THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Research on Virtualization In Cloud Computing For Security Threats and Solutions

Parminder Singh ,

Department of Computer Science and Engineering
Lovely Professional University, Phagwara, India
email: parminder.16479@lpu.co.in

ORCID: 0000-0002-0750-6309

Avinash Kaur

Department of Computer Science and Engineering
Lovely Professional University, Phagwara, India
email: avinash.14557@lpu.co.in

ORCID: 0000-0003-3534-4121

Abstract: Virtualization enables the heterogeneous architecture deployment on the same hardware with an abstraction of software and hardware resources. In virtualization, the essential components are hypervisors, virtual machines, and virtual network. Virtualization-based application is increased in a great amount from the past decade. It is a key to enable cloud computing for the illusion of unlimited resources. However, the vulnerabilities in this technology affect the confidentiality, availability, and integrity of the cloud services and resources. National Vulnerability Database (NVD) recorded many vulnerabilities related to VMware, Qemu and Xen. Kaspersky Lab described that the cost to overcome the attack become double if it affects the virtual infrastructure. Hypervisor attack is possible due to vulnerabilities like misconfiguration, poor design, week design, bug, week privileged, and so on. A potential target is a virtual machine to gain unauthorized access. VM can be in running, inactive or migration state, so the vulnerabilities are different in each state. Another security challenge in network virtualization is due to its dynamic requirement of reconfiguration in certain situations. It becomes complex when VM migration happened where the IP needs to be changed, in this situation the network infrastructure sharing becomes ill-protected for ARP and DHCP and IP protocols. These concerns add new challenges for traditional security solutions due to the

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

dynamic and migration nature. In this study, the analysis of security concerns and solutions are conducted. The identification of the research gap helps the research community to develop a secure platform for virtualization cloud computing.

1. Introduction

The abstraction of software and hardware resources is called virtualization which makes possible to run heterogeneous architectures on the same hardware [1]. Virtual networks, hypervisors and virtual machines (VMs) are the main components of virtualization. VM is a physical system-like software system that allows applications to run on operating system. Hypervisor offers a "physical resource abstraction" such as memory, CPU, Storage and network. and storage. Multiple operating system can run on single physical server concurrently. The installation can be done either by Type I or Type II operating system of the host. The network on virtualization allows virtual switch (vSwitch) communication between VMs.

2. Survey on Virtualization Security Threats

Virtualization-based apps have grown tremendously over the previous decade. The present virtualization implementations, however, present many vulnerabilities and challenges to security. Several vulnerabilities associated with famous hypervisors such as Qemu, VMware, XEN registered in the National Vulnerability Database (NVD). Successful exploitation of these vulnerabilities leads to many assaults that may influence the hypervisor's or its underlying VMs ' confidentiality, integrity, or accessibility (CIA). For example, the open-source Qemu hypervisor's critical VENOM vulnerability (CVE-2015-3456) makes the VM break, execute the code on host, and other VMs on host also accessed. Many virtualization products affected from VENOM vulnerability. The XenServer vulnerability of Foreshadow (CVE-2018-3646) enables to establish a side channel and information steal from physical server’s untrusted VMs. Kaspersky Lab has evaluated that virtual environment for the affect of attack, the price to recover from a safety incident twice. It is therefore necessary to secure the virtualization environment.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

This chapter discusses multiple vulnerabilities, safety threats, and virtualization-related assaults and classifies them according to the key elements of the infrastructure for virtualization. For each element, we investigate the current safety methods and methods and investigate the research gaps. The study gaps are considered to formulate the possible safety directions, this will assists the authors to develop the new trusted platforms for virtualization.

The main terms of paper described in Table 1. Section 2 presented the threats, vulnerabilities and assaults. Section 3 analyzes current safety solutions and virtualization difficulties. Section 4 discusses the study gaps in current alternatives and offers guidelines for future studies to improve virtualization safety.

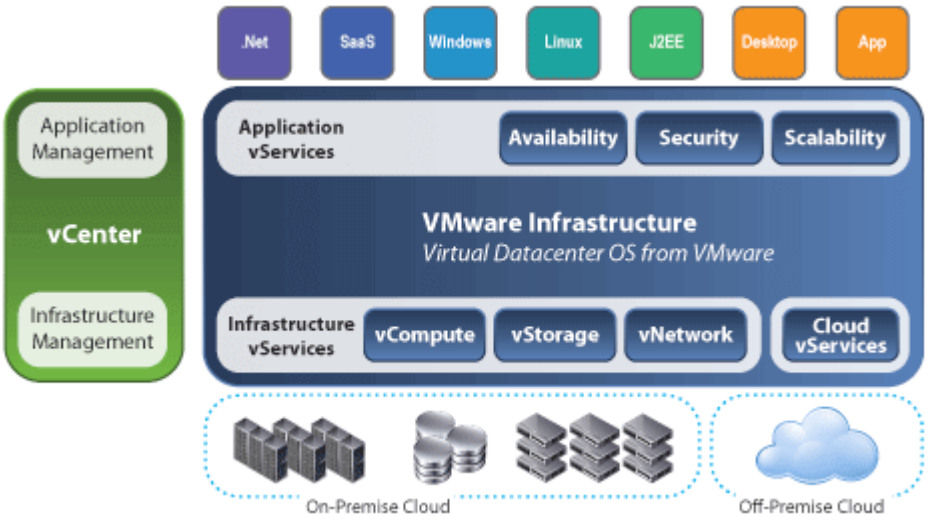

Figure 1: Components of Virtualization Environment

As shown in Figure 1, the virtualization environment consists of physical server, network, VM repository and management server. The physical server has five components: virtual network, VM, hypervisor, host OS (Type 1 hypervisor control VM), and underlying equipment. On the same hardware, a hypervisor operates various VMs. An end-user can access the VMs and applications through internet connection. Virtualization management is handled by the management servers.

It also store the repository of VM images and interacting with the management server, a VM proprietor send VM images for hypervisors. There are typically three different

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

networks in the network element: internal and external network, virtual network. VMs are linked to the virtual network via vSwitch. It is the responsibility of the internal network to connect management servers, physical servers and storage servers. The virtual environment connects with internal network with internal and external network.

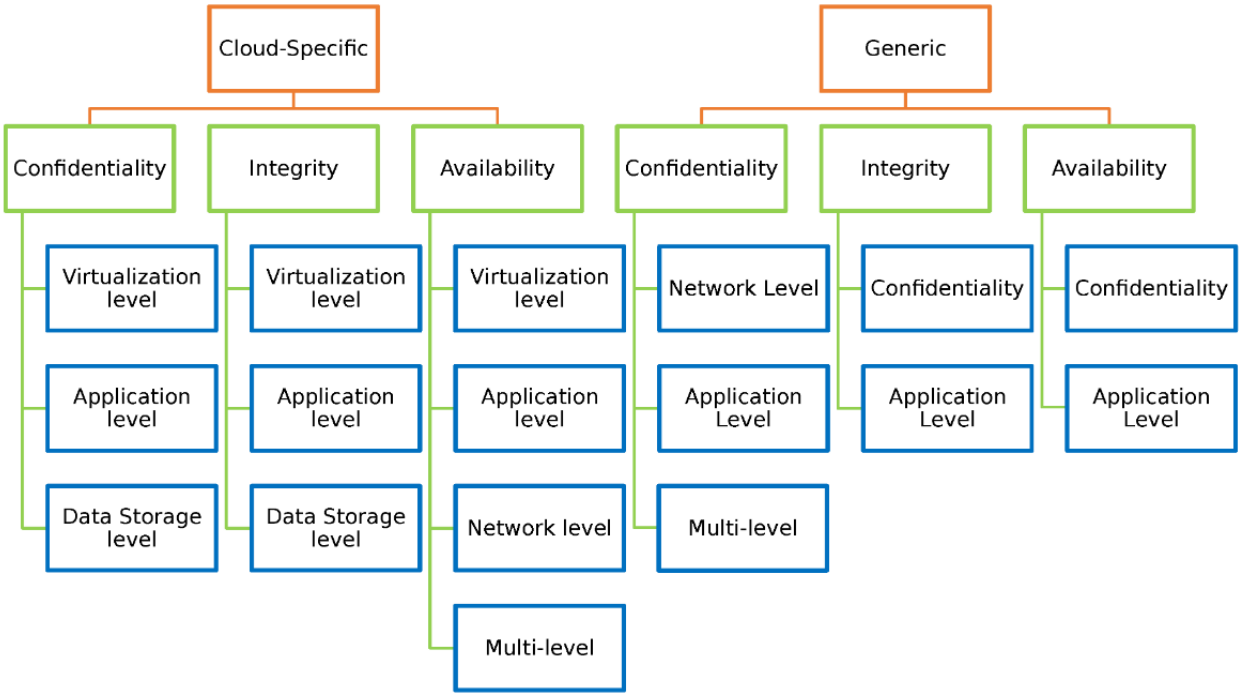

Figure 2: Taxonomy of Virtualization Safety

Vulnerabilities in various parts and activities in virtual environment generate the safety threats. Figure 2 demonstrates the virtualization safety taxonomy. The figure 1 and 2 is used for discussion of victims and source of the attacks.

Hypervisor makes virtualization possible, and it has various parts such as management the hypervisor Xen (Type 1) has a core hypervisor, interface/privileged domain (Dom0), Qemu, and VMs for guests. KVM (Type 2) hypervisor contact host's operating system, an extra KVM device. Hypervisor core regulates VM processes by offering abstraction of hardware. It conducts memory partitioning and CPU scheduling. Hypervisor's management interface holds all VM administration activities.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Type 1 hypervisor has control VM (Dom0), whereas it is a trusted process of the physical server's OS in a type 2 hypervisor. Dom0 is a prerogative VM with unique I/O resource access privileges. Xen supports two VM kinds: hardware VM (HVM) and paravirtualized VM (PVM) and equipment. Xen utilizes Qemu for clients running as HVMs, which operates in Dom0 to emulate devices. Qemu is regarded a big attack surface for Xen 3.3. It is therefore shifted outside of Dom0 and put for each guest in the stub domain. Unlike Xen, for device emulation, KVM depends completely on Qemu. It was noted that each of these parts can be targeted by an attacker. The underlying VMs can be controlled by a compromised hypervisor. Thus, gaining access from the hypervisor become a common attack in virtualization landscape.

2.1. Vulnerabilities: There are many vulnerabilities such as uncontrolled flexibility to generate VMs, bugs, bad design and misconfiguration, bugs, weak control of prerogative and administrative interfaces, out of control allotment of resources to VM, and other, where hypervisor-related attacks are feasible. Over the past 2 years (2016 to 2017) about 90 vulnerabilities have been recorded for Xen hypervisor. After the consideration of target and source, we have categorized Xen-related vulnerabilities as shown in Table 2. Table 3 shows the root cause of Xen vulnerabilities. There are 174 vulnerabilities similar to Qemu+KVM reported from 2016 to 2017.

We have categorized these vulnerabilities as objectives through attackers from remote and guest VMs shown in Table 4 and Table 5. Four types of vulnerabilities have been regarded here: Gain Privilege (GP), Denial of Service (DoS), Code Execution (CE) and Gain Information (GI).

2.2 Threats: Internal interface of hypervisor can report some vulnerabilities: management interface compromise, malware or rootkits insertion, unlawful access to resources of hypervisors, uncontrolled development of VMs, DOS attack through VM resources hunger, rouge Virtual Machine launch, network threats, and so on. The mentioned elements are interacted by a hypervisor: Insider/VM hypervisor management control/insider hypervisor code VMs, VM user and networks. The rootkits can infect a hypervisor code, which may infringe the hypervisor's integrity. Various interfaces such as

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

insider, remote access or drivers can be subjected to a control VM. A VM, remote attacker or malicious insider may jeopardize VM control. The lead control has given to the hypervisor's leadership functionality. The virtual machines are run by a hypervisor that can be launch on the Internet. A running VM may infringe the hypervisor's The confidentiality, availability and integrity can be infringe the hypervisors due to its running status. Hypervisor deploy a VM migration feature and the network's amalgamated hypervisor may adjust its method of consolidation. Hypervisor may be able to launch rouge VM.

2.3 Attacks: Hypervisor is the key element of virtualization, the related attacks have a severe effect on the safety of virtualization.

(1) VM-based rootkit hyperjacking (VMBR). It is intended for the violation of integration of hypervisor and started against type 2 hypervisors o. The hacker are trying to gain control of a hypervisor here. The hyperjacking can be done by injecting VMBR in OS and hardware. Slim rogue hyperviser and legitimate infrastructure can be execute. SubVirt [34], control the VM and host operating. In the virtualization allowed CPUs, Blue Pill [3] exploits hardware extensions and operates an harm servers into a VM.

(2) Committed leadership interface attacks / Malicious insider attacks. Qemu faults influence Xen safety KVM and Xen HVM mode. (CVE-2017-2615) and (CVE-2016-9603) can take the privileges from the privileged users. Recent research [4, 5] described that in the hypervisor TCB, the Dom0 can not entertained.

Thus, various assaults from the Dom0 are feasible for the hypervisor.

A. DomUs ' Compromising CIA. The VM supplier and hypervisor are always trusted by a VM consumer. If Dom0 is compromised, the DomUs ' CIA may be compromised.

B. Attack with Direct Memory Access (DMA). It is affects the integrity of hypervisors. Dom0 may directly able to communicate the resources through DMA transfer and main memory can be accessed. The data structure of hypervisors can write or read and root kit may be overwrite.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

C. Attack of VM sprawl. This is intended to violate the hypervisor's accessibility. A hypervisor can generate fresh VMs with flexibility. Such flexibility, however, carries a potential risk, such as sprawling of VM. A malicious insider can generate big numbers of VMs with administrative access to the management interface. Virtual machines are therefore hard in handle and monitor at a specific moment.

(3) The VM's attacks. Here we suppose that there is a secure VM picture initiated to the hypervisor. However, because of the insecure interface or malicious user at the running VM, the risk to the hypervisor remains. Author demonstrate most of the vulnerabilities of the hypervisor that the guest VM exploits.

D. Hypervisor code execution. The arbitrary code and heap memory can be changed by the malicious VM users.

E. Escape from VM (to Hypervisor). The pros of hypervisor and VMs isolation failure are used in it. The completely bypasses of rouge VM and hypervisor's function of access control and privilege gaining of hypervisor. Thus another VM and the hypervisor itself can be controlled by a rouge VM.

F. Crash of the hypervisor (DoS). The memory of virtual machine can affect the memory of hypervisor and trigger a crash of the hypervisor.

G. Leakage of data from hypervisor. A VM can access the hypervisor memory arbitrarily, resulting in hypervisor and other VM information leakage.

H. Hunger for the resource. This is intended to violate the hypervisor's availability. The uncontrolled allocation of resources give permission to physical server.

I. VM control attack. It can be attacked from the DomU as control VM resources. The control of hypervisor can be aimed.

J. VM hopping. It jumps from one VM to another VM. can lead to attacks on other VMs such as information leakage, traffic analysis, configuration file modification, service denial, and so on. Jasti et al. [6] have shown the VM get the access of maliciously hoping.

(4) Malicious hypervisors (networks) attacks. The migration of modules in the hypervisors implements the functionality of VM migration. Vulnerabilities such as stack overflow, buffer overflow. The migration modules exploited through the network in the

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

access control and authentication process to compromise a migration process's control plane and execute after assaults.

Migrating a vulnerable VM. A live migration is automatically initiated for dynamic load balancing. A malicious hypervisor attacker may misrepresent accessible resources and impact the trusted hypervisors.

(5) Start of rouge VM. A malware affects the trusted hypervisors through the malicious users. It presents the fresh source of an assault when the underlying infrastructure is guaranteed in the virtual environment.

3. Security Challenge to VM

The improper access to the hypervisor can be performed by VM. The three states of VM are: running, moving (moving to another host machine) and inactive. It is susceptible to various kinds of attacks in all these countries. The vulnerabilities related to VM are tracked by NVD such as "Xen 3.2.x through 4.4.x does not correctly clean guest-recovered memory pages, enabling local guest OS users to obtain sensitive data via unknown vectors (CVE-2014-4021)." "During the migration of virtual machines VNC are not maintained correctly byApache CloudStack before 4.5.2. Therefore, making an access to VNC passwords more easier. It causes service disruption and/or data leakage (CVE-2016-2084).

Running State security challenges for VM. The potential target during the execution phase is VM as its interfaces are easily exposed to attacker.

Vulnerabilities. In a operating state, VM involves vulnerabilities such as bad VM and hypervisor isolation, bad management device access control, uncontrolled VM rollback, fresh VMs default, bad shared resource isolation, network vulnerabilities, etc.

Threats. Possible threats include introduction of rootkits into a VM, network threats, unlawful access from malicious insider or the hypervisor management interface and also the isolation failure among different VMs. Model of threat for execution state of VM. The executing VM performs the interaction with various interfaces such as co-hosted VMs and networks,

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

management interface for hypervisors, hypervisor and kernel code for VMs. Different devices such as remote access interface, drivers and insider are subjected to command of VM. The control VM may be compromised by VM or remote attacker and a malicious insider ,.

Using the rootkit infected VM picture, a VM example can be developed that can affect a VM's kernel. A VM kernel can be compromised, which can manipulate a trusted application with execution. The Internet may be subjected to a VM co-hosted with other VMs. A co-hosted VM can carry out various assaults on various other VMs sharing the resources.

Operating condition attacks on a VM: Attacks by the hypervisor impaired. Type 1 and Type 2 hypervisor are considered here. We suppose that VM is in the hypervisor's TCB control in the event of type 1 hypervisor. Security risk lies with the VMs operating on a compromised hypervisor. The VMs ' CIA can be damaged here. Committed leadership interface attacks / malicious insider attacks. The Type 1 hypervisor, Trusted Computer Base (TCB) includes VM control along with remote access interface. Control VM uses other VMs (DomUs) to control access to physical assets or regulate all of DomUs. Therefore, we assume the trusted hypervisor; but, there are bugs in the software instantiating the Dom0 features. Furthermore, the remote access of the leadership interface is compromised by the web application assaults (SQL injection and CSS and SQL) for both the hypervisors. It exposes a distant attacker to the actions of the VM management. For example, vCenter Operations before 5.0.x, "CSS vulnerability (CVE-2012-5050) enables remote attackers to inject arbitrary web scripts via unknown vectors and take control of vCenter." Additionally, a malicious insider can directly access the management interface [7, 8]. A management interface's successful compromise can lead to subsequent assaults on VMs [9].

Violation of the privacy of VM: Because Dom0 have the capability to read access memory of executing VMs and also extract the sensitive data with snapshot of memory. A cryptography keys are malicious Dom0 enables us to get cryptography keys, confidential passwords, and files [8]. It can be done by cold boot attack by utilizes the target VM's memory snapshot. The permissive access of Dom0's to the memory of VM make a malicious insider enable for gathering data at specific intervals for reading the data exchanged among the VMs [10].

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Violation of the integrity of VM: Arbitrary code can be injected into VM as Dom0 has an access to a memory of VM.

Sniffing / spoofing: Before reaching DomU, Dom0 intercepts complete network traffic. Here a specific VM's network traffic can be sniffed to read the combined target communication or execute MITM assaults.

Rollback from VM: The procedure of rollback enables us to restore the prior state to a VM. It can disable earlier activated safety measures. A compromised hypervisor management interface uses the previous snapshot to execute the VM during the rollback attack [22]. A VM login password can be found by an attacker using a brute-force method.

VM relocation: The characteristics such as computer accesses and memory snapshots are disabled with the evolution of integrity protected hypervisors. However, virtualization offers a features to relocate VM to another server for the essence of load balancing. A malicious server can be relocated by an attacker where VM's memory snapshot can be taken [12].

Cloning of VM: VM is a kind of software, it can be transfer or backed up. Numerous executing cases of a virtual machine is not known to a VM proprietor. A malicious management interface for hypervisors can duplicate the VM to a hypervisor and begin it. It may get complete VM access.

Attack at kernel level: An example of VM generated through an infected VM picture can cause significant harm to a VM by viruses, rootkits, and other malware.

Rootkits of the kernel: It may have an administrative privilege and may be able to tamper with the VM kernel.

Rootkits ' seriousness.

—Modification of the kernel code: The kernel of operating system can be modified by intruder by inserting rootkits. At the administrators point, rootkits are operated.

—Modification of the kernel information structure. A rootkit can use the vulnerability of corrupted memory for damaging the information. It modifies the information and pointers in the data structures of the kernel directly. Direct Kernel Object Modification (DKOM)

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

attack [13]: It modifies the kernel accounting table data and bypasses the object manager entirely [14].

—Modification of the control flow kernel: The execution of running processes can be compromised and protection of code integrity can be bypassed by rootkit [15]. In order to execute a malicious code, memory corruption bug in the address space is exploited by Control-flow hijacking. Binary code reuse attack redirects control flow to present executable code.

Attack the Iago: It is presumed that the kernel is trusted in rootkit-based assault, and rootkits may interfere with its execution. More prone to compromise are the Commodity OS kernels and vice versa is also true where untrusted kernel exist and interferes with the running of a trusted application. A malicious kernel allows an application in an Iago attack [16] by communicating with it to act against its interests. The return values of kernel are not checked by Legacy applications.

Hypervisor to manipulate the request: An attacker with access to the hypervisor can manipulate the apps.

Unlawful access by co-hosted VMs: The failure of VM isolation results into unexpected covert channel and side channel between different VMs. The exposure of vulnerabilities because of assests shared and able to co-locate by an attacker for their destination VMs and also create covert/side channel with it [17].

Attack on the side channel of VM: A side channel enables us to obtain confidential target VM data. It utilizes shared resources such as memory,cache,CPU load etc. to obtain a target VM's sensitive data, resource usage status and cryptography keys. The hardware cache are shared by attacker for accessing the memory.

Target VMs co-location need to be identified for side-channel attacks. Authors [18] have recognized a specific likely virtual machine to be on the same host at that time. Also, a side-channel attack based on L2 cache is proved, which causes Amazon EC2 data leakage among VMs. Generally, to detect co-locationon Xen,HomeAlone [19] used the Prime+Probe method on

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

L2 cache. Authors obtained the ElGamal secret key used to decrypt [41]. A cache timing attack was aimed writers to remove ARM Cortex-A8 processor VM's operating system AES keys [20].

Flush+Reload [21] in X86 processors observes the leakage of information from the shared websites and are also responsible for extracting private keys from GNU Privacy Guard across different VMs. A mixture of Flush+Reload and lattice methods was used by the writers of Reference [22]. Flush+Reload technique obtains data and lattice methods for the reconstruction of secret keys for elliptical curves of 256 bit. In the VMware, the Flush+Reload technique is used to recover the "OpenSSL 1.0.1" AES keys executing within a target VM [43]. The memory deduplication in the VMware requires advantage of this. The authors [24] took advantage of L3 cache variations of access time. They utilized it without needing memory deduplication to operate across VMboundaries. In a multi-core situation, it needs only machine co-location. It is a variation of Prime+Probe attack based on L1 cache and works like Flush+Reload attack based on L3 cache. For attainment of co-location, L3 cache is monitored [25]. The noisy data from Amazon EC2 is extracted by Prime+Probe technique. It obtains the RSA encryption key of 2048-bit. Attack Prime+Probe performs the translation of the virtual addresses into physical addresses.

To transmit the VM information [26] , an address check is performed with another VM. GnuPG can use a Prime+Probe attack on the L3 cache to obtain the GnuPG crypto software key [27]. The flush instruction's execution time is used by Flush+Flush attack [28] . Since it makes no access to memory, there are no mistakes in the cache. The steady flush of the cache decreases the amount of hits of the cache. This attack can not therefore be identified using hits and misses from the cache. Physical memory is balanced by hypervisors by taking an advantage of balloon diver vulnerabilities by a shared memory-based side-channel attack [29].

Classic attacks on the network: The newly joined VM begins with active protocols, default services and open ports, allowing an intruder to ARP spoofing ,scan port, IP spoofing etc.

Inactive State security challenges for a VM: A VM image in the repository is considered to be an inactive VM during data launch on a physical device. To generate the VM cases, it involves configuration files.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Vulnerabilities: AVM image has many vulnerabilities in the virtual environment, such as weak access control, unsafe start-up channel, untrusted hypervisor, etc.

Threats: The inactive states of VM are affected from the uncontrolled uploading, creating, modifying, and using VM images. A threat model for inactive VM status plays a vital role. The distinct components used to watch the inactive state: repository image, picture startup, picture running, and virtual environment picture removal. The weak authentication and acess control results the VM pictures in the repository may be exposed to external users. The starting channel of the VM picture may be subjected to the network where an intruder may execute an assault on MITM. An untrusted hypervisor can launch a VM picture. Once the virtualization platform removes the VM image, another VM takes their hardware. By considering a fresh VM customer to be malicious, which may infringe the confidentiality of the information taken from VM? Moreover, it is possible to exploit the outdated software and known picture vulnerabilities. Various attacks are feasible as per the state of the VM picture.

VM picture content attacks— exploiting vulnerabilities of obsolete software and OS. Upon creation of a VM from the image, an attacker can exploit the well know vulnerabilities from virtual network and old sofwares on the system.. For instance, "TSWebProxy directory traversal vulnerability in recent Microsoft OSes such as Windows 7 SP1, Windows 8, Windows Server 2012 Gold and R2, etc., enables remote attackers to obtain privileges (CVE-2015-0016)."

The VM images attacks – The repository images can be accessed from a valid account. The VM images can be accessed with cryptography keys and user password which may leak the information [30, 31].

—Modify the VM picture. It is possible to infect the lawful VM picture by inserting rootkits or malware[40, 32]. A VM installed by such an picture can introduce virtual environment malware. Analyzing the VM activities and information can be leveraged, resulting in a violation of VM privacy.

—Uploading picture infected with malware. Because of the absence of adequate access control[40, 32, 31], an intruder can upload a VM picture with malware to a public repository.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

A Virtualization Exhaustive Survey on Security Concerns and Solutions— VM picture sprawl. VM image creation is code, it is possible to create multiple images in the repository by cloning, versioning, archiving, etc. Here, it can be generate by attacker, such numerous copies for very hard safety and maintenance. . This issue is referred to as the VM picture sprawl [33].

VM picture attack by MITM. During its launch from the repository to the target hypervisor, a MITMattack is feasible on aVMimage [32].

Attack on the VM picture of the hypervisor target. An untrusted hypervisor can launch a user's VM picture. Themalicious hypervisor may misrepresent the available resources and cause the VMowner to deploy a picture on it [32]. Since the VMowner is unable to determine the destination hypervisor's trustworthiness, its unsecured platform deployed these VM instances.

Attack fetch the information from VM. Physical depiction is known as data remanence of digital informaiton that even after removal stays on the physical device. If an attacker is assigned the same hardware, the attackers recreates the sensitive data of VM. It is feasible that both VM hard disk and VM memory information may be present [2].

Attacks on the network. Vulnerabilities in the network leads to the exploitation to various attacks such as port scanning, Denial of Service (DoS), spoofing of IP/MAC, sniffing, and so on.

4. Security Solutions for Virtualization

Large amount of work is performed on virtualization safety. The classification of current solutions in virtualization is performed as per the each element safety.

4.1 Hypervisor Security: It consists of hosted VMs, code, VM user interface and VM management. As depicted from hypervisor threat model, interfaces can become the medium of attack. In order to protect the hypervisor from, a specific interface, the assumption is trusting internal interfaces and developing solutions for the security. Hardware or nested hypervisors provides protection to software of hypervisor.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Securing Rootkits Hypervisor through Integrity Check: x86 processors's SMM is used by the HyperGuard [35] for verifying integrity of hypervisor's integrity. The integrity measuring code is protected by SMM. Also, rootkits are avoided by HyperCheck [36] that targets the integrity of Xen hypervisors. The integrity state assessment is outsourced. It can also perform monitoring and tracking of information and VM control code. The measuring agent in the SMM can be modified in HyperGuard [35] and HyperCheck [36].

HyperSentry [37] utilizes a software-based integrity measuring officer with access to the hypervisor's contextual data (right CPU state) to safeguard the measuring agent's manipulation. The agent is running in the context of the hypervisor and separated from the hypervisor by adopting a software and hardware component TCB.

HyperCheck[36], HyperSentry[37] and HyperGuard [35] are SMM-dependent, which the SMM rootkits can attack. Furthermore, as such a solution gives the integrity of static hypervisor software, the return-oriented rootkits can bypass it. The execution of type 1 hypervisor control flow integrity is verified by HyperSafe [38]. It involves two methods: indexing of limited pointers and non-bypassable lockdown of memory. Non-bypassable lockdown of memory offers integrity of hypervisor code by saving static information memory pages and hypervisor code. Reference authors [39] evaluated a runtime hypervisor checksum by modifying the CPU microcode. MGUARD [40] avoids any unlawful changes to the hypervisor code. A programmable guard can be integrated with the sophisticated "FB-DIMM" memory buffer. Continuously monitor the memory traffic between physical memory and CPU by the programmable guard.

HyperSafe [38] ,HyperGuard [35], MGUARD [40] , HyperCheck[36] and HyperSentry [37] focusses on control data from hypervisors and can not safeguard non-control information from hypervisors. An architecture supported by VM Hyperverify, tracks non-control hypervisor information using a VM control. The control VM records the hypervisor state using programmable device driver. It uses a memory analysis library to for performing the translation.

According to our observation, during its deployment and runtime, hypervisor must be embedded with an integrity checking mechanism.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

Conclusion

Virtualization in the construction of contemporary computing systems is regarded as the basis technology. However, the demand for safe virtualization is increasing owing to vulnerabilities existing in present virtualization platform applications. In this work, authors tried to demonstrate various threats, vulnerabilities and assaults on various virtualization parts, that genetically raise various privacy and safety issues of contemporary computer systems. We have recognized the study scope and some future difficulties by investigating the current safety alternatives.

A sophisticated safety model and better control of access and crypto algorithms are needed that can target distinct safety concentrations for various elements. This study assist the researchers for safety problems related to virtualization and make virtualization a platform for delivering secure and trusted computing systems.

REFERENCES

[1] Barham, Paul, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. "Xen and the art of virtualization." In *ACM SIGOPS operating systems review*, vol. 37, no. 5, pp. 164-177. ACM, 2003.

[2] King, S. T., P. M. Chen, and S. Virt. "Implementing malware with virtu—al machines." In *Proceedings of the 2006 IEEE symposium on security and privacy. W ashington, DC, USA: IEEE Comput—er Society*, pp. 3l4-327. 2006.

[3] Desnos, Anthony, Éric Filiol, and Ivan Lefou. "Detecting (and creating!) a HVM rootkit (aka BluePill-like)." *Journal in computer virology* 7, no. 1 (2011): 23-49.

[4] Colp, Patrick, Mihir Nanavati, Jun Zhu, William Aiello, George Coker, Tim Deegan, Peter Loscocco, and Andrew Warfield. "Breaking up is hard to do: security and functionality in a commodity hypervisor." In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, pp. 189-202. ACM, 2011.

[5] Shi, Lei, Yuming Wu, Yubin Xia, Nathan Dautenhahn, Haibo Chen, Binyu Zang, and Jinming Li. "Deconstructing Xen." In *NDSS*. 2017.

[6] Jasti, Amarnath, Payal Shah, Rajeev Nagaraj, and Ravi Pendse. "Security in multi-tenancy cloud." In *44th Annual 2010 IEEE International Carnahan Conference on Security Technology*, pp. 35-41. IEEE, 2010.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

[7] Kandias, Miltiadis, Nikos Virvilis, and Dimitris Gritzalis. "The insider threat in cloud computing." In *International Workshop on Critical Information Infrastructures Security*, pp. 93-103. Springer, Berlin, Heidelberg, 2011.

[8] Rocha, Francisco, and Miguel Correia. "Lucy in the sky without diamonds: Stealing confidential data in the cloud." In *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pp. 129-134. IEEE, 2011.

[9] Li, Chunxiao, Anand Raghunathan, and Niraj K. Jha. "Secure virtual machine execution under an untrusted management OS." In *2010 IEEE 3rd International Conference on Cloud Computing*, pp. 172-179. IEEE, 2010.

[10] Rocha, Francisco, Thomas Gross, and Aad Van Moorsel. "Defense-in-depth against malicious insiders in the cloud." In *2013 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 88-97. IEEE, 2013.

[11] Xia, Yubin, Yutao Liu, Haibo Chen, and Binyu Zang. "Defending against vm rollback attack." In *IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN 2012)*, pp. 1-5. IEEE, 2012.

[12] Shafieian, Saeed, Mohammad Zulkernine, and Anwar Haque. "Attacks in public clouds: Can they hinder the rise of the cloud?." In *Cloud Computing*, pp. 3-22. Springer, Cham, 2014.

[13] Butler, Jamie. "Dkom (direct kernel object manipulation)." *Black Hat Windows Security* (2004).

[14] Brasser, Ferdinand, Lucas Davi, David Gens, Christopher Liebchen, and Ahmad-Reza Sadeghi. "Can't touch this: Practical and generic software-only defenses against rowhammer attacks." *arXiv preprint arXiv:1611.08396* (2016).

[15] Checkoway, Stephen, Lucas Davi, Alexandra Dmitrienko, Ahmad-Reza Sadeghi, Hovav Shacham, and Marcel Winandy. "Return-oriented programming without returns." In *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 559-572. ACM, 2010.

[16] Checkoway, Stephen, and Hovav Shacham. "Iago attacks: why the system call API is a bad untrusted RPC interface." In *ASPLOS*, vol. 13, pp. 253-264. 2013.

[17] Seaborn, Mark, and Thomas Dullien. "Exploiting the DRAM rowhammer bug to gain kernel privileges." *Black Hat* 15 (2015).

[18] Ristenpart, Thomas, Eran Tromer, Hovav Shacham, and Stefan Savage. "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds." In *Proceedings of the 16th ACM conference on Computer and communications security*, pp. 199-212. ACM, 2009.

[19] Zhang, Yinqian, Ari Juels, Alina Oprea, and Michael K. Reiter. "Homealone: Co-residency detection in the cloud via side-channel analysis." In *2011 IEEE symposium on security and privacy*, pp. 313-328. IEEE, 2011.

[20] Weiß, Michael, Benedikt Heinz, and Frederic Stumpf. "A cache timing attack on AES in virtualization environments." In *International Conference on Financial Cryptography and Data Security*, pp. 314-328. Springer, Berlin, Heidelberg, 2012.

[21] Yarom, Yuval, and Katrina Falkner. "FLUSH+ RELOAD: a high resolution, low noise, L3 cache side-channel attack." In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pp. 719-732. 2014.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

[22] Benger, Naomi, Joop Van de Pol, Nigel P. Smart, and Yuval Yarom. "“Ooh Aah... Just a Little Bit”: A small amount of side channel can go a long way." In *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 75-92. Springer, Berlin, Heidelberg, 2014.

[23] Irazoqui, Gorka, Mehmet Sinan Inci, Thomas Eisenbarth, and Berk Sunar. "Wait a minute! A fast, Cross-VM attack on AES." In *International Workshop on Recent Advances in Intrusion Detection*, pp. 299-319. Springer, Cham, 2014.

[24] Irazoqui, Gorka, Thomas Eisenbarth, and Berk Sunar. "S $ A: A shared cache attack that works across cores and defies VM sandboxing--and its application to AES." In *2015 IEEE Symposium on Security and Privacy*, pp. 591-604. IEEE, 2015.

[25] Inci, Mehmet Sinan, Berk Gülmezoglu, Gorka Irazoqui Apecechea, Thomas Eisenbarth, and Berk Sunar. "Seriously, get off my cloud! Cross-VM RSA Key Recovery in a Public Cloud." *IACR Cryptology ePrint Archive* 2015, no. 1-15 (2015).

[26] Younis, Younis A., Kashif Kifayat, Qi Shi, and Bob Askwith. "A new prime and probe cache side-channel attack for cloud computing." In *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 1718-1724. IEEE, 2015.

[27] Liu, Fangfei, Yuval Yarom, Qian Ge, Gernot Heiser, and Ruby B. Lee. "Last-level cache side-channel attacks are practical." In *2015 IEEE Symposium on Security and Privacy*, pp. 605-622. IEEE, 2015.

[28] Gruss, Daniel, Clémentine Maurice, Klaus Wagner, and Stefan Mangard. "Flush+ Flush: a fast and stealthy cache attack." In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 279-299. Springer, Cham, 2016.

[29] Wang, Ziqi, Rui Yang, Xiao Fu, Xiaojiang Du, and Bin Luo. "A shared memory based cross-VM side channel attacks in IaaS cloud." In *2016 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pp. 181-186. IEEE, 2016.

[30] Hashizume, Keiko, Nobukazu Yoshioka, and Eduardo B. Fernandez. "Three misuse patterns for cloud computing." In *Security engineering for Cloud Computing: approaches and Tools*, pp. 36-53. IGI Global, 2013.

[31] Wei, Jinpeng, 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*, pp. 91-96. ACM, 2009.

[32] Grobauer, Bernd, Tobias Walloschek, and Elmar Stocker. "Understanding cloud computing vulnerabilities." *IEEE Security & privacy* 9, no. 2 (2010): 50-57.

[33] Reimer, Darrell, Arun Thomas, Glenn Ammons, Todd Mummert, Bowen Alpern, and Vasanth Bala. "Opening black boxes: using semantic information to combat virtual machine image sprawl." In *Proceedings of the fourth ACM SIGPLAN/SIGOPS international conference on Virtual execution environments*, pp. 111-120. ACM, 2008.

[34] Albelooshi, Bushra, Khaled Salah, Thomas Martin, and Ernesto Damiani. "Experimental proof: Data remanence in cloud vms." In *2015 IEEE 8th International Conference on Cloud Computing*, pp. 1017-1020. IEEE, 2015.

ISSN: 0971-1260
Vol-22-Issue-3-July-2019

[35] Rocha, Francisco, Thomas Gross, and Aad Van Moorsel. "Defense-in-depth against malicious insiders in the cloud." In *2013 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 88-97. IEEE, 2013.

[36] Wang, Jiang, Angelos Stavrou, and Anup Ghosh. "Hypercheck: A hardware-assisted integrity monitor." In *International Workshop on Recent Advances in Intrusion Detection*, pp. 158-177. Springer, Berlin, Heidelberg, 2010.

[37] Azab, Ahmed M., Peng Ning, Zhi Wang, Xuxian Jiang, Xiaolan Zhang, and Nathan C. Skalsky. "HyperSentry: enabling stealthy in-context measurement of hypervisor integrity." In *Proceedings of the 17th ACM conference on Computer and communications security*, pp. 38-49. ACM, 2010.

[38] Wang, Zhi, and Xuxian Jiang. "Hypersafe: A lightweight approach to provide lifetime hypervisor control-flow integrity." In *2010 IEEE Symposium on Security and Privacy*, pp. 380-395. IEEE, 2010.

[39] Zimmer, Vincent J., and Yasser Rasheed. "Hypervisor runtime integrity support." U.S. Patent 7,962,738, issued June 14, 2011.

[40] Liu, Ziyi, JongHyuk Lee, Junyuan Zeng, Yuanfeng Wen, Zhiqiang Lin, and Weidong Shi. *Cpu transparent protection of os kernel and hypervisor integrity with programmable dram*. Vol. 41, no. 3. ACM, 2013.

[41] Zhang, Yinqian, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. "Cross-VM side channels and their use to extract private keys." In *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 305-316. ACM, 2012.