Int. J. Communications, Network and System Sciences, 2015, 8, 260-273
Published Online July 2015 in SciRes. http://www.scirp.org/journal/ijcns
http://dx.doi.org/10.4236/ijcns.2015.87026

Security Challenges of Virtualization Hypervisors in Virtualized Hardware Environment

Gabriel Cephas Obasuyi, Arif Sari

The Management Centre of the Mediterranean, Nicosia, Cyprus
Email: gabrielcobasui@gmail.com, arifsari@acm.org

Received 16 January 2015; accepted 14 July 2015; published 17 July 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
http://creativecommons.org/licenses/by/4.0/
Open Access

Abstract

The concept of virtualization machines is not new, but it is increasing vastly and gaining popularity in the IT world. Hypervisors are also popular for security as a means of isolation. The virtualization of information technology infrastructure creates the enablement of IT resources to be shared and used on several other devices and applications; this increases the growth of business needs. The environment created by virtualization is not restricted to any configuration physically or execution. The resources of a computer are shared logically. Hypervisors help in virtualization of hardware that is a software interact with the physical system, enabling or providing virtualized hardware environment to support multiple running operating system simultaneously utilizing one physical server. This paper explores the benefits, types and security issues of Virtualization Hypervisor in virtualized hardware environment.

Keywords

Virtualization, Hypervisors, Virtual Machine, Virtual Machine Monitor, Security

1. Introduction

Virtual machine (VM) has been in existence since 1960s when IBM made the first ever VM to enable repeated interface access to a mainframe computer. Then each VM was an instance of the physical machine. It was a transparent way of enabling time-sharing and resource sharing on expensive hardware. The emerging of multiprocessing and cheaper hardware in the 1970s and 1980s almost pushed VM out of the IT world.

The increased demand for IT resources has created a vast predicament of deploying and managing IT resources in a larger scale. Cloud computing has transformed the way people use computers and how services are

How to cite this paper: Obasuyi, G.C. and Sari, A. (2015) Security Challenges of Virtualization Hypervisors in Virtualized Hardware Environment. *Int. J. Communications*, *Network and System Sciences*, **8**, 260-273.
http://dx.doi.org/10.4236/ijcns.2015.87026

run. The Cloud Service Providers (CSP) are able to provide IT infrastructures to meet the demand from the cloud users by simply leasing infrastructure from the infrastructure provider. This is achieved by the infrastructure provider using virtualization, where customers of the cloud service share the same physical services that are virtualized logically. Hypervisor can be run in two ways: It can run directly on the hardware this is called the (Type-1 or the bare-metal virtualization) or it can run on top of a host machine operating system this is known as the (Typw-2 or hosted virtualization) [1]. The native or bare-metal hypervisor is more robust, efficient and delivers greater scalability more than a hosted hypervisor, this is because the bare-metal hypervisor has direct access to the hardware resource of the host machine rather being on top of the host machine operating system. There are a number of ways of implementing virtualization, two approaches stand out which is the Full virtualization (FV) and the Para-virtualization (PV) [2]. Among the different commercial Hypervisors available the XEN hypervisor or virtualization solution can host both the Full and Para-virtualization [2] [3]. The purpose of this paper is to carefully study the security issues of Virtualization Hypervisor, the types, its implementation and benefit.

This paper is organized as follows: Section 2 describes virtualization hypervisors together with its implementation; Section 3 discusses the types of hypervisor and shows comparison of various commercial hypervisor; Section 4 presents the types of virtualization; Section 5 highlights several advantages of virtualization, while Section 6 shows the security features of virtualization, and finally a conclusion.

Hypervisor also known as Virtual Machine Monitor (VMM), is the basic software for providing virtualization. Virtualization creates the environment for various operating system to run on a single node or host computer or machine. The main purpose of the hypervisor (VMM) is to monitor the Virtual Machine (VM) that runs above the VMM. This enables more than one guest machine to utilize the hardware resources of a single host machine [4]. Hypervisors are classified into two: Native or Bare Metal, and Hosted hypervisors. Virtualization through the implementation of VMM reduces cost, space requirement etc. There are various model of virtualization available but it all depends on the hypervisors role. The unique security features and pros of virtualization has increased the research line in virtualization [5]-[14]. Several implementations of Virtual Machine have been seen to monitor and protect operating system kernel such as: Livewire which is applied for monitoring of Malware detection [5]-[15], SecVisor [13]-[15], NICKLE [11]-[15], VMwatcher [2]-[15], Lares [4]-[15], HookSafe [11]-[15], and SIM [9]-[15], they all utilize virtualization to protect the guest machines Operating system kernel integrity and the behavior of the kernel. Also some other utilizations of virtualization have been integrated into the debugging and analysis tool such as K-Tracer [6]-[15], PoKer [8]-[15], and After Sight [15], they all examine the system and the kernel anomaly.

Figure 1 shows a virtual environment, where the hypervisor is right above the host hardware, and virtualizing guest machine with the full capability or more of the host machine.

2. Virtualization Hypervisors

Virtualization enables or allows multiple applications or operations to gain access to the hardware resources/ software resources of the host machine. Virtualization is a layer between the hardware and the operating system

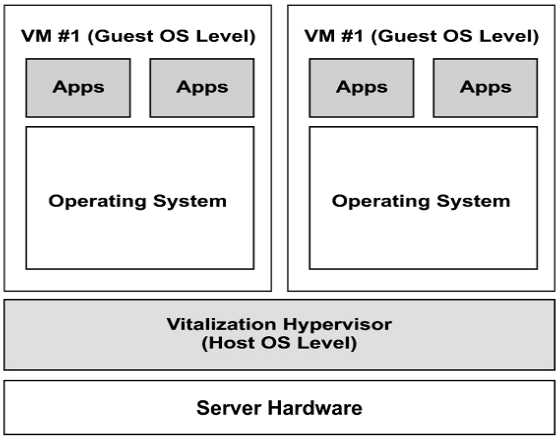

Figure 1. Virtualization architecture.

and it also provides access transparency. The hypervisors also known as the Virtual Machine Monitor (VMM), manages the applications and the operating system in general. There's a path created by the VMM which allows multiple of the same operating system to run on the host machine as well with the hypervisor managing the resources among the various operating system hardware requirement. In [16], Virtual Machine originated from the PR/SM hypervisor built for IBM 370 mainframes systems in 1970. In memory virtualization, an application or software in the computer gains access to more memory than what is originally installed physically, this way other IT infrastructures can also have memory virtualization to increase productivity, efficiency, and effectiveness of the corresponding application, such infrastructure includes: networks, storage, memory, server hardware, laptop hardware, operating system and applications alike. Figure 2 shows a host machine before virtualization; it runs a single Operating system image per machine, software and hardware resources per machine, resource are not used fully, it is not flexible and costly infrastructures, and running of more than one instance of an application on the same machine often causes conflicts of applications.

Figure 3 depicts a computer that is virtualized, that can host more than one operating system in its virtualized environment, running at the same time, and virtual machines can be deployed on any system, and it doesn't rely on the operating system nor hardware of the host.

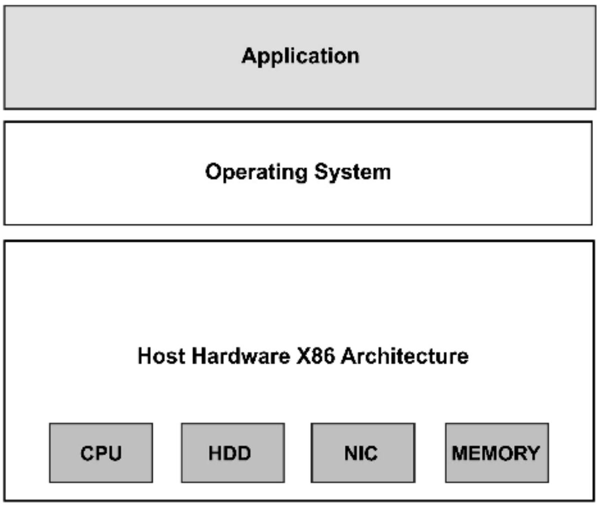

Figure 2. A machine before virtualization.

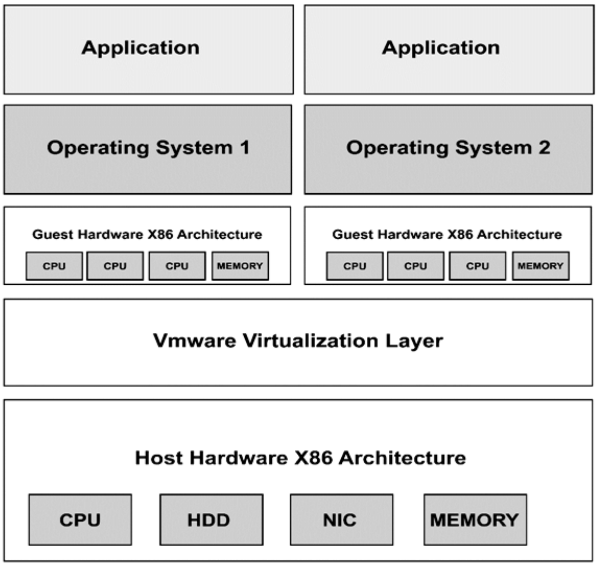
Application
Application
Operating System 1
Operating System 2
Guest Hardware X86 Architecture
CPU
CPU
CPU
MEMORY
Guest Hardware X86 Architecture
CPU
CPU
CPU
MEMORY
Vmware Virtualization Layer
Host Hardware X86 Architecture
CPU
HDD
NIC
MEMORY

Figure 3. A machine after virtualization.

Virtual infrastructure gives administration the upper hand in handling resources put together across the enterprise at large, allowing IT managers to focus and be more responsive to changing IT needs in an organization by utilizing the power of vitalization. Virtualization can be used and implemented in many ways among which are:

a) Server Consolidation: This combines or centralizes the workloads of various physical machines that are not fully used to lesser machines that can run safely and transparently over shared hardware infrastructure and also increase the overall utilization of the server from 5% - 15% to 60% - 80% [17]. **Figure 4**, illustrate the server consolidation virtualization, where different physical servers are virtualized into one physical server and then virtualized as different servers, increasing its efficiency, workability, speed etc.

b) Application Consolidation: This is giving legacy or outdated applications the environment to utilize new hardware and operating system by virtualizing the new hardware and providing access to other guest machines to utilize the application.

c) Multiple Execution: Virtualization can help create more than one environment for program or application execution ad also the quality of service can be increased by ensuring that specific amount of resources is allocated appropriately.

d) Virtual Hardware: The virtualization of hardware that is unavailable to users is achieved in virtualizing hardware. Examples of such hardware are: SCSI drivers, Virtual Ethernet Adapters, Virtual Ethernet Switches, and Hubs etc. as shown in **Figure 5**.

e) Debugging: The virtual environment can help in debugging of applications or software that are complicated such as an operating system or a device driver. This is achieved by allowing the user to execute the software in a virtualized environment with all the full control of the software available in the environment, giving the programmer or developer the perfect environment for debugging.

f) Multiple Simultaneous Operating System: Virtualization enables the facility of having and running more than one operating system simultaneously, and also having different applications according to the users demands as shown in **Figure 6**. The guest machine runs on the virtualized application or software that in turn runs above

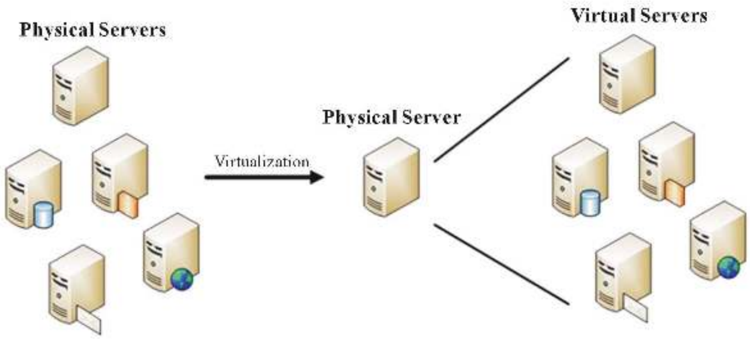

Figure 4. Server consolidation.

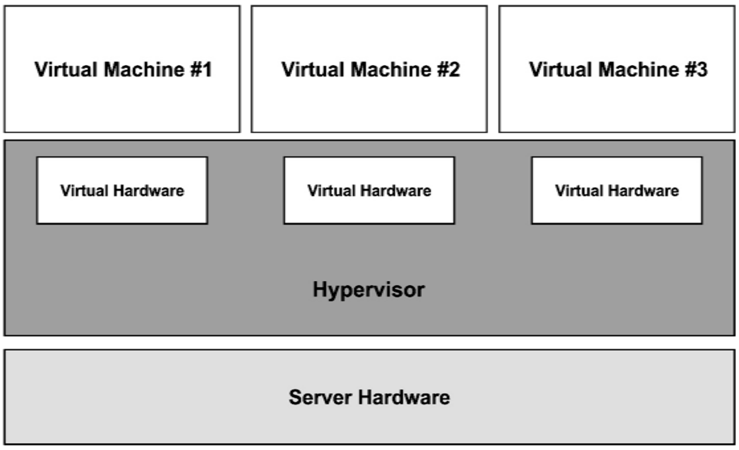

Figure 5. Hardware virtualization.

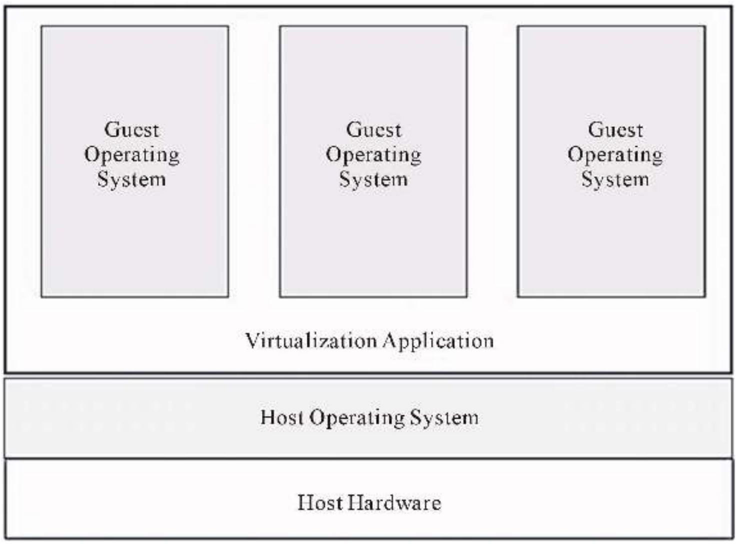

Figure 6. Multiple operating system virtualization.

the host machine operating system.

g) Business Continuity: This is achieved by putting the entire system files into a single file that can be replicated and restored on any server. This reduces downtime.

h) Sandboxing: Virtual Machine helps in providing secure and isolated environments for applications that are less trusted in the virtualized operating system. Virtualization helps in creating a secure computing environment.

i) Software Migration: This ease the migration or moving of software form one server to another, thereby helps mobility.

Virtualization has been part of the IT environment for decades. Today, Virtual Machine can be used in any system layer ranging from hardware, operating system, high-level languages virtualization etc.

3. Types of Hypervisors

Hypervisors as stated earlier is a software that manages different operating system or different instances of the same operating system in one physical computer or host machine, has two distinct types namely: Type 1: Native or bare Metal and Type 2: Hosted hypervisors.

3.1. Type 1: Native or Bare Metal Hypervisor

These are software that run directly above the hardware of the host machine. It also monitors the operating system that runs directly above the hypervisor and also monitors the operating system that runs on the guest machine. This is because the guest machine operating system runs on a different or isolated level that is directly above the hypervisor. Examples are Oracle VM, Microsoft Hyper-V, VMWare ESX and Xen [18], as shown in **Figure 7**.

3.2. Type 2: Hosted Hypervisor

The hypervisor is hosted or installed on an already existing operating system and it houses other operating system that is above it. In this type of hypervisor, any problem occurring with the host operating system will affect guest machine operating system that is running on the hypervisor and also it affects the hypervisor itself, although sometime the hypervisor running above the operating system might be secured but the guest operating system wouldn't be. As shown in **Figure 8**, the hosted operating system has an additional layer above it where the hypervisor resides and a third layer is above the hypervisor. Examples of such are Oracle VM Virtual Box, VM Ware Server and Workstation, Microsoft Virtual PC, KVM, QEMU and Parallels [18]-[20].

The hosted architecture of hypervisor, relies on host operating system for device support and physical resource management.

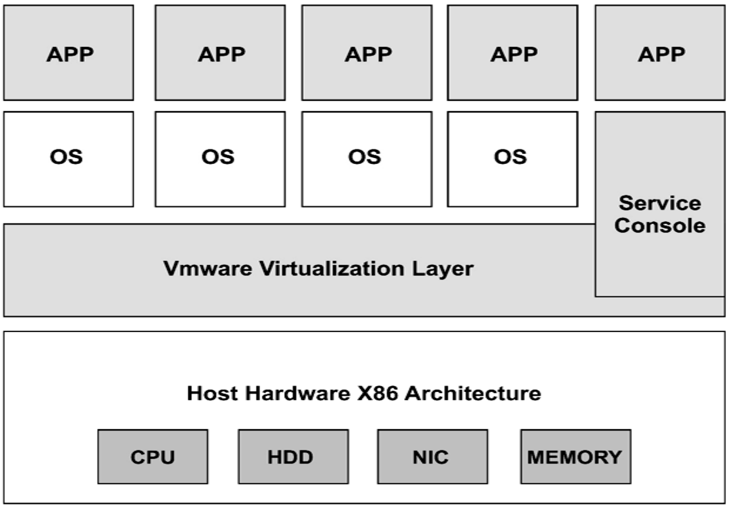

Figure 7. Native or bare metal hypervisor.

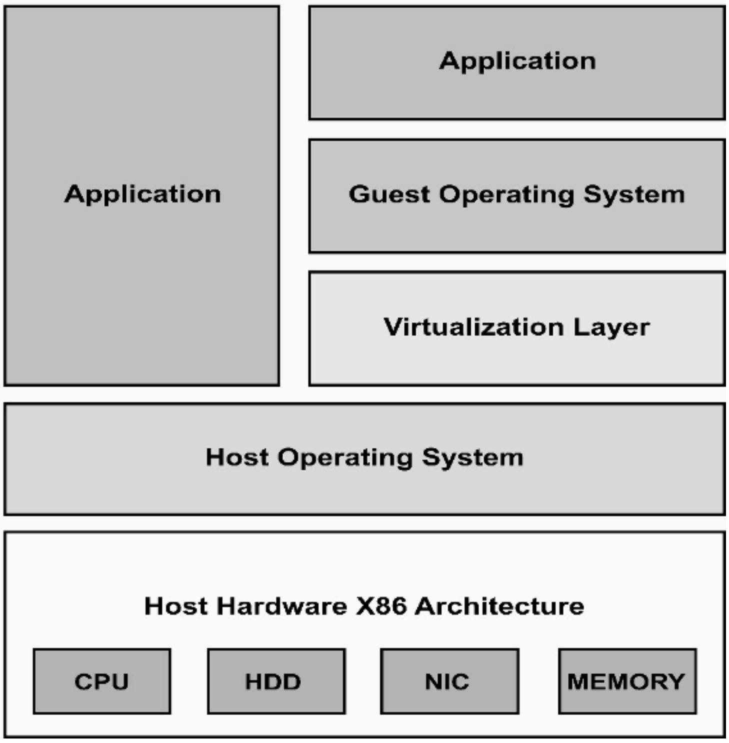

Figure 8. Hosted hypervisor.

Originally hypervisors were developed to suit server platforms, later on the virtualization of Desktop, PC operating systems were achieved. A challenge that held the virtualization of PCs operating system was the virtualization of the x86 based CPU architecture [21]. In virtualization, the x86 based CPU architecture a VMM is required below the host machine operating system above the hardware. Table 1 Shows different commercial hypervisors with common characteristics.

4. Types of Virtualization

There are numerous types of virtualization available in the IT world, in the cause of this research we are going to highlight some important ones that are currently applicable.

Table 1. Various hypervisor comparison.

Hypervisor	Host OS	Guest OS	Type of Hypervisor
Microsoft Hyper-V	Windows 2008 w/Hyper-v Role, Windows Hyper-V Server	Windows, SUSE	1
KVM	Linux	Linux, Windows	1 - 2
Xen	Linux, Solaris, NetBSD	BSD, Linux, Solaris, Windows	1
VMware ESX	None-bare-metal	Window, Linux, Novell, Netware, Sun Solaris, FreeBSD	1

a) Hardware Virtualization: This is the creation of a Virtual Machine that acts in the way of a real computer operating system. The software that's been installed is separate from the one on the hardware infrastructure [22]. For example, a host machine can virtualize a guest machine running on Linux operating system with the corresponding operating system software installed on it [23].

Types of Hardware Virtualization

- **Full Virtualization:** In this type of virtualization, a complete look-alike of the real hardware is virtualized to allow the software (consisting of guest operating system) to function without any modification [24]. As shown in **Figure 9**.
- **Partial Virtualization:** In this type of virtualization, not all of the host machine hardware are actually simulated (having a look-alike). This causes some programs to be modified in the guest machine to run in the virtualized environment [24]. This is shown in **Figure 10**.
- **Para-Virtualization:** This does not emulate the hardware environment in the software, instead it does organize the access to hardware resources in aid of the virtual machine [25]. The para-virtualized type of hardware virtualization offers possible performance benefits when a guest machine operating system is running in the virtualized environment with modifications done to the guest that is been virtualized [26]. Example of para-virtualization software is the Xen Open Source Virtualization software [25] [26]. This is shown in **Figure 11**.

a) Application Virtualization: This is the virtualization of applications in the host machine without modifying the host machine or the OS, File System, or Registry. With the application virtualization technology, organizations can easily deploy custom/commercial applications across the organization without installation conflicts, system changes etc. some benefits of application virtualization are:

- It ensures faster spread of the software
- ***Full Portability***: Applications that are virtualized can be accessed and shared from any network without the aid of a local server.
- Increased efficiency of application deployment
- ***Supportability***: Virtualized application does not require modifications of administrative or security permission for installation [27].

Figure 12 shows how application is virtualized.

b) Operating System Virtualization: In this technology the host machine desktop is totally moved from the real or physical operating system into a virtualized environment. The host machine is physically present but the virtualized operating system is hosted in another server elsewhere. Users of the virtualized operating system can conduct various kind of modification on their copy of the visualized operating system without other virtualized OS been affected [28]. **Figure 13** depicts how a virtual operating system is achieved. The host machine houses the host operating system, while the virtualized software runs just above the host operating system, and the virtualized OS is deployed above the virtualization software.

c) Nested Virtualization: This virtualization architecture enables the deployment of a virtual machine within another virtual machine [29] that is the running of one or more hypervisor within another hypervisor [30]. In nested virtualization, the hypervisor that is on the host machine is known as Level 0 or L 0; the hypervisor that runs on the guest machine LO is known as Level 1 or L 1; while the hypervisor that runs on the L 1 is known as the Level 2 or L 2 [31]. **Figure 14** illustrates the nested virtualization whereby one machine can host several server virtually.

d) Memory Virtualization: Memory virtualization enables applications to take advantage of a shared memory

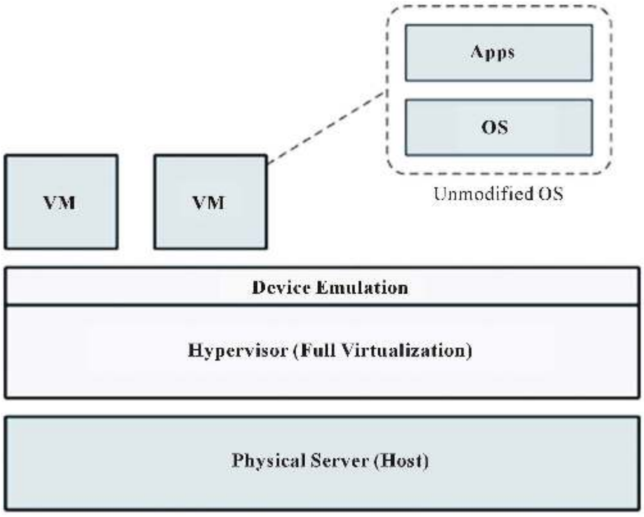

Figure 9. Full virtualization.

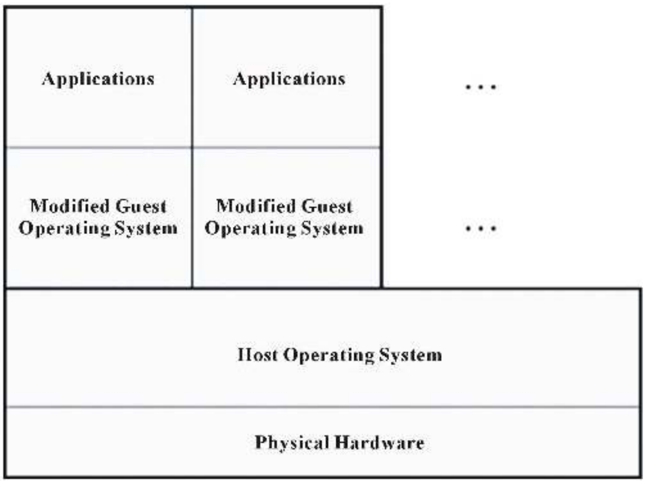

Figure 10. Partial virtualization.

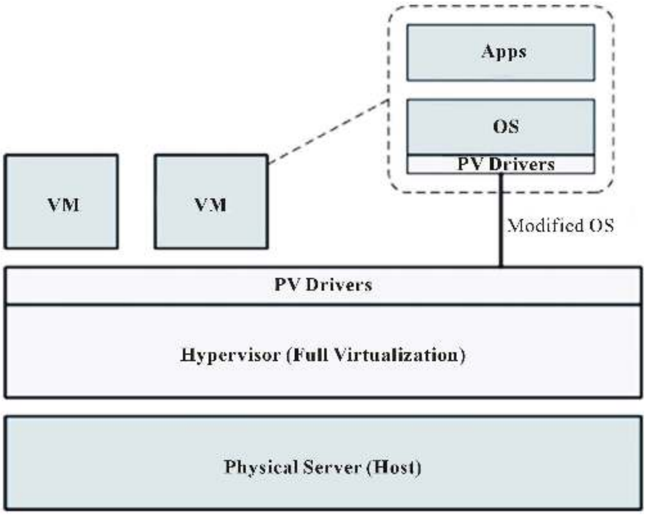

Figure 11. Para-virtualization.

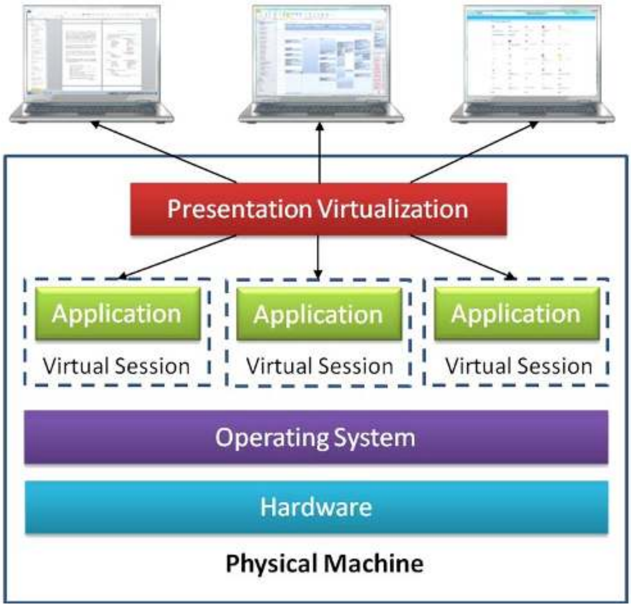

Figure 12. Application virtualization.

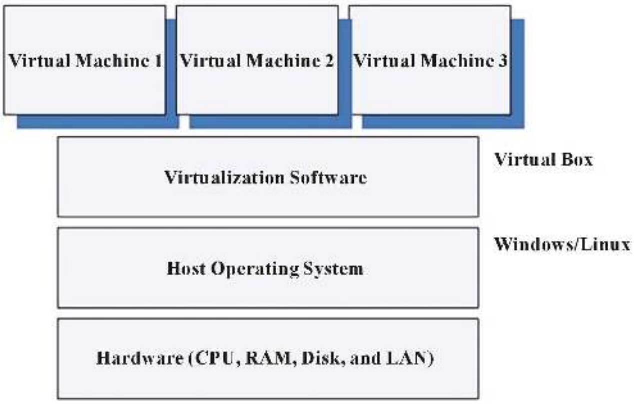

Figure 13. Operating system virtualization.

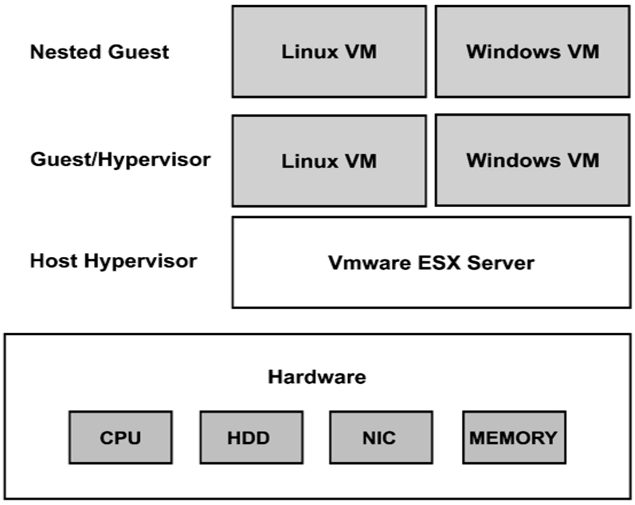

Figure 14. Nested virtualization.

pool to increase overall performance, usability, memory efficiency usage, stability etc. The memory virtualization shares a large pool of physical memory from more than one machine logically or virtually for applications to use [32]. **Figure 15** shows memory virtualization process from the host machine to the guest machine and to the application running in the guest machine (virtualized machine).

Some benefit of memory virtualization includes [32]:

- It improves the utilization of memory through the sharing of resources that are scarce.
- It increases the overall efficiency and reduces the run time data intensive application.
- It enables applications that run on various sever to share data without the reduction or decrease of the total memory needs.
- It provides faster access and reduces latency.

Figure 16 shows how an operating system connects to a memory pool and make available the pool memory to applications [32].

5. Advantages of Virtualization

- **Security:** A security breach on one of the virtual machines does not affect the other VM because of isolation. This is achieved by the different compact environment that have different or separate security measures in the

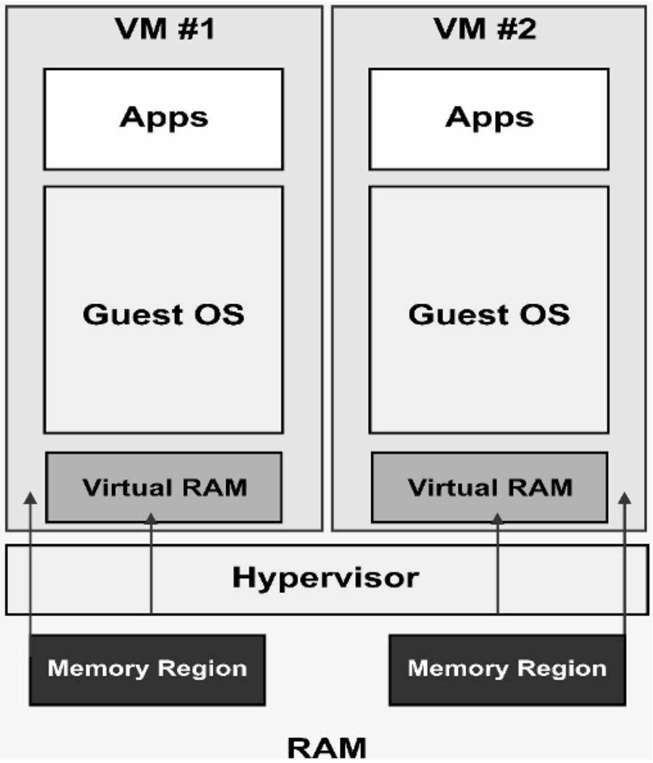

Figure 15. Memory virtualization.

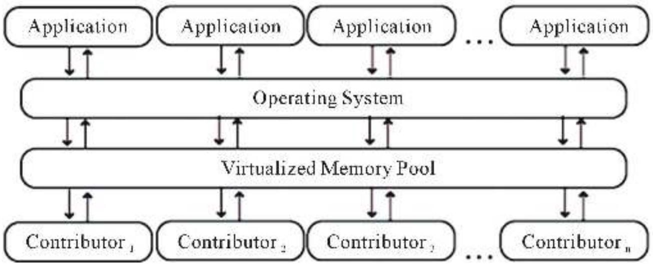

Figure 16. Memory pool virtualization.

different guest machines.

- **Reliability and Availability:** When there's a software failure in one virtual machine or guest machine, it doesn't affect other virtual machines.
- **Cost:** Virtualization is cost effective by combining small servers to secure a more powerful server. The cost effectiveness of virtualization runs down to the hardware, operations (man power), floor space, and software licenses. The cost reduction created from virtual machine ranges from 29% to 64% [33].
- **Adaptability to Workload Differences:** In virtualization when workload changes or varies, the workload degree can be optimized easily by shifting the resources and priority allocations between or among virtual machines [33]. Processors can also be moved from one virtual machine to another [34].
- **Load Balancing:** The software state of a VM is relatively condensed by the hypervisor, this makes it possible for migration of the entire virtual machine to another platform, it improves load balancing [35].
- **Legacy Applications:** This enables the running of legacy applications on old OD in the guest machines. For example if an enterprise decides to migrate to a different OS, it is possible to maintain the old legacy applications on the old VM or guest machine.

6. Virtualization Security

In a virtualized environment, various guest machines have liberated security zones which are not accessible from other VMs that also have their own security zone. Hypervisors also have their own security zone because it is the main controller of what happens inside the virtualization environment of the host machine. The functioning of a VM host can be affected by a hypervisor [36]. Multiple zones are available in a VM, all these zones occur within the same physical infrastructure. This can create a security problem when the hypervisor is attacked and is taken over by the attacker. When such attack is successful, the attacker gains full control over every data that's in the hypervisor environment. Another security problem is the access of the hypervisor from a guest machine or a VM level [37].

6.1. Abstraction

In Virtual Machine, the abstraction level adds additional security to the hypervisor. The OS restrict hardware access in VM by abstracting the hardware details. This is the reason why the same OS can be initiated on two machines with different configurations. VM creates an abstraction of the hardware and OS. **Figure 17** shows, the guest OS running inside a VM, can't tell the host machine OS or hardware configuration at all.

Because hypervisors are much simpler in operation than the native or traditional OS, it is much easier to secure [38].

6.2. Isolation

The hypervisor create segments of the physical resources and isolates them allowing each guest machine to run self-sufficiently. If an attack occurs on the VM it wouldn't affect other guest machines on the VMM or the host machine OS. The isolation of VM gives an additional level of security to the VM. When a Virtual Machine is compromised the hypervisor can restore the VM to an earlier state before the attack was done.

6.3. State Restore

Virtual Machines are capable of restoring a guest OS to an earlier time. On a time interval VMs take snapshot of the content in the virtual disk. State restore helps to guarantee data integrity and act as a virus removal.

6.4. External Monitoring

Virtual Machines run on separate hardware resource, this makes it possible to detect malicious software outside the VM unlike the physical installation of OS on a host, which requires an antivirus for protection. The hypervisor monitors the VM or a special Virtual machine that can view the systems activity and check for anomaly. **Figure 18** illustrates how a dedicated VM is used to monitor the activities of other VMs.

These dedicated monitors are used in intrusion detection system, integrity check, forensic analysis, etc. [39].

***Some other security benefits of virtualization are*:**

- The centralized storage used in virtual machine environments prevents loss of data when a device is either

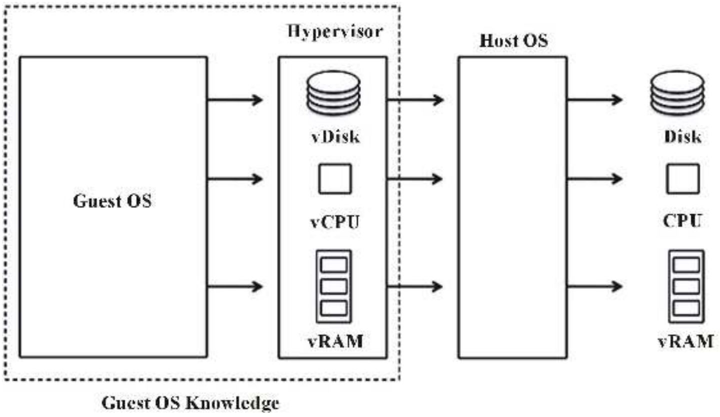

Figure 17. Abstraction of physical resources.

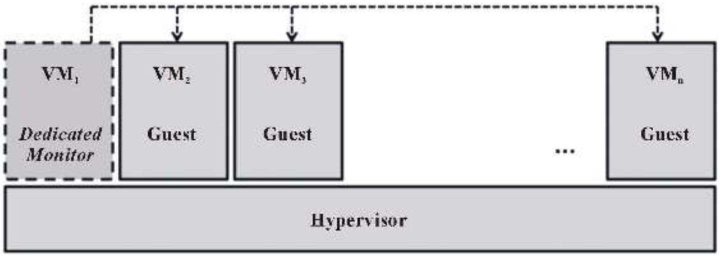

Figure 18. Dedicated external monitor.

lost or compromised.

- Virtual Machine provides flexibility.
- Reduction of physical hardware due to virtualization, reduces security risks because of few data centers.
- Server virtualization can lead to better handling of threat due to its ability to roll back to a working state before the attack or threat occurred.
- Desktop Virtualization helps to better control the virtualization environment. A better control of the OS is done using desktop virtualization to meet organizational needs.
- Virtual Switches is not open to inter-switch link tagging attacks because they does not carry out dynamic trunking; double encapsulation packets are dropped by the virtual switch, this prevents the double encapsulation attacks; virtual switch does not allow data packets to live its route or domain so that brute force attack does not work on them.

7. Conclusions

Today IT has expanded, the cloud environment runs various virtual machines for their infrastructure and applications. We discussed on the various type of virtualization that are available in the virtual environment, of which some are: Hardware virtualization, Application Virtualization, Memory Virtualization, Operating System Virtualization etc. These types of virtualization help in the utilization of virtualized resources or infrastructure. Also, in this research we found out the types of hypervisors and how they are deployed in the virtualized environment; the Native or Bare Metal Hypervisor runs directly on the host machine hardware, and the Hosted Hypervisor runs on a traditional OS or the host machine OS. A comparative table was drawn to show the differences and similarities between some commercial hypervisors available.

Virtualization technologies deliver numerous vital features that make it a powerful tool to be used in a wide array of applications. Some of them are server consolidation, application sandboxing, access to varieties of hardware and OS, debugging, mobile computing, packaging (for appliances), testing, easy system administration, and quality of service. These important features gave virtualization a widespread research area in academia as well as industry. Our future research would focus on mainly on the security challenges of Virtualization Hypervisors and how these challenges can be solved correspondingly.

References

[1] Gu, Z.H. and Zhao, Q.L. (2012) A State-of-the-Art Survey on Real-Time Issues in Embedded Systems Virtualization. *Journal of Software Engineering and Applications*, **5**, 277-290. http://dx.doi.org/10.4236/jsea.2012.54033

[2] VMWare (2007) Understanding Full Virtualization, Paravirtualization and Hardware Assist. http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf

[3] Fayyad-Kazan, H., Perneel, L. and Timmerman, M. (2013) Full and Para-Virtualization with Xen: A Performance Comparison. *Journal of Emerging Trends in Computing and Information Sciences*, **4**, 719-727.

[4] Expert Glossary. http://www.expertglossary.com/virtualization/definition/hypervisor

[5] Riley, R., Jiang, X. and Xu, D. (2008) Guest-Transparent Prevention of Kernel Rootkits with VMM-Based Memory Shadowing. *Proceedings of the* 11*th Recent Advances in Intrusion Detection*, **5230**, 1-20. http://dx.doi.org/10.1007/978-3-540-87403-4_1

[6] Jiang, X., Wang, X. and Xu, D. (2007) Stealthy Malware Detection through VMM-Based "Out-Of-the-Box" Semantic View Reconstruction. *Proceedings of the* 14*th ACM Conference on Computer and Communications Security*, Alexandria, VA, 29 October-2 November 2007, 128-138. http://dx.doi.org/10.1145/1315245.1315262

[7] Lanzi, A., Sharif, M. and Lee, W. (2009) K-Tracer: A System for Extracting Kernel Malware Behavior. *Proceedings of the* 16*th Network and Distributed System Security Symposium*, San Diego, February 2009, 83-91.

[8] Payne, B.D., Carbone, M., Sharif, M.I. and Lee, W. (2008) Lares: An Architecture for Secure Active Monitoring Using Virtualization. *Proceedings of the* 29*th IEEE Symposium on Security and Privacy*, Oakland, CA, 18-22 May 2008, 233-247.

[9] Rhee, J. and Xu, D. (2010) LiveDM: Temporal Mapping of Dynamic Kernel Memory for Dynamic Kernel Malware Analysis and Debugging. Tech. Rep. 2010-02, CERIAS.

[10] Riley, R., Jiang, X. and Xu, D. (2009) Multi-Aspect Profiling of Kernel Rootkit Behavior. *Proceedings of the* 4*th ACM European Conference on Computer Systems*, Nuremberg, 1-3 April 2009, 47-60. http://dx.doi.org/10.1145/1519065.1519072

[11] Seshadri, A., Luk, M., Qu, N. and Perrig, A. (2007) SecVisor: A Tiny Hypervisor to Provide Lifetime Kernel Code Integrity for Commodity OSes. *Proceedings of the* 21*st ACM Symposium on Operating Systems Principles*, Stevenson, 14-17 October 2007, 335-350. http://dx.doi.org/10.1145/1294261.1294294

[12] Sharif, M., Lee, W., Cui, W. and Lanzi, A. (2009) Secure In-VM Monitoring Using Hardware Virtualization. *Proceedings of the* 16*th ACM Conference on Computer and Communications Security*, Chicago, 9-13 November 2009, 477-487. http://dx.doi.org/10.1145/1653662.1653720

[13] Wang, Z., Jiang, X., Cui, W. and Ning, P. (2009) Countering Kernel Rootkits with Lightweight Hook Protection. *Proceedings of the* 16*th ACM Conference on Computer and Communications Security*, Chicago, 9-13 November 2009, 545-554. http://dx.doi.org/10.1145/1653662.1653728

[14] Yin, H., Liang, Z. and Song, D. (2008) HookFinder: Identifying and Understanding Malware Hooking Behaviors. *Proceedings of the* 16*th Network and Distributed System Security Symposium*, San Diego, 8-11 February 2008, 1-16.

[15] Chow, J., Garfinkel, T. and Chen, P.M. (2008) Decoupling Dynamic Program Analysis from Execution in Virtual Environments. *Proceedings of the* 2008 *USENIX Annual Technical Conference*, Boston, 22-27 June 2008, 1-14.

[16] Windows Server 2008 Hyper-V Technical Overview. http://www.google.com/url?sa=t&source=web&cd=1&ved=0CBcQFjAA&url=http%3A%2F%2Fdownload.microsoft.com%2Fdownload%2F4%2F2%2Fb%2F42bea8d6-9c77-4db8-b405-6bffce59b157%2FHyperV%2520Technical%2520Overview.docx&rct=j&q=hyper%20technical%20overview&ei=nA-RTZH9K_O10QHoneDfDg&usg=AFQjCNFLYm3D9izTVMzHZ_Nbe87WtEbAVg&cad=rja/

[17] www.vmware.com.

[18] https://docs.oracle.com/cd/E20065_01/doc.30/e18549/intro.htm.

[19] Rosenblum, M. and Garnkel, T. (2005) Virtual Machine Monitors: Current Technology and Future Trends. *Computer*, **38**, 39-47.

[20] Zhao, X., Borders, K. and Prakash, A. (2009) Virtual Machine Security System. *Advances in Computer Science and Engineering*, **1**, 339-365.

[21] Wang, Z., Jiang, X.X., Cui, W.D. and Ning, P. (2009) Countering Kernel Rootkits with Lightweight Hook Protection. *Proceedings of the* 16*th ACM Conference on Computer and Communications Security*, Chicago, 9-13 November 2009, 545-554. http://dx.doi.org/10.1145/1653662.1653728

[22] Virtualization in Education. IBM, October 2007.

[23] http://en.wikipedia.org/wiki/Virtualization

[24] http://securitywing.com/types-virtualization-technology/

[25] http://www.vmware.com/solutions/technology

[26] www.vmware.com%2Ffiles%2Fpdf%2Fapplication-virtualization-vmware-thinapp.pdf

[27] http://www.techadvisory.org/2013/07/4-types-of-virtualization-defined.

[28] http://en.wikipedia.org/wiki/Virtualization#Desktop_virtualization.

[29] http://wiki.xenproject.org/wiki/Nested_Virtualization_in_Xen

[30] VMware, Consolidating Mission Critical Servers. www.vmware.com/solutions/consolidation/mission critical.html

[31] http://en.wikipedia.org/wiki/Memory_virtualization

[32] Wasserman, O. and Hat, R. (2013) Nested Virtualization: Shadow Turtles. KVM Forum.

[33] Uhlig, R., Neiger, G., Rodgers, D., Santoni, A.L., Martins, F.C.M., Anderson, A.V., *et al.* (2005) Intel Virtualization Technology. *Computer*, **38**, 48-56. http://dx.doi.org/10.1109/mc.2005.163

[34] Texiwill, G. (2009) Is Network Security the Major Component of Virtualization Security?

[35] Bennani, M.N. and Menasce, D.A. (2005) Resource Allocation for Autonomic Data Centers Using Analytic Performance Models. *Proceedings of the* 2005 *IEEE International Conference on Autonomic Computing*, Seattle, 13-16 June 2005, 224-240. http://dx.doi.org/10.1109/icac.2005.50

[36] Sarna, D.E.Y. (2011) Implementing and Developing Cloud Computing Applications. Taylor and Francis Group, LLC, CRC Press, Boca Raton.

[37] Litty, L. (2005) Hypervisor-Based Intrusion Detection. Master's Thesis, Department of Computer Science, University of Toronto, Toronto.

[38] Sabahi, F. (2011) Intrusion Detection Techniques Performance in Cloud Environments. *Proceedings of the Conference on Computer Design and Engineering*, Kuala Lumpur, 12-14 August 2011, 398-402. http://dx.doi.org/10.1115/1.859797.paper64

[39] Gu, Z.H. and Zhao, Q.L. (2012) A State-of-the-Art Survey on Real-Time Issues in Embedded Systems Virtualization. *Journal of Software Engineering and Applications*, **5**, 277-290.