A FRAMEWORK FOR SECURE TRANSCODING OF VIDEOS IN CLOUD INFRASTRUCTURES

A DISSERTATION

Submitted in partial fulfillment of the
requirements for the award of the degree
of

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING

By
JITENDER KUMAR

(14535020)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE – 247 667 (INDIA)

MAY, 2016

DECLARATION OF AUTHORSHIP

I declare that the work presented in this dissertation with title **“A Framework for Secure Transcoding of Videos in Cloud Infrastructures”** towards the fulfillment of the requirement for the award of the degree of Master of Technology in Computer Science & Engineering submitted in the Dept. of Computer Science & Engineering, Indian Institute of Technology, Roorkee, India is authentic record of my own work carried out during the period from July 2015 to May 2016 under the supervision of **Dr. Sateesh K. Peddoju**, Assistant Professor, Dept. of CSE, IIT Roorkee.

The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

DATE: ……………….. SIGNED: …………………….

PLACE: ………………. (JITENDER KUMAR)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

SIGNED: …………………….

DATE: ……………….. (**Dr. Sateesh K. Peddoju**)

Place: Roorkee

Assistant Professor

DEPT. OF CSE IIT Roorkee

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep sense of gratitude and regards to my research supervisor **Dr. Sateesh K. Peddoju,** Assistant Professor, Dept. of Computer Science & Engineering, Indian Institute of Technology, Roorkee, for his trust in my work, his invaluable guidance, regular source of encouragement and assistance throughout the course of dissertation work. His wisdom, knowledge and commitment to the highest standards inspired and motivated me. He has been very generous in providing the necessary resources to carry out my research. He is an inspiring teacher, a great advisor and most importantly a nice person.

I would like to thank **Railtel Corporation of India Ltd. (RCI-763(3)-ECD)** for providing resources to setup cloud in High Performance Computing Lab. On a personal note, I would like to thank Animesh Kuity for helping me in setting up of cloud.

Lastly, I thank almighty for the wisdom and perseverance that he bestowed upon me during my research.

JITENDER KUMAR

ABSTRACT

Cloud computing provides organizations with benefits like flexibility, scalability, reliability, automatic load balancing, reduction in cost of set-up of infra-structure etc. But organizations are not still very open to cloud platform. The main reason behind this is security of data stored and processes running in the cloud. All the threats into enterprise systems are automatically repackaged into cloud systems.

There can be insider as well as outsider attacks in the cloud. Outsider attacks are done by the malicious persons who don’t have proper authority to use cloud resources or data stored into clouds. There can be various motivations behind these outsider attacks. Insider attacks are done by malicious person who has authority to operate the cloud system. These persons may be any administrators, employees in the cloud service provider’s organization, employees in the trusted business partner’s organization or any contractor. Insider attacks are more difficult to prevent than outsider attacks because insider attacks are performed by the persons who have authority to access the system.

The report discusses about the various types of Insider attacks and the scenarios where these attacks can happen. There are three types of Insider attackers- (1) Malicious Administrators, (2) Insiders who exploits the vulnerabilities into the cloud and (3) Insiders who uses cloud resources to attacks the others resources in the cloud or local resources of the an organization. Malicious administrators may have different level of capabilities depending upon at what level they are working. Approaches to detect and mitigate insider attack are hardware as well as software based.

This thesis presents a security framework which provides a computation environment which is free from a specific type of insider attack. Currently this framework is used to secure the transcoding process in a video streaming application.

The Video Streaming Application presented in this report is a Railtel project. Both insider and outsider attacks are possible into this Video Streaming Application. The security work presented in this thesis is more focused on insider attacks which can be launched by malicious administrator who uses management VM to control other VMs. To mitigate such

type of Insider attack, Input Output Memory Management Unit (IOMMU) is used in this security framework which makes it difficult for the administrator to know which memory area is the actual memory area of the target user. The IOMMU connects the main memory and input/output (I/O) buses of the physical hardware devices through DMA remapping. Secure computation is done in IOMMU based cloud.

CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Overview

The main reason behind the fact that organizations are not much comfortable to move their business on cloud is security of data stored and processes running in the cloud [1] [2]. Attacks on data stored into cloud may be insider attack or outsider attack. Insider attacks are more serious than outsider attacks [3] [4]. The report presents how insider attacks can take place on the data into cloud. Data during the insider attack may be residing into cloud storage or may be residing in memory for processing. The work presented in this report is basically on providing a secure framework where computations can be done securely. Both insider and outsider attacks are possible in this application. The app stores the videos into cloud and does processing on these videos in memory of virtual machines. So video can be stolen by the outsider attacker as well as insider attacker. The work is focused on how to mitigate the theft of videos from the cloud. Insider attacks are possible because in cloud more than one virtual machine run on the same hardware which shares processor time and resources [5] [6]. Available means for memory like a memory management and input output memory management provide spatial separation for these VMs [7]. IOMMU based clouds can conceal the security critical information from the insider attackers and can provide a secure computation environment [5] [6] [7] [8] [9]. To provide secure computing environment IOMMU based cloud has been built which can be used to mitigate insider attacks [8] [9]. Cloud is built using open nebula in coordination with KVM hypervisor to provide virtualization. Open Nebula used KVM (Kernel Based Virtual Machine hypervisor) to build and run virtual machines. Cloud has two types of physical resources- (1) IOMMU based and (2) General Resources. IOMMU based cloud is used for operations which require higher level of security like authentication, transcoding and encryption and decryption etc. Simple cloud is used for operations which require moderate security level like storage and streaming etc. Security Implementation mainly focuses on Insider attacks.

1.2 Motivation

There are numerous ways in which an insider can harm an organization. In the Video Streaming Application the Videos are stored in encrypted form in the cloud storage but are in plain form during the transcoding process. An administrator who uses management VM to manage the virtual machines can copy the video which are in plain form during transcoding. The motive is to stop the administrator using the management VM from copying the videos during transcoding.

1.3 Problem Statement

A secure cloud computing based framework to protect the computation, which is going on in virtual machine, from insider attacker is proposed in this thesis.

1.4 Organization of Report

Section 2 describes the background and related work. Section 3 introduces Video Streaming Application with high level and low level design and security issues in the Video Streaming Application. Section 4 describes the proposed framework and security analysis. Section 5 describes the implementation and results. Section concludes the work and describes the future work in this project.

2 LITERATURE REVIEW

2.1 Insider Attackers

Insider Attacks are usually done by cloud provider’s employees, contractors or business partners who have authorized access to resources [3] [4]. The motive behind Insider attacks may be financial benefit, personal revenge with employer’s or career enhancement by stealing intellectual property of the employer and providing it to the future employer [3]. There are three types of Insider Attackers which are described as follows:

1. A fraudulent administrator hired by cloud service provider
2. A fraudulent employee who uses vulnerabilities in the cloud to access resources
3. A fraudulent employee power of cloud resource to target other resources

2.1.1 Fraudulent administrator hired by cloud service provider

Researchers are interested to address mostly this type of cloud Insider [3]. As a result of this attack the sensitive information of the organization may be stolen which consequently results into risk of confidentiality and integrity. The motivation behind this attack may be financial benefits. Some more common motivation factors for Insider may be theft of intellectual property or fraud. But IT sabotage may be another motivation behind this attack, where an insider wishes to destroy IT infrastructure of his/her employer. This type of attack may be done in cloud environments. This attack is done by administrators. Even if the insider may not have any intension of revenge with the victim organization, he can damage the cloud service provider’s reputation by doing harm to a victim organization. Following is an example of this type of attack. A victim organization employed an Insider from a data-mining firm to process information related to customers. Insider could access the server and data of the victim organization although he could do his job without accessing the data and server. Administrator accessed an encrypted file which had passwords of the customers who belong to the victim organization. Fortunately the insider got arrested before he could sell stolen information.

A rogue administrator sometimes can attack to impact the customer data availability. This type of attack is also overlooked. There is one more example of attack similar to described above where a system administrator who was to manage data and operations for some companies. He didn't steal data but simply removed some required software from the server. This way system was not able to respond to the customer request. In this case customer data was intact and confidential but customer could not access the resources and important information for a long time.

2.1.2 Administrator who uses vulnerabilities Introduced by Use of the Cloud

This type of attack is also overlooked most of the time by security researchers. This is second type of insider threat which is related to cloud architecture [3]. This is done by the insider who takes benefit of vulnerabilities which is because of use of cloud services. These vulnerabilities can be exploited to get into systems and/or steal data which belongs to company. This can be intentional (malicious) or unintentional (accidental). This threat may also be successful because of the inability of the organization to respond quickly. This type of attacker tries to steal the information including intellectual property of the organization which can be sold. The cloud may provide the easiest way to compromise security level with minimum probability of detection. But sabotage attacks must not be ignored. An attacker can easily target the local system to destroy some resources into it if the attacker is not willing to steal the sensitive information. The example of this type attack is- An employee was tricked to open a document by the malicious outsider. The employee belonged to the victim organization and the document consisted of a malware. Outsider attacker used that exploit to get access to the email system of victim organization the email service was hosted on cloud. Even Though the victim organization had come to know of the attack while it was going on, the victim organization couldn't abort the email service very quickly to prevent loss of sensitive data. The victim couldn't validate its identity to the cloud service provider in the given time so this inability increased the severity of the attack.

This attack actually comes under an accidental insider attack, where the employee had no intention to do something harm. However, the outside attacker got credentials of that administrator who was employee in the victim organization, and miss used those credentials to harm the victim organization. The attack was looked like done by an inside administrator but actually he didn’t. The weakness exploited in this way could harm the victim because victim had no direct control over the email service which was hosted by a cloud service provider.

2.1.3 Using the Cloud to Conduct Nefarious Activity

An employer may be attacked in one another way. An insider who uses this type of attack comes under third type of attacker [3]. This type of attacker uses power of cloud services and resources to carry out attack employer’s infra-structure. This attacker is similar to the other attackers in the sense that it also targets the system and data of the organizations. This attacker may target cloud systems itself by using the cloud resource as power or he can target the systems which are not part of cloud. This type of attack can happen in the following cases:

- To gain access to organizations bank account an insider can use power of cloud resources to hack the password file.
- An unsatisfied insider can use cloud power to do distributed denial of service attack on organization.
- An insider who has to leave his organization can use cloud power to consolidate company’s information.

2.2 I/O Virtualization Methods

The idea of virtualization is not new to Computer Science. It was used in Operating Systems to implement multitasking where it is called process virtualization. Before cloud computing in old days the resource sharing was achieved by executing the processes under the control of a supervisor i.e. Operating System. Now with cloud computing the picture of virtualization has become somewhat complex. The idea of supervisor is same but in cloud computing supervisor is not the Operating System. Here operating systems run as processes under a supervisor i.e. Hypervisor (Supervisor of supervisor) on same hardware [10]. This type of virtualization is known as System Virtualization. Virtualization is the abstraction of computer resources such that these resources can be multiplexed to many consumers; consumers may be any software or hardware component. With virtualization more than one user can be served with single computer system. So the motive behind virtualization is to abstract the hardware components of computer system and offer them to multiple operating systems for resource sharing. Virtualization is the technology to make the cloud computing efficient and scalable. Server consolidation is possible with virtualization. Any hardware resource like processor, network card or hard disk can be virtualized.

Popek and Goldberg [11] mention three requirements which a hypervisor must satisfy. First, VM which is running inside a hypervisor should have the same effect if it were running directly on hardware as a host operating system. Second, Degradation in the performance of the virtual machine running under a hypervisor should be small. Third, hypervisor should have the control of hardware resources.

To maintain the performance, Hypervisor must allow the virtual machines to execute the instruction directly on the physical hardware of the machine. But this direct access should not violate the property of isolation and correctness i.e. a virtual machine must produce the correct results and must not interfere with the resources which are assigned to some another virtual machine [11]. Hypervisor doesn’t interfere when a VM tries to execute a non-privileged instruction on physical hardware. But when VM tries to execute a privileged instruction on physical hardware this instruction is trapped into hypervisor to maintain the isolation and protection. The virtualization of I/O devices is much more complicated than the virtualization of processor or memory because [11]:

1. Arrival rate of I/O events is high because I/O devices are being used by many virtual machines.
2. Speed of CPU is much higher than that of I/O devices. The speed of CPU has been increasing with time but increase in speed of I/O devices has not much increased. So there a mismatch in speed of CPU and speed of I/O devices.
3. One I/O device may be used by many virtual machines so multiplexing and de-multiplexing of these I/O devices is required which is not so easy. The multiplexing and de-multiplexing of these devices must be safe.

4. Hypervisor intervention is required which results in delay of I/O operations and CPU overhead.
5. Usage of I/O bandwidth must be fair. There should not be any starvation.

The virtualization of Network Interface Card is more complicated than other I/O devices because of the following points:

1. Most of the I/O devices use “pull mechanism” for operations. For example, when an application has to read or write from a particular location of memory or disk it will make a request for that location. So the operating system has the knowledge about I/O requests in advance and hence optimization is possible. The same is true in case when an application has to send some data via Network Interface Card but picture changes when Network Interface Card receives data from the network. Operating System doesn’t have knowledge of these received packets in advance. So interrupts are required as a result guest virtual machine will trap into hypervisor.
2. Rate of arrival of packets is high. In a group of packets received in a particular window of time it is observed that packets belong to different VMs.
3. These received packets have to pass through additional layer of software, Hypervisor.
4. Many applications need to be latency compliant and hence can’t tolerate long delays.

There are mainly 4 methods for I/O for virtual machines in a Cloud Computing Environment. These are classified as follows:

2.2.1 Fully Virtualized Device Model

In this model of I/O physical devices are controlled and multiplexed by hypervisor [11] [12]. Multiple virtual device interfaces are emulated by hypervisor. The guest virtual machine’s code will remain unchanged in this model and guest VM will not know about the virtualization. In this model when an application has to do some input or output operation the application will call the function written in guest driver. The guest driver will execute a privileged I/O instruction but this instruction will not directly run on hardware. As result of this privileged instruction a trap will be sent to hypervisor. Hypervisor will then select the required hardware device for this operation by decoding this instruction. Then Hypervisor initiates the required I/O operation. As far as efficiency and transparency is concerned this approach is fine. But in this approach physical devices are controlled by hypervisor. Instructions received from guest driver are translated into another format by the hypervisor. This translation is known as binary translation. This approach requires more maintenance work. For example if guest driver is updated or modified then the physical device driver resident in the hypervisor is also modified so that binary translation may be correct. This model is implemented in VMWare ESX server. Figure 2.1 shows this model pictorially.

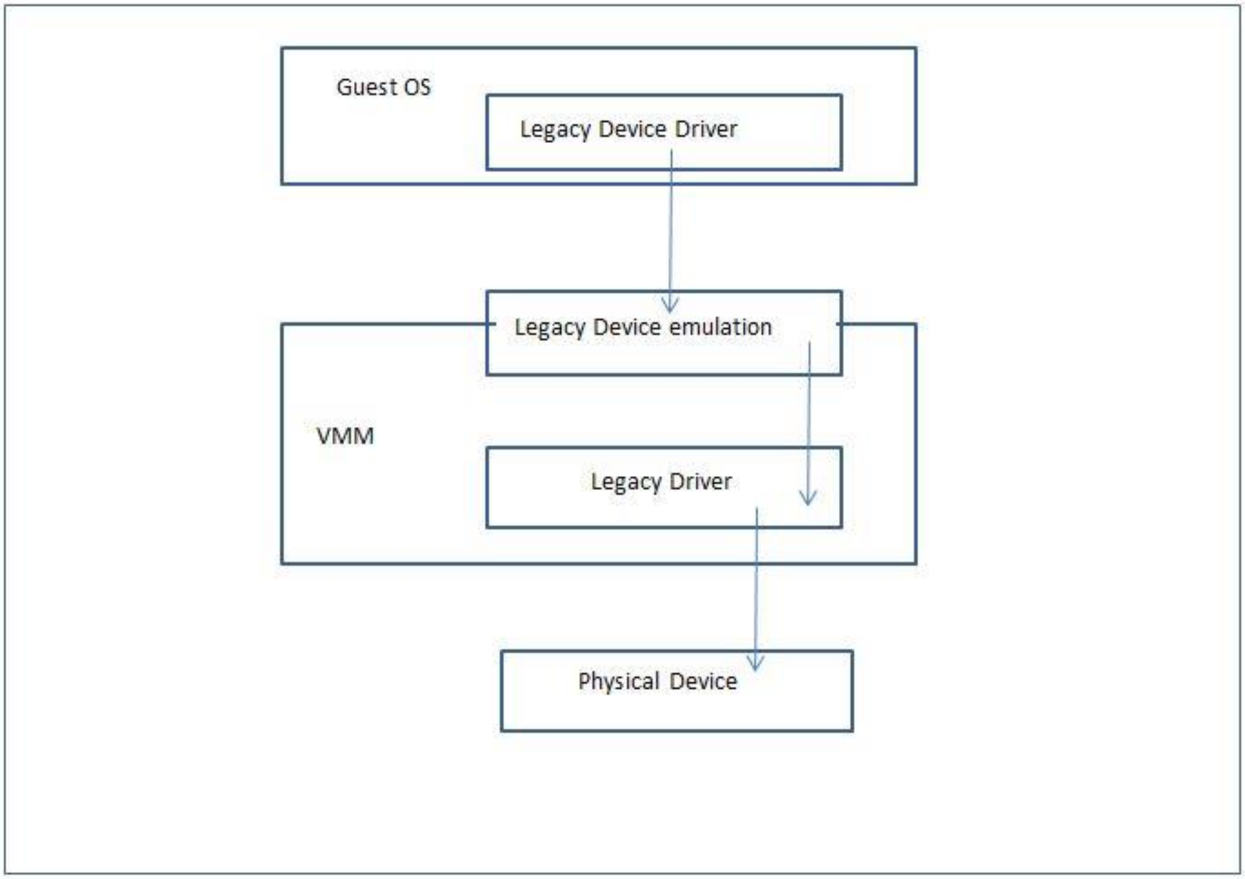


Figure 2-1 Fully Virtualized Device Model [12]

2.2.2 Para-virtualized Device Model

In this model device driver is comprised of two parts- Front End Driver and Back End Driver [5] [11] [12]. Guest VM runs the front end driver while the back end driver is placed into a specialized VM which has more privilege than other VMS.

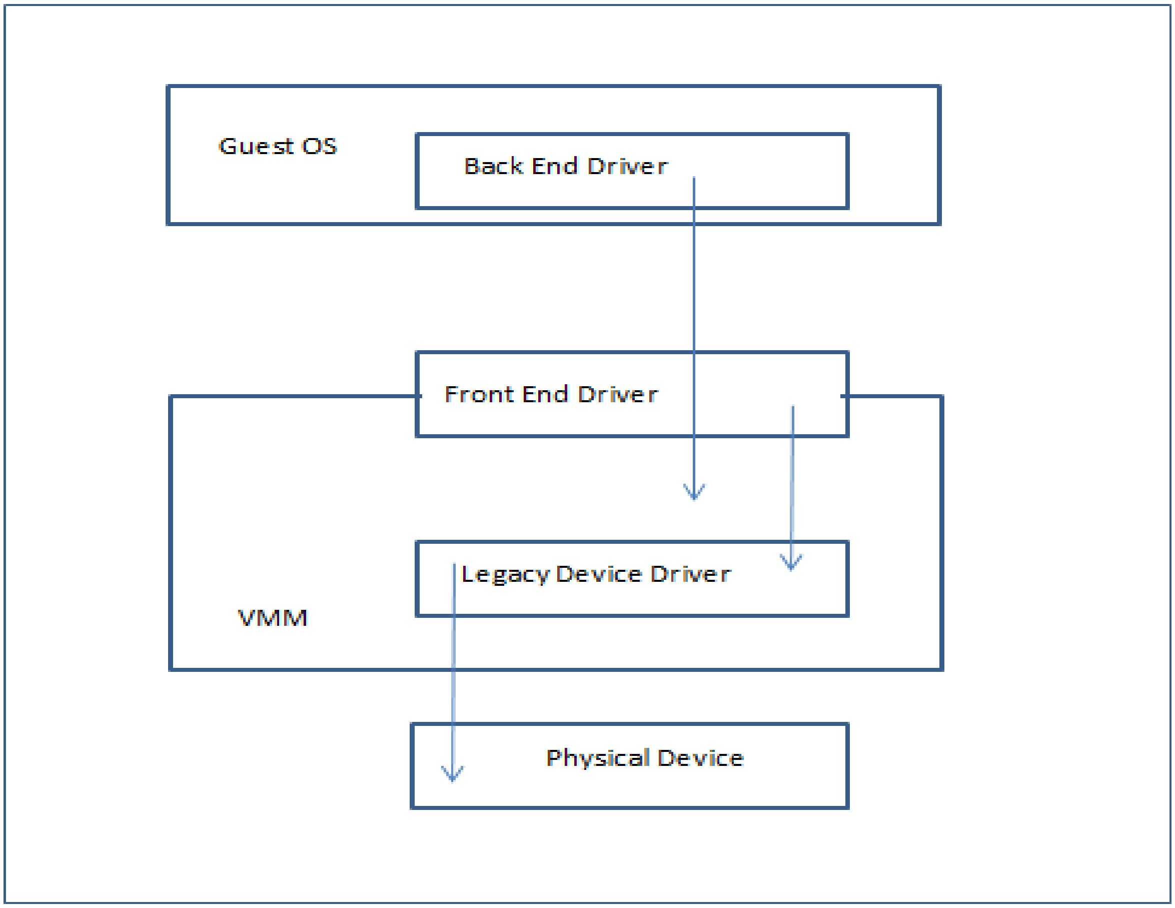


Figure 2-2 Para-virtualized Device Model [12]

This VM is called Driver Domain or Management VM. The operating system which is run by the guest VM is modified. In this model when an application has to do some input or output operation the application will call the function written in front end part of the guest driver. The front end driver will then send a request to the backend driver. The backend driver will then selects the proper hardware device for this operation by decoding the instruction received from front end driver. After this Physical driver in the driver domain initiates the I/O operation. The physical devices are controlled by the Driver Domain. Resource management task can be done by the backend driver. In this technique guest operating system is modified. This technique is different from the pure virtualization technique in the sense that device drivers are not controlled by hypervisor. Xen hypervisor uses this technique. Figure 2.2 shows this model pictorially.

2.2.3 Emulated Device Model

This model is not much efficient. Host based hypervisor implements this model. Host machine treats the hypervisor as an application [11] [12].

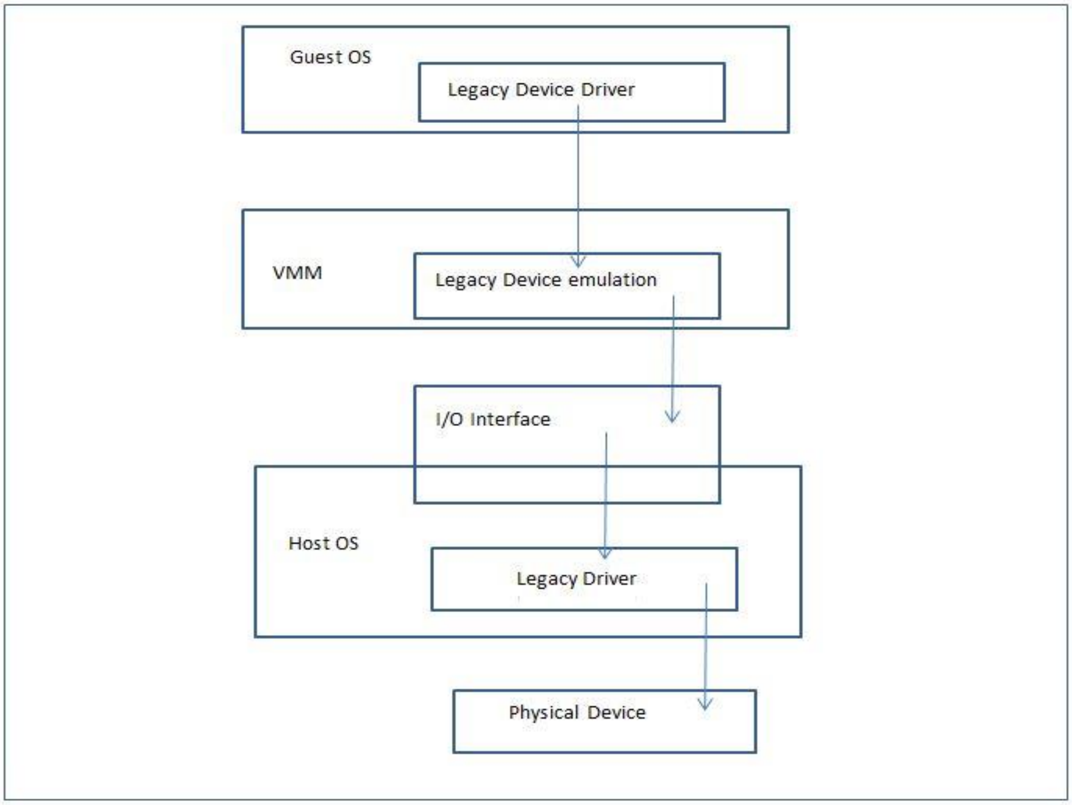


Figure 2-3 Emulated Device Model [12]

Host operating system has full control over the physical device drivers and hypervisor uses the service of host operating system for I/O operations. When an application has to perform some I/O operation it sends a trap into hypervisor which in turn passes this request to the host operating system by invoking a system call on host operating system. Host operating system initiates the required I/O operation. In this technique guest operating system is not modified.

But there are so many context switches in this approach which needs some optimization. Figure 2.3 shows this model pictorially.

2.2.4 Direct Device Assignment Model

In this model hypervisor is not a part of I/O operation. Hypervisor doesn't control the I/O devices [11] [12]. Researchers noticed the overhead caused by the involvement of hypervisor in the I/O operations of guest VM. So to increase the efficiency of guest VM's I/O operations intervention of VMM was removed from the I/O operations of guest VM. Implementation of this approach puts some challenges which are listed as below:

1. Data Transfer between device and guest is done with DMA so as to increase the efficiency of I/O operations. DMA poses a challenge in this model. DMA capable devices needs physical address of the physical memory but guest has no idea of which physical memory is allocated to it. Guest VM only knows about pseudo physical address which is generated by guest operating system after page table lookup. Only Hypervisor can translate this pseudo physical address into actual machine physical address but hypervisor is not a part of I/O operations in this model. So if guest VM is allowed to use DMA capable device directly it can access the memory of other guest VMs because there is no protection check on the memory access by DMA capable devices. In this way Isolation property gets violated so this is a challenge to maintain Isolation property without involving hypervisor or Driver Domain in the I/O operations.
2. Second problem is related to scalability i.e. no. of guest virtual machines supported to perform I/O operations. This constraint is posed by the no. of physical device present on the computer system.
3. Third challenge is related to the Interrupts. Interrupts are important part of DMA operations. Guest VM sends interrupt request when it has to do I/O operation. On completion of I/O operation physical device sends a notification to the guest virtual machine via Interrupt. So physical device needs to know which guest VM to send the notification to. The virtualization of interrupts is done by Hypervisor.

But hypervisor is no longer part of I/O operations. So virtualization of Interrupts without involving the hypervisor in the I/O operations is a challenge.

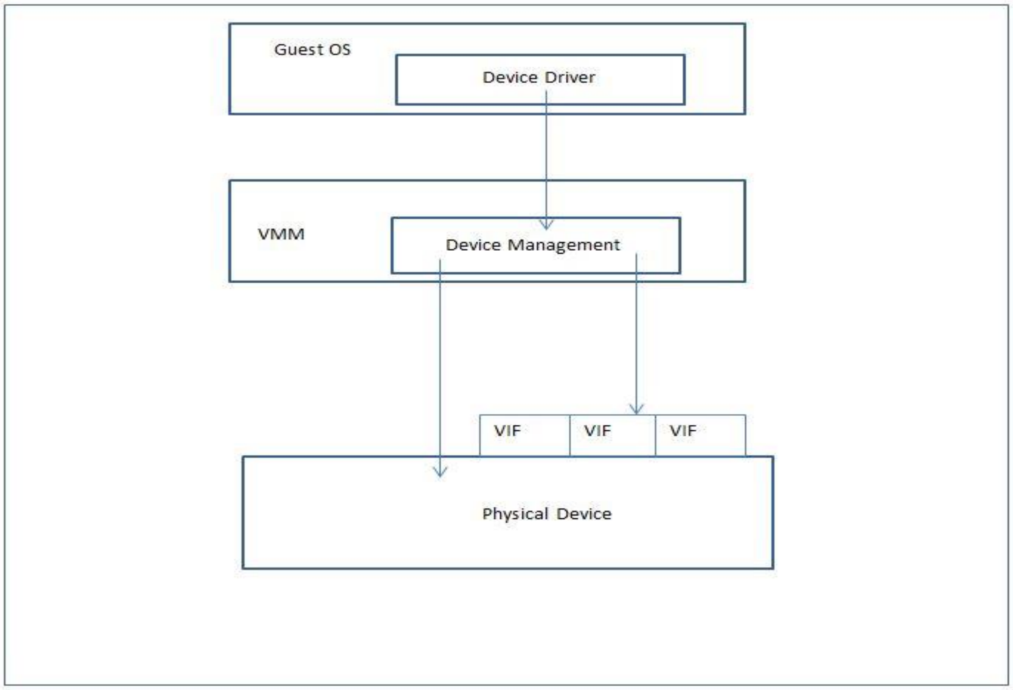


Figure 2-4 Direct Device Assignment Model [12]

So there are two main problems to be solved in this model- 1. Isolation: means guest virtual machine must not access the physical memory which is not allocated to it, 2. sharing of physical I/O device by the guest operating systems.

This approach can be implemented efficiently and safely with the help of some hardware component in the I/O access path. Figure 2.4 shows this model pictorially.

Following hardware components can be used:

1. IOMMU: Input Output Memory Management Unit is a hardware which provides isolation and translation feature. Both Intel and AMD have launched their IOMMUs. Intel named it as VT-d and AMD call it as IOMMU.
2. Network Interface cards with Single Root Input Output Virtualization and Multi Root Input Output Virtualization.

This model works fine from the efficiency point of view because VMM and Driver domains involvement is removed from I/O operations. But there are some disadvantages also of this model which are listed below:

1. It can support only a few virtual machines. No of guest virtual machines supported is limited by the available units of hardware. Therefore scalability is the issue.
2. Hypervisor cannot manage the network traffic generated by guest virtual machines.
3. Live migration of VM is a challenging problem in this model.

2.3 Virtual Machine Monitor aka Hypervisor

A hypervisor or VMM is a software layer which is responsible for enabling the multiple VMs to run on single physical machine [10] [13]. It is the responsibility of hypervisor to host and manage VMs on the same physical machine. Here host refers to the physical machine and VMM. Guest virtual machines are monitored by hypervisor that is why it is also called Virtual Machine Monitor. Hypervisors have been classified into two classes- Bare Metal or Type 1 and Hosted Hypervisor or Type 2.

2.3.1 Type 1

This is also called bare metal hypervisor because it directly sits on the physical hardware [10]. There is no operating system needed to run this type of hypervisor. Direct communication between hardware and hypervisor is the prime benefit of this type of hypervisor. Xen is an example of bare-metal hypervisor. Allocation of resources to the guest operating systems is the responsibility of this hypervisor. This type of hypervisor has small code and requires less hardware resources to run. This type of hypervisor needs hardware drivers for the target machine on which to install them. If driver is not available for any specific hardware then this type of hypervisor cannot be installed on this hardware. Currently there is only limited driver database. Some of the hypervisors of this category may need an extra virtual machine which is also called driver domain. In server virtualization this type of hypervisor is used. Data centers heavily use Type 1 hypervisor.

2.3.2 Type 2

This type of hypervisor runs as an application on some operating system [10]. This type of hypervisor is also called hosted hypervisor. KVM is an example of hosted hypervisor. It has an advantage over Type 1 hypervisor. The hardware driver issue does not exist in this type of hypervisor. This is because host operating system on which this hypervisor is installed is responsible for its interaction with hardware. But from performance point of view this type of hypervisor may be slow because of extra overhead incurred.

2.4 Xen

Xen is Type 1 hypervisor [14]. It was launched in year 2000 and now it is an established bare metal hypervisor. Para-virtualization which Xen provides makes it a well-known hypervisor. Due to Para-virtualization it provides near-native performance. With Xen virtual machines can be dynamically instantiated. Xen provides both Full Virtualization and para-virtualization. Installation of Xen hypervisor provides a specialized virtual machine which has high privilege than the other guest virtual machines [14]. This VM is used by administrator to manage the other virtual machines so called management VM or Domain 0 or Dom 0. This virtual machine has all the device drivers and hence is called Driver Domain. Dom 0 is automatically started by Xen hypervisor. But now Xen uses the hardware support (Virtualization extension in processor) provided by x86 architecture and supports full virtualization too. This makes it possible to run guest OS as virtual machine without any modification to the kernel of guest virtual machine. Figure 2.6 shows this model pictorially.

2.5 KVM (Kernel-based Virtual Machine)

In the past few years a new hypervisor has gain popularity which is known as KVM (Kernel-Based Virtual Machine) [15]. It was released in January 2007. It is not a standalone hypervisor instead added as a module to the linux kernel.

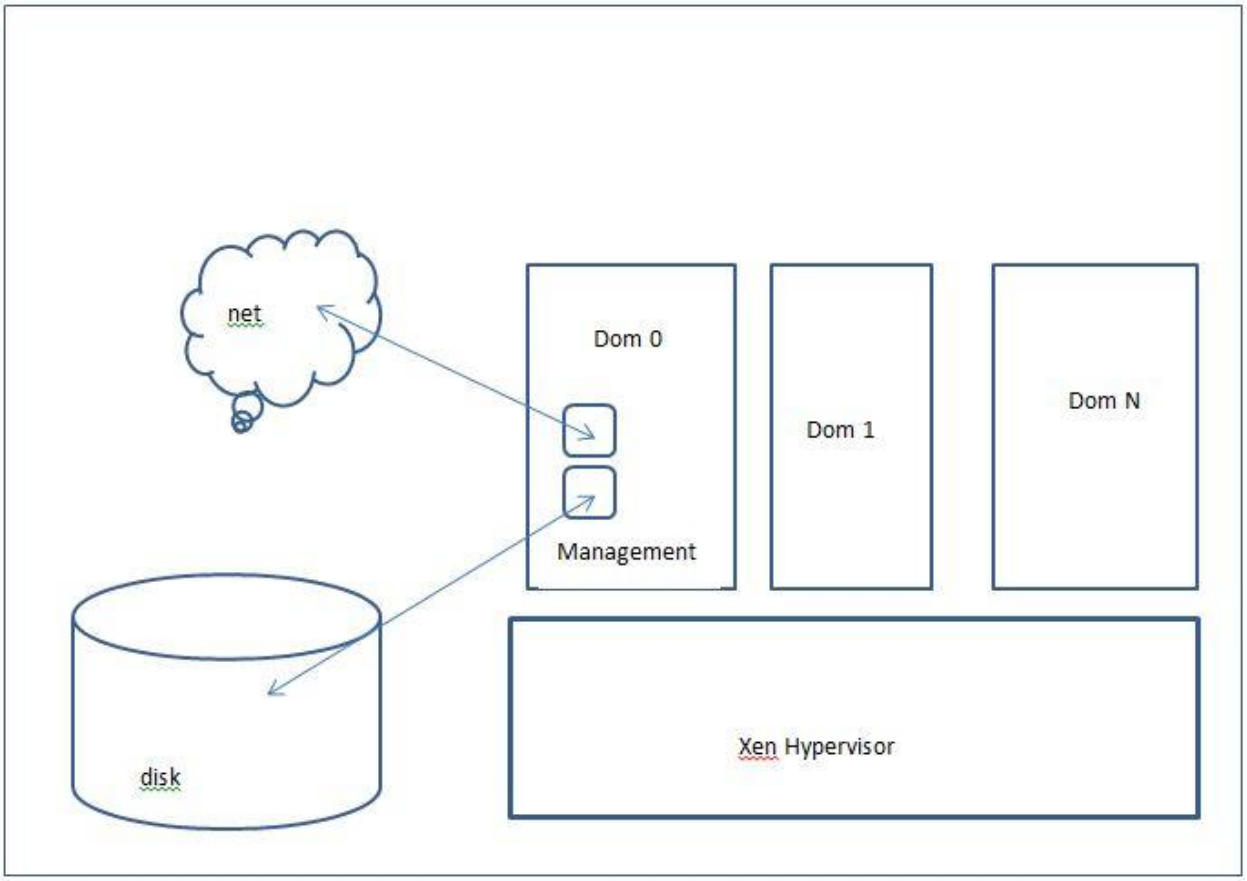


Figure 2-5 Virtualization under Xen [14]

Popularity wise it has not reached up to a status which Xen has but it has some attractive features which make it different from Xen. Upstream linux kernel 2.6.20 comes with support for both host and guest VMs. Kernel becomes the bare-metal hypervisor on loading the kernel module for KVM hypervisor. KVM is optimized to use the virtualization extension provided by x86 hardware. This is because Hardware assisted virtualization extension was developed and released before the design work of KVM was started. So KVM was designed keeping the “Hardware assisted virtualization extension” in mind. This is reason that KVM hypervisor requires Intel virtualization technology or AMD-V (Advanced Micro Devices Virtualization) to be enabled. Figure 2.7 shows this model pictorially.

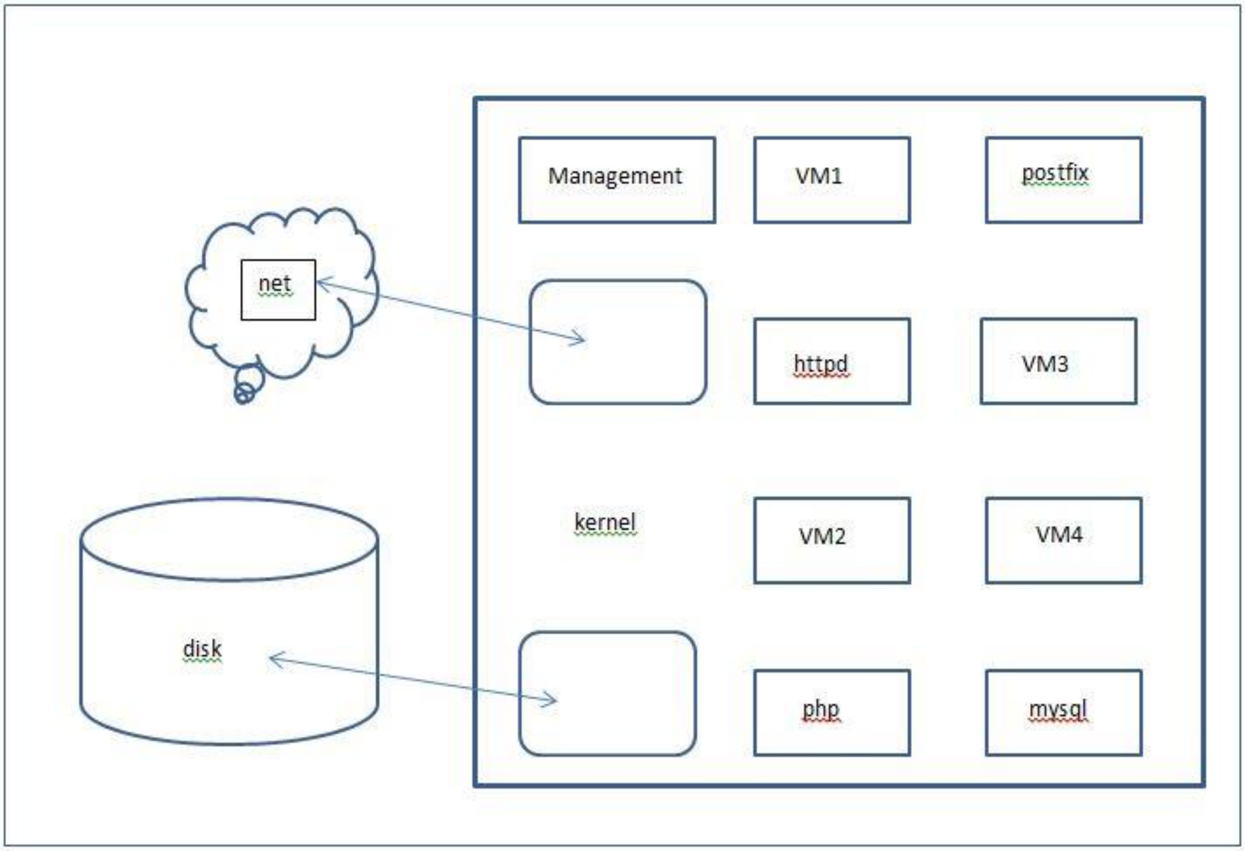


Figure 2-6 Virtualization under KVM [15]

Being a loadable kernel module KVM can leverage many advantages of Linux kernel's other modules. KVM didn't have to implement many features which are already present in the Linux kernel. For example instead of building the memory manager and process scheduler from the scratch KVM can use the memory manager and scheduler which are already present in the Linux kernel.

Update process of KVM is very simpler as compared to that of Xen because KVM is loadable kernel module. In case of Xen both hypervisor and Dom 0 needs to be updated because of split architecture of Xen hypervisor. Apart from that any changes to the upstream kernel are required to be back-ported to the Dom 0 of hypervisor. From the host machines perspective all virtual machines of KVM run as Linux process. These processes are managed by the kernel the same way as the other standard processes in the system. For the purpose of emulation of various hardware KVM uses the QEMU (Quick emulator). With Quick emulator, various physical hardware like Network card. PCI (Peripheral Component Interconnect) or disk controllers can be emulated.

Para-Virtualization can be done with KVM by installing an extra drivers which come with VirtIO. VirtIO is installed in guest operating system[16] [17]. VirtIO is a stanadard library which can be installed on linux like any other software package. VirtIO enables transfer of virtual machines across different hypervisors provided these hypervisors have support for VirtIO. The I/O operation in case of VirtIO are very fast as compared to those in case of emulation model (e.g. qemu). VirtIO can be installed on both Windows and Linux. Linux kernel 2.6.20 provides support for VirtIO.

2.6 Direct Memory Access

A Direct Memory Access is an operation in which data is transported (copied) from one hardware resource to another hardware device in a computer system without the involvement of CPU. The execution of Data copy operation is handled by a hardware unit which is called Direct Memory Access Controller (DMAC). The copy operation is executed between a source hardware and destination hardware. Hardware may be any of the I/O device or memory. Source and destination may be anything i.e. I/O device or memory. Direct Memory Access Controller does the work of CPU in case of I/O operations to and from memory. CPU does this work with programmed I/O which consumes a lot of CPU time. Figure 2.8 represents the block diagram of DMA.

2.7 Direct Memory Access Attack

Although good from throughput and CPU utilization point of view, DMA makes a system vulnerable to attack. DMA can bypass any protection provided by the OS or processor [18]. So a malicious DMA capable device can access any area of system memory [18] [19] [20]. This memory area can be read or written by the DMA capable device which may be illegal. This way data from memory can be stolen. A malicious code can be executed on the system. Figure 2.4 shows the Block Diagram of DMA.

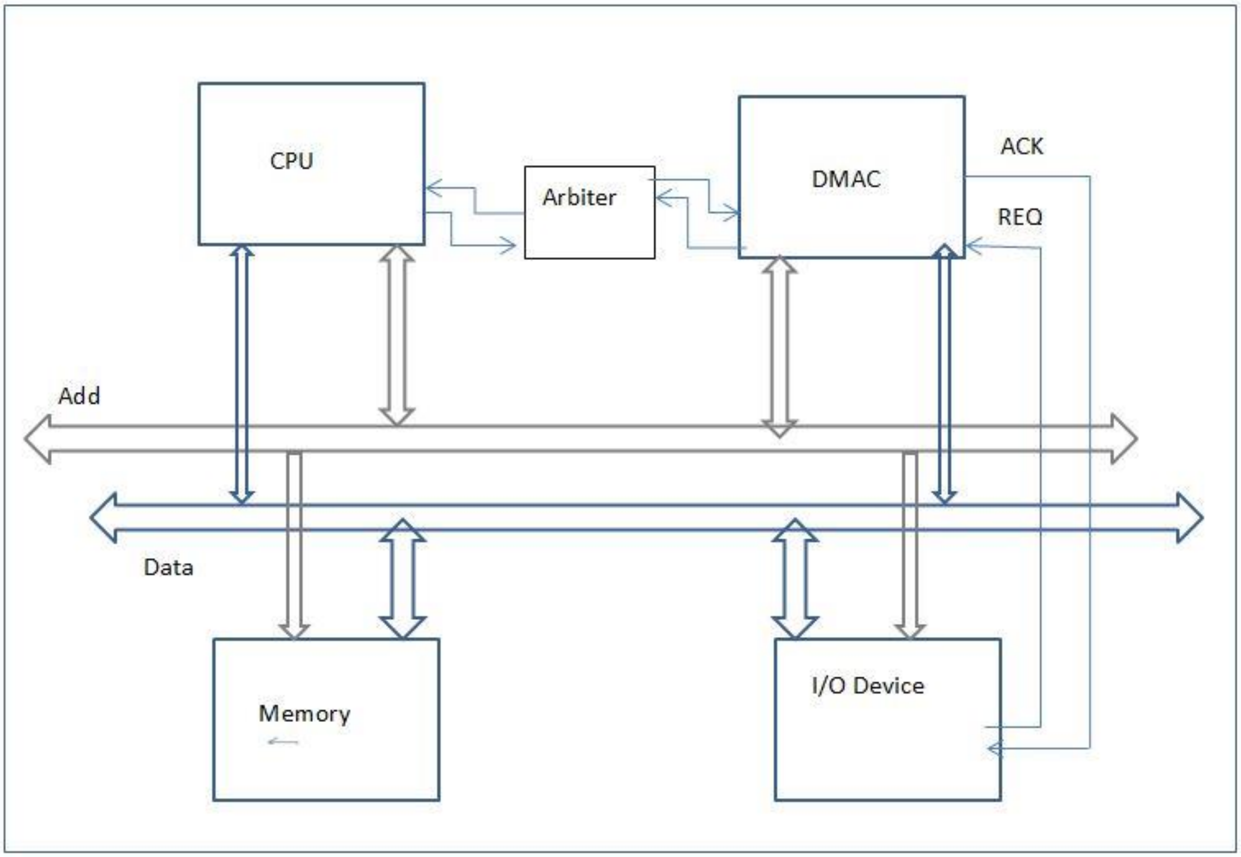


Figure 2-7 DMA Block Diagram

2.8 Insider Attacks in Virtualized Environment

The following sub-sections describe the ways in which insider attacks can be launched in a virtualized environment.

2.8.1 Insider Attacks with Malicious DMA Usage

In virtualized environment a malicious user can use a DMA capable device to read or write the memory which is not actually assigned to it [18] [19]. In virtualization environment many VMs run on the same hardware as shown in figure 2.9. They share the processor time and physical memory. The memory allocated to the VMs may not be contiguous. A user using any VM can launch DMA attack to access the memory which is allocated to some other VM. This way he can read the memory or write other VMs data.

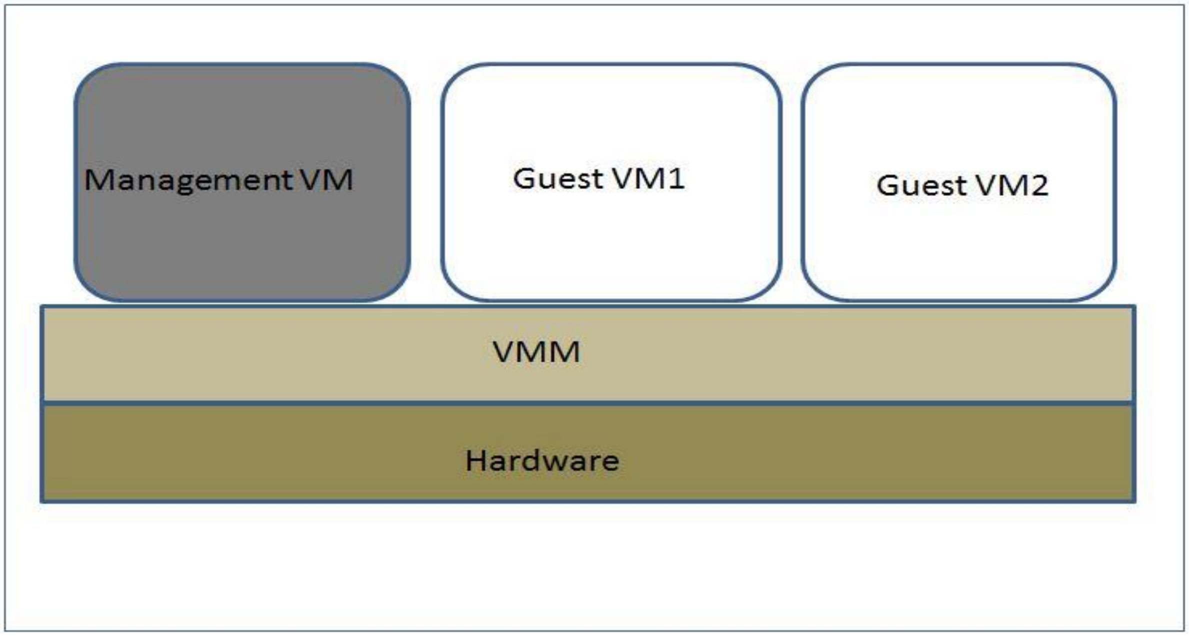


Figure 2-8 Multiple VMs running on same hardware

2.8.2 Insider Attack with Exploitation of Device Drivers

In Para-virtualized environment the physical device driver are installed in management OS. Guest VM can’t directly call the functions in driver. Split driver model is used for input output [5]. The driver is split in two parts- back end driver and front end driver. Back end driver resides in guest VM while front end driver runs in driver domain or management OS. When a guest VM has to do some input output then it interacts with frontend driver which then interacts with back end driver which is in driver domain. The back end driver then interacts with actual physical driver. Figure 2.10 shows this process pictorially. Figure shows how data packets from guest VM are sent to NIC.

2.9 Related Work

This report focuses only those attacks which are launched by the administrator who uses a privileged OS to manage other VMs in the cloud. His job is to create new VMs and managing them.

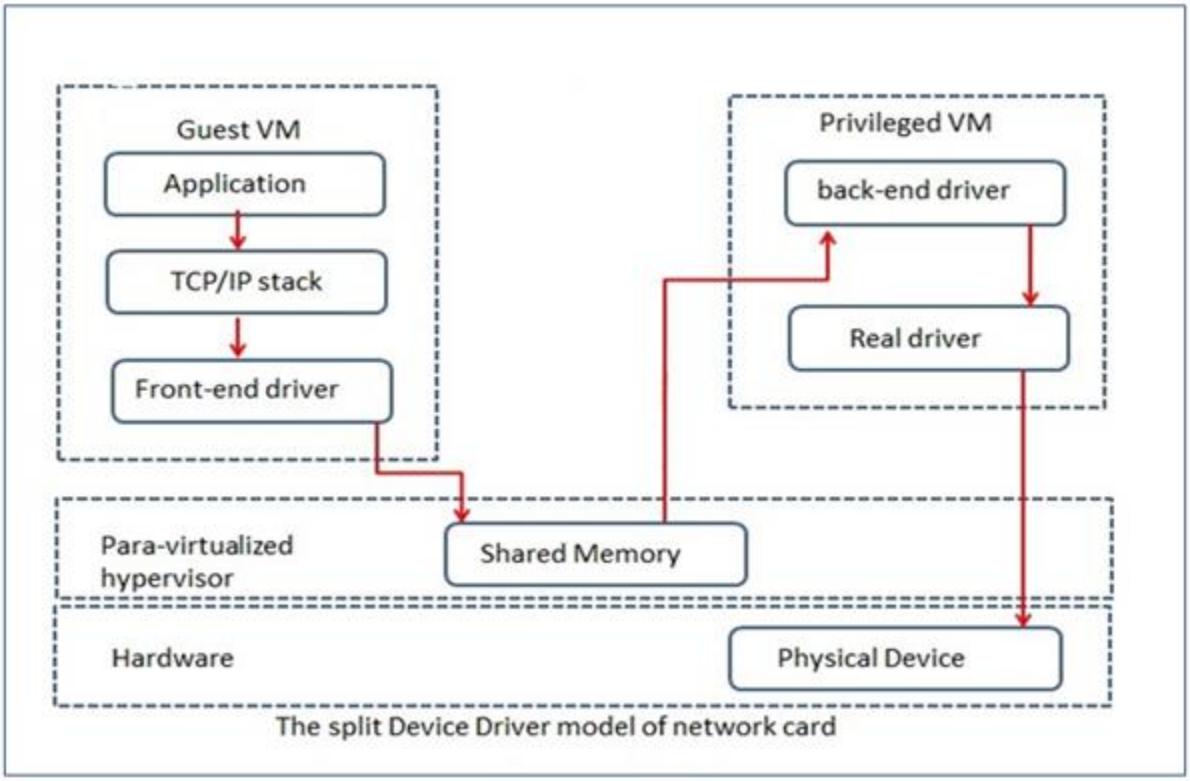


Figure 2-9 Exploitation of physical device drivers

Usually the guest VMs have to use the services provided by this privileged OS. For example in a Para-virtualized environment if a guest VM has to do some I/O operation then the guest VM cannot perform the I/O itself rather it has to make a request to the privileged OS. The privileged OS or management VM has all the device drivers. So data flow through this privileged OS. A malicious administrator who uses this OS can read or modify this data.

Apart from that a malicious administrator can launch DMA attacks on guest VMs by using malicious DMA capable I/O devices to read memory area of the guest VMs. He can also modify the memory of guest VM which may contain sensitive data. He can also put some code into the memory so that guest OS is compromised. Researchers have worked to cope such type of malicious activities.

TPM (Trusted Platform Module) is used in developing secure systems [9]. TPM is a hardware device which can be used in conjunction with peripheral bus and provides facility of secure booting but doesn't secure the data and processing in the physical memory [9]. But limitation of TPM is that it cannot be used to protect the data in memory of guest operating systems.

In [7] a hardware based solution (HyperCoffer) is proposed. It also protects the memory with memory encryption and integrity checking. This approach needs some changes in the hypervisor code. A mechanism called VM-Shim runs in between hypervisor and VM.

In [7] hardware based architecture is proposed to provide a secure virtual environment for guest VM. This approach requires some modification in the processor chip. Apart from that hypervisor code should also be modified in order to use that architecture.

In [6] Hardware Assisted SecMM was proposed which intends to mitigate insider attacks. This approach uses extra hardware to mitigate insider attacks. It provides strong isolation between guest VM and management with the help of FPGA-based (Field Programmable Gate Array based) virtualization. In this approach memory contents are always encrypted by secure memory manager and decrypted only when the VM which belongs to this memory area accesses this memory. Management VM can’t manage to decrypt the memory contents. The secure memory manager maintains two types of tables- PPT (Page Permission Table) and KT (Key Table). PPT stores information about the pages a VM is allowed to access while the KT is used to store the keys (encryption and decryption keys) which are used during encryption and decryption.

2.10 Research Gap

IOMMU based cloud infra-structure is able to mitigate Insider attacks which are initiated by Virtual Image Administrator using the management. As virtual image administrator can’t know the actual memory area of the target user. Till date there is no IOMMU based cloud service provider [5]. The proposed idea is to set up IOMMU based cloud infra-structure.

3 VIDEO STREAMING APPLICATION

Figure 3.1 shows the high-level architecture of the video streaming application which hosts videos and provides streaming with DASH (Dynamic Adaptive Streaming over Http). Users can watch these videos from their devices like smartphone, tablet, laptop or desktop.

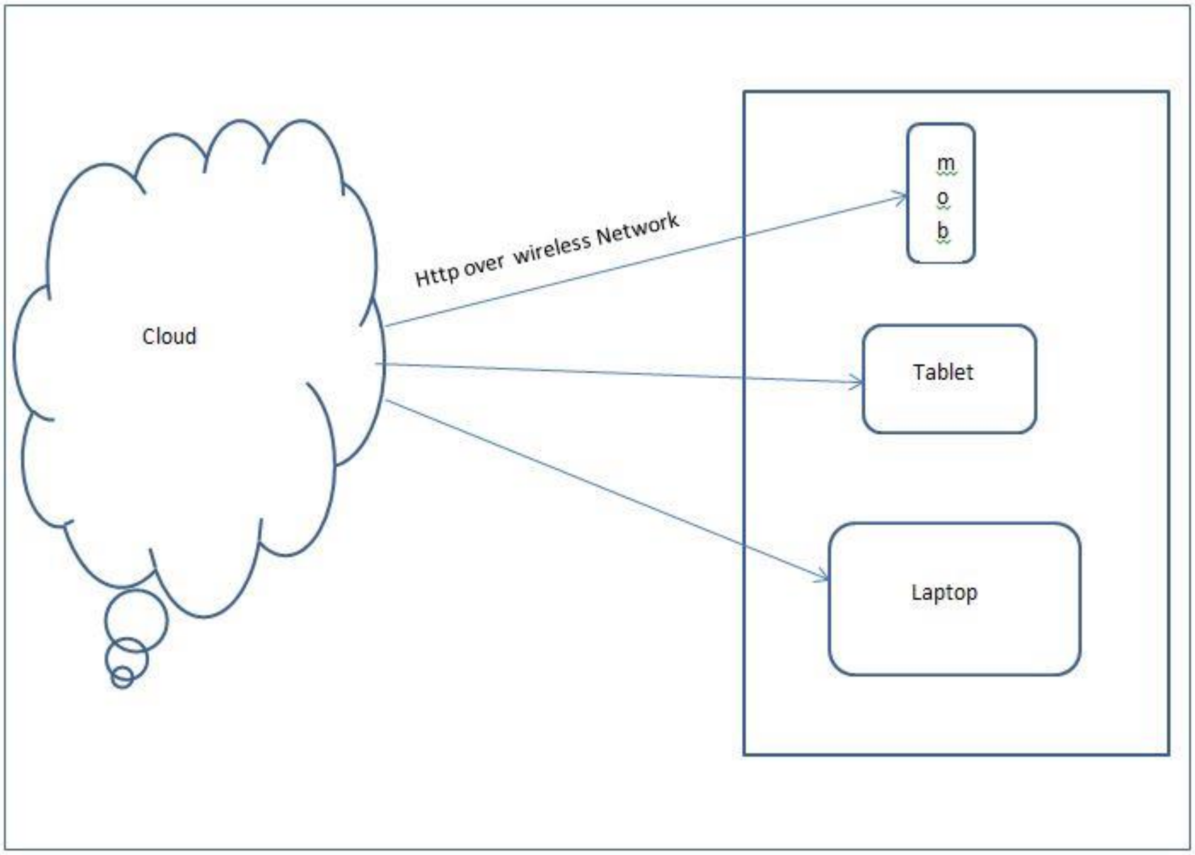


Figure 3-1 Architecture of Video Streaming Application

This is a cloud based application which uses the cloud for storage and computation. It uses DASH (Dynamic Adaptive streaming over HTTP) for streaming the videos to user's devices. Compute intensive function can be done in cloud which otherwise are not possible in mobile devices with limited computing power. The High Level Architecture of this application is shown in figure 3.1. Rest of the chapter will be spent on low level design of this application. Users can watch the contents on various digital devices provided they have Google chrome Brower installed on them. The devices need not to be of high computation power. So functions like authentication, ciphering-deciphering, encoding-decoding can be done in cloud itself.

3.1 Low Level Design of Video Streaming Application

The cloud of figure 3.1 provides many services which includes storage, streaming, authentication, transcoding, en/decryption as shown in figure 3.2. Storage for the application is provided by the cloud.

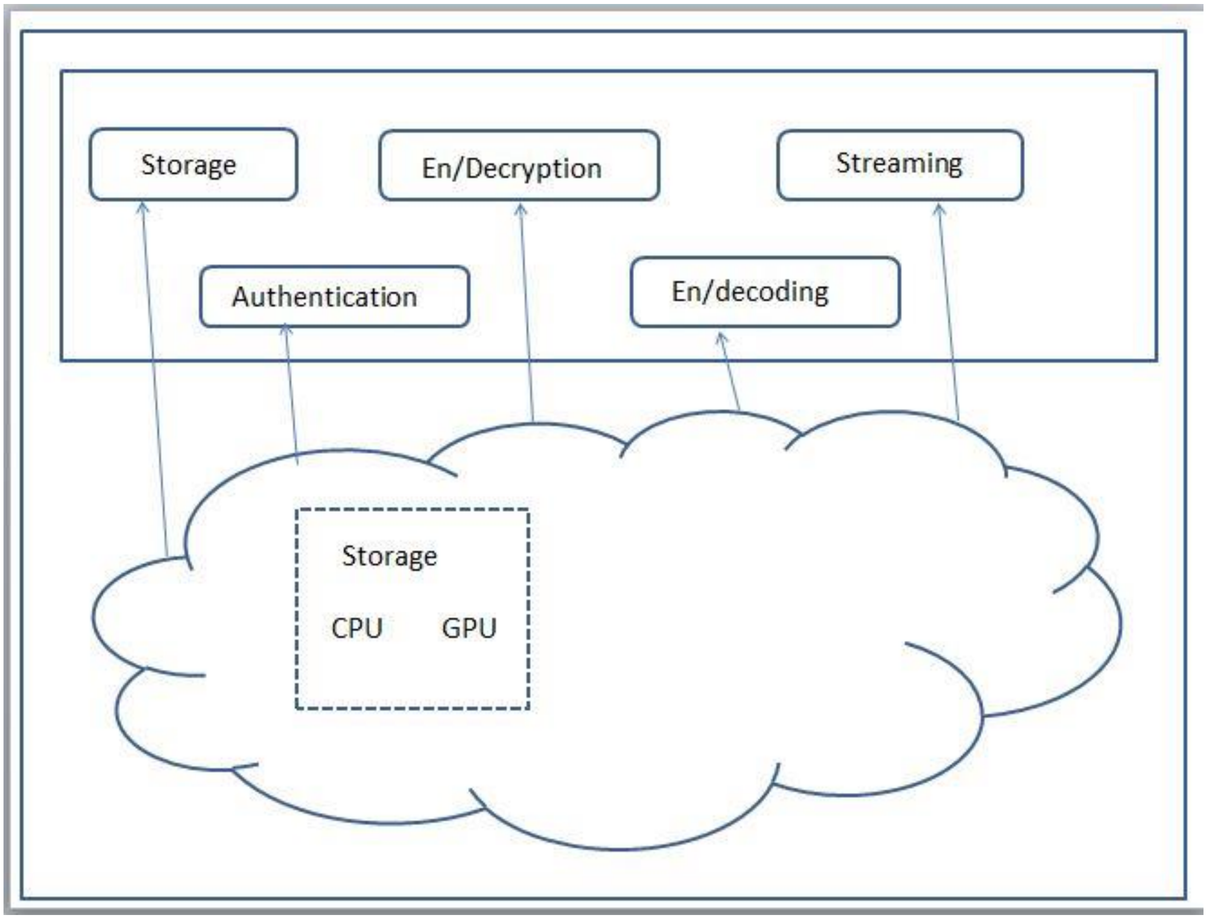


Figure 3-2 Services provided by cloud infrastructure to Video Streaming Application

Encryption service is provided by the third party algorithm running in the cloud. There are bunch of encryption algorithms available to use which may be Block Cipher or Stream Cipher. AES (Advance Encryption Standard) and DES (Data Encryption Standard) are examples of Block Cipher. Streaming service is required to serve the video contents to the users.

Without the streaming functionality a user has to download the video and then play it. Downloading may take long time if the video is big and user has to wait till the downloading of video completes. Apart from that if a user has to watch only a part of video he will have to download the complete video. So streaming service is required for efficient usage of network bandwidth. User experience is also better with streaming. Transcoding is required to convert the video from one format to another. Authentication is process of verifying that someone is really one who he claims to be.

3.2 Working of Video Streaming Application:

First a video is uploaded to the cloud which may be in any of the formats available. If uploaded video is in MPEG format it is encrypted and stored into the cloud. If video is in some other format it is first converted to the MPEG format by the transcoder and the transcoded video which is in MPEG fort is encrypted and stored in the cloud. When a user tries to watch video, the user is first authenticated. If user is a registered user then video is streamed to the users' device otherwise users is forbidden from accessing the video. The Work flow of the application is explained in the following figure:

3.3 Security issues in Video Streaming Application

The owner of the video may be worried about the data leakage which may happen in the cloud [4]. Video files may be stolen by someone who is not an authentic person in the media cloud which comes under outsider attack or videos may be stolen by the employees of the cloud service provider which is an insider attack. He/she can decrypt the video if he/she knows the key needed to decrypt the video. But insider attacks are more serious issue because video sometimes needs to be decrypted into the cloud. Decryption is required when video stored into the cloud is not an MPEG video. So the video needs to be transcoded into the MPEG format which needs actual video contents. During this transcoding decrypted video is residing into the memory so management VM can copy this decrypted video. Both insider as well as outsider attacks can cause commercial damage to the broadcasting service provider.

3.3.1 Outsider Attack

An outsider is someone who has not authorized access to the resources but uses some vulnerability in the cloud to access the resources in unauthorized manner. In my scenario the outsider can steal the video files and can play or distribute.

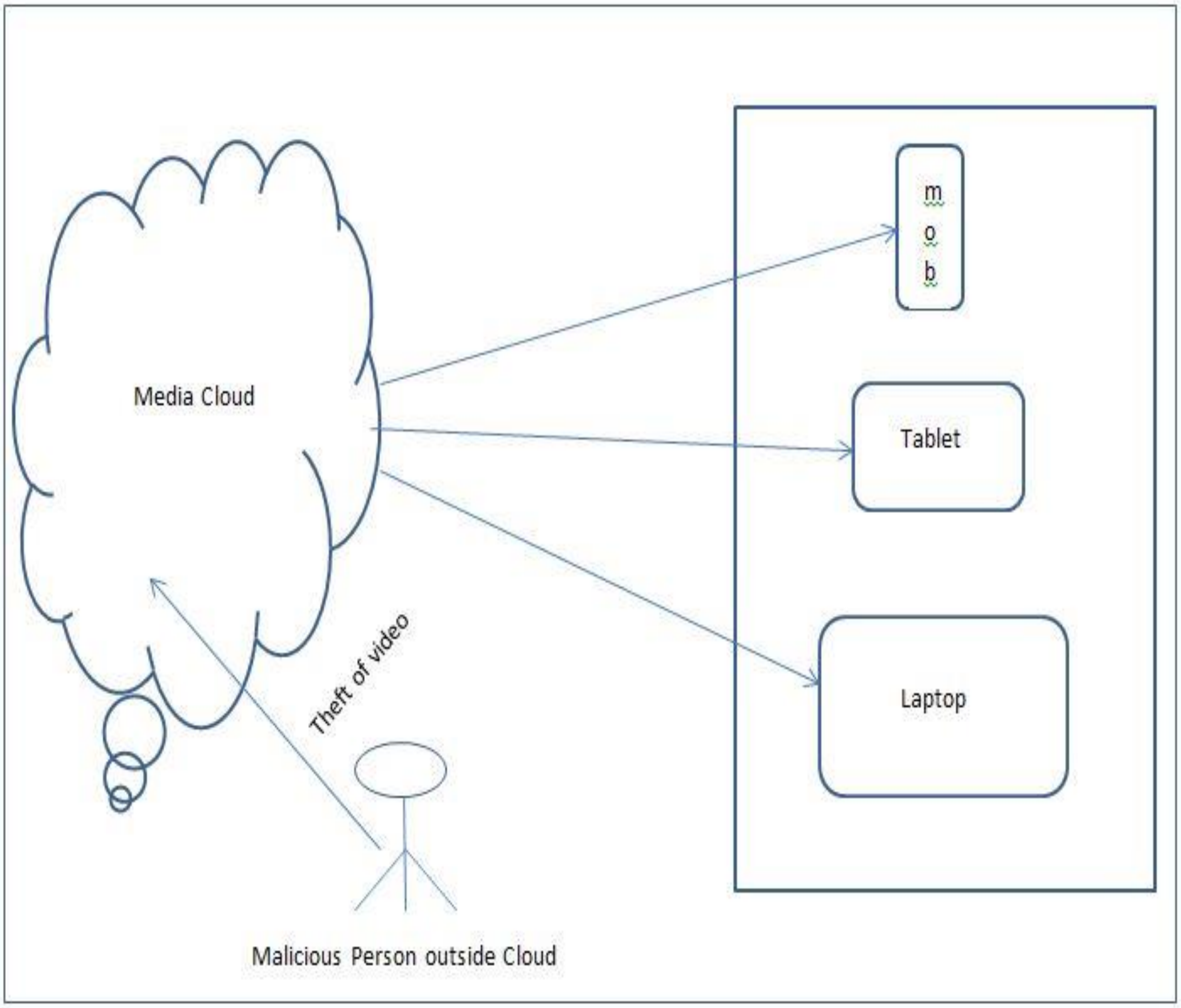


Figure 3-3 Outsider attack in Video Streaming Application

The person may be part of cloud provider organization or may not be part of organization but in outsider attack the attacker doesn't use the authorized access to target the cloud resources but used the vulnerabilities. Figure 4.1 depicts the scenarios pictorially.

3.3.2 Insider Attack

Insider attacks are the attacks which are launched by the person which are employees or former employees or contractor in the cloud service provider organization.

The insider attacker has authorized access to the organization's resources and uses this access to launch attacks for their personal benefit or to harm the organization. Figure 4.2 depicts the scenarios pictorially.

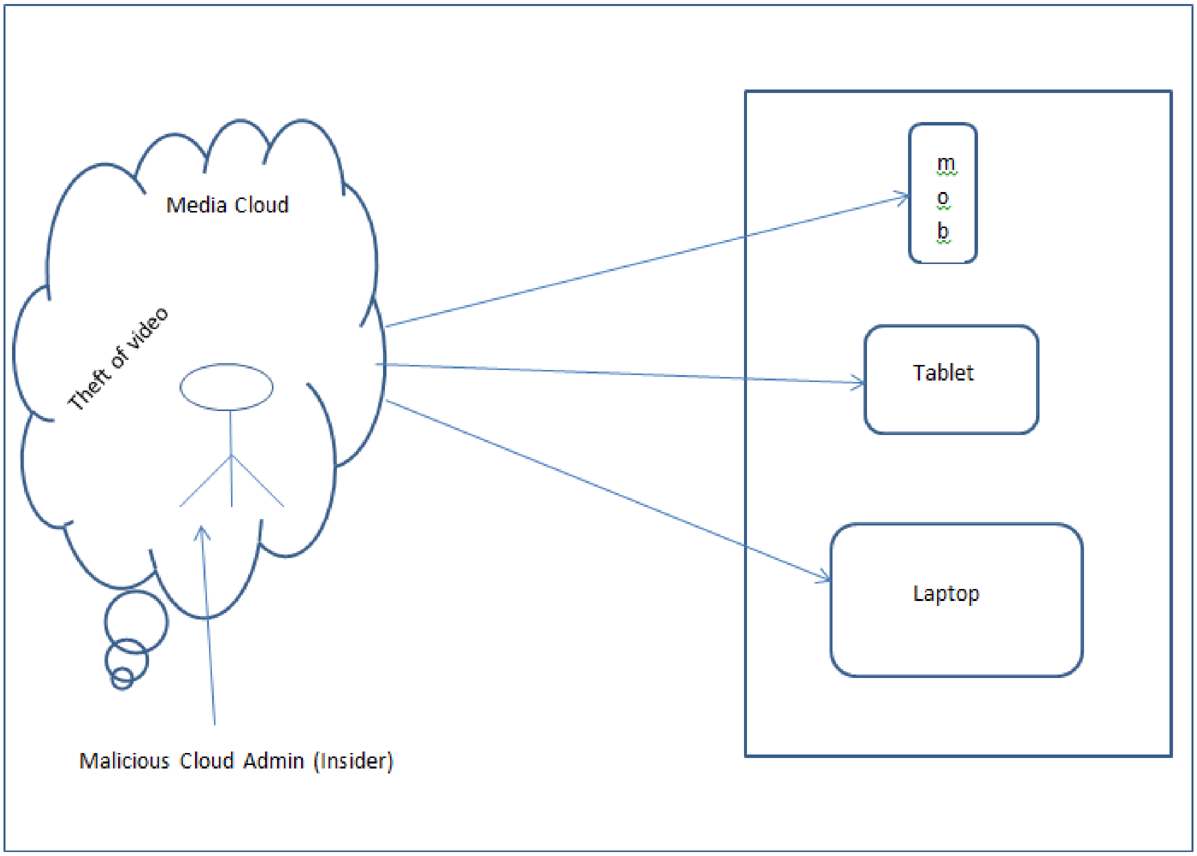


Figure 3-4 Insider attack in Video Streaming Application

In my scenario an Insider can steal the video easily if video is in plain format. Even if the video is stored in encrypted form malicious insider can manage to decrypt it with some effort. An Insider can steal the secret key form physical memory during the encryption or decryption process. Once the secret key is compromised he can use this key to decrypt the video which he has stolen by using authorized access.

3.4 How Insider attack is launched

The video streaming application uses MPEG-4 format to transmit the video to the user's mobile device. MPEG-4 is necessary to use DASH (Dynamic Adaptive Streaming over Http). If a video is not in the MPEG-4 format already then it is transcoded. Videos are stored in the cloud in encrypted form. But transcoding process cannot be done on encrypted file so the file must be decrypted. After decryption the file is transcoded and encrypted again. So during the transcoding process the secret key is loaded into memory to decrypt the video file. A malicious insider can launch DMA attack during this period and can get the secret key or decrypted video. In both cases he can get the video.

4 PROPOSED FRAMEWORK

4.1 Architecture

Section 3 shows that Video Streaming Application is prone to Insider as well as Outsider attacks. So a secure architecture is designed with these attacks in mind. Outsider attack i.e. theft of video by the outsider is mitigated by storing the video in encrypted form in the cloud. For encryption AES (Advanced Encryption Standard) is used.

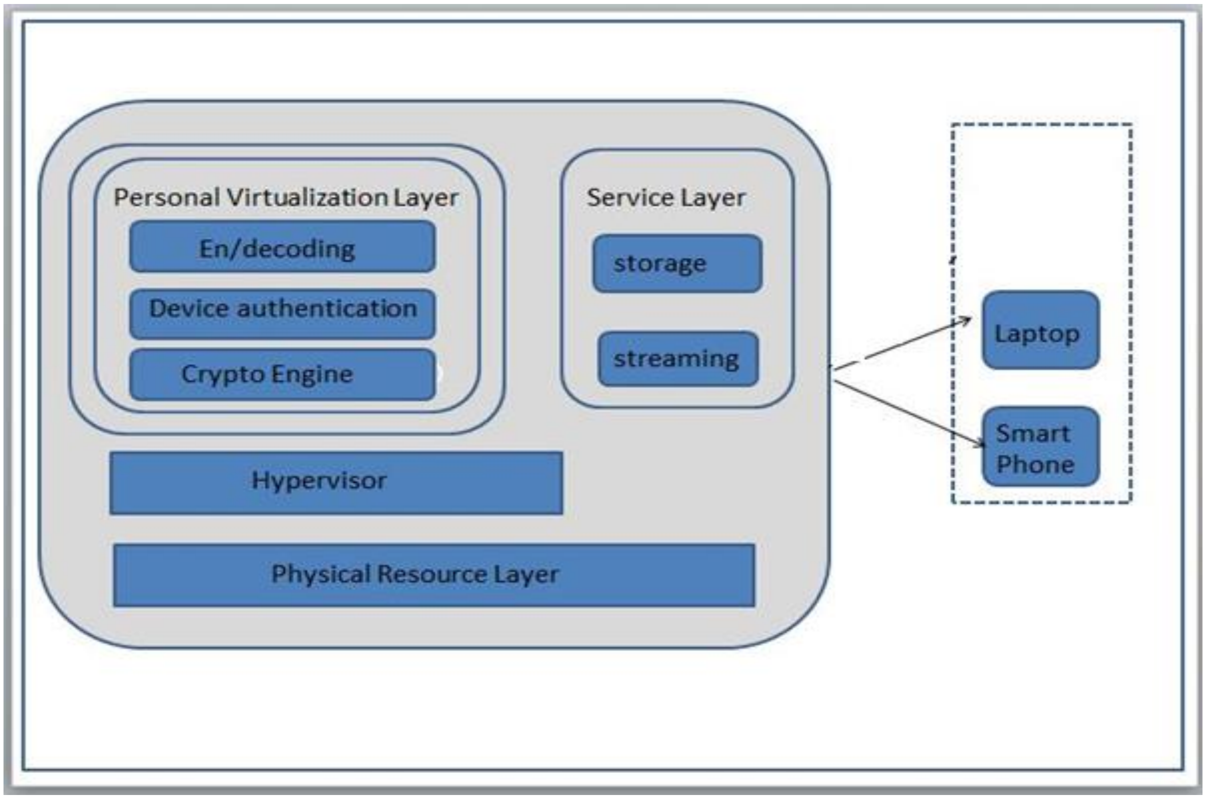


Figure 4-1 Secure Video Streaming Application

If an outsider manages to compromise the cloud infra-structure and steal the video he cannot watch the video as he requires the Encryption key which was being used during encryption to encrypt the video. To mitigate the Insider attacks, the virtualization layer of cloud is split into two different parts. One layer is general cloud and other is IOMMU based. With IOMMU in place it is very difficult for an administrator using the privileged OS to know the actual memory space of a user. An IOMMU provides Isolation and protection of memory []. It can remap the DMA addresses and Interrupts. This chapter gives the architecture of secure video streaming with IOMMU. Figure 4.1 shows the architecture of secure video streaming. The architecture consists of following components.

4.1.1 Physical Resource Layer

Physical Resource Layer composed of storage and compute resources into the cloud. These are described as follows:

4.1.1.1 Hardware

Hardware Provides efficiency through managing schedules of physical hardware such as CPU, Memory, etc. needed for data computing.

4.1.1.2 Storage

Storage can reduce costs by raising the use rate of storage resources and provide easy scalability and availability.

4.1.2 Hypervisor Based on IOMMU

This hypervisor is used to mitigate insider attacks. The IOMMU [5] [6] [23] connects the main memory and Input /Output buses of the Devices through DMA remapping. This scheme transfers/manages the memory address for the guest operating system (OS) a client that uses virtualized resources of the cloud that can directly access the unique memory address [8].

4.1.3 Personal Virtualization Layer

This is IOMMU based cloud. This is used to perform operations like authentication, transcoding etc. which require higher layer of security. For transcoding an open source java API called JAVE (Java Audio Video Encoding) is used.

4.1.4 Service Layer

This is general cloud which is used to perform operations like storage and streaming etc. which requires less secure computation.

4.1.5 Encryption algorithm

Any third party encryption algorithm can be used in the framework. Currently AES encryption algorithm is used to encrypt the video contents.

4.2 Security Analysis

4.2.1 Outsider Attack

The report focuses on outsider attack where theft of video can take place. To mitigate outsider attacks video files are stored after encryption. So if someone manages to steal to video file from the storage area in the cloud then he will get only encrypted file i.e. he can’t play the video file because to play the video file encryption key is required which was used in encrypting this video.

4.2.2 Insider attack

In insider attack a malicious administrator was launching the attack. Administrator was stealing the video while it was in plain (decrypted form) form during the transcoding process. This was possible in two ways- 1 by using a maliciously programmed DMA capable device which could access any area of physical memory. This way he can access either the full video or just encryption key which is used in encryption and decryption of video. Transcoding operation is performed in an IOMMU based cloud so copy or write operation at an arbitrary location by a DMA capable device is not possible. So with the use of IOMMU it is not possible for a malicious administrator to copy either encryption key or video itself.

4.2.3 How IOMMU Provides Strong Isolation among VMs

4.2.3.1 What is IOMMU

An IOMMU is a hardware unit which provides two main functions- Translation and Protection of memory [17]. IOMMU connects the main memory to IO devices.

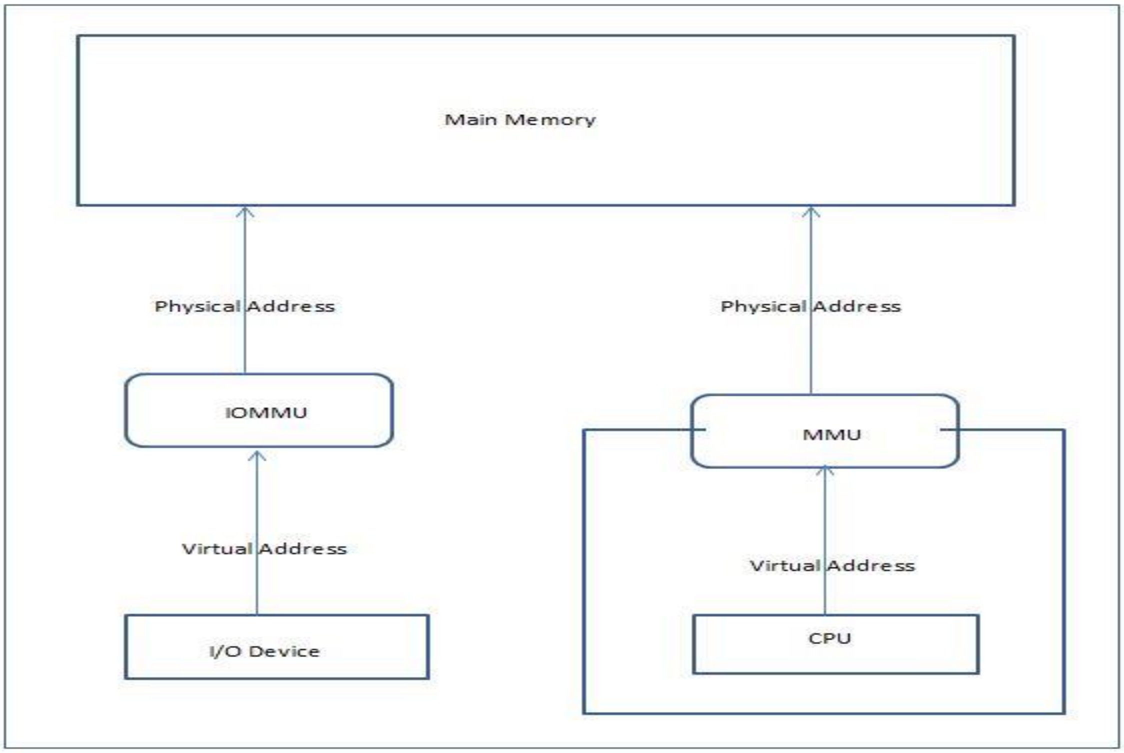


Figure 4-2 IOMMU in Computer System

An IOMMU translates the device addresses (The address generated by the I/O physical devices during DMA operations) into physical addresses which can be used to read or write data from and to a physical memory location. Figure 6.1 shows interaction of IOMMU with

other components of the computer. There are many Implementations of the IOMMUs available in the market. Intel has its own IOMMU which is called VT-d (Virtualization Technology for Directed Input Output). AMD also has its own IOMMU which is called IOMMU only.

4.2.3.2 Isolation

It creates one or more than one exclusive address spaces. These address spaces are used to regulate how a DMA operation accesses memory of host physical machine. It is somewhat similar to MMU (Memory Management Unit). An MMU translates the virtual addresses generated by the CPU into physical addresses which can be used to read or write data from and to a physical memory location. An IOMMU translates the device addresses (The address generated by the I/O physical devices during DMA operations) into physical addresses which can be used to read or write data from and to a physical memory location

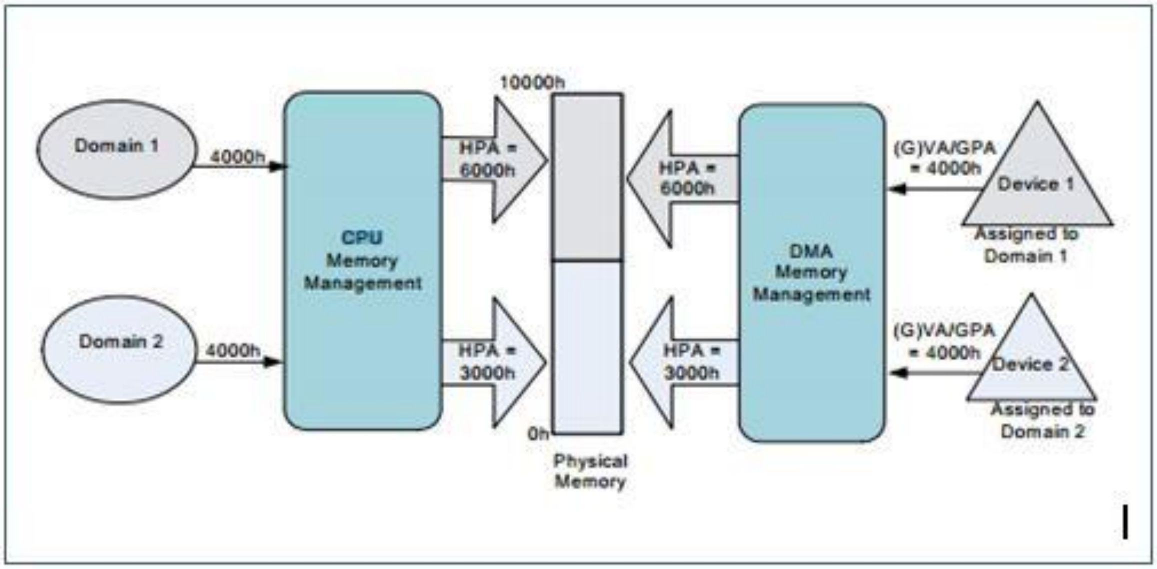


Figure 4-3 Address Translation by DMA [Source: Intel VT-d specification]

In case of virtualized environment an IOMMU can restrict the VMs to read or write to specified area of memory location. Like the operating system which maintains the page table for MMU to use in translation process, VMM (Virtual Machine Monitor) or hypervisor keeps a table for IOMMU to use in translation process. Apart from page table VMM maintains another table which is used to map an Input Output device to a page table. Every I/O device must be mentioned in this table which maps I/O device to some page table. It is not mandatory for a device to have a separate page table. So device can access only that physical address which has an entry into the page table.

5 IMPLEMENTATION AND RESULTS

5.1 Implementation

Implementation consists of two types of clouds- General Cloud and IOMMU based Cloud. The following sub-sections describe the whole picture:

5.1.1 General Cloud

For Implementation of general cloud the project uses a cloud of 15 hosts. Hosts include Xeon processors. Each host has 4 GB of RAM and 250 GB of hard disk. Implementation was done with open Nebula to build cloud. KVM hypervisor is used to virtualize the hardware resources. Some VMs have been created on these physical machines. VMs include Ubuntu VM, Cent OS VM, Open Suse VM and Windows XP VM. One of the VM is hosting apache tomcat server which is working as streaming server. The streaming server transmits the video contents using DASH (Dynamic Adaptive Streaming using HTTP). The videos are stored in a directory and are referenced from the html page. Html 5 is used to play the video on the website. Some video clips are used as test data. The videos can be streamed on any mobile device running chrome as browser. Mobil device may be having any OS like android, IOS or windows.

5.1.2 IOMMU Based Cloud

To implement the IOMMU based layer of cloud an hp machine have been used. The machine has 18GB of internal memory, 250GB of secondary storage and Intel processor. Processor has 8 cores. Each core runs at 3.4 GHz speed. Machine has VT-d which is IOMMU implementation by Intel.

For virtualization KVM (Kernel Based Virtual Machines hypervisor) is used. Para-virtualized I/O device model which is similar to Xen’s split driver model is used. For using Para-virtualized model virtio library is used. There are two VMs created on this machine. One VM uses IOMMU to secure the computation from insiders. This VM runs the transcoding and encryption processes.

For Transcoding an Open Source JAVA API called JAVE (Java Audio Video Encoding) is used. JAVE supports many audio and video formats. JAVE calls the FFMPEG which is an encoder which is going to encode the videos in to required format. FFMPEG is written in C language, So JAVE uses it as native code. For encryption of Videos AES (Advanced Encryption Standard) have been used.

5.1.3 Communication between Two Types of Clouds

General cloud uses transcoding service provided by the IOMMU based cloud. Video file is transferred from general cloud to IOMMU based cloud.

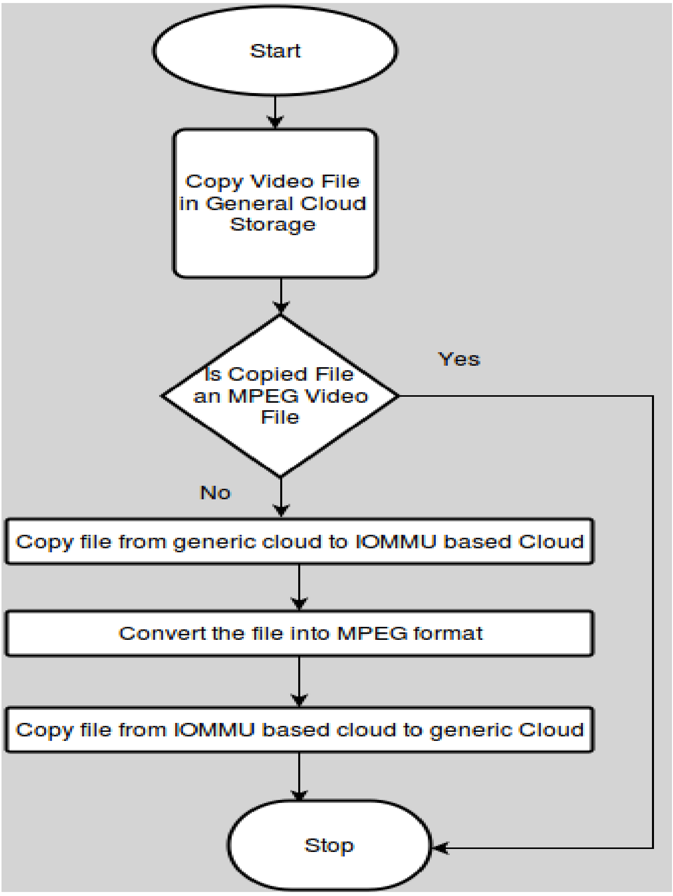

Start
Copy Video File in General Cloud Storage
Is Copied File an MPEG Video File
Yes
No
Copy file from generic cloud to IOMMU based Cloud
Convert the file into MPEG format
Copy file from IOMMU based cloud to generic Cloud
Stop

Figure 5-1 Integration Steps

After transcoding the MPEG file is again transferred to general cloud. This communication is handled with ftp and SOAP. ftp is used to transfer file between two clouds and SOAP is used to provide transcoding as web service. For ftp vsftp ftp server is used. For SOAP apache axis2 library is used on tomcat server. Flowchart in figure 5.1 shows the steps required for integration of transcoding process on IOMMU based cloud with other processes in general cloud. Figure 5.2 shows the design which can execute the steps written in flowchart of figure 5.1 and the components which are required to support this design.

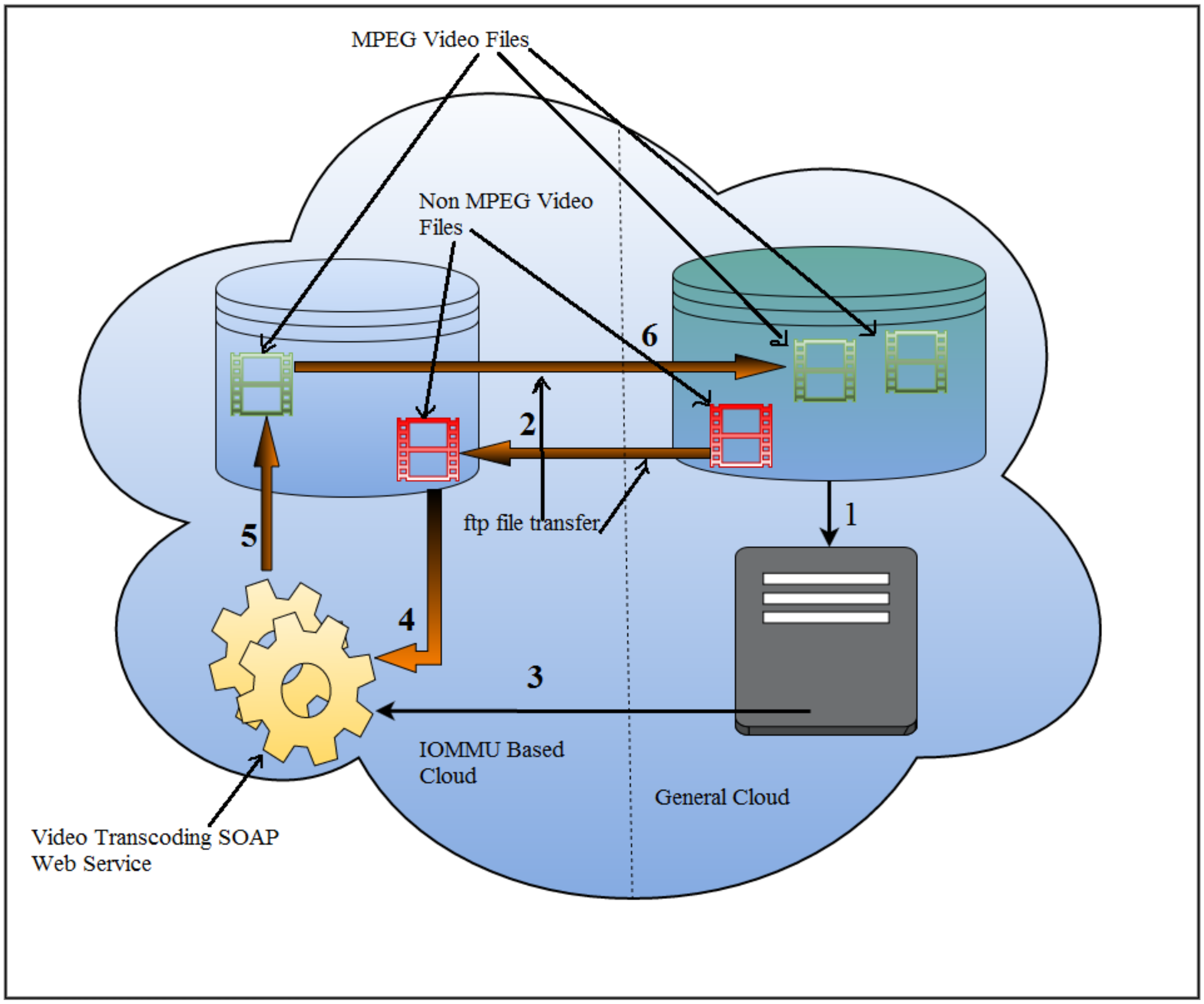


Figure 5-2 Communication Between Services in Two types of clouds

The video file is uploaded to a directory in general cloud. Then video file is transcoded if it is not an MPEG file. The transcoding process completes in following 6 steps:

1. A directory listener is used which generates an event when a new file is copied to the directory.
2. Non-MPEG file is transferred from general cloud to IOMMU based cloud using ftp protocol.

3. On completion of ftp file transfer the code running on general cloud calls the soap web-service running on IOMMU based cloud which transcodes the video and transfers the file to general cloud.

5.2 Results

IOMMU provides sufficient security from Insider attacks points of view [4]. But this extra security comes at the cost of extra hardware resources and computing time. IOMMU uses memory for tables which are used during translation process. During the memory access this table lookup takes some extra time also. Following sub-sections shows the memory usage and extra time consumed by the IOMMU based VM.

5.2.1 AES Computation Time

The graph of Figure 5.1 shows the time taken by the AES encryption algorithm on IOMMU based VM and VM without IOMMU support. It can be observed that for same file size the time taken by the encryption algorithm is more on IOMMU based VM. This extra time is the overhead in TCE Table lookups.

5.2.2 Transcoding Computation Time

The graph of Figure 5.2 shows the time taken by the Transcoding on IOMMU based VM and VM without IOMMU support. It can be observed that for same file size the time taken by the encryption algorithm is more on IOMMU based VM. This extra time is the overhead in TCE Table lookups.

5.2.3 Host Physical Machine’s Main Memory Usage

The graph in Figure 5.2 shows that an IOMMU based VM reserves the memory equal to total memory allocated to this VM all the time after the VM is started. So this VM has full control over the memory allocated to it. The memory is reserved because the pages are mapped by the hypervisor when this VM starts.

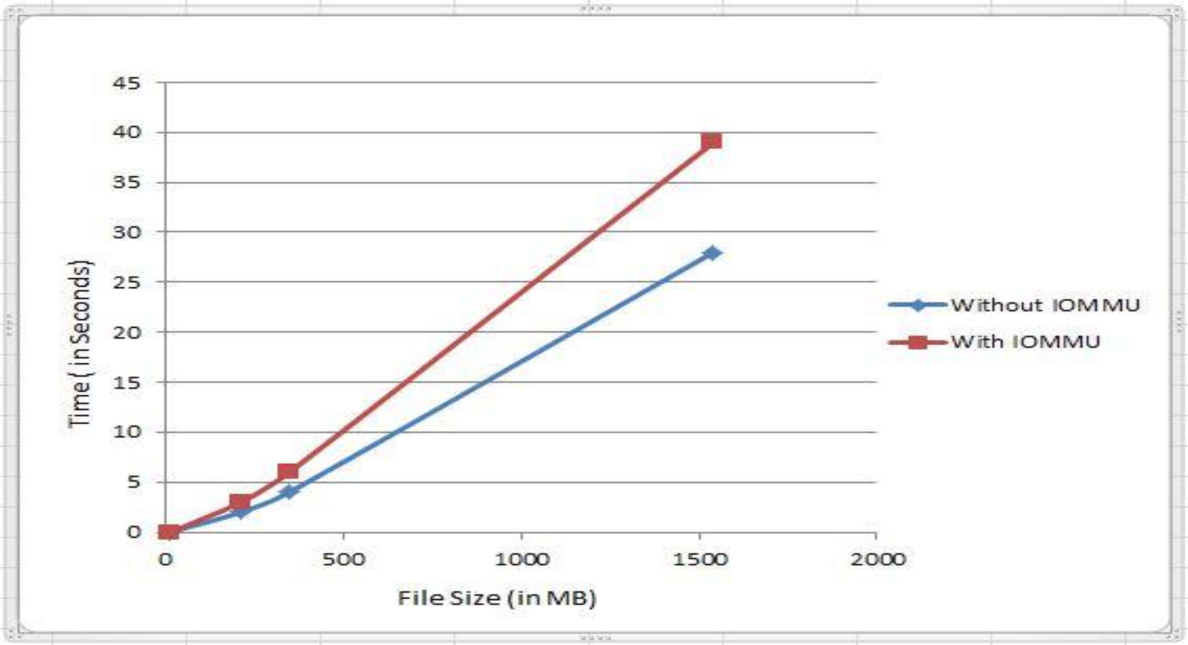


Figure 5-3 AES Computation Time on IOMMU Based VM vs. VM without IOMMU Support

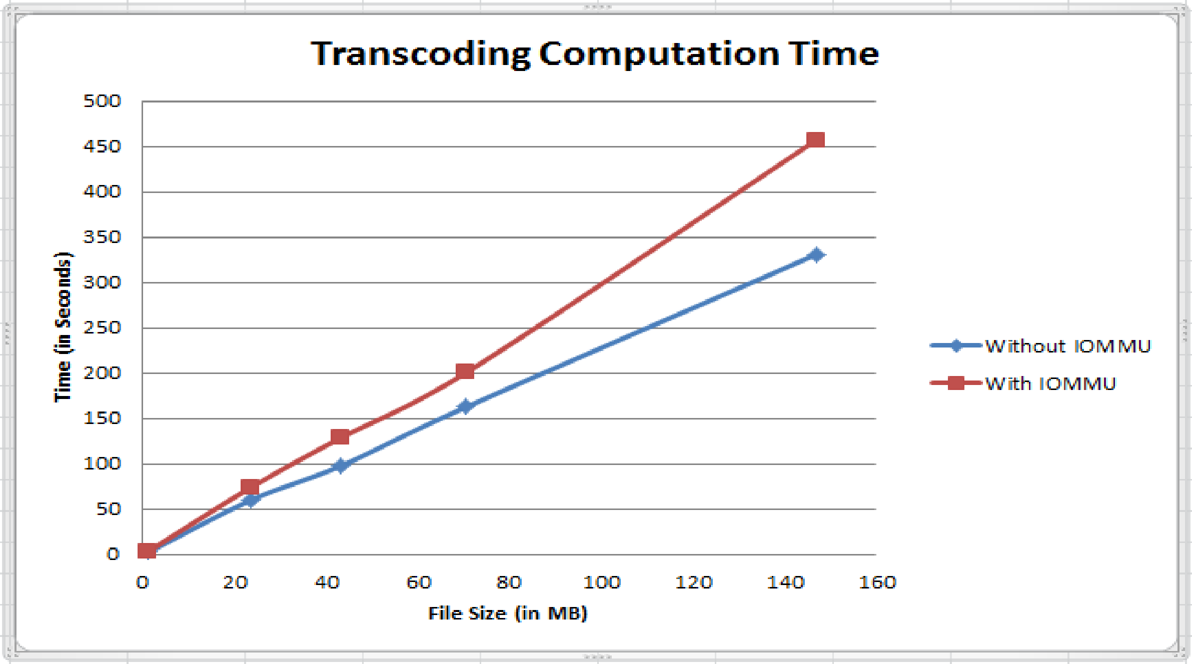
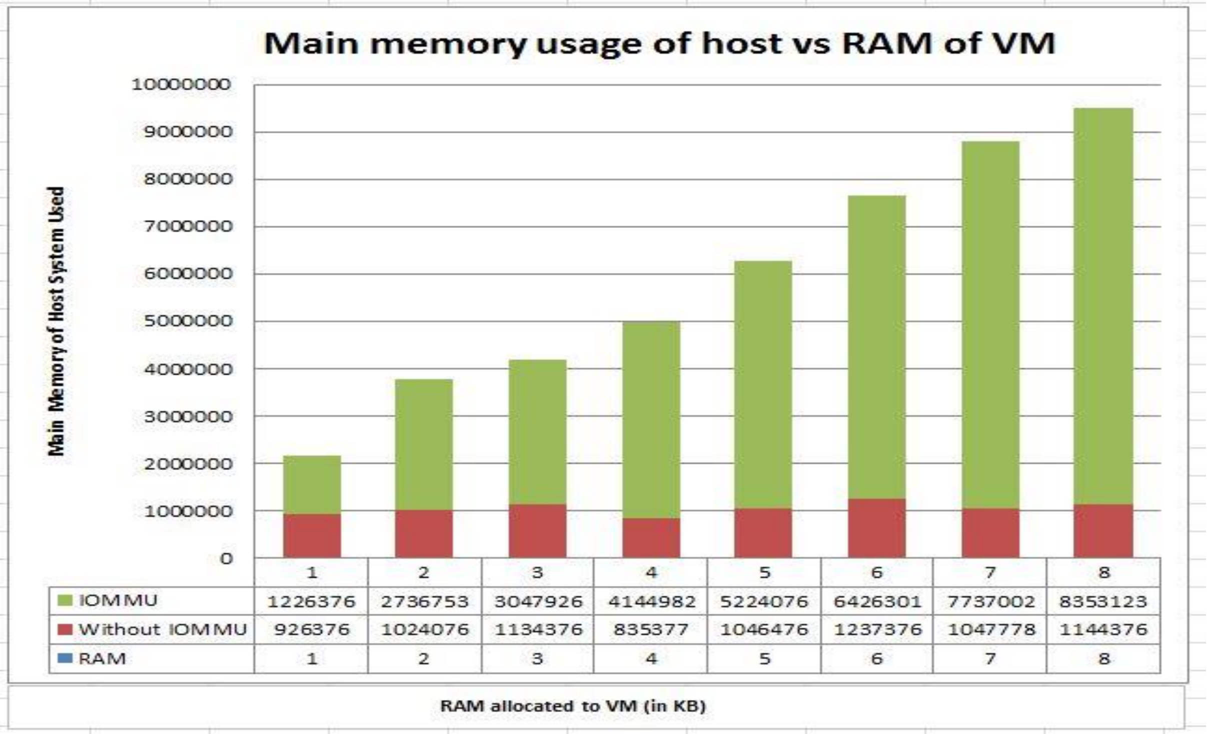


Figure 5-4 Transcoding Computation Time on IOMMU Based VM vs. VM without IOMMU Support



	1	2	3	4	5	6	7	8
🟩 IOMMU	1226376	2736753	3047926	4144982	5224076	6426301	7737002	8353123
🟥 Without IOMMU	926376	1024076	1134376	835377	1046476	1237376	1047778	1144376
🟦 RAM	1	2	3	4	5	6	7	8

Figure 5-5 Main Memory Usage of Host vs. RAM of VM

6 CONCLUSION AND FUTURE WORK

The proposed framework provides a secure computation environment for the transcoding of videos. In fact the framework can be used to protect any type of computation from the Insider attacks which are launched by administrator with malicious DMA devices or exploitation of physical device drivers. To mitigate insider attacks IOMMU based cloud has been used. The cost of IOMMU based cloud infra-structure goes high so only a portion of cloud is made IOMMU based while the other portion is made general cloud. The results presented in the report shows that IOMMU based VMs take more time than the VMs without IOMMU with same configuration to complete the same task. So this security gain is achieved at the cost of computation time. Service Layer provides services like storage and streaming. The Service layer was implemented on open nebula. AES algorithm is used to encrypt videos. The results show that computation on IOMMU based cloud is slow. This is because of the translation overhead incurred during memory access. Techniques to mitigate IOTLB bottleneck can be used in future to improve the performance of IOMMU based clouds. For Key management DRM (Digital Rights Management) can be used.

REFERENCES

[1] Gibson, J.; Rondeau, R.; Eveleigh, D.; Qing Tan, "Benefits and challenges of three cloud computing service models," in Computational Aspects of Social Networks (CASoN), 2012 Fourth International Conference on , vol., no., pp.198-205, 21-23 Nov. 2012

[2] P. K. Chouhan, S. Sezer, Y. Choi, I. Kim and C. Jung, "Secure virtualised environment," *Internet Technology and Secured Transactions (ICITST), 2014 9th International Conference for,* London, 2014, pp. 112-117.

[3] Claycomb, W.R.; Nicoll, A., "Insider Threats to Cloud Computing: Directions for New Research Challenges," in Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual , vol., no., pp.387-394, 16-20 July 2012

[4] Duncan, A.J.; Creese, S.; Goldsmith, M., "Insider Attacks in Cloud Computing," in Trust, Security and Privacy in Computing and communications (TrustCom), 2012 IEEE 11th International Conference on , vol., no., pp.857-862, 25-27 June 2012

[5] C. Li, A. Raghunathan, and N.K. Jha, “A trusted virtual machine in an untrusted management environment,” IEEE Trans. on service computing, vol. 5, no. 4, pp. 472-483, Oct. 2012.

[6] Dinesh Chandrasekaran, Hardware-Assisted Secure Memory Management, Master of Science thesis, Polytechnic Institute of NYU, 2010.

[7] L. Xu; J. Lee; S. H. Kim; Q. Zheng; S. Xu; T. Suh; W. W. Ro; W. Shi, "Architectural Protection of Application Privacy Against Software and Physical Attacks in Untrusted Cloud Environment," in IEEE Transactions on Cloud Computing , vol.PP, no.99, 2015, pp.1-1.

[8] Junggab Son; Hussain, R.; Hunmin Kim; Heekuck Oh, "SC-DVR: a secure cloud computing based framework for DVR service," in Consumer Electronics, IEEE Transactions on , vol.60, no.3, pp.368-374, Aug. 2014.

[9] Junggab Son; Hoonjung Lee; Heekuck Oh, "PVR: a novel PVR scheme for content protection," in Consumer Electronics, IEEE Transactions on , vol.57, no.1, pp.173-177, February 2011.

[10] Rinku Shah, Network I/O Virtualization: Challenges and Solution Approaches, phd seminar report, Indian Institute of Technology, Bombay, 2014.

[11] Gerald J. Popek, Formal Requirements for Virtualizable Third Generation Architectures. In Comunications of ACM Volume 17 Issue 7,1974.

[12] Binbin Zhang, Xiaolin Wang, Rongfeng Lai,Liang Yang, Yingwei Luo , Zhenlin Wang, and Xiaoming Li. A Survey on I/O Virtualization and Optimization. In Chinagrid, 2010.

[13] R. Adair, R. Bayles, L. Comeau, and R. Creasy, “A virtual machine system for the 360/40,” 1966.

[14] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauery, Ian Pratt, and Andrew Wareld. Xen and the Art of Virtualization. In SOSP, 2003.

[15] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori. kvm: the Linux Virtual Machine Monitor. In OLS, 2007.

[16] Rusty Russell. virtio: Towards a De-Facto Standard For Virtual I/O Devices. In ACM SIGOPS Operating Systems Review - Research and developments in the Linux kernel archive, Volume 42 Issue 5, July 2008, Pages 95-103.

[17] M. Tim Jones. Virtio: An I/O virtualization framework for Linux. In IBM developer Works, January 2010.

[18] M. Ben-Yehuda, J. Mason, O. Krieger, J. Xenidis, L. Van Doorn, A. Mallick, J. Nakajima, and E. Wahlig. Utilizing IOMMUs for virtualization in Linux and Xen. In Ottawa Linux Symp. (OLS), pages 71–86, 2006.

[19] M. Ben-Yehuda, J. Xenidis, M. Ostrowski, K. Rister, A. Bruemmer, and L. van Doorn. The price of safety: Evaluating IOMMU performance. In Ottawa Linux Symp. (OLS), pages 9–20, 2007.

[20] http://www.intel.com/content/www/us/en/embedded/technology/virtualization/vt-directed-io-spec.html

[21] D. Abramson, J. Jackson, S. Muthrasanallur, G. Neiger, G. Regnier, R. Sankaran, I. Schoinas, R. Uhlig, B. Vembu, and J. Wiegert. Intel virtualization technology for directed I/O. Intel Technology Journal, 10(3):179–192, Aug 2006

[22] Patrick Stewin, and Iurii Bystrov, "Understanding DMA Malware", Security in Telecommunications-Technische Universit¨at Berlin

[23] Y. Xia, Y. Liu and H. Chen, "Architecture support for guest-transparent VM protection from untrusted hypervisor and physical attacks," High Performance Computer Architecture (HPCA2013), 2013 IEEE 19th International Symposium on, *Shenzhen*, 2013, pp. 246-257.

[24] R. Sailer, T. Jaeger, E. Valdez, R. Caceres, R. Perez, S. Berger, J.L. Griffin, and L. Doorn, "Building a MAC-based security architecture for the xen open-soruce hypervisor," in Proc. The 21st Annual Computer Security Applications Conference, pp. 276-285, Dec. 2005