THREAT MODELING OF PARA-VIRTUALIZED ENVIRONMENTS

A Thesis by

Rajeev Nagaraj

Bachelor of Engineering, BNMIT, VTU, 2008

Submitted to Department of Electrical and Computer Engineering
and faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2011

THREAT MODELING OF PARA-VIRTUALIZED ENVIRONMENTS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

_____
Ravi Pendse, Committee Chair

_____
Linda Kliment, Committee Member

_____
Abu Asaduzzaman, Committee Member

DEDICATION

I dedicate this thesis to my father, who has taught me the value of perseverance, to my mother who has taught me the value of patience, to my brother for teaching me the value of kindness, and finally to the Almighty for providing me the strength.

ACKNOWLEDGEMENTS

This journey to my Masters Degree has been a privileged one, here at Wichita State University. I would like to directly thank those people in the department who have been instrumental in contributing to my learning and knowledge experience at Wichita State University. I would like to thank my advisor, Dr Ravi Pendse, for his invaluable guidance and constant involvement in every matter throughout my journey as a graduate student. I have had a tremendous learning experience under his guidance. His experience and expertise, especially in the fields of Computer Networking and Computer Architecture have allowed him to focus on the critical aspects of any problems encountered during my thesis. It has indeed been a privilege to be able to work under his guidance, and undoubtedly, the credibility and prestige of his name will continue to open doors for me in the future, in addition to keeping me directed towards learning ahead.

I would specially like to thank Mr Amarnath Jasti, Mr Nagaraja Thanthry and Mr Vijay Ragothaman for their valuable input and support during the course of my thesis.

Further, I would like to thank the members of my thesis committee, Dr Abu Asaduzzaman and Dr Linda Kliment for their valuable feedback in backing up my research foundation.

Special thanks to all my fellow students, for everything and in helping me keep my calm throughout. My heartfelt gratitude to my family for the constant support they have given me all the way.

ABSTRACT

Organizations are looking at various cost effective methods to reduce the overall cost of data storage systems. This measure is taken essentially to reduce the hardware that is currently being used for hosting servers. In recent years, the organizations around the world have looked at various options such as parallel computing and grid computing. However, these techniques have not been implemented in organizations due to their limitations. Virtualization is a new technique that is being adopted by system administrators to overcome the hardware issues within a computer network. Virtualization has the main advantages such as secure logging and terra architecture which enhances overall performance of the server and effectively reduces the cost.

Virtualization can be broadly classified into 2 types: Full Virtualization and Para-Virtualization. As, with every new technology that comes into existence, there arise the security concerns associated with it. This thesis addresses the growing security concerns associated with Virtual Machines (VM's) in a Para-virtualized environment. Some of the most common threats are Denial of Resource Attack, Sniffing Attack, and Authentication and Authorization issues. Thus, it becomes essential to derive a threat model so that these issues are identified based on their severities and addressed more effectively with appropriate security algorithms. This thesis provides the readers an insight to modeling threats, analyzing threat parameters, deriving risk equations, and validating the results.

TABLE OF CONTENTS

Chapter Page

TABLE OF CONTENTS (cont.)

Chapter Page

LIST OF FIGURES

LIST OF FIGURES (cont.)

CHAPTER 1

INTRODUCTION

1.1 History of Storage Area networks

The modern computer age has made its way towards faster and reliable computing. This evolution has transformed the entire system of storage systems and parameters, affecting them in a new direction. Since pre-historic times the storage of information has been essential to mankind. As time evolved, mankind has realized the importance of data storage and has found newer and advanced techniques to store data. With time, the focus on storage and backing up data has increased. In this modern age of computing, the emphasis to restore and backup data is seen more profoundly [1]. In the mid 1970's, IBM came up with industry standards for data storage. This standard was not well accepted as the cost for reliable data backup was very expensive to the budding industry.

As the years passed, organizations had an upscale resurge of data usage and thus backing up data was of utmost importance. When the term “backing up” is used, the focus shifts towards copying of data and restoring it for further use in case of disaster. Everyone in the industry has been exposed to situations where they have accidentally lost work without backing it up. This eventually results in a loss of time and information, thereby heavily costing the individual and the company as a whole. Hence, securing data would be of importance to everybody.

Initially when computers came into existence, the method of backing up data was usually on magnetic tapes or punch cards. As computers evolved, newer technologies such as floppy disks and hard drives came into existence which used the same technology as the magnetic strips. In the mid 1980's and the early 1990's, these solutions to backup data were insufficient to withhold the ever-evolving computing standards and thus data backup was one of the most

researched area during this period [2]. Backing up data using a common network became a mantra for the whole industry because everybody thought data could be stored at a common place and reduce the entire cost of the system. This revolution for data storage and reduction of access time gave way to a new technology called Storage Area Networks.

During the 1970's, organizations came up with a dedicated and centralized management system to control the operation of storage elements. These centralized management systems were commonly referred to as Controllers and would essentially do the brain work for all hard-drives attached to the processors. However, as the industries grew, Controllers were not an efficient solution to manage hard-drives. This became a limitation on the entire network since the Controllers acted as the bottle-neck between the network and the storage devices. In the 1980's, focus shifted from the Controllers towards an efficient system of management of data which involved a server/client model. In this model, issues were resolved by distributing data among the servers instead of Controllers. The clients would connect to these servers to operate on the storage devices. This was an efficient solution. However, the management of these clients/servers was much harder and posed other issues to the storage devices.

In the early 1990's, research was completed and a new model was developed to reduce the management access issue that the server/client model posed. In this model, the network was divided into two parts, the Network Segment and the Storage Segment. The Network Segment essentially contained all the clients which would access the servers through a LAN/WAN network. The information processed by these servers was then stored on the storage devices which would be the data banks for the processed information. This model proved to be much more efficient than the previous model since it provided the network administrators with the flexibility of globalizing the data stored on the storage devices. The other important feature of

this model was the access to the management server which would control the rest of the devices, thus providing a broader management perspective to the network. This technology was further developed into Storage Area Networks.

The Storage Area Networks cannot be defined by a single definition. This is due to the fact that every individual using it has their own definition. In general, Storage Area Networks (SAN) can be defined as networks dedicated for the use of mass storage and data backup. The storage components can be disk arrays and servers on remote sites. The hosts connect to these remote sites and perform read/write operations over a common network. This entire system could be summed up as a SAN. In other words, Storage Area Networking can also be defined as a mass storage solution for data backup in enterprise networks. This is designed to provide faster, reliable and scalable data storage for data processing industries. Prior to the discovery of the SAN, researches looked into similar technologies such as Direct-Attached Storage (DAS) and Network-Attached Storage (NAS).

1.2 Introduction to SAN

The current networking industry can be broadly classified into three categories: hosts, server/client and storage. Figure 1 shows the basic storage model [18]. The hosts are used to generate requests and carry out tasks. The servers are used to process this information and storage is used to store the information that has been processed for further reference. It is also essential that these components have a connectivity channel to communicate between each other [8].

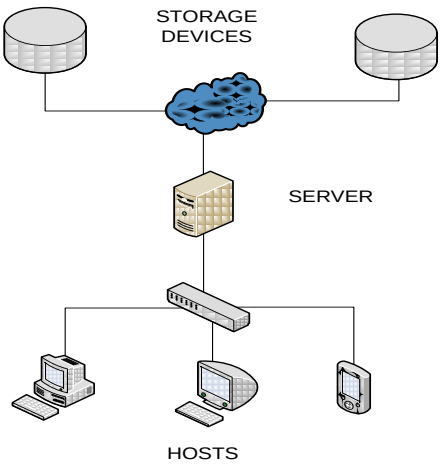

Figure 1: Basic Storage Model [18]

With the increase in the use of storage networks, researchers emphasized faster connectivity between the hosts and the storage devices. The traditional methods of the directly attached storage network evolved into a common switched network where an ‘island’ of storage devices was created. A network was used to connect these ‘islands’ to the server/clients for access. This proved to be a very cost-effective solution for major organizations since the implementation saved a lot of time and resources to provide complete connectivity between their server/client and storage devices. This solution also had the advantage of scalability since both the internal LAN network and the storage network were completely independent of each other.

Even with these advantages, SAN was not very popular among organizations because of the fact that the data transfer required a very fast and reliable network to have optimum efficiency between the storage devices and server/client. Thus, these situations led to the discovery of number of protocols such as FCIP, SCSI, and iSCSI. Every networking protocol had its own pros and cons. After the discovery of these protocols, there was a rapid implementation of SAN across the globe among network administrators since the protocols

provided them with the flexibility of multiple virtualization and transactional points. SAN also provided engineers with the advantages such as scalability and reliability within a network [8].

1.3 Understanding DAS and NAS

A Direct Attached Storage (DAS) is as shown in Figure 2 in a network [3]. These are the most fundamental levels of storage systems. The storage devices are a part of the host connecting as a driver or a server with RAID as a technology to bind them. All the hosts on the network must connect to the server to gain access and perform operations on the disks. Unlike SAN or NAS where the users have to connect to a common network to access the storage devices, DAS provides an easier option for users to connect to the storage drivers. Although this was very popular during the early 1980's and 1990's, this model had major drawbacks in terms of speed and performance [3].

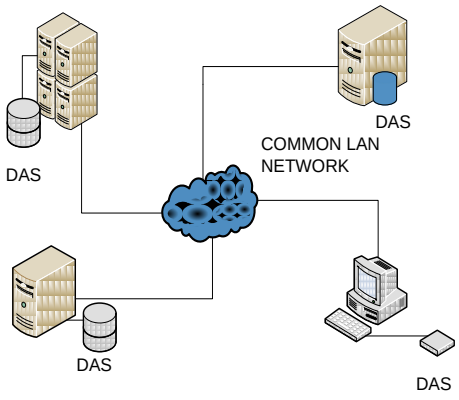

Figure 2: Directly Attached Storage topology

Figure 3 shows a diagrammatic representation of a Network Attached Storage (NAS) within a network. NAS was introduced to eliminate the drawbacks of DAS. In NAS, the server is dedicated only to perform the storage operations and file serving, unlike in DAS where the server was burdened with both file sharing and application serving. Therefore, NAS is more effective in nature and eliminates the performance issue. Although this technology was introduced to serve

the Enterprise level of organizations, it has become popular among the small and medium organizations because of its flexibility. With this structure in mind, the main focus of this thesis is about the work that has been done in this field of study and why this is essential.

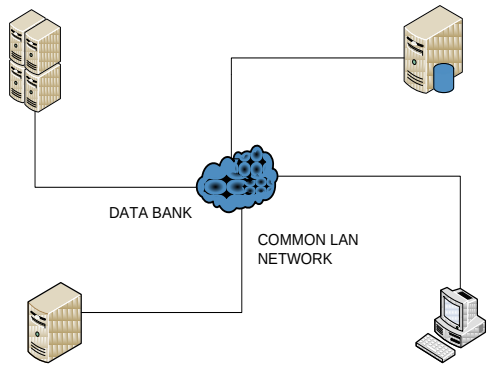

Figure3: Storage Area Network Topology

1.4 Importance of SAN

Storage Area Networking provides wide ranges of advantages such as scalability, reliability, flexible and heterogeneous connectivity and system simplification. In general, the storage is connected to its servers through a dedicated channel. The servers are then interconnected by LAN or WAN technologies; thereby providing redundancy within the network. The workstations connecting to these servers could range from desktop computers to handheld devices.

The workstations thus run different kinds of operating systems such as Windows, Linux, MAC and so on. Each operating system would have a different file formatting technique which would lead to different database systems. Managing a multi-format, multi-vendor system would be more expensive and more complex. Recent studies also have shown that distributed computing is cheaper than a centralized storage system. It also provides more flexibility to the administrators. The Storage Area Network (SAN) provides a simple solution since it has the

advantage of storing elements in the form of blocks. Block storage is discussed further in the architecture section of this thesis. This functionality of SAN can be explained by considering an example.

In traditional computing or in a client/server model, deploying servers for the clients to connect to would create islands of server/client information on the network. This would essentially result in isolated information frames, making the sharing of data between clients a tedious task. The data transfers can be carried out in different forms, such as backup, file transfer, and inter-process communication. This would also mean that organizations would have to invest in additional media to copy or use the resources stored on the servers.

However, with the above method, sharing out-of-date information would increase the overhead of updating information between the users. The SAN provides an effective solution to this issue by providing on-the fly data sharing between users which would result in a reduction of time and cost in file transfer between peers.

The basic SAN model uses the principle of “any data at all time”. This means that any single host would not be isolated to a particular server or storage. If one server needs information from another, it does not have to copy it or transfer it. Instead it can access a common media where the information is stored and can be used with the permissions from the other server. Thus, the basic requirement for file access would be permissions and time for access on the common storage media. The feature is also called the Universal Storage Connectivity. These discussed features make SAN as a very powerful technology to have universal connectivity in networks for storing and accessing data within an organization.

The advent of Storage Area Networking also brought about the major advantage of cost-effectiveness to organizations. Since all the hosts and workstations can access the storage

directly, time is saved. This also means that no extra temporary storage is required to share the data produced by one machine with another. Similarly, storage elements such as magnetic tapes and hard drives, which are expensive and exclusive to one host, can be accessed by the entire network. This greatly reduces the cost of implementing the storage devices without diminishing the quality of service [9].

The Storage Area Networking also provides administrators the feature of failover in case one of the devices fails on the network. This provides the Storage Area Network with the feature of redundancy, increasing the fault tolerance level of the entire network.

1.5 The Classic Storage Model

The Classic Storage Model was developed initially to facilitate the vendor needs. The vendors, designers, and customers approached storage elements as mere boxes or wires without proper explanation of what their requirement was. This form of representation increased confusion among the developers and vendors since the information associated with the design was inconsistent [11].

Figure 4 describes the Classic Storage Model [18]. Things were made even worse when there was a large network involved and this inconsistent data was relayed from the vendors to the storage providers. Thus, the performance could not be maintained at an optimum level. In addition, the efficiency of the network would reduce drastically scalability was required.

These drawbacks were overcome with the SNIA shared model of storage systems. SNIA is an autonomous institution that sets industry standards for storage providers. This organization came up with a shared model for all vendors and providers so that they can communicate with the same standards and, therefore, improve the overall efficiency of the storage network. The

other main advantage of the SNIA shared model is that this model is more "function" specific than "box-specific" and thus propagated information would be more consistent in nature.

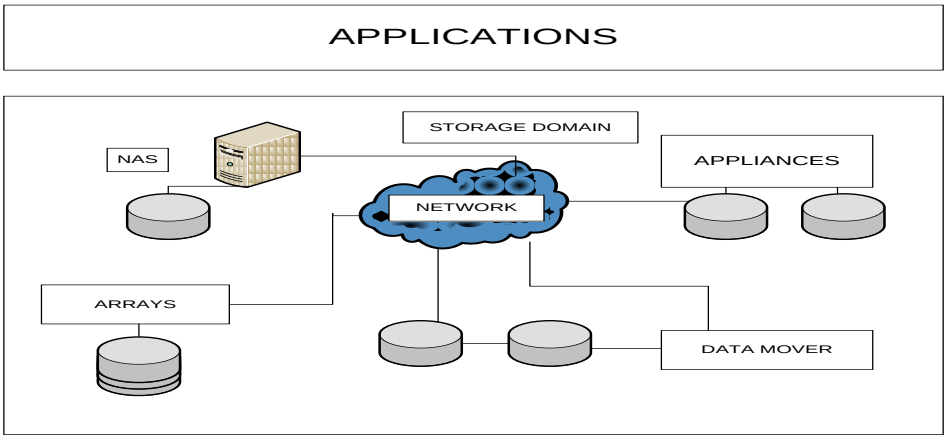

Figure 4: The Classic Storage Model

1.6 SNIA Storage Model

Figure 5describes the SNIA Storage Model [18]. The main components that are described by the SNIA Storage Model are:

i. Interconnection: The interconnection network refers to the components that connect the different storage elements within a network. This primary network provides connectivity between the clients and the storage elements in the network. The common hardware that are used for all the interconnecting networks are Fast-Ethernet, Gigabit-Ethernet, Fiber Channels and ServerNet. The protocols used by interconnecting networks include SCSI, iSCSI, TCP/IP, CIFS and many more. It is also desired to have a secondary connection between the lines in case of a failure within the network. This secondary connection provides redundancy and is used to direct traffic around a failed component. The other main advantage of this secondary connection would be to perform load balancing between the storage elements and the hosts in an I/O path. The main idea behind the redundant path is to provide connectivity and share the load in case of a failure on the primary connection.

ii. Host Computers: In the classical model of storage systems, the hosts were considered external to the storage environment. Thus, the host running on the network would act as an anomaly to the storage environment. However, in the SNIA model, any host that is having some or all of its needs supplied by the shared storage system is considered a host. These hosts are responsible for function mappings and thus are considered an integral part of the network subsystem. The host is typically attached to the storage system by a network interface card (NIC) or a host-bus adapter (HBA). These hosts are mostly unaware of others connected on the network; however, they participate in sharing resources from a common bus.

iii. Physical Storage Devices: These devices are essentially the non-hosts or the storage elements that are a part of the storage network. These devices may be directly attached to the storage network or can be associated with a host participating in the subsystem. The commonly seen storage devices are magnetic tapes, array controllers, disk arrays, hard drives, and disk drives. These devices have the capability of storing data persistently.

iv. Logical Storage Resources: This is an abstraction from the service subsystem that divides a storage element logically. The common logical resources are volumes, blocks, files and data movers.

v. Storage Management: The storage management functions to observe, control, and implement logical resources to a network. These are performed by specialized softwares that segment the physical drives.

With the understanding of the storage components, it would be much easier to explain the different layers in the SNIA storage model. The SNIA shared model can be divided into three main blocks: File/Record layer, Block aggregation layer and the service subsystems [11].

i. The File/Record layer: This layer is responsible for packing the small things such as byte vectors and records into bigger components such as block-level volumes and storage device logical units. The File/Record layer can be further divided into the database management system and the file system. These layers function to index files, thereby allowing ease of access to the storage system for faster caching. The volumes act as a base for both database management and the file system. However, both functions provide mappings to different entities. This layer is also responsible for indexing, primary fetching, and caching to improve the overall efficiency of transactions between the storage elements and the hosts.

ii. The Block Aggregation Layer: The Block Layer acts as an interpreter to the higher layers. This layer is mainly responsible for providing low-level storage to network elements. In SCSI, the logical address spaces are called Logical Units (LU's) and the storage device is capable of storing many such LU's. These LU's are stored on the drive as "native" elements and can be aggregated to one or more blocks to provide redundancy. The Block Layer also performs the function of indexing in a simple form such as Logical Unit Names (LUN's) and caching to provide faster access to the data stored on the blocks.

iii. Service Subsystems: The Service Subsystem is essentially used to manage the storage network within a storage networking environment. It performs duties such as the discovery of storage elements, the monitoring of storage elements, resource management, redundancy, and capacity planning. These are responsible for carrying out the most vital operations within a storage network. The other main feature of the service subsystem is to enhance and optimize the use of the storage network to provide the best efficiency for the storage network.

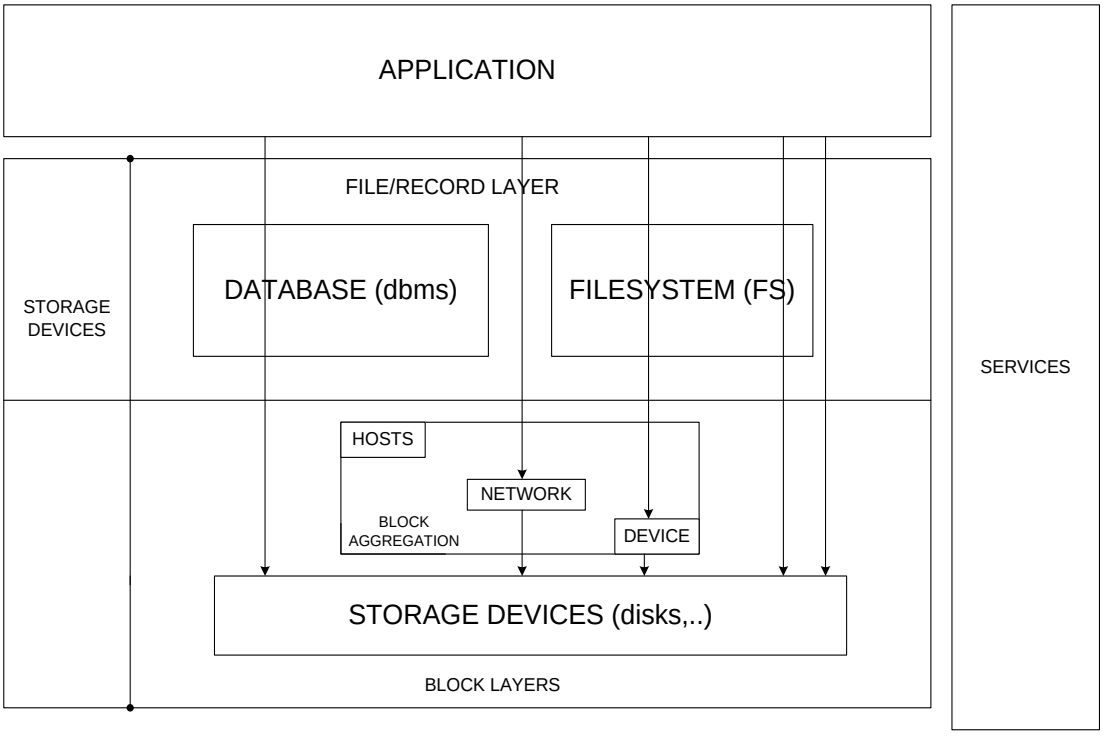

Figure 5: SNIA Storage Model

1.7 Motivation for Thesis

There is an ever growing demand for faster and reliable communication in the Storage Area Networking industry. The client/server model proved to be the best solution in the early 1990's. This technology could cater to most of the demands at that point. The major drawback of the client/server model was the physical resources that were needed and, with, an ever-increasing demand, the scalability was limiting the growth of the industry. This drawback was overcome with virtualization, which required that the network administrators use many such client/servers on one physical machine. This proved to be an effective solution to the scalability issue that the client/server model posed.

As in the case of any technology, the security concern of virtualization needs to be remedied to make it the most effective form. The security implementation has to be a systematic approach and proper measures need to be taken to understand the threats faced by virtualization.

In the research carried out until now, the planning of security implementations is not considered. Therefore, this thesis is focused on providing administrators a basic threat model to implement security features and mitigate some basic threats faced by virtualization.

1.5 Organization of Thesis:

This thesis provides the readers a detailed description of modeling threats in a Para-Virtualized (PV) environment. Chapter 1 presents a brief introduction to the technologies such as Storage Area Networking (SAN) and virtualization. This chapter also provides the reader with details regarding the importance of different architectural styles in SAN. Chapter 2 renders a detailed description to virtualization and Xen. This would constitute the technology aspect and the threat modeling that is presented in Chapter 3 along with the discussion on threats in a PV environment. Risk equations are derived for these threats in Chapter 4 and results are discussed to validate these risk equations in Chapter 5. The Chapter 6 provides a conclusion followed by future work.

Chapter 2

VIRTUALIZATION AND XEN

2.1 Introduction to Virtualization

The history of virtualization dates back to the late 1960's and early 1970's when IBM came out with the first virtualization technology in the x86 architecture. However, this technology did not achieve much success during this period since the cost involved to implement it was very high. Although this technology has been in place for the past 40 years, it came to prominence only during the past decade when the x86 architecture became a common feature among computer networks [9].

Virtualization is defined as a technology that allows users to remove all physical dependencies from the server operating system, thereby allowing them to move and recover from any faults. Virtualization provides an abstraction layer that separates the hardware and the applications. This isolation improves the overall reliability of the IT infrastructure by maximizing the use of the resources available.

Before proceeding with virtualization, an understanding of the concepts of centralization and decentralization of server segments within a network is required. Figures 6 and 7 show the implantation of a centralized server and decentralized servers within a network environment. In a decentralized environment, network peripherals are chosen in such a way that they perform only a specified task, thereby isolating the possibility of overlapping functions. Each task is processed as a separate entity and the processor is the only device that would integrate all the functional results from different entities. This provides benefits such as security and stability by utilizing most of the resources that the machine has to offer [10].

Unlike the decentralized server, the centralized server integrates all the resources on a single server. The centralized is much more expensive than the decentralized server, but the physical server would have the drive to operate more sub-servers on one physical server. The processor within the centralized server would share the resources between all the sub-servers that would be running on it. The hardware resources are shared among the sub-servers by the process. Hence, the overall cost for implementing a centralized server would be much cheaper than the decentralized server.

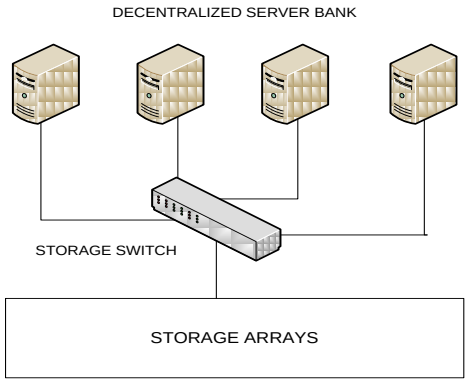

Figure 6: Decentralized Server

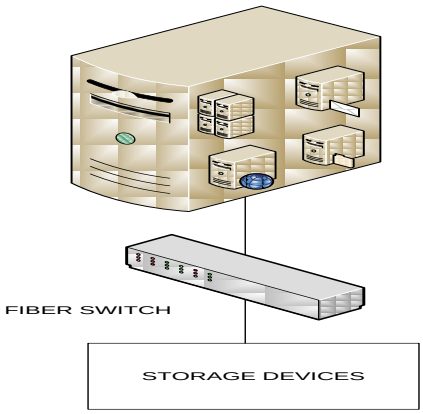

Figure 7: Centralized Server

Virtualization is a paradigm of both these techniques. It has the features of both centralized and decentralized server management techniques. It inculcates the advantages from both techniques and provides cost effective solutions to vendors.

2.2 Benefits of Virtualization

The main benefits of virtualization are:

i. Consolidation: The consolidation of a server would increase the server utilization. This also allows the vendor to streamline the test and the development environment. The main advantage of consolidation of hardware would be the ability to use multiple operating systems on a single physical platform.

ii. Reliability: Virtualization is comparably more reliable since it has the ability to create dedicated failover partitions on a physical disk. This also has the benefit of isolating software faults and reallocating an existing partition.

iii. Security: Since virtualization uses the concept of volumes in a partition, the fault detection and correction is much simpler as compared to traditional methods. This also prevents digital attacks and each part of the partition can have a different security setting. These advantages assist in making the entire system more secure against more common types of attacks [10].

2.3 Virtual Machine Monitor

The main feature of virtualization is the ability to create multiple virtual servers on a single physical server. To understand this, it is necessary to understand the four ring structure of the x86 processors. Each of these rings represents a privilege level numbered from 0 to 3. The innermost ring or the ring 0 has complete control over the processor and ring 3 has restricted access. Each ring structure represents the data segment that the processor can use for storing data. The processor uses this ring structure to determine what type of code needs to be written in each segment. Figure 8 shows the ring structure for different operating levels [7].

The Virtual Machine Monitor (VMM) is a thin software layer that assists in creating the virtual partitions on the server. The VMM runs directly on the physical hardware and provides a platform for running multiple operating systems. This operates in the ring-1 of the x86 architecture and provides the processor with all the necessary information regarding the Virtual Machines hosted by it [12].

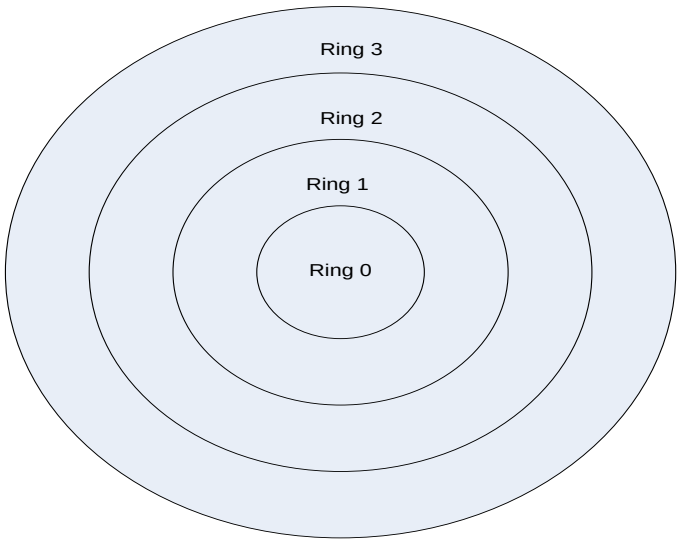

Figure 8: Ring Representation of Different Operating Levels

The Virtual Machine Monitors are generally referred to as hypervisors. These hypervisors are primarily responsible for interfacing the processor operation with the requests made by the Virtual Machines. The hypervisor places the requests made by the hosts on the ring 0 and presents it to the processor, while the hypervisor handles the actual interaction of the CPU, I/O and memory with the physical hardware.

Figure 9 shows the basic model of virtualization. There are 2 main types of Virtual Machine Monitors.

a. Type-1 VMM: The type-1 VMM is software that runs directly on the hardware platform or has direct access to the ring-0. The guest operating systems run on top of this type-1 VMM, above the hardware layer abstraction, allowing isolation between each Virtual Machine.
b. Type-2 VMM: In the tpe-2 VMM, the hypervisors run on the operating systems, usually on the ring-3. The hypervisor acts as a pseudo ring-0 to all the Virtual Machines that it hosts. Hence, the hosts are not accessing the direct hardware. The main drawback of this type is that the Virtual Machines are at a distance from the actual processor and thus the calls that are sent to the processor have to traverse through multiple layers before they are processed. In

both of the above types, the Virtual Machine Monitor is the center of server virtualization. It presents itself as the control for all Virtual Machines and the physical hardware resources. It presents the Virtual Machines with a virtual set of CPU, I/O and memory. Hence, the VMM would be the main aspect for any virtualized environment.

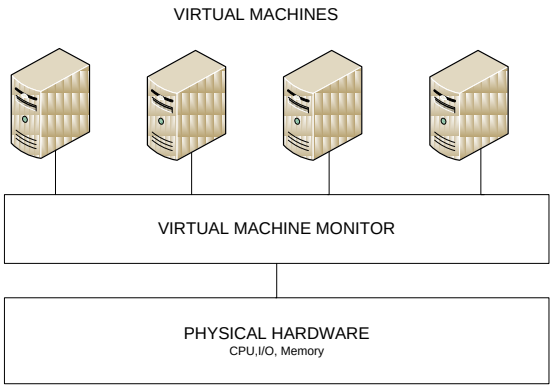

Figure 9: Basic Model of Virtualization

2.4 Types of Virtualization

The current research is producing newer techniques to implement virtualization to other forms of traditional computing. This is due to the fact that virtualization provides network administrators with the benefits such as reliability and scalability at a low cost. Virtualization can be broadly classified into four main categories

a. Server Virtualization: The server virtualization is most common one that is in use today. This is used extensively in the current networks because there is no additional hardware required to implement this technology. All the x86 platforms support the feature, and currently, the Advanced Micro Devices (AMD) has also come up with their own processor that supports server virtualization. This thesis deals with this form of virtualization, which is discussed extensively in the next section.

b. Storage Virtualization: This virtualization technology in storage area networks is one of the hottest research topics. The discovery of "Cloud Computing" brought a huge boost to storage virtualization. Over the past decade the storage vendors have been offering high-performing storage solutions to their clients. The most basic form of storage includes physical disks and spindles presented to the hosts and servers with the implementation of RAID technology. Although an assembly of multiple physical disks is considered here, this would be acting as a single logical entity. The hosts connecting these storage elements are promised their own dedicated physical storage. However, in actuality, the vendor would have just allocated a small amount of drive space to this virtual partition and provided to the host. This would be more beneficial for both the vendors and the clients since the vendors do not have to worry about indexing the volumes on the drive to provide privacy and security to the host's information. The clients have the advantage of a more economical and quicker drive space and do not have to worry about the unused space on the drive.

c. Network Virtualization: Network virtualization has been in place for a very long time. However, this is not recognized as the server or storage virtualization. The most popular forms of network virtualizations are VPN and VLAN. These technologies are fairly simple in nature. The VLAN, or Virtual Local Area Network, is a technology wherein the switch is divided into multiple broadcast domains. The VPN, or Virtual Private Networks, is a mechanism used to communicate confidentially over a public network.

d. Application Virtualization: This is the newest form of virtualization. In application virtualization, applications are not operating-system-dependent or driver-dependent. These can be instantly activated, deactivated, or reset based on the need of the user. This has the

advantages such as the elimination of application conflicts, compatibility issues, and instant provisioning [12].

2.5 Virtual Machine

The Virtual Machine is a single segment hosted by the physical machine. As described earlier, virtualization is a technique that partitions a single physical machine to a multiple sub system capable of handling different operating systems and applications running simultaneously without any interruptions. Each of these subsystems runs independently, acts as a single entity of its own, and each single entity can be termed as a Virtual Machine. The main advantage of these Virtual Machines is that although they belong to one physical machine they would be operating as an independent machine, thereby increasing overall utilization of the physical machine.

It is important to understand the working of the Virtual Machine and Virtual Machine Monitor since the technology of virtualization revolves around these two key features. The server virtualization would be an ideal example to explain these two terms in depth. As described in the types of virtualization, the server virtualization can be categorized further as

i. Full Virtualization ii. Para-Virtualization

ii. Operating System virtualization iv. Native Virtualization

1. Full virtualization: The hypervisor provides a full simulation of the underlying hardware. Therefore, the guest operating system is unaware of the virtual environment around it. All applications that are compatible with the operating system would run as though they are the raw hardware of the physical machine. The major advantage of this type of virtualization is that it provides complete isolation between the hypervisor and the guest Virtual Machine. The operating system on the guest Virtual Machine (VM) can be installed without

performing any modifications on it. This also exhibits near-native CPU and memory performance. On the other hand, the drawback of the virtualization technique is the compatibility between the hardware and software that is being used. It is hard to attain optimum efficiency due to the fact that x86 architecture uses a set of privilege calls that cannot be trapped. Thus additional trap and patch emulators are required.

2. Para-Virtualization provides a partial simulation of the physical hardware to the operating systems on the guest VM. Hence, the guest operating system is completely aware of the environment under which it is operating. The main feature of this type of virtualization is that the hypervisor provides each Virtual Machine with its own address space, thus making the transactions occur directly on the physical hardware. The main advantage of a Para-Virtualized environment is the implementation of this technique. Unlike full virtualization, the Para-Virtualization does not face the issue of compatibility with respect to hardware and software. This does not require any hardware assistance to operate since the guest operating system would be modified to work on a virtualized environment. However, this can also be accounted as a drawback since the Para-Virtualization requires some form of modification to work on the hardware that is already present. The other major drawback of this technique is that the VMs may not be portable and are not backward compatible.
3. Operating System Virtualization: This technique is similar to the full virtualization and the Virtual Machine Monitor would be running on a host VM. Thus, this model can be described as a technique which involves a single operating system instance. This technique does not involve a great deal of processing time since the operating system runs on the physical machine at a single instance. The Operating System Virtualization also benefits greatly during the upgrade of the entire machine and can support the machine to operate at their

native speeds. The major flaw with this technique is the fact that this type of virtualization does not support a mixed operating system facility. Since the VMs are running on a single instance of operating system, the security of guest machines becomes a prime concern in this technique since they are not isolated as in the other techniques.

4. Native Virtualization: This form of virtualization uses the benefits of both Para-Virtualization and Full Virtualization and is commonly referred to as Hybrid Virtualization. The main advantages of this technique are that it uses the architecture of Para-Virtualization where the Virtual Machine Monitor would essentially perform the ring-0 operations. However, unlike the Para- Virtualization, this technique does not involve modification of the operating system for the guest VM. Thus, this can operate at higher speeds as compared to the full virtualization. Although native virtualization attempts to combine the benefits of both virtualization techniques, it still involves a bit of modification on the operating system of the guest VM. Thus to achieve optimum efficiency, the CPU architecture should be assisted by the hardware accelerator.

2.6 What is Xen?

Xen is an open-source Virtual Machine Monitor that supports 32-bit, 64-bit and ARM architectures. It classifies itself under the Type-1 VMM and runs directly on top of the physical hardware. This feature enables the user to run multiple guest operating systems on a single host computer. All the Virtual Machines operate individually with near-native performance.

The Xen VMM was first started at the University of Cambridge Computer Laboratory. Later, this project was funded by the UK's Engineering and Physical Sciences Research Council (EPSRC). The main purpose of this project was to use Xen to provide public infrastructure for

WAN distributed computing. The Xen VMM was held as the core for this project. Later, this project was handed over to an open-source community and now is called XenSource [16].

2.7 Why Xen?

When virtualization came into existence was this technology not implemented by many people. This was primarily because either the softwares that were available were proprietary, or the ones which were open source were not secure enough to be implemented within a critical work environment.

Xen, after its invention, was a revolution by itself. This was an open-source project with a sophisticated infrastructure that provided security and reliability to an enterprise network. This piece of hypervisor was packed with many features and many people were interested in understanding and implementing this technology within their network.

Xen provided administrators with benefits such as dynamic resource allocation, compatible VMM for almost all hardware, live and zero-downtime migration of running Virtual Machines. Xen also supports the Virtual Machines running on it with near-native performance. It supports different processors such as x86, x86 (32 bit) with Physical Address Extension (PAE), and x86 (64 bit) with extensions.

Xen operates in both Para-Virtualized and full virtualized environments. However, for the Para-Virtualization mode, the guest operating system has to be modified and the kernel needs to be re-compiled to support the Xen VMM. This greatly affects the version of operating systems that could be used. It presents better performance in return. In full virtualization mode, there is no need to modify the operating system and this would be a much clearer option. The drawback of this mode would be that this is supported only on the Intel-VT and AMD-V processors.

2.8 Xen Architecture:

The main aspect of implementation of any new technology would be the compatibility of the existing hardware in an organization. It would not make any sense to implement a technology that cannot work with the current existing environment. Administrators also have to consider consolidation, optimization, increased flexibility, and scalability before implementing the technology. Thus, in such circumstances, Xen would be an ideal choice due to the fact that Xen has a very unique and robust architecture.

In recent years, researchers have come up with different techniques to host multiple applications on a single server in a shared-service pattern. The issue of hosting files and processes on a single server is, hosting them on a single instance of the operating system. This does not provide adequate security and does not isolate the processor from the dedicated hardware. The other main issue with the "stacked" approach would be the competition of resources. Since applications are being executed concurrently and are contending for the CPU, memory and I/O, processes can easily cause the host machine to be oversubscribed [12].

This effect would cause applications to believe that they are associated with dedicated hardware. Applications will attempt to utilize the hardware as needed. This can be overcome by creating containers within an operating system to help assure each application is receiving the correct amount of resources. However, this technique is not reliable.

Although the stacked approach would provide enterprises with benefits such as fault and performance isolation on the host running a single instance of the operating system, this method proves to be expensive. To overcome such issues, the simplest solution would be virtualization. Virtualization provides the users with multiple options, each having their own benefits and

strengths. Xen is a very viable method, taking advantage of x86 architecture and using Para-Virtualization as the primary technique.

Xen supports processors such as x86, IBM's power PC, Intel's Itanium and AMD's AMD-V. The main advantage of Xen is the function of modifying the operating system to run on any hardware. This also provides the user with the flexibility of customizing the operating system as per the requirements. Xen's virtual I/O architecture optimizes the performance and increases the efficiency [12].

Xen has a very simple design for a complex architecture. The main goal for Xen architecture is to provide complete isolation between the fault and performance of the hardware. The Xen hypervisors handle the CPU scheduling and access-controls. This allows the guest operating system to access the higher level controls of the physical machine directly. Xen also proves to be efficient by providing hardware level isolation similar to any physical machine without duplicating it on the processor. This technique is commonly referred to as Para-Virtualization.

There are a few other parameters that Xen needs to cater to in order to work as a complete Para-Virtualization machine. These parameters are memory management, CPU, and I/O. Xen handles these requirements to optimize efficiency within a Virtual Machine.

Considering memory management, Xen performs operations such as segmentation and paging by inclusion of a privileged level access that cannot overlap with the top end of the linear address space. Although guest operating systems have access to read and write directly, updates are batched individually and validated by the hypervisor.

Similarly, CPU operations such as exception handling, system calls, and interrupts are handled by Xen in a unique manner. The guest operating systems are not allowed to operate in

ring-O, thereby securing the host machine. The guest operating system also has to update a register table for exception trap handlers. System calls are handled directly by the guest operating system without the intervention of the hypervisor. The hardware interrupts are replaced by a notification event mechanism.

Finally, I/O is handled by Xen in a similar way as in the case of any virtualizer software. The virtual devices are very easy to access and data transference takes place using asynchronous I/O rings. Similarly the I/O interrupts are handled by event notifications.

2.9 Xen Domains:

The Xen hypervisor divides the physical layer into two layers: a management layer and a virtual hardware called as Application Programming interface (API) that includes a control interface for guests to interact directly with the underlying layers. The interaction with management APIs takes place using a control region that has privileged access to hardware and has the user mode management code. Both control regions and the Virtual Machines are referred to as domains.

During the booting of the device, the system initiates a separate domain that has access to use, control interfaces, called the Domain-0. This is responsible for loading the user mode management code that is used to administer the Xen. This also performs the operation of starting and shutting down other less privileged domains known as Domain-U. The Dom-0 controls the Dom-U CPU scheduling, memory allocations, and accesses the I/O devices. The Dom-U is commonly referred to as the Xen Virtual Machine.

The next chapter discusses security aspects along with threat modeling concepts for Xen. This would be the basis for writing the risk equations and assessing risks caused by different types of attacks on the Xen virtualizer [12].

2.10 Security in Virtualization

Virtualization has been an ever-growing technology over the past decade. This technology has spanned from web servers to database servers and storage devices to user desktops in the recent years. As in the case of any upcoming technology, virtualization also faces security threats and proper provisioning has to be placed in case of a security breach.

Virtualization basically adds another layer on top of the operating system, called the Virtual Machine Monitor to host the Virtual Machines. This layer accesses the kernel on a regular basis to process the tasks that are requested by the Virtual Machines. Thus, it is essential to understand the vulnerable areas within the virtualized environment. Identifying alone would not do any good in preventing a possible security breach. It is also necessary to take adequate measures to prevent these breaches from happening on the network.

This thesis emphasizes identifying such vulnerable points within a Para-Virtualized environment by considering Xen as an example and providing a threat modeling structure in case of a possible attack.

Chapter 3

THREAT MODELING

3.1 Introduction to Threat Modeling

With the advent of newer and faster technologies such as Storage Area Networking and virtualization, securities in these areas have become a prime concern to all network administrators. Security by itself would be the major factor for restricting the growth of the industry. Hence, threat modeling would play an essential role in identifying the risks associated with any network and mitigating these risks to reduce the impact of any attack against the network.

Threat modeling by definition means to identify the risks of implementing a technology within a network. Threat modeling involves three main functions:

i. Threat Identification

ii. Risk Assessment

iii. Mitigation Techniques

These three functions form the core for any security implementation that happens within a network. The administrators have to make sure they have a perfect balance between adequate security of a network along with the reliability of the network. There would be no question of security if the data is not available to legitimate users and thus the use of any security techniques should involve parameters like confidentiality, integrity and availability [14].

During the process of development, focus is always towards the design and features of an application and much time is wasted during the testing phase to identify the associated threats. Within any organization, the developers use security as an afterthought which would result in disastrous circumstances. In such cases, vulnerability is undetected during the development

phase of an application. This is one of the typical situations any developer would face during the development of an application. The challenge that any organization would face would include the security aspect during the developing stage to reduce the risk of vulnerabilities going undetected. The solution to this challenge would be threat modeling [1]. Threat modeling can be used as a tool during the development phase to set standards. During the design phase, the application is tested for these standards and checked for possible vulnerabilities that an attacker can exploit. This gives the developer the flexibility to change the code and secure the application against any possible vulnerability. This would reduce the time involved during the testing phase making the application and the network more secure. The threat modeling also provides advantages such as

i. Defining a security of a system
ii. Documentation of security standards that the application has passed
iii. Identifying bugs at the architectural level
iv. Multiple layers of security

The work done for this thesis is inspired by several studies in the field of virtualization and Storage Area Networking [9]. It is necessary to understand the types of virtualization to obtain the maximum efficiency when they are implemented. It is also necessary to recognize the work done by the researchers in this field of study as this could be one of the driving forces in the near future. This research work can be divided into two main parts, namely threat modeling and implementation of this threat model into a Para-Virtual environment. Thus, it is necessary for the reader to understand the concepts of both virtualization and threat modeling. This would greatly benefit the reader to understand the implementation of threat modeling in a virtualized environment.

In the paper written by G.Kbar [4], a detailed description of modeling of threats is introduced to a computer network. This model can also be considered a prototype for risk assessment of a computer network. The author introduces concepts such as security gates and failure analysis in his paper and, hence, it becomes one of the most important papers for the understanding of security implementation within a network. The work by Longstaff and Haimes [5] provides a very good overview of information assurance and classification of threats within a network. This paper also provides a basic introduction to information safety and risk assessment techniques that can be utilized to safe guard the information within an organization.

The threat model of any organization, when implemented, would assess, forecast and mitigate network issues during real-time operations. The research done by Liao, Li and Song [6] provides a strong modeling structure for real-time network. This paper also puts forward the concepts of forecasting and mitigation techniques by considering risk equations. These are a few papers that provided vital information regarding the threat modeling of a computer network. These papers were also useful in understanding the security considerations during modeling risks for various network implementations. Along with these, the work done by researchers in the field of storage virtualization and resource management in a virtualized environment acts as the base for this thesis work [25]. The work done by Zhang, Sun and Gu [7] provides an insight in optimizing the use of inter-domain communication in a Para-Virtualized environment. The authors explain the working of inert-domain channels and their importance in a Para-Virtualized environment.

3.2 Threat Modeling Methodology

Threat modeling is a very systematic approach towards securing the network against possible attacks at different layers within the network. The primary focus for any security expert

within an organization would be to identify the assets and the threats that are associated with them. The assets could be at the application layer, the medium that carries the data, or the entire network itself [16].

In recent years, many organizations have researched the methodologies to model threats and have come up with various methods to secure their assets against attackers. Some of the most common methodologies are data flow, call flow, and trust flow. Any researcher would not be in a position to identify a particular method as the best method since all of these methods have their own pros and cons. It would be in the interest of an administrator to pick or form their own method for assessing threats and model the security infrastructure of the network.

In recent years, organizations such as Microsoft and IBM have come up with threat modeling to design the security infrastructure of their organizations [13]. One of the most popular methods was STRIDE (Spoofing, Tampering, Repudiation, and Information disclosure, Denial of service and Elevation of privileges) which was considered to be a very sophisticated threat model for an application.

By identifying these vulnerabilities, the impact of an attack is estimated. This methodology uses the data flow form, wherein each vulnerability variable is assigned a value automatically by a program and any changes made would recalculate the risk value for a threat. Similarly, other methods such as DREAD (Damage, Reproducibility, Exploitability, Affected Users and Discoverability) are also used by Microsoft as risk estimation for any given threat.

3.3 Threats in Para-Virtualized Environments:

The understanding of the definition of threat modeling would serve as the ideal platform to discuss the common threats that are found in a Para-Virtualized environment. By definition, a Para-Virtualized system is a category under the virtualization which modifies the guest operating

system to execute operations directly on the hypervisor so that no performance loss is observed during the emulation of the complete hardware [3].

Since the Para-Virtualization is a technology that was initiated by an open-source community, security concerns with this technology are of prime concern. Therefore, understanding the threats faced by this technology and providing adequate mitigations against these threats is essential. Many organizations have come up with their own forms of Para-Virtualization like Xen, KVM, and VMware. However, some of these Para-Virtualization applications are not open source. Xen is one such Para-Virtualization environment that is open-source and was developed by Citrix systems. As in the case of any threat modeling structure, threats that are affected by these Para-Virtualized hypervisors are considered. This would essentially form the base structure for the risk assessment and mitigation techniques.

The Xen hypervisor provides the user with some benefits that the other virtualization techniques does not provide. However, it is essential to understand the security threats associated with Xen and mitigate these threats to make it more secure against possible attacks.

The major security threats associated with a Para-Virtualized Environment such as Xen are:

i. Authentication Level Attacks
ii. Hypervisor Level Attacks
iii. Kernel and Disk Level Attacks.

3.3.1 Authentication Level Attacks

Authentication Level Attacks that are caused due to ignorance or inadequate protection provided to a Virtual Machine. Although this is not a major security concern, this type of attack can lead to other attacks of larger proportions on the network. When an authentication level

attack is carried out, the attacker tries to gain unauthorized access into a Virtual Machine. Once the attacker gains access to the Virtual Machine, important information can be extracted thereby compromising with the integrity of the environment. The authentication level attack is one of the most common types of attacks that can be found on a networking environment. The authentication level attack can be classified into two types

i. Attack caused by an internal user

ii. Attack caused by an external user

The attack by an internal user is mainly due to ignorance or violation of a security policy by an employee of an organization. When an authentication level attack is performed on a host machine, the attacker would be able to control the privilege level of a guest Virtual Machine, thereby allowing the guest user to misuse the resources allocated to it. The impact of such an attack is lesser than other forms of attacks since, the fault detection is simple. Likewise, securing a Virtual Machine from an authentication level attack is comparatively simple.

Most of the authentication level attacks are caused by an external user trying to get access into a Virtual Machine. The attackers on the outside try to sniff the packets from outside and perform a brute force attack on the Virtual Machine to gain access [13].

3.3.2 Hypervisor level attacks

The hypervisor level attack is an attack that is performed by targeting the Virtual Machine Monitor or hypervisor of the virtualizer. The attacker primarily would try to gain access to a Virtual Machine on a physical server. Then, a sniffer program would be installed on the Virtual Machine to collect information regarding the communication that takes place between the Virtual Machine and the hypervisor. With this information, the attacker tries to modify the channel information and attack the communication between other Virtual Machines and the

hypervisor. The attacker would be either degrading the performance of the Virtual Machines or trying to gain more than the allocated resources for this machine. The impact level of such an attack is much higher than the authentication level attacks [5].

3.3.3 Kernel Level Attacks

This type of attack is not that common in nature since the time taken to perform such an attack is much longer than any of the above discussed attacks. The impact of a kernel level attack is very high on the network since the attack is carried out on the host Virtual Machine. The attacker gains access to the host Virtual Machines and installs "rootkits" on the kernel of the host machine. These rootkits are snippets of a sniffing program that transmit information to the attacker who sits remotely. The rootkits work in the background and do not identify themselves to any user that is operating on the Virtual Machine. The rootkits record almost all the data that is passing through the machine and sends it to the attacker. The attacker with this information would be able to track information that is getting processed on the kernel and would be able to perform several attacks on the Virtual Machine, eventually bringing down the entire physical server and the Virtual Machines on it.

This thesis discusses the risk involved when such attacks take place on a network and the methods of mitigating these attacks to improve the overall efficiency of the virtual environment. The next chapter deals with assessing the risks, vulnerabilities, and impact of each type of risk by providing mathematical formulates to calculate risk [6].

Chapter 4

RISK EQUATION

4.1 Definition of Risk

Risk is defined as the analysis of all the threats that impacts the performance of an asset. It is important to understand the risks involved while implementing a new technology so that adequate security measures can be taken to mitigate any possible attacks. Hence, it is also necessary to formulate these risks to properly plan out the security implementations on the network. Mathematically [4][17],

$$\text{Risk} = \text{Threat} * \text{Vulnerability} * \text{Impact}$$

Threats for Para-Virtualized Environments can be classified into three main categories:

1. Authentication Level Attacks
2. Hypervisor Level Attacks
3. Kernel and Disk Level Attack

4.2 Authentication Level Attack:

This is the most basic form of attack. This attack is carried out in order to gain control of the Virtual Machine [13]. The attacker tries to break into the VM by compromising the security features incorporated in VM. The security features may include a single authentication level or authentication at different levels. The authentication can be performed either at local access level or on a remote server. The different authentication methods are:

1. Encrypted Passwords
2. Active Directory
3. Secure Login

Thus the risk equation for an authentication can be written as [3],

$$P\ (A) = \sum_{x=1}^{n} x\left[P(F) * [P(i) + P(e)]\right] \quad (4.1.1)$$

Where, *P (A)* = Probability of authentication level attack, *P (F)* = Probability of attack due to failure in authentication, *P (i)* = Probability of attack due to internal user, *P (e)* = Probability of attack due to external user, *x* = Number of users trying to attack the network.

4.2.1 Probability of attack due to failure in authentication

It is essential to calculate and understand the components that constitute this type of attack, because with the clear understanding of the attack, fault detection and fault isolation becomes very simple. The probability of failure of attack is constituted by two factors:

a. Method of Authentication: There are different forms of authentication that can be performed to secure a Virtual Machine. Some of the examples are encrypted passwords, secure login, active directory etc.
b. Authentication Level: The level signifies the point at which authentication is carried out. This can be at several points. The higher the number of such points, the higher would be the security. *P (F)* can be represented as,

$$P\ (F) = \sum_{y=1}^{n} [P(Fp, y) \| P(Fs, y) \| P(Fa, y)] \quad (4.1.2)$$

Where, *P (F)* = Probability of attack due to failure in authentication, *P (Fp,y)* = Probability of failure of authentication method when encrypted password is considered at level *'y', P (Fs,y)* = Probability of failure of authentication method when secure login is considered at level *'y', P (Fa,y)* = Probability of failure of authentication method when active directory is considered at level *'y', y* = authentication level (local database, remote server).

4.2.1.1 Probability of attack due to internal user:

In this form of attack, the attacker can be the employee of an organization or a guest user trying to get access to the company resources. The vulnerability of this kind of attack is very high due to the fact that most inside users are either just ignorant or very naïve about the operations that they perform over their Virtual Machines. *P (i)* can be expressed as,

$$P\ (i) = \sum_{j=1}^{i} P(Hj, Gj) \tag{4.1.3}$$

Where, j = Number of attack attempt made on guest or host VM, *P (Hj,Gj)* = Probability of attack on host or guest VM on the j^{th} attempt by an internal user, *Hj* = Attack rate on the host VM on the j^{th} attempt, G_j = Attack rate on the guest VM on the j^{th} attempt, j = Number of successful attacks on the VM.

Further, using Baye's theorem [2], the effect of an attack on any VM, along with the effect on the other VM can be calculated as

P (Hj,G) = Probability of guest VM getting affected due to attack on host VM by an internal user

$$P\ (Hj, G) = \frac{P(G|Hj) * P(Hj)}{P(G)} \tag{4.1.3.1}$$

P (Gj, H) = Probability of host VM getting affected due to attack on guest VM by an internal user

$$P\ (Gj, H) = \frac{P(H|Gj) * P(Gj)}{P(H)} \tag{4.1.3.2}$$

The second term *P (Gj, H)* can be neglected in this equation since the effect on a host VM from an attack on the guest VM is minimal.

4.2.1.2 Probability of attack by an external user:

$$P\ (e) = \sum_{k=1}^{i} P(Hk, Gk) \tag{4.1.4}$$

Where, k = Number of attack attempt made on guest or host VM, $P\ (Hk, Gk)$ = Probability of attack on host or guest VM on the k^{th} attempt by an external user, Hj = Attack rate on the host VM on the k^{th} attempt, Gj = Attack rate on the guest VM on the k^{th} attempt, k = Number of successful attacks on the VM.

Further, using Baye's theorem, the effect of an attack on any VM, along with the effect on the other VM can be calculated as

$P\ (Hk, G)$ = Probability of guest VM getting affected by attack on host VM by an external user

$$P\ (Hk, G) = \frac{P(G|Hk) * P(Hk)}{P(G)} \tag{4.1.4.1}$$

$P\ (Gk, H)$ = Probability of host VM getting affected by attack on guest VM by an external user

$$P\ (Gk, H) = \frac{P(H|Gk) * P(Gk)}{P(H)} \tag{4.1.4.2}$$

The second term $P\ (Gk, H)$ can be neglected in this equation since the effect on a host VM from an attack on the guest VM is minimal. Therefore the probability of authentication level attack can be written as:

$$P\ (A) = \sum_{x=1}^{n} x \left[\sum_{y=1}^{n} [P(Fp, y) \| P(Fs, y) \| P(Fa, y)] * \left[\sum_{j=1}^{i} P(Hj, Gj) + \sum_{k=1}^{i} P(Hk, Gk)\right]\right] \tag{4.1.5}$$

4.2.2 Vulnerability equation for Authentication level Threat

Vulnerability is defined as the product of exploitability of an asset to the reproducibility of the attack [4]. Reproducibility is be defined as the frequency at which an attack is carried out, and exploitability is defined as the level of security in place to repel a threat.

Vulnerability= Exploitability * Reproducibility

Vulnerability can be expressed as the value obtained from the equation. Rank assessment is done by considering the set of values assigned to exploitability and reproducibility of the threat

$$V(A) = \left|\sum_{i=0}^{n} Ei * \sum_{j=0}^{n} Rj\right| \quad (4.2)$$

Where, *Ei* = Estimated Exploitability value, *Rj* = Estimated Reproducibility value.

$$\begin{cases} i = 0 - \text{The system is highly secure with proper authorization levels and timeout values in place when a user does not logout} \\ i = \text{n} - \text{The system does not provide any form of security in case of an illegitimate user trying to gain access the VM} \end{cases}$$

$$\begin{cases} j = 0 - \text{No attacks being carried out} \\ j = \text{n} - \text{Attacks being carried out at regular intervals of time} \end{cases}$$

A SAN administrator can take all possible security precautions on the network, but these types of attacks are frequent due to the fact that these attacks are caused within the network and it would be really hard for the administrator to track such attacks. The vulnerability values are really high in case of simple attacks because of the fact that these attacks can be easily reproduced on any guest or host VM. The attacker does not require a high level of understanding of the network to perform such an attack.

4.2.3 Impact of an Authentication Level Attack:

Impact plays an important role in risk estimation. The impact of a threat is basically defined as the damage caused due to that particular threat by the attack that was carried out. Hence, impact can be defined as the product of damage caused by the threat to the users affected by the threat [4].

Impact= Damage caused by the threat * Number of users getting affected by the threat.

Impact of an authentication level attack can be expressed as,

$$I\,(A) = \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.3}$$

Where, *Di* = Estimated value of damage on impact, *Uj* = Estimated value of users affected by attack

$$\begin{cases} i = 0 - \text{No effect on host or guest VM's} \\ i = \text{n} - \text{If both host and guest VM performance is affected} \end{cases} \qquad \begin{cases} j = 0 - \text{No users are affected} \\ j = \text{n} - \text{All users are affected} \end{cases}$$

The impact for such an attack is low since the number of users affected is few in number. The damage caused by a simple attack is restricted to that particular VM and hence the impact is very low as compared to higher level of attacks. The authentication level attacks become effective when it is combined with a higher level of attack. The risk equation for an Authentication Level attack can be derived as the product of probability of an authentication level attack, vulnerability caused due to the threat, and the impact of the attack.

$$Risk = \sum_{x=1}^{n} x\left[P(F) * [P(i) + P(e)]\right] * \left|\sum_{i=0}^{n} Ei * \sum_{j=0}^{n} Rj\right| * \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.4}$$

4.3 Hypervisor Level Attack:

Under hypervisor level attacks, the attacker can either be a malicious user on the guest Virtual Machine (VM) or a malicious user who has gained access to the host VM through a sniffing attack or man-in-the-middle attack [5]. The hypervisor is merely an operating system designed with the purpose of abstracting hardware from one or more Virtual Machines running above it. The hypervisors use hypercalls to communicate between VMs and the host OS.

These hypercalls can be modified on either end of the VM to initiate and terminate connections. Unlike an authentication level attack, hypervisor level attacks cause a much more severe impact on the network. As, mentioned above, an attacker, to perform any changes on the hypervisor, first needs to gain access to the host machine or the guest machine to launch an attack and disrupt services.

In case of hypervisor level attacks, the attacker tries to modify the existing code or exploit the defects in the existing code used to communicate between VM's and gain control over the communication. Therefore, the attacker can get more resources from the host VM than the allocated amount of resources. Thus the vulnerability to such an attack is lesser than that of the authentication level attacks since the attacker first has to gain control over the VM to perform any modifications. This type arises a potential threat for the virtualized environment. Thus hypervisor level attack can be represented as,

$$P\ (H) = \sum_{x=1}^{n} x\left[P(C) * \left(P(H, G)\right)\right] \qquad (4.5.1)$$

Where, *P (H)* = Probability of successful hypervisor level attack, *P (C)* = Probability of successful attack on the communication channel of the hypervisor, *P (H, G)* = Probability of

attack being carried out on either host VM or guest VM or both, *x*= Number of users successful in trying to attack the communication channel of the hypervisor.

4.3.1 Probability of attack during communication between host VM and guest VM:

In a hypervisor level attack, the main focus of attack for an attacker is the communication channel between the host and the guest VM. During this communication, the guest VM initiates a communication channel requesting for resources. This request is received as an interrupt by the host VM and, based on the priority of the interrupt, the host VM will allocate resources for the requested task to be performed. The host sets up a channel where resources are allocated and controls are transferred to the guest VM and this is called an event channel. The event channel is responsible for proper communication between the host and the guest VM

The event channel maintains all the values of buffers, registers, and page tables required by the guest VM to operate. A malicious guest VM can trace these values to take advantage of the event channel to launch an attack on the network. Apart from an attack on the event channel, an attacker can also exploit the defects in the code of the hypervisor to launch an attack and gain more resources than what is allocated to that VM. *P(C)* can be represented as,

$$P(C) = P(E) + P(Dh) \tag{4.5.2}$$

Where, *P (C)* = Probability of attack during communication between host and guest VM, *P (E)* = Probability of attack on event channel, *P (Dh)* = Probability of attack due to defect in hypervisor code. Further, *P (E)* can be written as

$$P(E) = P(A) \,\&\&\, P(S) \tag{4.5.3}$$

Where, *P (E)* = Probability of attack on event channel, *P (A)* = Probability of attack due to authentication on a VM, *P (S)* = Probability of sniffing attack on a VM. As in the previous case of authentication level attacks, the values of vulnerability and impact of hypervisor level attacks are calculated.

4.3.2 Vulnerability equation for a Hypervisor Level Attack

The vulnerability equation for a hypervisor level attack can be written as

$$V(H) = \left|\sum_{i=0}^{n} Ei * \sum_{j=0}^{n} Rj\right| \quad (4.6)$$

Where, *Ei* = Estimated Exploitability value, *Rj* = Estimated Reproducibility value.

$$\begin{cases} i = 0 - \text{requirement of advanced programming concepts and networking concepts to perform an attack} \\ i = \text{n} - \text{Simple programming techniques which can be modified by users to launch an attack} \end{cases}$$

$$\begin{cases} j = 0 - \text{No attacks being carried out} \\ j = \text{n} - \text{Attacks being carried out at regular intervals of time} \end{cases}$$

Such types of attacks are harder to be carried out since the attacker needs prior knowledge of programming the source code. However, bugs within the source code can also be used to launch attacks on the VM. If the developer chooses to use simple techniques to develop the source code, then the users on the VM will be able to compromise the code to affect the performance of the network. The vulnerability value of such an attack is much less, compared to an authentication level attack, since the reproduction of such an attack would be not being easier.

4.3.3 Impact equation for a Hypervisor level Attack

Impact of a hypervisor level attack can be expressed as,

$$I\,(H) = \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.7}$$

Where, *Di* = Estimated value of damage on impact, *Uj* = Estimated value of users affected by attack

$$\begin{cases} i = 0 - \text{No effect on host or guest VM's} \\ i = \text{n} - \text{If both host and guest VM performance is affected} \end{cases} \qquad \begin{cases} j = 0 - \text{No users are affected} \\ \text{j} = \text{n} - \text{All users are affected} \end{cases}$$

This type of attack has a higher impact since the number of users affected will be higher. When there is a modification of source code involved, then the damage to the asset increases. This is because, it requires more time to identify the fault and debug the fault to eliminate the threat. Risk equation for a hypervisor level attack can be derived

Risk = (P (H) * V (H) * I (H))

$$Risk = \sum_{x=1}^{n} x\left[P(C) * \left(P(H,G)\right)\right] * \left|\sum_{i=0}^{n} Ei * \sum_{j=0}^{n} Rj\right| * \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.8}$$

4.4 Kernel and Disk Level Attack:

The kernel level attack can be carried out only when the attacker gains access to the host VM to compromise the security of the entire network. In case of kernel level attacks, the severity of attack is much higher and thus proper measures have to be taken to mitigate such attacks. In kernel level attacks, the attacker gains control of the host VM by either performing an

authentication level attack or a combination of authentication level attack and a hypervisor level attack on the host VM [6].

Once the attacker gains access to the host VM, rootkits are installed on the kernel of the host VM. The rootkits are similar to worms or viruses which are short programs that work in the background away from the administrator's domain. The rootkits record values of all the data, such as data communication parameters, of VM's, registry values, buffer levels and relay the data acquired to the attacker.

The attacker, with this knowledge, can launch an attack from the remote machine and thus affect the entire communication channel of the environment. With many attacks, such as the denial of resource attack, buffer interception can be carried out with this information. These attacks are much harder to identify since the rootkits act at the root-level or kernel level.

However, the vulnerability of such an attack is much less in value since this attack can be carried only from the host operating system environment. In the driver level of the attack, the attacker uses the information obtained by the help of rootkits to identify the location of arrays. During the data transfer between the VM's and the driver, the attacker can intercept and modify the existing data. Thus, the probability of a kernel and disk level attack can be written as,

$$P\ (K) = \sum_{x=1}^{n} x\left[\left[P(R) + \ P\ (D)\right]\right] \tag{4.9}$$

Where, *P (K)* = Probability of attack on the host VM kernel and disk array attached to machine, *P (R)* = Probability of successful installation of rootkits on the host VM kernel, *P (D)* = Probability of attack on the disk array during data transference and data storage.

4.4.1 Probability of successful installation of rootkits: P(R)

$$P\ (R) = [P(A)\|P(A)\&\&P(C)] + \ P(Dh) \tag{4.9.1}$$

Where, *P (R)* = Probability of successful installation of rootkit, *P (A)* = Probability of attack due to authentication failure on host machine, *P (C)* = Probability of attack on communication channel between the host VM and the guest VM, *P (Dh)* = Probability of attack due to errors in kernel code of the host VM.

Probability of attack on disk array as; the two main regions where data can be intercepted at disk levels are data in flight and data at rest. Thus *P (D)* can be written as:

$$P\ (D) = \sum_{i=1}^{n}[P\ (fi) + \ P\ (ri)] \tag{4.9.2}$$

Where, *P (D)* = Probability of attack on disk array, *P (fi)* = Probability of successful interception of data during flight on the i^{th} attempt, *P (ri)* = Probability of successful attack on data during rest on the i^{th} attempt. Therefore,

$$P\ (K) = \sum_{x=1}^{n} x\left[\left[[P(A)\|P(A)\&\&P(C)] + \ P(Dh) + \sum_{i=1}^{n}[P\ (fi) + \ P\ (ri)]\right]\right] \tag{4.9.3}$$

4.4.2 Vulnerability of a Kernel and Disk Level Attack

$$V\ (K) = \left|\sum_{i=0}^{n} Ei * \ \sum_{j=0}^{n} Rj\right| \tag{4.10}$$

Where, *Ei* = Estimated Exploitability value, *Rj* = Estimated Reproducibility value.

$$\begin{cases} i\ = 0 - \text{ requirement of Complete knowledge of working of kernel and network topology to identify the storage devices} \\ i\ = \text{n} - \text{ security provided at kernel level to modify source code and Sending and storing data in clear text} \end{cases}$$

$$\begin{cases} j = 0 - \text{No attacks being carried out at kernel and disk level} \\ j = \text{n} - \text{Attacks being carried out at both kernel and disk level} \end{cases}$$

Such types of attacks are harder to be carried out since the attacker needs prior knowledge of programming the kernel and also need to be aware of the disk level storage in the network. Root kits can be installed at the kernel to perform such attacks. The kernel and disk attacks involve a combination of several types of attacks. The vulnerability value of such an attack is much less compared to a Hypervisor level attack since the reproducing such an attack would not be possible.

4.4.3 Impact of a Kernel and Disk level Attack

Impact of a kernel level attack can be expressed as,

$$I\,(K) = \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.11}$$

Where, *Di* = Estimated value of damage on impact, *Uj* = Estimated value of users affected by attack

$$\begin{cases} i = 0 - \text{No effect on host or guest VM's} \\ i = \text{n} - \text{If both host and guest VM performance is affected} \end{cases} \qquad \begin{cases} j = 0 - \text{No users are affected} \\ j = \text{n} - \text{All users are affected} \end{cases}$$

This type of attack has a higher impact since the number of users getting affected will most times be 100%. These types of attacks are very slow in nature and thus require a lot of time to detect. Such type of attacks cannot be eliminated completely, however, only measures can be taken to reduce the probability of such attacks occurring. Risk equation is written as,

$$Risk = \sum_{x=1}^{n} x \left[\left[[P(A) \| P(A) \&\& P(C)] + P(Dh) + \sum_{i=1}^{n} [P\,(fi) + P\,(ri)]\right]\right] * \left|\sum_{i=0}^{n} Ei * \sum_{j=0}^{n} Rj\right| * \left|\sum_{i=0}^{n} Di * \sum_{j=0}^{n} Uj\right| \tag{4.12}$$

Chapter 5

RESULTS AND MITIGATION TECHNIQUES

5.1 RESULTS

In the previous chapter the risk equations for various types of attacks were put down to assess risk for different kind of threats found in a Para-Virtualized environment. It is important to analyze these risk equations and provide proper reasoning to validate them.

The risk equations can be validated theoretically using a simulation tool such as MATLAB. The MATLAB code was written to understand the behavior of each of the attack by considering a normally distributed random value set. These random values can be seeded as per the user and network requirements, hence providing the user with flexibility of controlling any variable at all times. The random variables can be seeded with an upper limit and a lower limit thus providing a complete range of all possible values. In all the graphs that are discussed, the attack attempt is plotted along the x-axis and its corresponding value of attack is plotted along the y-axis. The average value of each attack attempt is depicted in the graph.

5.2 Analysis of Authentication Level Attack

The authentication level attack mainly considers the type of attack and point of attack. As mentioned in the previous chapter the risk equation considers the probability of the attack, vulnerability to the threat, and the impact of such an attack on the physical machine.

The total risk equation for the authentication level attack is expressed in equation 4.1.1

The probability of an authentication level attack involves the parameters of failure of authentication method, probability of attack due to an internal user and probability of attack due to an external user. The simulation for this type of attack is considered by assigning a random variable list for all the three types of probabilities and the following are the graphs obtained.

5.2.1 Probability of Authentication Level Attack

From Figure 10, it is evident that the probability of attack due to an authentication level is very high. This is because the authentication attack is a fairly simple form of attack. To perform this attack the attacker does not need a great deal of expertise in coding. The attacker can perform this type of attack even with a simple random key generator.

The graph shows the number of attacks to the probability of a successful attack. The probability of an authentication level attack depends on the attack by an internal user and attack by an external user. Hence it is essential to take a look at both the graphs to understand the total probability of the authentication level attack.

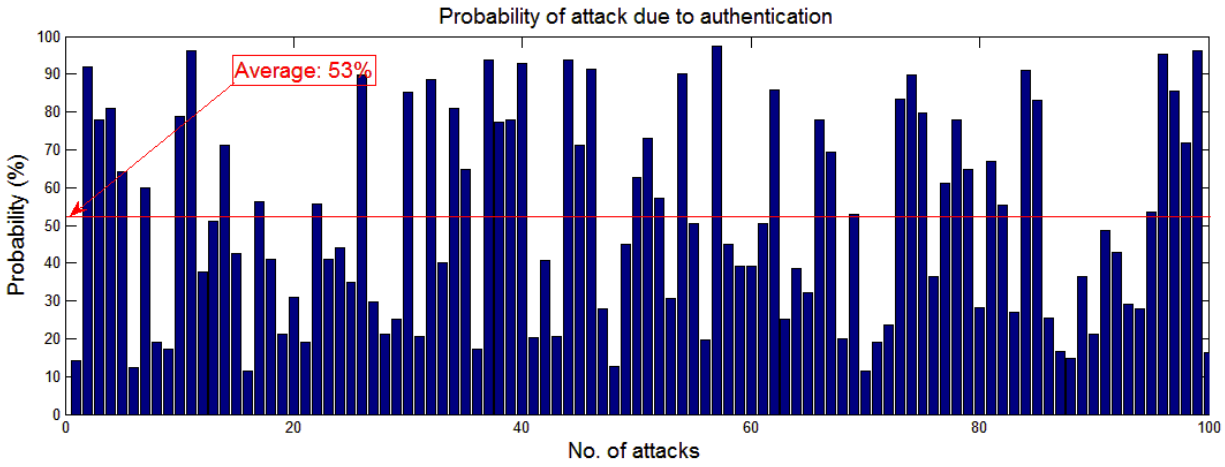

Figure 10: Probability of Attack due to Authentication Level Attack

The two figures 11 and 12 indicate the probability of attack due to an internal and external user, respectively. These are based on the equations 4.1.3 and 4.1.4 as discussed in Chapter 4. The samples are considered for 100 attack attempts and random values are chosen to simulate these graphs. From the graphs, it is clear that the attack due to an internal user would be much higher than an attack due to an external user. This is due to the fact that, if an attack is carried out on the inside of a Virtual Machine, then the attacker would have more access to the resources of the machine directly, and the machine is more vulnerable against these attacks.

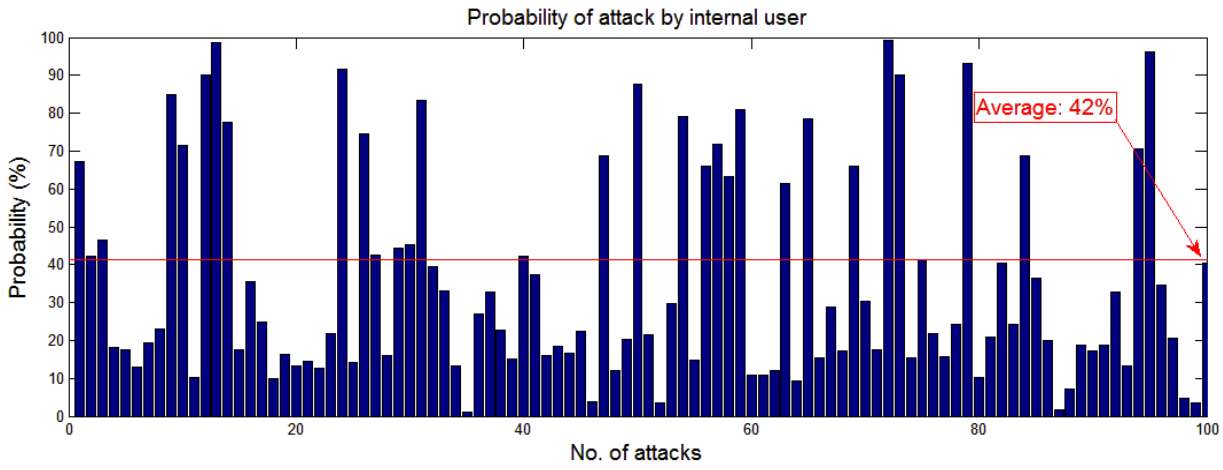

Figure 11: Probability of Attack due to an Internal User

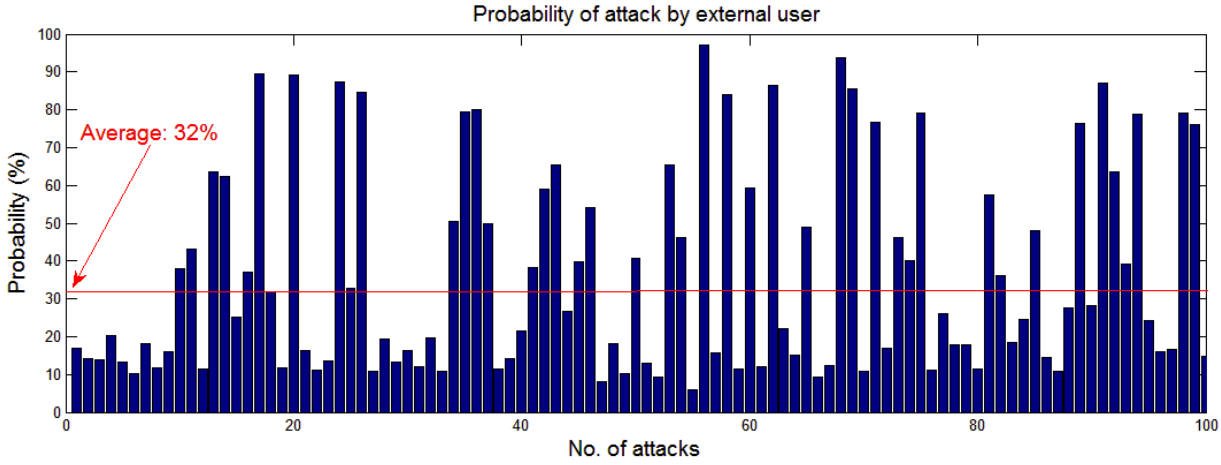

Figure 12: Probability of Attack due to an External User

5.2.2 Vulnerability due to an Authentication Level Attack

The figure 13 shows that the Virtual Machine is more vulnerable to an authentication level attack because this form of attack can be carried out very easily. The plot considers the number of attacks on the x-axis and the vulnerability due to each attack attempt on the y-axis. The equation 4.2 is used to obtain the above graph. The graph shows that, on average, 50% of the time the attack is successful. This proves that any Virtual Machine would be vulnerable to such forms of attack. Hence, proper care needs to be taken to avoid such attacks.

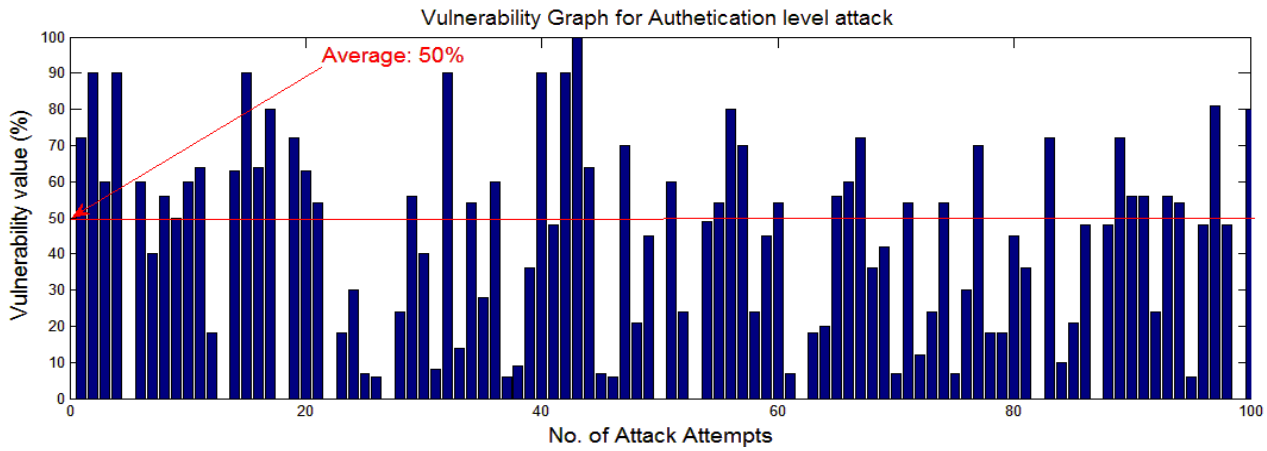

Figure 13: Vulnerability of Authentication Level Attack

5.2.3 Impact of an Authentication Level Attack

Figure 14 describes the number of attack attempts made to the impact of each attack attempt. The impact of an authentication level attack is considerably less as compared to the other forms of attack due to the fact that the authentication level attack does not disrupt the operation of the entire physical machine. Although the authentication level attack has a very high vulnerability value, the impact of this attack is considerably less and thus, with proper security gates in place, this form of attack can be mitigated. The impact equation is discussed in equation 4.4 from chapter 4.

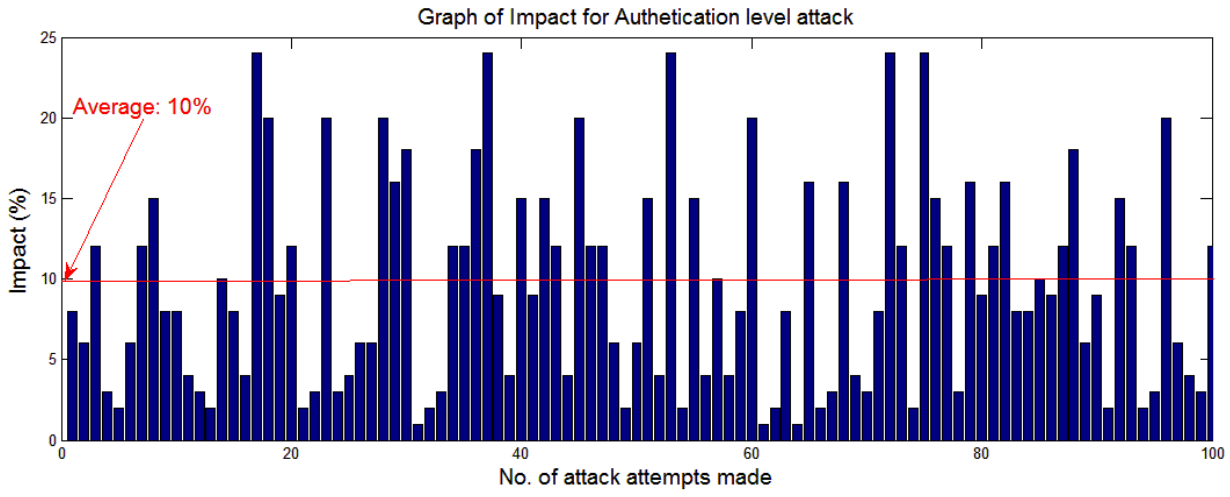

Figure 14: Impact of an Authentication Level Attack

5.2.4 Risk of an Authentication Level Attack

The risk is generally calculated as the product of threat, vulnerability and impact of an attack. All the three components are measured individually by considering the parameters that affect them and the corresponding values of these parameters are substituted in the equation to obtain the total risk value. With these risk values, a graph is generated to assess the risk from a particular threat. Figure 15 shows the graph generated from the equation 4.5 in chapter 4. The probability of risk is considered by increasing the value of number of users linearly.

The risk of an authentication level attack being carried out is comparatively on the higher side due to the fact that the machine has a vulnerability value for these forms of attack. Although they are more prone to these forms of attacks, the impact of this attack would be much less and thus would not be a major factor during the risk assessment process. However, it is necessary to assess the risk of such attacks because of the fact that these forms of attack can be used to launch more impactful attacks, such as the hypervisor level attack.

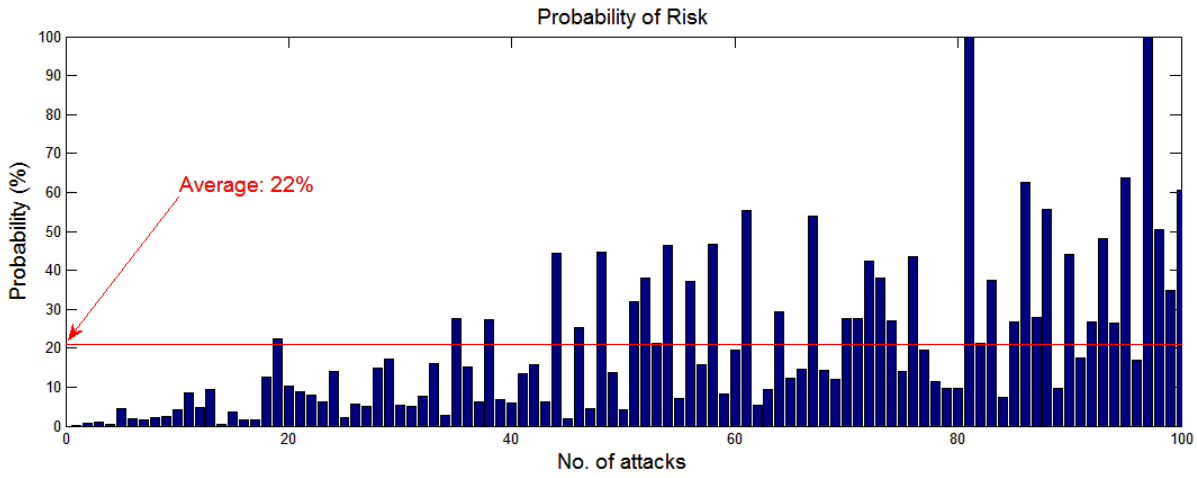

Figure 15: Risk due to an Authentication Level Attack

5.2.5 Mitigation of Authentication Level Attack

The attack mitigation is a technique to identify the impact of any given threat and provide adequate security measures to reduce the impact of the threat. It is a known fact that no threat

can be completed mitigated. However, the impact of any threat can be reduced drastically with proper security measures.

As per the risk equation for authentication level attack in chapter 4, the following techniques can be used to mitigate the authentication level attacks.

1. Mitigating attack caused due to Internal Users: Stringent policy measures and logging would help the network administrators to reduce the impact of any internal attacks on the Virtual Machines. Educating the users regarding the security compliance and risk management would greatly assist in any chances of attack caused by internal users [20].
2. Mitigating attack caused due to External Users: The use of firewalls and authentication servers would reduce the attack rate for an authentication level attack in case of an external user trying to attack the physical machine. Deployment of security management systems such as SSH for console access and filter rules to limit the administration from the outside of the network would make sure only legitimate users are accessing these Virtual Machines and thus the external attacks can be minimized.
3. Zoning of Virtual Machines: This technique can be used to eliminate the threat of sniffing attacks and limiting the number of users accessing the Virtual Machines. Auditing any changes made on the server would assist the administrator in tracking the changes made and the type of change made on the Virtual Machine. This greatly reduces the time to detect fault and isolate this fault from spreading across the server and causing more damage [21].

It is also necessary to find the perfect balance between the security gates and the availability of resources, since security to a machine comes at a price. If there is high security and not enough availability of data to the users then it would not be of any help and, hence, such measures are not implemented within the network.

5.3 Analysis of Hypervisor Level Attack

The hypervisor level attack can be defined as an attack made to gain control over the hypervisor in a virtual environment. Once, the attacker gains control, attacks such as spoofing and denial of resources can be carried out and degrade the performance of the Virtual Machine. Therefore, it is necessary to understand and assess the risk involved in such attacks. The graphs are drawn by considering the number of attack attempts made on the x-axis to its corresponding value on the y-axis.

5.3.1 Probability of a Hypervisor Level Attack

Figure 16 shows the probability characteristics of a hypervisor level attack. This form of attack has a lower probability of success than the authentication level attack due to the fact that to perform a hypervisor level attack the attacker must have gained control over the virtual or host machine. In addition to the control, the attacker must also have prior knowledge of modifying hyper calls to launch a successful attack on the hypervisor. Thus the probability of a successful attack on the hypervisor is around 18%.

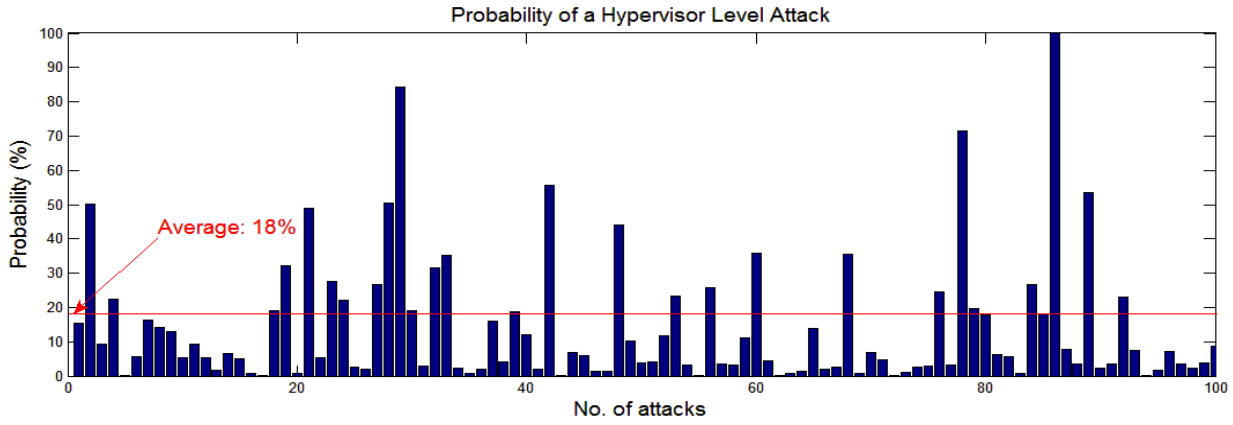

Figure 16: Probability of a Hypervisor Level Attack

5.3.2 Vulnerability for a Hypervisor Level Attack

Figure 17 shows the vulnerability graph due to a hypervisor level attack. The vulnerability due to a hypervisor attack is much less than to the authentication level attack because of the complexity involved to perform this attack. The average vulnerability value is around 25% which is much less than the authentication level attack.

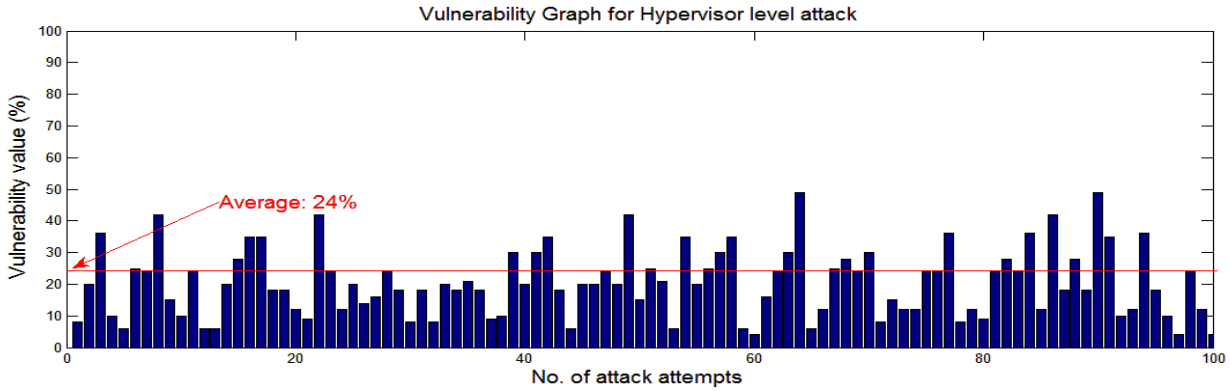

Figure 17: Vulnerability due to a Hypervisor Level Attack

5.3.3 Impact of a Hypervisor Level Attack

The impact graph from figure 18 shows the average value of a hypervisor level attack is much higher than the authentication level attack since it involves two or more Virtual Machines. The average impact of a hypervisor level attack is around 30% and adequate protection has to be employed to reduce this attack.

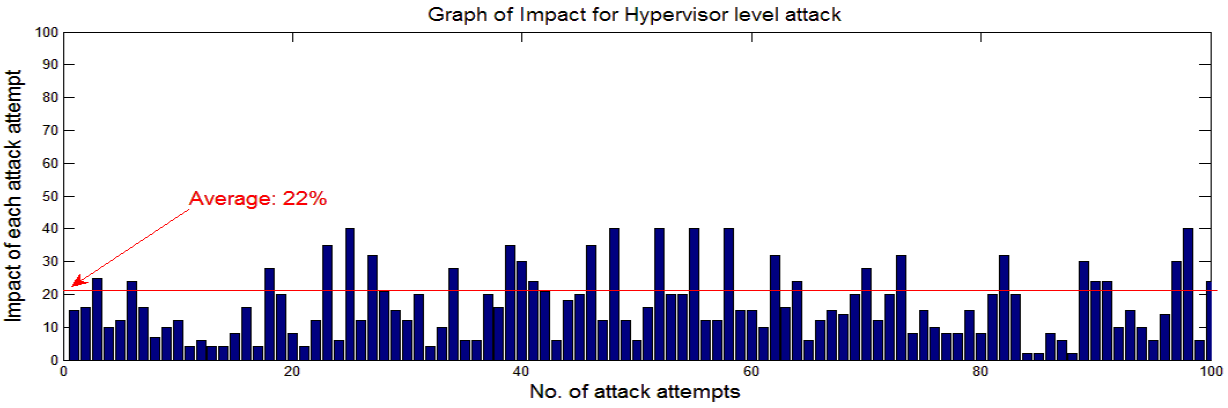

Figure 18: Impact for a Hypervisor Level Attack

5.3.4 Risk of a Hypervisor Level Attack

The Figure 19 depicts the average risk assessed for a hypervisor level attack. The average risk level of a hypervisor level attack is around 25% which is quite a bit lower than the authentication level attack .Although the risk involved is less than the authentication level attack, the impact of such an attack is much higher and thus could lead to a major security flaw if overlooked.

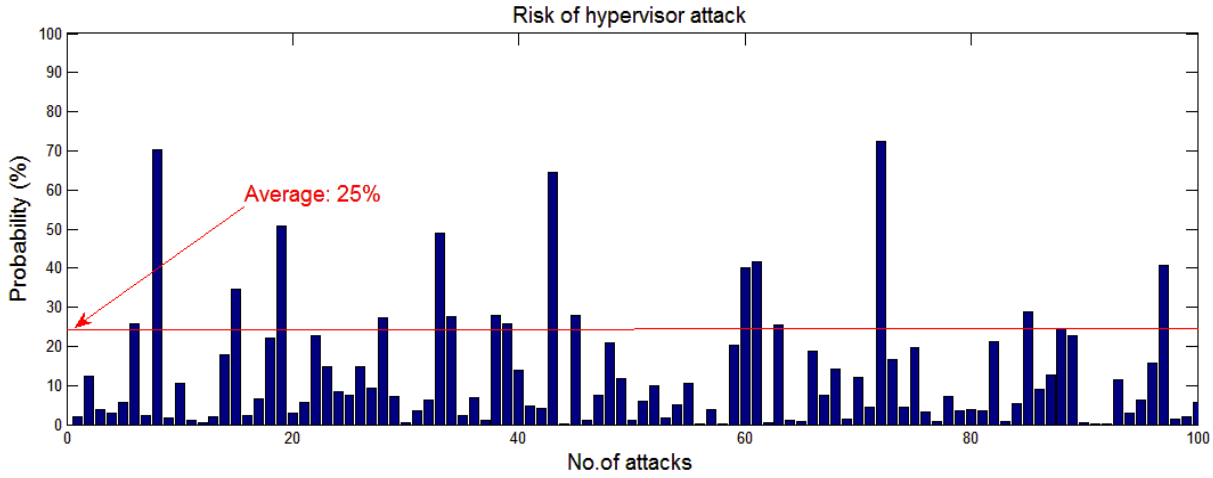

Figure 19: Risk due a Hypervisor Level Attack

5.3.5 Mitigation of Hypervisor Level Attack

From the risk equation for the hypervisor level attack from Chapter 4 the following techniques to reduce the impact of a hypervisor level attack.

1. Mitigating Authentication Level Attacks: The hypervisor level attacks can be mitigated by the use of proper authentication and authorization for legitimate users. Dynamic allocation of resources can be employed. However, this could lead to a denial of resource attack if an illegitimate user overloads the Virtual Machine. This issue can be resolved by limiting the resources to one particular Virtual Machine.
2. Mitigating Sniffing Attacks: The attacker normally would try to gain access by sniffing the conversation happening between VMs, and thereby performing a man-in-the-middle attack.

To avoid such an attack the communication between the Virtual Machines and the hypervisor should be encrypted. It is also essential for the administrators to make sure that the timeline for configuration is set so that the default settings of any Virtual Machines are not being used. This method ensures that the transactions between the Virtual Machines are not the same and the attack rates of sniffing attacks are reduced.

3. Software Maintenance: Maintenance of the operating systems which include updating the monitoring system, patching the operating system with security updates and installation of virtualization aware tools are needed to be done frequently to reduce such types of attacks. The hypervisor layer attacks can be reduced further by encapsulating the hypercalls and providing a separate secure block of addresses for the communication between the guest VM and host VM [12].

5.4 Analysis of Kernel and Disk Level Attack

Kernel and Disk level attacks are carried out from the host Virtual Machine. The kernel level attacks are carried out basically on the kernel of the host operating systems. The attacker gains control of the host Virtual Machine and installs a small program which gains information from the host machine kernel and relays that information to the remote attacker. The users on the host machine are unaware of this program that runs on the kernel, and with this information that is relayed, the attacker launches the attacks from the outside. With this information the attacker will also be able to identify the locations of the disks on the network and would be able to access them, thereby enabling the attacker to modify or destroy data.

5.4.1 Probability of a Kernel and Disk Level Attack

The probability of a kernel and disk level attack on a Virtual Machine is quite low as compared to the other forms of threats. This is due to the fact that the kernel level threats need to

be performed on the host Virtual Machine to be effective and require a combination of several attacks. Figure 20 suggests a probability of around 14%. The kernel and disk level attacks cannot be repeated at regular intervals since they need a very large amount of time to perform one such attack. The attacker also needs to have a sound knowledge about the working of the kernel to make any modifications to the code to disrupt services.

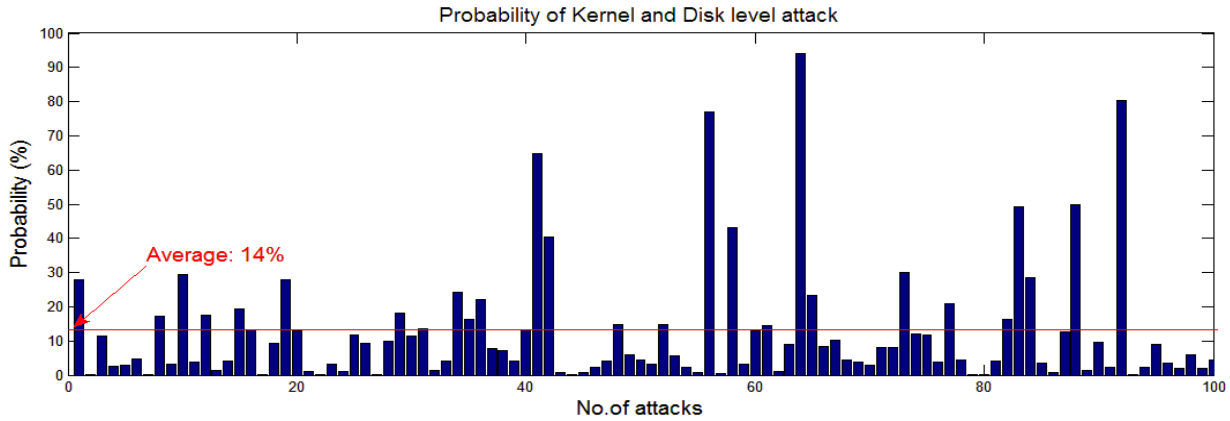

Figure 20: Probability of Kernel and Disk Level Attack

5.4.2 Vulnerability due to a Kernel and Disk Level Attack

The vulnerability value for a kernel and disk level attack is very low since this type of attack needs to be carried out in conjunction with several other forms of attacks. Thus the success rate of such an attack is very low. From Figure 21 shows that the attack success rate is less than 5% for every 100 attempts and thus the vulnerability to such an attack is very low.

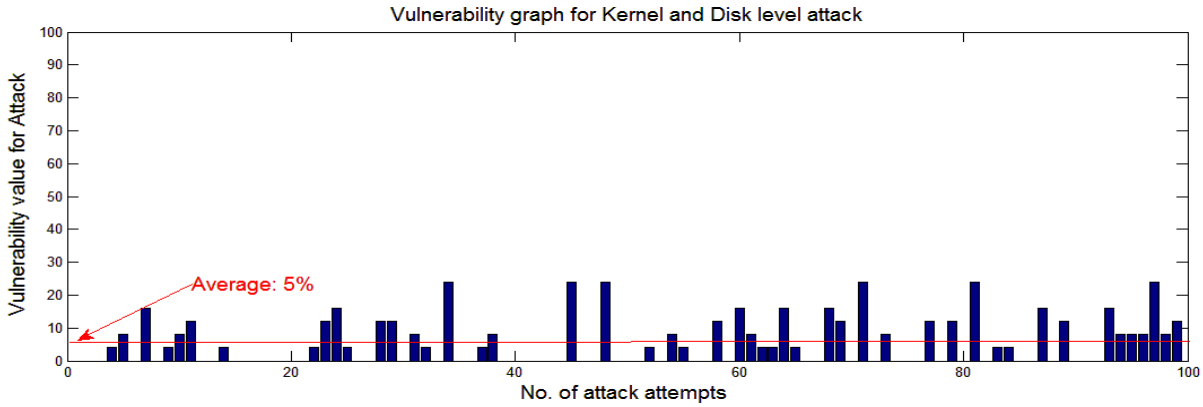

Figure 21: Vulnerability of Kernel and Disk Level Attack

5.4.3 Impact of a Kernel and Disk Level Attack

Figure 22 shows the impact due to kernel and disk level attack. Equation 4.11 from chapter 4 is used to obtain this graph. The impact of a kernel and disk level attack is relatively higher than any forms of attack in a Virtual Machine. This is due to the fact that this form of attack is not detectable to the user and the attacker is at a remote location. Once the attacker is able to perform this attack successfully it would be able to completely gain the access of the virtual environment. The impact level of this form of attack is around 60% which essentially means every time a successful attack happens on the Virtual Machine; it affects at least 60% of all the operations that is being performed on the Virtual Machine.

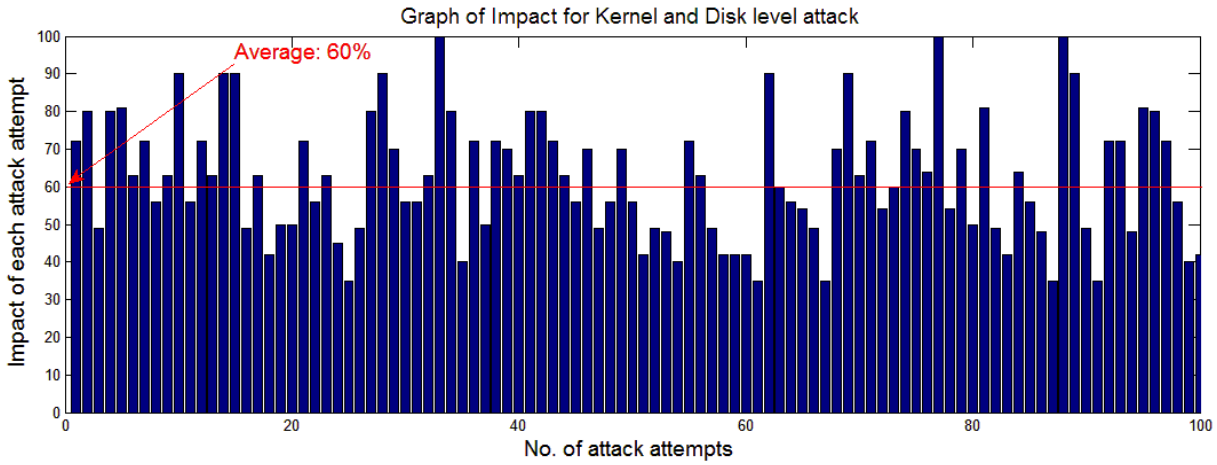

Figure 22: Impact of Kernel and Disk Level Attack

5.4.4 Risk due to Kernel and Disk Level Attack

Figure 23 shows the graph for the probability of risk due to a kernel and disk level attack. This is generated using the equation 4.12 as discussed in chapter 4. The probability of risk is relatively less as compared to other forms due to the fact that this attack is not very simple to carry out. The success rate of carrying out such an attack is much less and requires a very long time to carry out such an attack. However, once an attack is successful it is very hard to find out the source of the attack and it has a very great impact on the virtual environment.

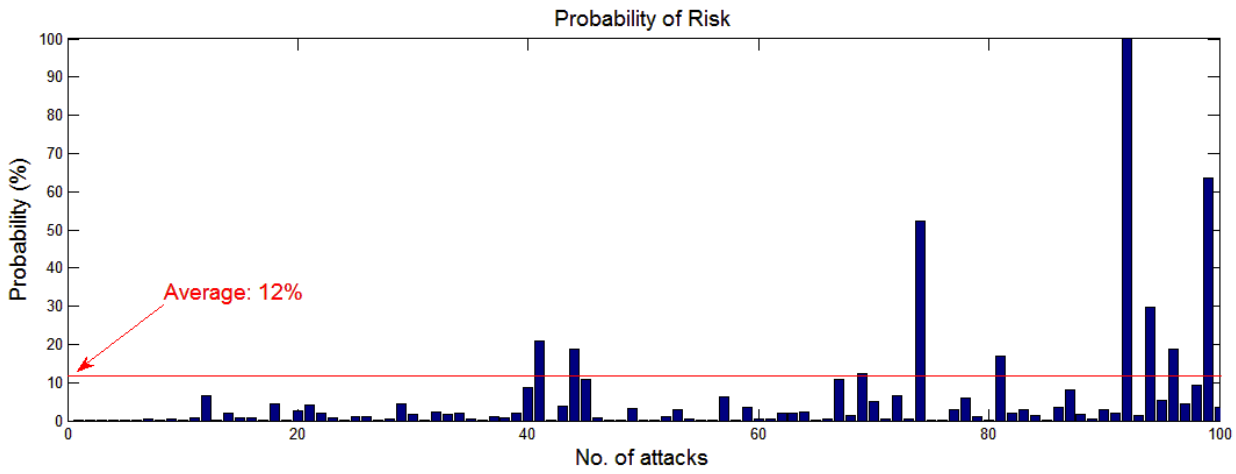

Figure 23: Risk due to a Kernel and Disk Level Attack

5.4.5 Mitigation of Kernel and Disk Level Attack

The mitigation of kernel and disk level attacks involves two different techniques.

1. Mitigating Kernel-based Attacks: The kernel level attacks can be mitigated by periodically updating the firmware on the host Virtual Machine. It is also necessary to update the operating systems with security patches that are available to avoid all security loopholes. Since Xen uses the Linux frame work for installation of the guest and host VM's, securing the kernel with the traditional Linux security features would greatly reduce the effect of a kernel level attack. However, with the installation of Xen VM, a significant modification of the operating system is made. These modifications need to be audited and made sure that the security features are not compromised. In addition to the Linux security features, strategies such as anti-malware, directory service security, file system security, web server security (in case the VM is hosting a web server) should be applied to reduce or eliminate any possible attacks on the VM [21].
2. Mitigating attacks at Disk level: The Disk level attack can be mitigated by the use of secure protocols such as SecureFCP for the data when in flight. The data can also be encrypted at

the transmission side and decrypted at the receiving side thereby adding more security for the data in flight. The data in rest can be secured by randomizing the data storage instead of the traditional sequence storage. Securing data at rest can be done by the use of secure drives at the data centers. The secure drive key management system needs to be time-lined so that the same key is not shared over a long period of time. The most effective method for securing data at rest would be storing the data in multiple copies over different drives. This method had the benefit of redundancy and in case of a possible attack the drive can be powered down without the interruption of service to the legitimate user [24].

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

This thesis provides a basic structure of risk assessment in a virtualized environment by considering Xen as an example. The basic threats, such as the authentication level attack, the hypervisor level attack, and the kernel and disk level attack are studied to analyze the performance aspects such as vulnerability and impact. The results from the studies provide an understanding of the security breaches and the methods of protection that need to be taken to reduce the impact of attacks in a virtual environment. The risk assessment methodology that is considered can be extended to other forms of attacks just by replacing the parameters that constitute the threat. The results obtained clearly show that the vulnerability value of a threat is inversely proportional to the impact of any threat. Thus, based on this inference the security aspects can be planned to avoid any possible threat to a virtual environment.

6.2 Future Work

This study can be extended in the future to other forms of virtualization. It is essential to understand the risks associated with any form of threat before implementing a security feature within the network. There has not been much research in the field of risk assessment and planning, so this study would be an ideal platform for future research in the field of virtualization.

REFERENCES

LIST OF REFRENCES

1. D. Alabi, "Choosing the Right Storage Technology for Your Organization", http://www.storagesearch.com/xtore-art1.html, Cited May, 2004

2. O. Al-Rabayah "Virtualization Concept and History", http://www.remoteitservices.com/content/virtualization-concept-and-history, Cited January, 2010

3. R. Barker, P. Massiglia, "Storage Area Network Essentials: A Complete Guide to Understanding and Implementing SANs" Publisher: John Wiley & Sons, Published: November 2002, ISBN: 0471034452.

4. Dr G. Kbar, "Security Risk Analysis for Asset in relation to Vulnerability, Probability of Threats and Attacks," Proc. Innovations in Information Technology, 2008. IIT 2008. International Conference, Dec.2008, pp 668-672, doi: 10.1109/INNOVATIONS.2008.4781631

5. T. A. Longstaff and Y. Y. Haimes "A Holistic Roadmap for Survivable Infrastructure Systems," Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions, vol 32, issue 2, pp 260-268, Nov.2002, doi: 10.1109/TSMCA.2002.1021113

6. N. Liao, F. Li and Y. Song "Research on real-time network security risk assessment and forecast," Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference, July 2010, pp 84-87, doi: 10.1109/ICICTA.2010.273

7. H.Zang, K.Gu, Y.Li, Y.Sun, D.Meng " A Highly Efficient Inter-domain Communication Channel" CIT '09 Proceedings of the 2009 Ninth IEEE International Conference on Computer and Information Technology, Vol 02, doi: 10.1109/CIT.2009.50

8. T. Clark, "Designing Storage Area Networks: A Practical Reference for Implementing Fibre Channel and IP SANs", Second Edition, Published by Addison Wesley, March 2003

9. J.Tate, F. Lucchese, R. Moore, "Introduction to Storage Area Networks", Published by IBM Redbooks, July 2006

10. D.E Williams, J. Garcia "Virtualization with Xen", Published by Syngress Publishing Inc., May 2007

11. J. Wilkes, "Shared Storage Model", Published by SNIA Technical Council, revised edition, April 2003

12. D. Ruest, N. Ruest, " Beginner's Guide to Virtualization", Published by McGraw-Hill Professional Publishing, January 2009

13. A. Householder, A. Manion, L. Pesante, G. Weaver "Managing the threat of Denial-of-Service Attacks", Published by CERT coordination Center, October 2001

LIST OF REFRENCES (cont.)

14. Microsoft Corporation "Threat Analysis & Modeling", https://www.owasp.org/index.php/Threat_Risk_Modeling, cited March 2007

15. A. Aylward, "The Classical Risk Equation", http://infosecblog.antonaylward.com/2010/05/19/the-classical-risk-equation/, cited May 2010Z

16. C. Takemura, L.S.Crawford "The Book of Xen", published by W. Pollock, October 2009

17. A. Asosheh, B. Dehmoubed, A.Khani, "A new quantitative approach for information Security risk assessment", Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on, pg 222-227, DOI: 10.1109/ICCSIT.2009.5234391

18. J.D. Hietala, "Top Virtualization Security Mistakes (and How to avoid them)", A SANS Whitepaper, sponsored by Catbird and McAfee, August 2009
.
19. B. Hau, "Virtualization and Risk: Key Security Considerations for Your Enterprise Architecture," published by McAfee Corporation, 2007
.
20. C.H. Le, "Protecting Xen hypercalls", The University of British Columbia, July 2009

21. J.Harauz, L.M.Kaufman, B.Potter, "Data Security in the World of Cloud Computing", Aug 2009 Security 7 Privacy, IEEE, vol.7, Issue-4, pg 61-64, Aug 2009, DOI: 10.1109/MSP.2009.8

22. D.Kamat, P. Thaler et al,. "Edge Virtual Bridge Proposal," edited by Hewlett-Packard and IBM, Ver 0, Rev 0.1, Apr 2010, http://www.ieee802.org/1/files/public/docs2010/bg-joint-evb-0410v1.pdf

23. O.B.Sien, A.Samsudin, R.Budiarto, "A new image-database encryption based on a hybrid approach of data-at-rest and data-in-motion encryption protocol," Information and Communication Technologies, Proceedings. 2004 International Conference, Sch. of Computer. Sci., University Sains Malaysia, Penang, Malaysia, Apr 2004, pg 603-604, DOI: 10.1109/ICTTA.2004.1307907

24. Z.Wang, X.Jiang, "HyperSafe: A Lightweight Approach to Provide Lifetime Hypervisor Control-Flow Integrity," Security and Privacy (SP), 2010 IEEE Symposium, July 2010, pg 380-395, DOI: 10.1109/SP.2010.30.

25. Y.C. Cho, J.W. Jeon, "Sharing Data between Processes Running on Different Domains in Para-Virtualized Xen," International Conference, Automation and Systems, ICCAS, 2007, pg 1255-1260, DOI: 10.1109/ICCAS.2007.4406528.