the virtualized environment. Security measures for virtualized systems cannot always be applied in the same manner as in physical systems. Virtualized environments may have multiple virtual systems on the same physical machine, so different levels of security one needed. The hypervisor is the controlling program that provides the virtual systems' effective isolation and security. In this article, we present a comprehensive review of the existing security ideas and architectures for virtualized environment and some of the open issues in virtualization security.

1. Introduction

Since the 1960s, with the advent of the first virtual machine-enabled operating system for mainframes by IBM, virtualization has become very popular [1]. Currently, virtualization can be deployed on high end servers, normal personal computers, or even mobile devices such as mobile phones or handy tablets [2]. It divides the physical system into multiple virtual systems and gives the end user an illusion that he is working on the actual physical system. Nowadays, several information services and new technology paradigms, such as infinite computing (grid computing, cloud computing), web services, and other on-demand services, use virtualization as their core technology because of its economy and efficiency [3]. Most of the datacenters today rely on this technology for their functioning. Several big players in the market are introducing novel features and aspects to this technology on a continuous basis, such as VMware, Citrix, and KVM [4]. A report on Server Virtualization MCS 2010, by Kaspersky (server virtualization shipment forecast 2005 to 2014), stated that in 2010 more than half of the industry's installed workload was virtualized and in 2013 it is expected to surpass two-thirds of the installed workload [5]. Figure 1 show the tremendous increase in the number VMs (virtual machines) used in the industry for their different workloads (development/test/critical).

It is a fallacy to believe that virtualized settings are more secure than physical ones. Unfortunately, even though this idea has no basis in fact or logic, it might mislead some organizations into a false feeling of security when it comes to security requirements for virtualization initiatives. A virtual computer "looks" and performs exactly like any physical machine from the perspective of everything that interfaces or interacts with it. The hypervisor is usually the only thing that knows the machine is virtual. As a result, it is a basic fact that virtualized environments must still deal with all of the possible security issues that physical environments must deal with.

The virtualization provider operates, manages, and controls all components from the bare metal host operating system and hypervisor virtualization layer down to the

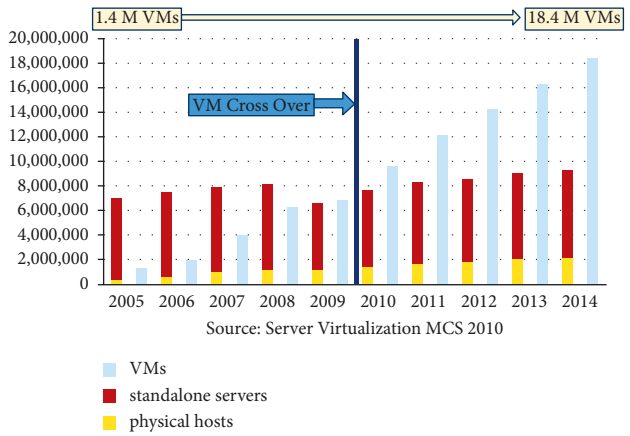

Figure 1: Increased number of VMs over the timeline.

physical security of the facilities where the services are provided under the shared responsibility paradigm [6]. It means that the virtualization provider oversees safeguarding the global infrastructure that underpins all of the services provided.

Physical security of datacenters with controlled, need-based access, location in nondescript facilities with 24/7 security guards, two-factor authentication, access logging and review, video surveillance, and disc degaussing and destruction are all things that your virtualization provider is responsible for. Provider's hardware infrastructure includes servers, storage devices, and other appliances. Operating systems, service applications, and virtualization software are all hosted on software infrastructure. Routers, switches, load balancers, firewalls, and cabling are examples of network infrastructure. Virtualization provider also keeps an eye on the network from the outside, safeguards access points, and offers redundant infrastructure with intrusion detection.

While the virtualization infrastructure is secured and maintained by provider, customers are responsible for security of everything they put in the cloud.

The customer is responsible for what is implemented by using provider services and for the applications that are connected to AWS. The security steps that customer must take depend on the services that they use and the complexity of the system.

Customer responsibilities include selecting and securing any instance operating systems, securing the applications that are launched on virtualization resources, security group configurations, firewall configurations, network configurations, and secure account management.

When customers use virtualization services, they maintain complete control over their content. Customers are responsible for managing critical content security requirements, including

(i) what content they choose to store on infrastructure

(ii) which provider services are used with the content

(iii) in what country that content is stored

(iv) the format and structure of that content and whether it is masked, anonymized, or encrypted

(v) who has access to that content and how those access rights are granted, managed, and revoked.

Customers retain control of what security they choose to implement to protect their own data, environment, applications, IAM configurations, and operating systems.

Along with its benefits and increased popularity, virtualization brings with it several security concerns. These are very critical and if not addressed properly the security of most datacenters and information services are at risk. Several security protection programs are emerging in research communities and markets, which emphasizes various aspects of virtualization.

In this paper, we organize these security architectures, threats, and solutions in a consolidated manner. Section 2 presents a brief overview of virtualization and virtual infrastructure. Section 3 discusses the vulnerabilities in a virtual infrastructure. Section 4 is a discussion on the various attack surfaces of virtualization. Sections 5 and 6 comprise brief discussion on the various attacks, mitigation, and other proposed architectures. Section 7 is an overview of future trends and various open issues in the virtualization security.

2. Background

In addition to the development and test workloads, organizations are now in the process of virtualizing even their most critical workloads. Also, virtualization is the core technology for many modern computing paradigms such as cloud computing [7, 8]. Commercial cloud providers use virtualization to provide and host different cloud services such as IaaS (Infrastructure as a Service). This allows users to benefit from virtualization in the form of reduced costs, easy availability, disaster recovery, and greater agility. They however need to realize that there are security risks that come along with these, such as easy creation/deletion/modification of VM and the fact that there is an entirely new infrastructure layer that needs to be secured. Attack on the virtual infrastructure can result in damage to the business drastically. The focus of this paper is to survey the current security vulnerabilities of virtualization.

2.1. Virtualization. Virtualization is the term commonly used to define abstraction of the underlying physical resources with logical objects [9]. This enables running multiple logical servers on a single physical server. Use of virtualization saves physical space, reduces power consumption, and saves network and storage cabling. Apart from this, virtualization also helps in providing hardware independence to operating systems and applications, dynamic provisioning to logical objects (operating systems or applications), better business continuity, better utilization of the physical resources, increased efficiency and responsiveness, better platform for legacy applications, and better isolation from other logical objects. All this is provided by a layer which lies in between the physical entity and the logical object known as the Virtual Machine Monitor (VMM). This layer decouples the physical resource from the multiple

logical resources. Mainly three approaches are used to implement virtualization:

(i) *Full Virtualization.* In this approach, the logical object (may be VM) need not be modified to run over the VMM. Also, the VM or guest OS is not aware of the virtualization, as the whole physical system (BIOS, memory, storage, processor, etc.) is emulated to logical objects.

(ii) *Paravirtualization.* In this, the guest OS is aware of the presence of virtualization, as it is modified before being used in the virtual environment. This results in better performance of the guest OS than that in full virtualization.

(iii) *Hardware-Assisted Virtualization.* In this approach, virtualization is enabled at the hardware level. Intel's VT-x and AMD's SVM technology are examples of this. This runs the VMM at the higher privilege mode or ring than that of the guest OS and the VMM is responsible for all resource allocation and memory management using extended/nested page table.

In Figure 2, different types of virtualization techniques are shown. These are represented based on protection ring or hierarchical protection domains and privileges of different parts of virtualization.

2.2. Virtual Machine Monitor. Virtual Machine Monitor or VMM is the software layer used to provide a virtual environment [2]. This control program is responsible for monitoring and managing the virtual machines. VMM keeps track of the happenings in the VM such as resource allocation, device redirection, and policy enforcement. There are two types of VMM:

(i) Type 1: this VMM runs directly on the bare metal or hardware. This does not require any hosting OS. Type 1 VMM is popularly called hypervisor [10]. This type of VMM itself runs as an OS and then spawns the virtual machine after booting. Citrix XenServer, KVM, VMware ESX/ESXi, and Microsoft Hyper-V are examples of modern hypervisors. Bare metal architecture in Figure 2 uses type 1 VMM.

(ii) Type 2: this VMM runs on top of the hosting operating system and then spawns the virtual machines. It relies on the hosting OS for device support and physical resource management. This VMM runs as software inside the hosting OS. VMware Workstation and VirtualBox are example of this type of VMM. Hosted architecture shown in Figure 2 uses type 2 VMM.

2.3. Virtual Machine. The concept of virtual machine came into picture when IBM Corporation in the 1960s first created logical instances of the large mainframe computers for concurrent access. This logical instance or virtual machine gave an environment of the actual physical machine. Virtual machines are the OS installed in the virtual environment onto the VMM. VMs are the set of files which are used by the hypervisor/VMM for giving the end user an illusion of the physical machine. These VMs are easy to move or copy or manage. VMs provide isolation as the root of a guest virtual machine cannot access the host OS or host hardware or other VM.

3. Vulnerabilities

Virtualization improves resource utilization, eases the management of VM, and provides isolation and greater agility. At the same time, it adds vulnerabilities to the infrastructure. In virtualization, multiple VMs reside on the same physical host.

This may lead to compromising the VMs if any of the VMs or hypervisor or physical infrastructure is compromised, thus putting virtualization at higher risk of attack. Intra-VM communication is also one of the vulnerabilities, as the communication is through the hypervisor and does not need to go through the external network security solutions. Dynamicity of the infrastructure also adds to the vulnerability as the static security policy enforcement is uncertain over dynamic provisioning, decommission, and migration. "VM escape" is also one of the vulnerabilities; if the attacked VM can bypass the hypervisor and access the underlying host directly, it can attack at the hardware level in several ways such as DoS (Denial of Service) as it will get the root privileges. One of the vulnerabilities is "VM capturing," if an attacked VM can access the other VM and attack through it such as DDoS (Distributed DoS). Modifications to the hypervisor are also one of the vulnerabilities worth securing.

4. Attack Surface

Despite providing several benefits to the industry, the virtualization infrastructure also exposes a larger attack surface to the attacker, which is shown in Figure 3. The following are the major attack surfaces:

(i) Hypervisor

(ii) Virtual machine

(iii) Host machine

(iv) Management console.

As the hypervisor is the control point of the virtualization ecosystem, it has direct access to the underlying physical hardware and hosted virtual machines. It typically has full access to the environment which enables it to violate the security policies, privileges, and aggregation of duties. This makes it a crucial attack surface. According to IBM X-Force's Trend and Risk report [10], hypervisor is the largest attack surface. Also, the hypervisor has a very large number of lines of codes, which somewhat make it more vulnerable to attacks. Attack on the hypervisor, be it on the single point of failure gives the attacker root level (highest privilege) access to the hardware. This attack is analogous to the "man-in-the-middle" attack as it gives the attacker place

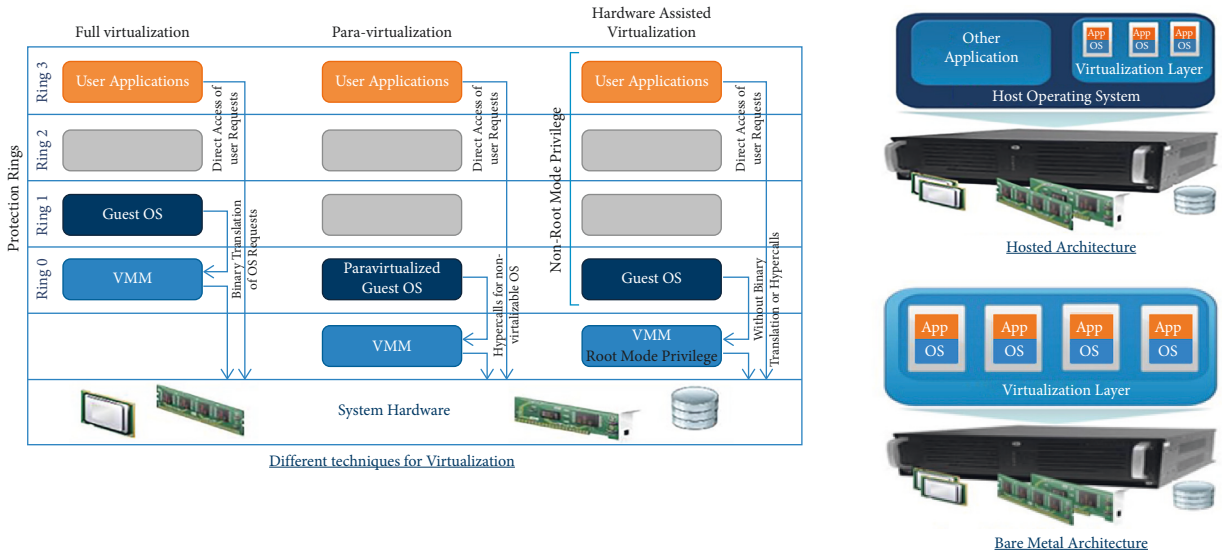

Figure 2: Different techniques and architectures in virtualization.

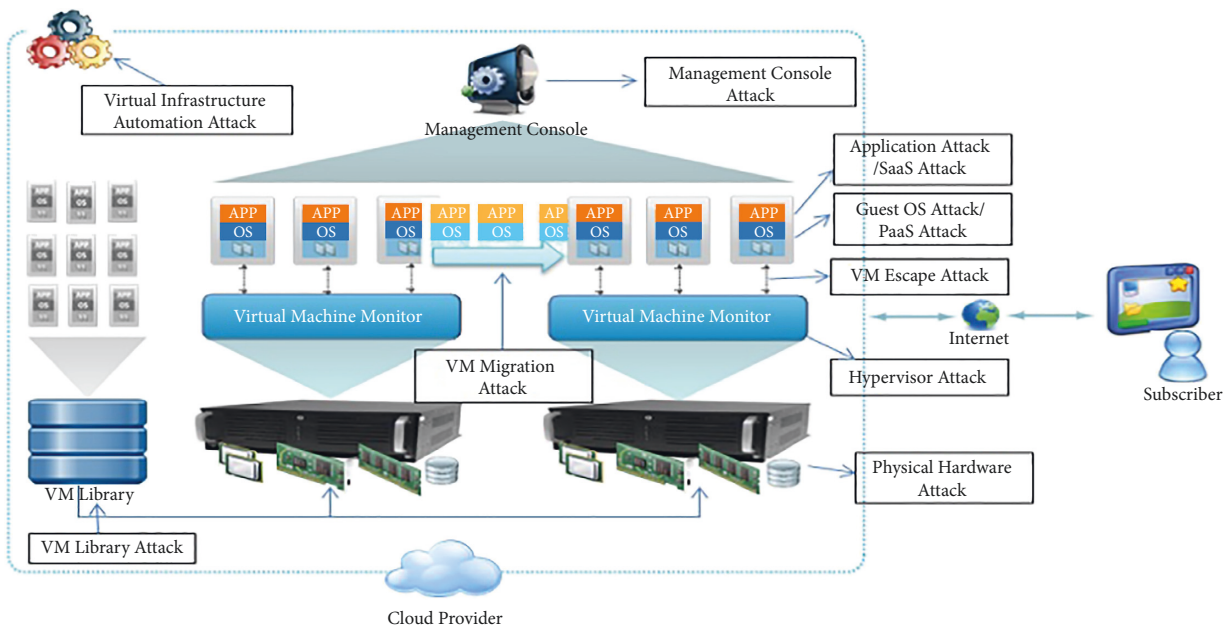

Figure 3: Different attack surfaces in virtual environment.

between the VMs and/or the hardware. This enables it to eavesdrop or modify the communication or system calls or register's/memory's value between VMs and/or hardware as the attacker. Attack on the hypervisor will also enable the attacker to attack actively by crashing the hypervisor or shutting down the VM. It may attack passively by modifying/controlling the VM or enabling the communication channel between the VMs such that they need not go through the hypervisor or also by changing resource allocation and usage.

Attack on the VM involves attack on the installed guest OS. The attacker may take advantage of the well-known vulnerabilities of the operating system and exploit them in a virtualized environment. Also, the unique configuration of each OS will have security concerns which if not addressed strictly may act as a channel for the attack. Even the state restoring capability of the VMs can be exploited for an attack; the VM can be restored to the state at which it was compromised, which will nullify the effect of the applied security patches. This would also enable the attacker to perform VM to VM attack, which may turn to DDOS attacks or use them as the attack launcher.

The host machine provides an attack surface mainly to the attacker that has direct access to the host, for example, through SSH, Rlogin, or by having access to the network of the host machine. This enables the attacker to launch DOS or to install the rouge hypervisor or change the network flow. The management console also provides a significant attack

surface if it is not hardened properly. The management console can be accessed through SSH or web interface or custom console, which once compromised can control the hypervisor or the VM or the underlying hardware. Other than this, there is need to secure Guest operating system (Virtual Machines (VMs)).

All physical resources are controlled by the VMM, and it may create multiple logical objects to serve each VM. These logical objects dynamically bound to physical resources are several other attack surfaces which need to be hardened for a secure virtual infrastructure, such as attack on the VM Library which can put the security of the VM image or the whole repository at risk and attack on the VM in transit (during VM migration or VM deployment from the VM library) which also gives way to the man-in-the-middle attack. The VM in transit attack may be passive, that is, by only sniffing the contents of the VMs or creating an illegitimate copy of the VM. Attack may be active, that is, by modifying the VM state or contents or causing DOS attacks. An attack surface may have poorly written automation APIs (Application Program Interface). Virtualization vendor offers customizable APIs for better management, automation, and customization of the virtual infrastructure according to the customer's need. The APIs, if poorly written, can prove to be a significant avenue of attack. Besides the mentioned attack surfaces, there are also ones related to the vulnerabilities of the kernel used in the hypervisor.

5. Threat and Mitigation

With increasing reliance on virtualization and the hypervisor, cost is reduced, and management is getting easier; however, the threats are evolving. Most of the known threats are categorized as "hypothetical" rather than real. Hypothetical threats are those that are realized in the lab considering the worst-case scenario, while real threats are operational threats in practical scenarios [11]. A few of the operational threats or real threats are VM sprawl [12], lack of visibility [13], separation of duties (of users or devices or applications) at the virtualization layer, and too many rights (direct or indirect control of the whole infrastructure is given at different levels). Operational threats are more challenging and should be mitigated carefully. However, we cannot neglect hypothetical threats as they also have got very strong proof of concept. Here, we categorize different threats (including hypothetical as well as real) based on core principles of security.

5.1. Confidentiality. Confidentiality is said to be violated if information is disclosed to an unauthorized system or person. In a virtual infrastructure, if the communication between VMs or between VMs and the host is intercepted or modified by an unauthorized system, the confidentiality is said to have been compromised. Multiple VMs can be hosted on a single physical hardware, which may serve a different purpose, namely, a web server, a DNS server, or an FTP server. Such VMs may need to communicate with each other and/or the host. This communication may involve some common shared memory area or the system calls (hypercalls otherwise). This communication channel may also provide a gate to the attacker for "man-in-the-middle" attack. The attacker may modify the common memory area or may intercept and masquerade the system calls (hypercalls). In turn, this can result in malicious behavior of the VMs. The communication between the VM and the underlying host may involve hypercalls, which run in a higher privilege mode than the VM. Modification to these hypercalls can result in the control of the hardware and can lead to other threats.

This threat can be mitigated by maintaining isolation wherever possible. Further mitigation can be ensured by allowing all communication to happen only through the hypervisors and managing efficient MAC—Mandatory Access Control—policies [14] at the hypervisor such as Biba, Bell-LaPadula, Caernarvon, Type Enforcement, and Chinese Wall policies (as used in IBM's sHype [7]). This also controls resource sharing (e.g., virtual resource—shared memory, event channel, local resource—vLANs, vDisks). This in turn will monitor and minimize suspicious communication. Further, security of the communication is ensured by using HTTPS, TLS, SSH, or encrypted VPNs [15]. Other than the above mitigation strategies, one more way may be by conducting all communication through the physical networking devices which are already well hardened using various security policies rather than the virtual network and vSwitch. This is however done at the cost of compromising performance and increased network traffic.

Another well-known threat to confidentiality is the virtual machine-based rootkit (VMBR) [16] which is popularly known as the "Blue Pill" attack. Blue Pill attack is an advanced form of VMBR that installs a VM underneath an existing operating system and hoists the original operating system into the virtual machine. In Blue Pill attack, the running VM is intercepted or monitored by running it under the thin hypervisor which is malicious and remains undetected by the VM. This malicious hypervisor runs in the lower layer and can control and monitor the higher layers. The lower layers comprise the inner ring of the hierarchical protection rings, which run at the highest privileged level (kernel level) and can control the hardware. The malicious thin hypervisor can intercept any of the system calls of the VM. At the same time, the OS of the VM can reference all of its existing system calls, files, or devices and is unable to detect the presence of any of the rootkit. This malicious hypervisor can see all the states and events in the VM, such as keystrokes, network packets, disk states, and memory states [16]. Hyperjacking is also one of the attacks, which involves installing a malicious hypervisor underneath and taking full control of the server based on the Blue Pill or VMBR [17].

VMBR can be mitigated only after it has been detected. VMBR can be detected by using detector software that run below the VMBR that can view the system (physical memory or disk) and look for the signatures of the VMBR [16]. Other ways to control the VMBR are secure boot, use of secure hardware, use of secure VMM [5], and observing overhead caused due to VMBR at the native system resources, such as system clock, paging activity, and virtual I/O device

behavior. Red Pill [18] is also able to detect the rootkit. Red Pill detects the presence of a malicious hypervisor by executing the nonprivileged instructions like sidt, at the lower privilege level. GuardHype [17] also provides mitigation from hyperjacking (hijacking of hypervisor).

VM fingerprinting is also a threat worth considering; an adversary may know about the VM by analyzing different registers values, memory dumps, etc. Also, details on older VMs may be recovered if the allocated memory to the deleted VM is not cleared properly, using memory recovery operations.

5.2. Integrity. Integrity is said to be violated in the context of virtualization if the modification has been done to the virtual infrastructure in a manner that is undetectable. In a virtualized environment, if the code of the hypervisor is modified or else a malicious kernel module is installed in the hypervisor, the integrity of the hypervisor is compromised. The idea behind this attack is to increase the complexity and size of the OS kernel, which also gives way to the higher security vulnerabilities. The main component of the hypervisor is its kernel, and it installs the other kernel modules to enable virtualization. This makes the system vulnerable as the kernel runs at a high privilege mode and it can control the whole system, thus compromising the kernel will bring the whole system's security at stack. Also, the compromised hypervisor's kernel can give up control of the hypervisor, which can provide a means to launch other attacks such as VM manipulation (start/stop, allocated resources) and zombie attack (controlling a VM for further attacks). This category of attacks is also known as external modification to the hypervisor [3].

This threat can be mitigated by only allowing user approved code in the kernel privilege. This can be done by checking all the code against the supplied user policy. This also ensures that once approved code cannot be further modified [15].

Other attacks related to exploiting zero day or other well-known vulnerability in the kernels are injecting malicious code and performing kernel buffer flow. Another kind of attack can be done by controlling the peripheral DMA and corrupting the kernel memory by frequent DMA writes or by manipulating IOMMU's translation. Such attacks are mitigated or minimized by making use of hardware memory protection schemes and AMD's Secure Virtual Machine (SVM) [15].

5.3. Availability. For any system availability is an important property. Availability is necessary to meet the requirements in SLAs and for ensuring continuity of business relationship. Any type of DoS (Denial of Service) or DDoS (Distributed Denial of Service) attack is a threat to availability. Such attacks are the result of vulnerabilities in the system. If the attacker gets access to the VM or hypervisor, it may make the service currently running on the VM unavailable either by stopping the VM or crashing the hypervisor or deleting the required files of the VM. Also, the external modification of the VM can be a threat to its availability. These types of attack are critical as the unavailability of the system can cause damage to the financial, business, and social reputation of the industry.

Other types of attack could be improper configuration of the hypervisor. An improper or careless configuration of the hypervisor provides one VM to capture all the physical resources. Suppose a compromised VM consumes all the processing power or networking resources and makes other VMs to starve. In such a situation, legitimate VMs will not have sufficient hardware resource to carry out their tasks. This will lead to DOS for users of those VMs. These types of attack can be mitigated by proper security policies at hypervisor level.

5.4. Authorization. Authorization is said to have been compromised if the system is able to perform a task that it is not allowed to. In the context of virtualization, threat to authorization is huge. If an attacker escalates its privileges and performs a task that it is not authorized, authorization is said to have been compromised. This is the most common form of attack in virtualized environment.

"VM escape" is one of the major attacks seen in a virtualized environment. In this attack, the VM can completely bypass the hypervisor and gain direct access to the hardware. When a program running on the VM gains the root privilege of the hardware, it may misuse the root access for active attacks like external modification of VM/hypervisor, or passive attacks such as VM monitoring from the host. Even the data cache is susceptible to being monitored and modified. This attack can lead to complete collapse of the security framework [3].

Other attacks along with this principle include obtaining access to the root in the management console and performing unauthorized tasks, such as VM modification, monitoring VM externally. These can be mitigated by properly managing different management consoles—local and web access.

In general, to mitigate various attacks and threats, focus should be towards hardening the infrastructure, proper configuration, timely patching, and change management. Various external tools and technologies may be adopted like virtualization-aware firewalls/IDS/IPS/antimalware, external vSwitch, and Virtual Encryption.

6. Other Secure Architectures

Several breakthroughs and models have been suggested for securing virtualization environments [19]. We did a comparison of various security models with security applied at the hypervisor level. Security implemented at the hypervisor level may reduce the resources required. These models comprise varied parameters of security and consider various threat models, as shown in Figure 4. Table 1 is a tabulation of these models.

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}. \quad (1)$$

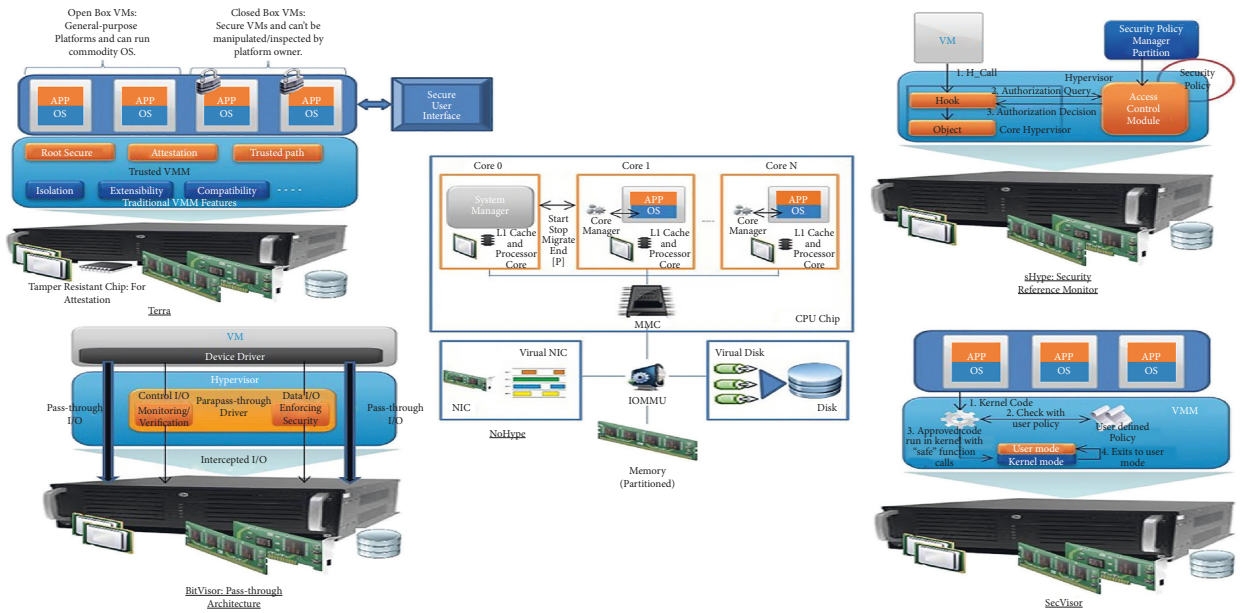

Figure 4: Comparison between different security models for virtualization [11, 21–23].

Table 1: Comparison between different security models for virtualization.

	SecVisor [15, 20]	sHype [7]	NoHype [21]	BitVisor [11, 22]	Terra [2]
Basis	Kernel occupies a privileged position in software stack and compromising it will give the attacker complete control of system.	Malicious OS can compromise the security of other OSes. Thus, secure cooperation of OS is needed at hypervisor layer.	Virtualization layer gives large attack surface to adversary.	Eliminating device drivers and device models from hypervisor may minimize hypervisor size.	Need for diverse security requirements in government, consumer, and enterprise application.
Concept	Ensure only user approved code should execute with kernel privilege.	Implementation of security reference monitor interface in hypervisor to enforce information flow constraints between VMs.	Elimination of hypervisor layer and giving virtualization at the hardware level directly.	Minimizing the hypervisor size by using the device drivers of guest OS to handle devices and I/O.	Flexible architecture for trusted computing with variable security requirements.
Threat model	(i) Misuse of modularization support from the kernels in the form of malicious code injection. (ii) Exploitation of software vulnerabilities in the kernel code. (iii) Malicious devices may corrupt kernel memory through DMA writes.	(i) Application set with conflicting security requirement may compromise other's security. (ii) Security above OS level can be bypassed by many threats, e.g., trap doors, malicious developers, and boot-sector viruses. (iii) Uncontrolled information flow between VMs.	(i) Large KLOC of hypervisor may have more vulnerability. (ii) Customer may run any software on the VM without any restriction. (iii) Compromised hypervisor may disturb the functionality of the whole infrastructure.	(i) Security of Virtual Machine Monitors is crucial if security is enforced at VMM. (ii) Vulnerabilities of hypervisor with large code may compromise security of the whole system.	(i) OS are complex programs with low assurance, to provide trusted computing base. (ii) Poor isolation of different applications may cause compromising the entire platform. (iii) Absence of trusted path between user and application.

Table 1: Continued.

	SecVisor [15, 20]	sHype [7]	NoHype [21]	BitVisor [11, 22]	Terra [2]
Security benefits	(i) Reduced code size. (ii) Reduced attack surface at kernel interface. (iii) Customized security policy of user.	(i) Strong isolation. (ii) Better access control. (iii) Boot-time and run-time guarantee.	(i) Confidentiality. (ii) Availability. (iii) Integrity. (iv) Reduced side channel attacks.	(i) Reduced code size of hypervisor. (ii) Improved reliability of hypervisor. (iii) Better I/O device security.	(i) Secure isolation. (ii) Better privacy, confidentiality, and integrity. (iii) Better application security assurance.
Assumptions in model	(i) Hardware has the virtualization support on which model is running. (ii) User and kernel share address spaces. (iii) Kernel does not make BIOS calls after initialization.	sHype was for single hypervisor infrastructure and all the communication was through virtual network not real network.	Hardware has virtualization capability; as vendor of network switches, multicore memory controller, IOMMU, and processors provide virtualization support.	Hardware has virtualization support and also platforms are equipped with IOMMU. Disk image of hypervisor cannot be modified externally. Firmware and BIOS are trusted entities.	(i) Attestation relies on security of standard SSL session key exchange protocol. (ii) Hardware platform with tamper-resistance, virtualization-enabled technology.
Brief design	SecVisor design based on mainly two principles: (i) CPU only executes the approved code in the kernel mode. (ii) Approved code should only be modified by SecVisor and its TCB (trusted computing base). For these, SecVisor used hardware memory protections and properly managing all kernel mode entries and exits.	sHype implemented different policies and modules; for example, for isolation, ring-based security is implemented. For access control enforcement, security hooks are inserted into code path inside the hypervisor, to guard the access to the virtual resources. Access control module (ACM) is implemented for policy management, making policy decision and other security decisions.	Implement virtualization at the hardware level with features: (i) one VM per core. (ii) Hardware enforced memory partitioning. (iii) Dedicated virtual IO devices to each VM. This design ensures that the working of one VM does not interrupt other VMs and isolate VMs from one another.	BitVisor implemented "parapass-through architecture," in which most of the access to the hardware from guest device driver is passed through the hypervisor. Part of I/O accesses are intercepted to (i) protect hypervisor from guest OS and (ii) enforce security policy.	Terra design is based on using a traditional VMM, to allow properties like isolation, extensibility, efficiency, and compatibility. Along with this, some more additional capabilities are included for making trusted VMM: (i) root secure, (ii) attestation (using tamper resistant hardware), and (iii) trusted path (using secure user interface).
Limitations	(i) Only provide integrity of code but not the integrity of control flow. (ii) Only single CPU support. (iii) No measures for self-modifying codes.	(i) Information leakage through covert channel. (ii) Lack of scalability: only for single hypervisor. (iii) Security of VM in transit.	(i) Lack of scalability. (ii) Underutilization. (iii) Performance may degrade. (iv) Security of VM in transit.	(i) Information leakage through covert channels. (ii) Hypervisor has limited functionalities. (iii) No support for USB and ethernet devices. (iv) Security of VM in transit.	(i) No measures for untrusted device drivers. (ii) Information leakage through covert channel. (iii) Security of VM in transit.

7. Future Trends and Concerns

From the time of IBM's first virtual machine-enabled mainframe until today, virtualization is evolving continuously, unfortunately though, so are attacks on virtual environments. Several sophisticated attacks are still at large and need to be properly understood and researched. An example of these attacks is "VM in transit," VMs when brought from VM library to the VMM or VM migration from one VMM to another VMM [24]. If the network for VM migration is compromised or, in future, when the customer has the flexibility to move his VMs between other machines via any network, attacks on VM in transit will become more sophisticated.

One other concern is related to the popularity of mobile phone virtualization. Though mobile virtualization enables the user to have two different OSs on the same device, on one OS personal data may reside, while on the other corporate data, with efficient separation by virtualization. However, this road to mobile virtualization is not smooth, because mobile platforms have several limitations as compared to servers and desktops, in terms of available resources, real-

time computation, power limitations, and dependence on other technologies. Although mobile virtualization is a good way to provide security at the enterprise level (in form of BYOD—Bring Your Own Device—security), mobile virtualization in itself is prone to many challenges, such as limitation of computing resources, variable connectivity to the network, performance overhead by the virtualization layer, lost device issues, ruggedized devices, data ownership conflicts, rapid changes in mobile industries, and foremost jailbreaking, in which warranty from the vendor may become void owing to the installation of third-party software.

VMception or "virtualization inside virtualization" is another issue that needs to be addressed. We noted that most available measures cater to physical machines, meaning that models and measures require physical hardware for deployment. However, the attacks in nested virtualized environments need a different approach. The hardware would be virtual hardware and that too is hosted on a virtual machine, which will make traditional mechanisms inefficient.

More concrete security guidelines are needed for nested virtualization and mobile virtualization, which however would result in a trade-off between security and performance.

8. Conclusion

In this article, we have articulated various security related ideas and architectures in context of virtualization. In true sense, it is a virtualization-aware security implemented over the virtualized framework. Existing techniques that have been numbered in the article prove to secure the core of various information services. This is however not sufficient for ensuring security holistically. Computing paradigms such as cloud computing pose several other vulnerabilities and require addendum security measures. Also, security measures must be chosen to balance with the functional requirement and considerations of "Security vs Performance vs Economy."

Virtualization emerges as a powerful yet economic solution to reduce operational expenses in current computing paradigm. It easily becomes a threat to the environment if the configuration is not integrated with fine security. A full-proof virtualization security model to withstand probable attacks is the need of the hour. As quoted at several instances through the run of the paper, significantly monitoring new developments in this domain is important. A summarized state of the art and projecting newer strategies are the scope of the work presented in the paper.

Data Availability

No data were used to support the findings of the study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] P. M. Chen and B. D. Noble, "When virtual is better than real [operating system relocation to virtual machines]," in *Proceedings of the 18th workshop on hot topics in operating systems*, pp. 133–138, IEEE, Bertinoro, Italy, May 2001.

[2] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and L. Anthony, "kvm: the Linux virtual machine monitor," *Proceedings of the Linux symposium*, vol. 1, no. 8, pp. 225–230, 2007.

[3] L. Wang, G. von Laszewski, A. Younge et al., "Cloud computing: a perspective study," *New Generation Computing*, vol. 28, no. 2, pp. 137–146, 2010.

[4] J. P. Walters, A. J. Younge, I. K. Dong et al., "GPU pass-through performance: a comparison of KVM, Xen, VMWare ESXi, and LXC for CUDA and OpenCL applications," in *Proceedings of the 2014 IEEE 7th international conference on cloud computing*, pp. 636–643, IEEE, Anchorage, AK, USA, June 2014.

[5] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, and D. Boneh, "Terra," *ACM SIGOPS-Operating Systems Review*, vol. 37, no. 5, pp. 193–206, 2003.

[6] E. Keller, J. Szefer, J. Rexford, and R. B. Lee, "NoHype," *ACM SIGARCH-Computer Architecture News*, vol. 38, no. 3, pp. 350–361, 2010.

[7] C. Liu, C.-L. Sia, and K.-K. Wei, "Adopting organizational virtualization in B2B firms: an empirical study in Singapore," *Information & Management*, vol. 45, no. 7, pp. 429–437, 2008.

[8] K. Adams and O. Agesen, "A comparison of software and hardware techniques for x86 virtualization," *ACM SIGOPS-Operating Systems Review*, vol. 40, no. 5, pp. 2–13, 2006.

[9] S. Chiueh, T.-C. Nanda, and S. Brook, "A survey on virtualization technologies," *Rpe Report*, vol. 142, 2005.

[10] IBM, "IBM X-force 2010 trend and risk report," 2011, http://public.dhe.ibm.com/common/ssi/ecm/en/wgl03007usen/WGL03007USEN.PDF.

[11] A. Seshadri, M. Luk, N. Qu, and A. Perrig, "SecVisor," *ACM SIGOPS-Operating Systems Review*, vol. 41, no. 6, pp. 335–350, 2007.

[12] M. Lindner, F. McDonald, B. McLarnon, and P. Robinson, "Towards automated business-driven indication and mitigation of VM sprawl in Cloud supply chains," in *Proceedings of the 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, pp. 1062–1065, IEEE, Dublin, Ireland, May 2011.

[13] S. J. Vaughan-Nichols, "Virtualization sparks security concerns," *Computer*, vol. 41, no. 8, pp. 13–15, 2008.

[14] R. Sailer, T. Jaeger, E. Valdez et al., "Building a MAC-based security architecture for the Xen open-source hypervisor," in *Proceedings of the IEEE Computer Security Applications Conference 21st Annual*, p. 10, Washington, DC, USA, December 2005.

[15] J. Kirch, "Virtual machine security guidelines" the center for Internet Security," 2007, http://www.cisecurity.org/tools2/vm/CIS_VM_Benchmark_v1.0.pdf.

[16] S. T. King and P. M. Chen, "SubVirt: implementing malware with virtual machines," in *Proceedings of the IEEE Security and Privacy Symposium*, p. 14, Berkeley/Oakland, CA, May 2006.

[17] M. Carbone, D. Zamboni, and W. Lee, "Taming virtualization," *IEEE Security and Privacy Magazine*, vol. 6, no. 1, pp. 65–67, 2008.

[18] Y. J. Rutkowska, "Red Pill... or how to detect VMM using (almost) one CPU instruction," 2005, http://invisiblethings.org/papers/redpill.html.

[19] S. Jin, J. Ahn, S. Cha, and J. Huh, "Architectural support for secure virtualization under a vulnerable hypervisor," in *Proceedings of the 44th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 272–283, Porto Alegre Brazil, December 2011.

[20] K. Lab, "Rethinking security for virtual environments," 2012, http://media.kaspersky.com/documents/business/brfwn/en/IDC-technology-spotlight_Kaspersky-Security-for-Virtualization-white-paper.pdf.

[21] R. Sailer, E. Valdez, T. Jaeger et al., "Secure hypervisor approach to trusted virtualized systems," *Techn. Rep.*, vol. RC23511, 2005.

[22] T. Shinagawa, H. Eiraku, K. Tanimoto et al., "BitVisor: a thin hypervisor for enforcing i/o device security," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pp. 121–130, Washington, DC, USA, March 2009.

[23] R. Perez, L. Van Doorn, and R. Sailer, "Virtualization and hardware-based security," *IEEE Security and Privacy Magazine*, vol. 6, no. 5, pp. 24–31, 2008.

[24] M. Christodorescu, R. Sailer, L. S. Douglas, D. Sgandurra, and D. Zamboni, "Cloud security is not (just) virtualization security: a short paper," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 97–102, Chicago, Illinois, USA, November 2009.