A MODEL TO MITIGATE SECURITY VULNERABILITY OF LIVE MIGRATION IN VIRTUALIZATION

BY

JAMES .N. NDUNG'U

MASTER OF SCIENCE IN DATA COMMUNICATIONS AND NETWORK

KCA UNIVERSITY

2018

A MODEL TO MITIGATE SECURITY VULNERABILITY OF LIVE MIGRATION IN VIRTUALIZATION

BY

JAMES .N. NDUNG'U

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN DATA COMMUNICATIONS AND NETWORK IN FACULTY OF COMPUTING ANDINFORMATION MANAGEMENT AT

KCA UNIVERSITY

MAY, 2018

DECLARATION

I declared that the dissertation was my original work and had not been previously published or submitted elsewhere for award of a degree. I also declared that it did not contain material written or published by other people except where due reference is made and author duly acknowledged.

Student Name: James Njenga Ndung'u Reg. No.: 16/00340

Sign: _____ Date: _____

I do hereby confirm that I have examined the master's dissertation of

James Njenga Ndung'u

And have approved it for examination

Name of Supervisor: _____ Date: _____

Sign: _____

ABSTRACT

The concept of a virtual machine is not new as its popularity and usage has increased overtime. Live migration is a feature in virtual machine that allows resources from one physical server to be moved to another with little or no interruption in the processes of the guest operating system. It is a feature that is widely used in modern data centers, since servers and applications need to be available all time even when there is a system maintenance window. However there is a cost associated with live migration that needs further examining. During live migration data on transit is in clear text and can be intercepted by hackers by performing a man-in-the-middle attack. In order to improve data confidentiality, integrity and authenticity during live migration, we need to address this issue. This dissertation seeks to establish the security vulnerabilities of live migration in virtualization technology by performing a lab simulation experiment and propose a solution to mitigate the problem.

Keywords: Virtualization, Live Migration, Hypervisor

ACKNOWLEDGEMENTS

I acknowledge my supervisors, for their guidance during the development of this research proposal. I further extend gratitude to other members of staff in the University for the support they accorded me as I undertook my studies.

TABLE OF CONTENTS

DEDICATION

I dedicate this research project to my colleagues and family; their support and encouragement has brought me this far.

ACRYNOMS AND ABBREVIATIONS

CPU- Central Processing Unit
IaaS- Infrastructure as a service
PaaS- Platform as a service
SaaS- Software as a service
OS- Operating Systems
VMM- Virtual machine monitor
VM- Virtual Machine
ESXi- Integrated Elastic Sky
KVM- Kernel-Based Virtual Machine
ARP- Address resolution protocol
DNS- Domain name system
SAN- Storage Area Network
NAS- Network Area Storage
ISCSi- Internet Small Computer System Interface
VMFS- Virtual Machine file System
MAC-Media access control
DDoS-Distributed Denial of Service
GNU- GNU's Not Unix
IPS- Intrusion Prevention Systems
IDS-Intrusion Detection System
IT- Information Technology
IEEE-Institute of Electrical and Electronic Engineers
IET-Institution of Engineering and Technology
Vmotion-Virtual Motion
VBox-Virtual Box
SSL-Secure Socket Layer
USB-Universal serial bus
VLAN-Virtual local area network
KB-Kilo byte
MB-Mega Byte
GB- Giga Byte
Vmdk-Virtual machine disk file
HP-Hewlett Packard
DL-Density Line
G8- Generation 8
GHz- Giga Hertz
QEMU- quick emulator
IPsec-Internet Protocol security
NSE-Network Security Engines (NSE)
NSE-H Network Secure-Hypervisor

CHAPTER ONE

INTRODUCTION

1.0 Introduction

Virtualization is increasingly being used in portions of the back-end of Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and SaaS (Software as a Service) providers. The technology has the ability to isolate co-resident Operating Systems (OSs) on the same physical platform as well as perform live migration of VMs from one location to another. Live migration is a feature that enhances systems performance and management and reduces on systems downtime.

1.1 Background of the Study

Virtualization is the process of creating multiple virtual environments on a single server. According to (Bhaskar Prasad Rimal, Eunmi Choi, 2009), virtualization can also be defined as the abstraction of logical resources away from their underlying physical resources in order to improve agility and flexibility, reduce costs and thus enhance business value. In a virtualized environment, VMs can be dynamically created, expanded, shrunk or moved as demand varies. Virtualization does this by using virtualization software, known as a virtual machine monitor (VMM) or hypervisor which separates compute environments from the physical hardware (Noorafiza, Maeda, Uda, Kinoshita, & Shiratori, 2015) and makes it possible to run several operating systems on one computer at the same time.

It has been observed by (Duncan, Creese, Goldsmith, & Quinton, 2013) that, the use of Virtual Machines (VMs) and Infrastructure-as-a-Service (IaaS) has risen dramatically and, according to Gartner, is set to continue rising with a compound annual growth rate predicted to be 41.7% over the four years to 2016. By using Cloud providers, organizations are reducing their capital expenditure on hardware, software and support, however, these same organizations are putting a great deal of trust in the provider offering a safe and secure platform for their data and resources. Virtualization should never be confused with cloud. The confusion with cloud comes

from the fact that cloud computing uses virtualization. In a cloud data center, the large numbers of servers are virtually partitioned, but it is not the virtualization that makes the cloud. Instead, cloud is a means of delivery of shared computing resources, software or data as a service (SaaS) via the Internet. Cloud computing, therefore, allows businesses online access to complex applications and enormous computing resources as well as providing elasticity, scalability and automated management on a pay-per- use basis.

According to (Duncan et al., 2013) hypervisors are divided into two categories. Type one hypervisors, at times called bare-metal hypervisors, are installed directly above the hardware and have direct communication with the hardware, hence eliminating the need for an operating system. These hypervisors act as a medium through which the guest OS communicates with its resources. Type two hypervisors, on the other hand, require a base operating system to be installed, since they are in fact adding the virtualization feature to the base operating system. Although this might be a positive point allowing for further flexibility in regards to policies and configurations, any security issue in the base operating system can affect the entire system.

Modern data centers are virtualized to make them dynamic. Virtualization is therefore extremely well suited to a dynamic cloud infrastructure, because it provides important advantages in sharing, manageability and isolation for instance, multiple users and applications can share physical resources without affecting one another. Therefore data center administrators should be encouraged to have an environment that supports a dynamic infrastructure in an evolutionary new model in order to provide an innovative, efficient and flexible approach in helping aligning IT with business goals.

With the introduction of live migration of virtual machines, the weakening of isolation boundaries protecting machine state has increase to a whole new level(Oberheide, Cooke, &

Jahanian, 2008). Now, instead of physical access or access to a VM, an attacker simply needs snoop on a network where these migrations are occurring. Exposing this information to the network without authentication or confidentiality guarantees a significant security risk.

Originally the physical machines were running a single operating system. The state of the machines were protected by hardware mechanisms. However, attacks were still possible against this model. Nonetheless, gradually the technology evolved to the virtualization, where the state of a machine was no longer protected by hardware, but by a software layer, namely the hypervisor/VMM. This software has its own vulnerabilities that can be exploited. As this new technology was still in its infant stage, Live Migration was introduced as a new feature of virtualization that complicated the already compromised security architecture of virtualization.

1.2 Problem Statement

Migrating operating system instances across distinct physical hosts is a useful feature in data centers and clusters. By carrying out the majority of migration while OSes continue to run(Clark et al., 2005), we achieve impressive performance with minimal service downtimes. It allows a clean separation between hardware and software, and facilitates fault management, load balancing, and low-level system maintenance.

Live migration is an essential feature of virtualization that allows transfer of virtual machine from one physical server to another without interrupting the services running in virtual machine. (Shetty, 2012) observed that disclosed vulnerabilities of live migration pose significant security risks. Because of these security risks the industry is hesitant to adapt the technology for sensitive applications. An attacker can easily hijack the device module process or hypervisor where these migrations occur. If the process is hijacked, the information of the migrated virtual machine including states of operation system kernel, applications and services running within the operating system, the sensitive data currently being used by those applications and even the

inputs from keyboard are accessible to the hackers. The migration functionality implemented by vendors such as XEN, VMware ESXi, and KVM now exposes the entire machine state of a VM to device module which listens to the incoming live migration requests from remote platforms(Oberheide et al., 2008).

Most of the previous works have focused on the implementation of live migration with little or no consideration towards its security. A lot of recent research has focused on improving the security of the hypervisor and securing the intra-migration traffic. The inter-migration traffic security model is still at its infant stage and it is an area that requires a lot of attention since many businesses are moving their services to cloud and inter-migration traffic need to be secure. The major security concern of live migration (Intra or Inter) is that the migration protocol does not encrypt migration data(Oberheide et al., 2008), (Shetty, 2012). All migration data i.e. kernel memory, application state, sensitive data such as passwords and keys etc. are transmitted as clear text. The reasons for plaintext transmission could be two fold. One is the assumption that all these migrations happens inside an already trusted network. Secondly, encrypting live migration data will be expensive in terms of computation and resource utilization. This dissertation seeks to establish vulnerabilities of VMs during live migration for inter-site traffic and propose a secure model for the migration.

1.3 Main objective

The main objective of this study is to develop a model to secure the data-on-transit of a VM during live migration.

1.3.1Specific Objectives

The specific objectives for this study shall be;

1. To review the vulnerabilities of live migration in virtualization technologies.
2. To design model to secure the live migration data.
3. To implement the design model

1.3.2Research Questions

The following research question will inform the study;

1. What are the vulnerabilities of live migration?
2. What are some of the technological solutions implemented to secure live migration?
3. Is the data-on-transit clear text during live migration?
4. Can you intercept traffic data during live migration?
5. Can data-on-transit be encrypted during live migration?
6. How do we ensure security of data-on-transit?

1.4 Significance of the Study

The absence of studies on vulnerabilities of live migration as feature in Virtualization, denies organizations knowledge on the benefits or lack of, integrating Virtualization technologies in their Data Center operations.

1.4.1 Contribution to Body of Knowledge

The contemporary research on live migration so far is performance oriented and security issues have not received much attention. Study in this direction is important since there is scarcity of literature on the existing security measures of cloud and virtualization technologies especially in developing countries such as Kenya.

1.4.2 Contribution to Virtualization Industry

It will enhance the confidence of new entrants and consumers of virtualization technologies since they would have hands on information on how best to implement the technology as well as measures of securing their environment. Nonetheless live migration is still in an early stage of implementation and its security is yet to be evaluated. The security concern of live migration is a major factor for its adoption by the IT industry.

1.4.3 Develop a secure model

There is a clear shift towards delegating computation to the Cloud and virtual systems. To benefit from the full potential of these systems, features like secure live migration seems to be crucial for both system perform ability and availability.

1.5 Motivation of the Study

Virtualization has made a huge impact in a very short time in the IT and networking worlds and has already provided huge cost savings and returns on investments for data centers, enterprises and the Cloud. What seems to be less substantial and lagging is the understanding of virtualization and virtualized environments from a security point of view. Some people think that virtualization is more secure than traditional environments because they've heard of isolation between virtual machines (VMs) and because they haven't heard of any successful attacks on hypervisors. Others think that the new virtualized environment needs security just like traditional physical environments and therefore apply the same long standing approaches to securities that are already in place. The bottom line is that the new environment is more complex, and virtualization approaches added to current networks creates a new network that needs a new approach to security. This should include traditional security as well as additional security for virtualization.

Virtualization has made computers to no longer be idle or performing below their capabilities because there are fewer connected users, or because the hosted application happens to be less demanding than the server can handle. With most virtualization solutions it is possible to move a virtual machine from one physical machine in the environment to another. With physical servers this was originally possible only if both physical machines ran on the same hardware, operating system and processor. In the virtual world, a server can be migrated between physical hosts with entirely different hardware configurations. Live Migration is typically used

to improve reliability and availability; unfortunately several vulnerabilities are disclosed in the implementation of live migration.

1.6 Scope of the Study

The study will focus on vulnerabilities of live migration for VMs. It will test live migration of VMs on VMware ESXi hypervisors. Live migration will be tested between VMs on the same hardware and VMs on different hardware on the same network. The study will only be concerned with active VM migration, or migration that does not interrupt the operation of migrating VMs.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

Today's enterprise level computer resources are so powerful that they often have excess capacity. By virtualizing the hardware and allocating parts of it based on the real needs of users and applications, the available computing power, storage space and network bandwidth can be used much more effectively.

The introduction of virtualization by middleware software called Virtual Machine Managers (VMMs) or hypervisors come with different vulnerabilities. (Sheinidashtegol & Galloway, 2017) noted that these vulnerabilities are adding to previously existed vulnerability of Networks and Operating systems and Applications.

2.1 Related Work

The process involved in performing a live migration includes copying the guest virtual machine memory state and CPU register state from a hypervisor on one server to another. Service downtime is inevitable in migrating ongoing virtual machine. It is obvious that service downtime impacts the quality of live virtual machine migration. It is important to take stock of the uses of virtualization and more specifically on live migration, and to provide a framework for similar studies in future.

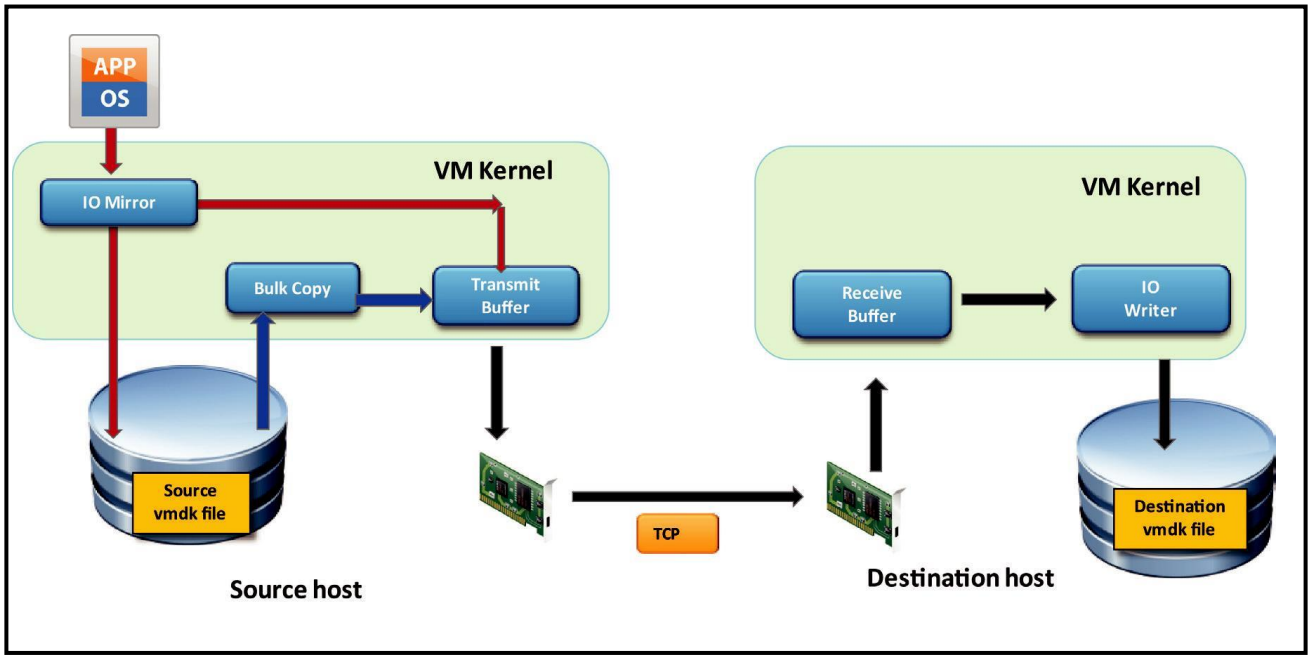
2.1.1 VMotion

VMware VMotion enables the live migration of running virtual machines from one physical server to another with zero downtime, continuous service availability, and complete transaction integrity. It is transparent to users. So how Does VMotion work?

First, the entire state of a virtual machine is encapsulated by a set of files stored on shared storage such as Fiber Channel or iSCSI Storage Area Network (SAN) or Network Attached Storage (NAS)(Study, 2012). VMware's clustered Virtual Machine File System (VMFS) allows

multiple installations of ESX Server to access the same virtual machine files concurrently. Second, the active memory and precise execution state of the virtual machine is rapidly transferred over a high speed network, allowing the virtual machine to instantaneously switch from running on the source ESX Server to the destination ESX Server.

Figure 1: vSphere 5.0 vMotion Data Path during Storage Migration



Source: (Study, 2012)

VMotion keeps the transfer period imperceptible to users by keeping track of on-going memory transactions in a bitmap. Once the entire memory and system state has been copied over to the target ESX Server, VMotion suspends the source virtual machine, copies the bitmap to the target ESX Server, and resumes the virtual machine on the target ESX Server. This entire process takes less than two seconds on a Gigabit Ethernet network. Third, the networks being used by the virtual machine are also virtualized by the underlying ESX Server, ensuring that even after the migration, the virtual machine network identity and network connections are preserved. VMotion manages the virtual MAC address as part of the process. Once the

destination machine is activated, VMotion pings the network router to ensure that it is aware of the new physical location of the virtual MAC address. Since the migration of a virtual machine with VMotion preserves the precise execution state, the network identity, and the active network connections, the result is zero downtime and no disruption to users.

2.1.2 Performance Impact of DDoS Attacks on Three Virtual Machine Hypervisors

A DDoS attack is one that attempts to saturate a targeted resource of the victim's system. The goal of DDoS attacks is to make these targeted resources or services unavailable to the legitimate user by flooding the victim's machine with requests. (Sheinidashtegol & Galloway, 2017) observed that machine running the VBox hypervisor became completely unresponsive under CPU DDoS attack.

Using SSL protocol, the attacker attempts to saturate the CPU by sending encrypted packets to the victim's system. Since the pressure decryption can put on a CPU is fifteen times the pressure of encryption, given the appropriate amount of packets, the victim's CPU becomes flooded and hence saturated by the attack.

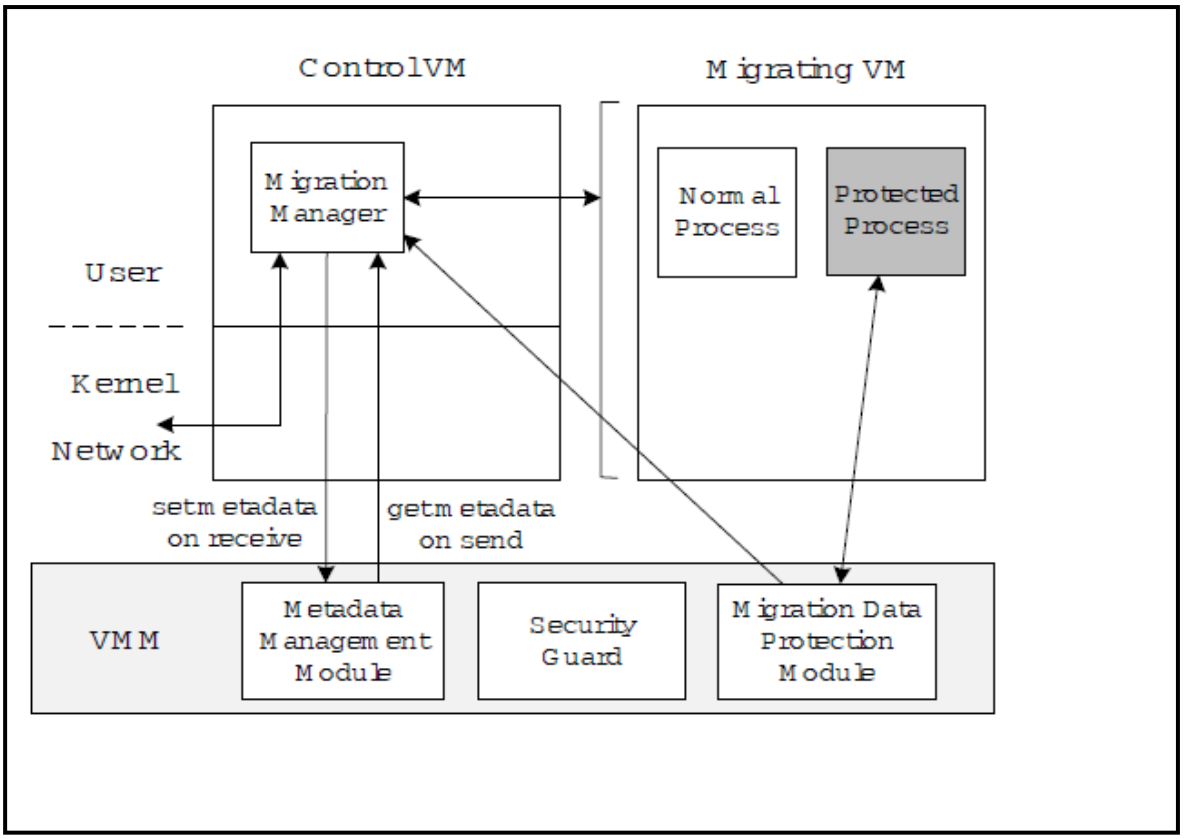
Knowledge Gap- The experiment targeted only DDoS attack on VMM. The researchers clearly demonstrated how a DDoS attack could happened on a VM, and did not mention if the same attack can happen to a VM in Live Migration.

2.1.3 PALM: Security Preserving VM Live Migration for Systems with VMM-enforced Protection

AVM cannot be straightforwardly migrated if there are protected processes inside. As the VMM extends its protection to the process granularity, the VM is no longer a black box. Live migration of a VM for these systems may impose critical security issues. Simply using cryptographic means and hashing to protect the sensitive data and meta-data is not enough. Critical issues lie in the VM live migration, in which the migrating VM is still running while the migration is in

process. Time-of-check to time-of-use (TOCTTOU) [2, 13] attack and replay attack could be launched if the protection for migration is not carefully designed.(Zhang, Huang, Wang, Chen, & Zang, 2008)presented a security preserving system that guaranteed security strength during and after the migration.

Figure 2: Overall Architecture of a Secure VMM



Source:(Zhang et al., 2008)

A prototype system called PALM (Protection Aegis for Live Migration of VMs) was implemented, which is based on Xen VMM and GNU Linux on x86 architecture. The protected pages were migrated the same way as normal pages from the perspective of the migration manager, a phase is added to get and migrate metadata that Xen provides on the sender side, and receive it and pass it to Xen on the receiver side.

Research Gaps- The framework is based on hypervisors included with Network Security Engines (NSE) and hence the whole system is called Network Security Engine-Hypervisors (NSE-H). NSE includes firewall, intrusion detection systems (IDS) and intrusion prevention system (IPS) to provide security to virtualized environment and to eradicate intrusions occurring in virtual networks. The NSE firewall works in a state-full way and it includes built-in intelligent packet processing capabilities.

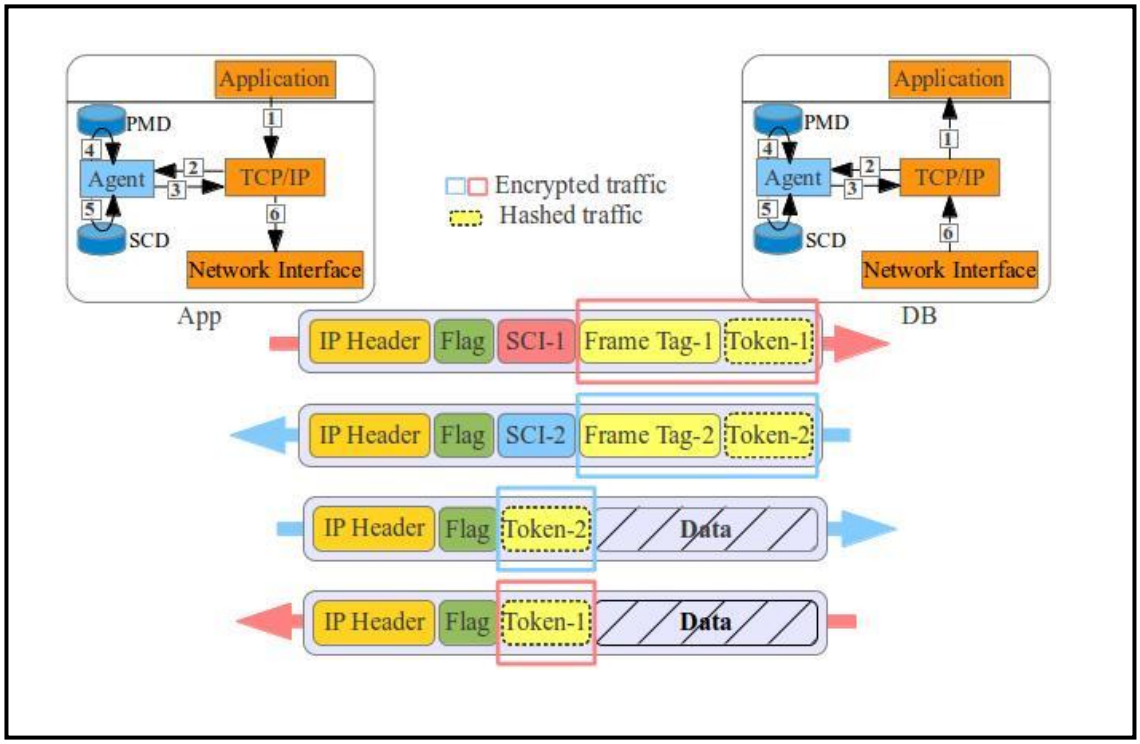
The framework enables the traditional security approaches like firewall, IDS, IPS present inside NSEs to work in context of live migration. It transfers the security context along with migration data so that the VM can be restored at the destination. Despite the fact that the VMM has inbuilt protection software, this still could not guarantee the security of data on transit during the live migration. VMM-enforced protection systems extend protection to the processes in the virtual machine (VM), it also breaks the mobility of VMs since a VM is more closely bound to the VMM.

2.1.4 Secured Architecture for Inter-VM traffic in a Cloud environment

In a traditional IT environment, network traffic can be monitored, inspected and filtered using a range of server security systems to try to detect malicious activity. But the problem with virtualized environments provides limited visibility to inter-VM traffic flows. This traffic is not visible to traditional network-based security protection devices, such as the network-based intrusion prevention systems (IPSs) located in network, and cannot be monitored in the normal way.

(Benzidane, Khoudali, & Sekkaki, 2013)aims to control and analyze a particular traffic which is the inter-VM traffic by introducing a security structure characterize by a frame called frame tag.

Figure 3: IP Packet Processing in a Secured Architecture



Source: (Benzidane et al., 2013)

It is added via a sender agent in the payload of the IP packet ensuring high-level integrity, so that the receiving VM's agent will detect, analyze and authenticate the incoming traffic then respond by accepting or rejecting this IP packet according to the compliance of the information on the frame tag. The proposed solution gives identity to this particular traffic. This identity is about the where and the who sends the request. The frame tag holds the proper credentials which are the tenant and the application that sends the IP packet, providing data origin authentication, and integrity.

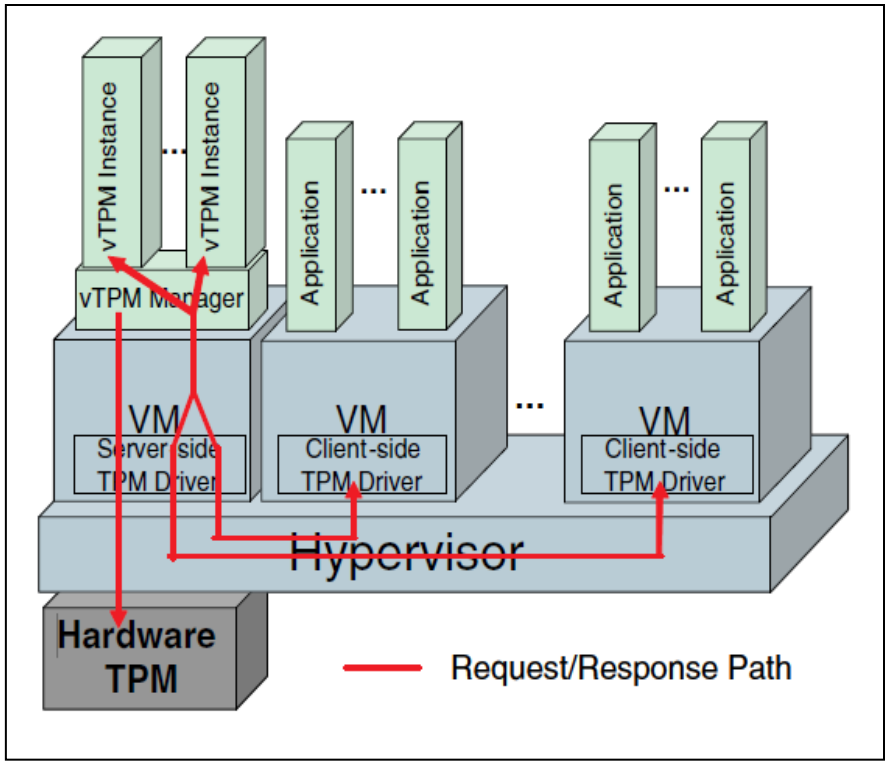
2.1.5 An Improved vTPM Migration Protocol Based Trusted Channel

One of the most important benefits of virtualization is Virtual Machine (VM) migration. While the semantics and performance of live VM migration are well explored, the security aspects have received very little attention. (Wan, Zhang, Chen, & Zhu, 2012) Observed that extension of

trusted computing to virtual systems using vTPM's allows applications in the VM to use the vTPM for secure storage and reporting platform integrity. The TPM is a security specification defined by the Trusted Computing Group. Its implementation is available as a chip that is physically attached to a platform's motherboard and controlled by software running on the system using well-defined commands.

Trusted computing and Trusted Platform Modules (TPMs) facilitate secure storage of sensitive information and allow verification of system integrity. The application of trusted computing into virtual systems could be significantly enhance system security. This extension of TPMs for use with virtual machines is called TPM virtualization and it results in a virtual TPM (vTPM) design.

Figure 4: vTPM architecture



Source:(Schiffman, Berger, Sailer, & Goldman, 2006)

(Schiffman et al., 2006) propose to virtualize the hardware TPM by equipping every VM with its own software TPM. This software-TPM only uses the underlying hardware TPM for certain operations.

With the incorporation of trusted computing into virtualized systems, some secure migration protocols have also been introduced along.

Knowledge Gaps- Secure VM-vTPM: Live migration is not supported in vTPM based migration protocol rather it supports migration of suspended VMs. Keys of vTPM are also stored outside the TPM, therefore prone to leakage and unauthorized modification. The vTPM state is also migrated, so it is an overhead and it increases the migration time.

CHAPTER THREE

METHODOLOGY

3.0 Introduction

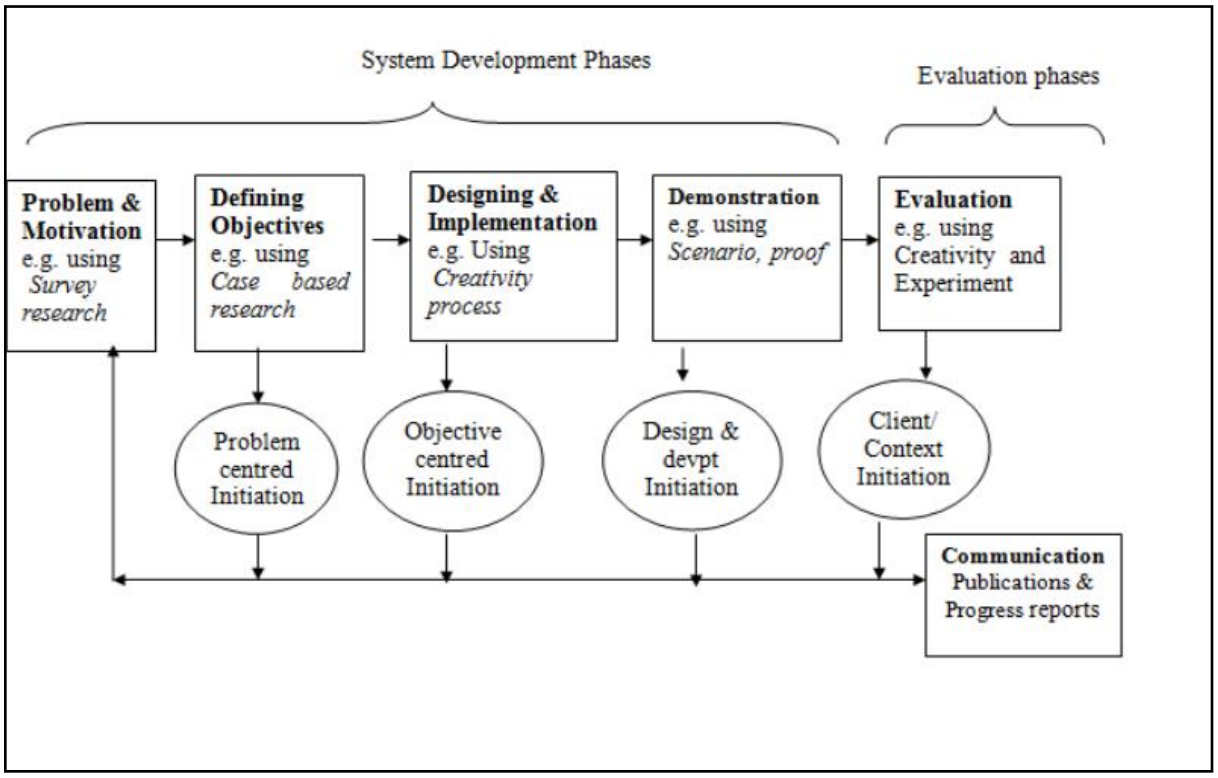
This chapter clearly defines the research methods to be used to conduct the study. The methodology explains how the necessary data and information to address the research objectives and questions was collected, presented and analyzed. Reasons and justifications for the research method is given.

3.1 Methodology

In order to achieve my main objective of designing and implement the model, simulation was used to perform live migration of VMs. VMs were created on each hypervisor and live migration tested for vulnerabilities. Since getting some of these software and hardware was costly, simulating the real networks was a good alternative. The conjecture was that the analytical model is still a reasonable representation of the real system. Since the study was depicted to solve organizational problems and to fully test the model, design science research methodology helped in prototyping and evaluating the resultant model(Nyaga, 2016).

1. Creativity process was used for guiding the designing and development of the proposed secure live migration model as well as evaluating its effectiveness.
2. True Experimental design was applied for guiding the execution of evaluation activity.
3. Observational study was adopted to carry out problem analysis and formulation by studying the live migration experiment and collecting data by changing different parameters solutions for different virtualization technologies

Figure 5: Design Science Research Methodology



Sources: (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

3.2Concepts Combination

Concept combination was also incorporated to help achieve the objective of reviewing the vulnerabilities of live migration in virtualization technologies. It entailed integrating aspects ideas retrieved from pertinent information and scholarly articles live migration. The sources of this information included systematic literature review from online databases e.g. IEEE, Google Scholar and IET, Lab simulation results as well as problem analysis results of different live migration technologies that are not more than 5 years old.

3.3Data Analysis

A lab was prepared for test bed. The experiment was conducted using servers, switches and laptops connected to simulate a network. VMMs were installed on different server and VMs created and moved across the different hardware. Iperf and Wireshark were used to capture traffic as it passes across. Data analysis activity comprised three main tasks.

1. Pre-experiment data analysis-Live Migration with no security,
2. Post-experiment data analysis -Secure Live Migration,
3. Comparison of pre-experiment and post-experiment results.

CHAPTER FOUR

DATA ANALYSIS FINDINGS AND DISCUSSION

4.0 Introduction

Perfuming simulation of live migration in a testing environment, helped in the data collection. Two physical servers were installed with VMM and VMs created. Wireshark software was used for packet capturing an analysis.

4.1 Low-Technology Attacks

The purpose of this sub-topic was to achieve the following objective and associated research questions;

Objective 1-To review the vulnerabilities of live migration in virtualization technologies.

Research Question 1-What are the vulnerabilities of live migration?

Research Question 2-What are some of the technological solutions implemented to secure live migration?

The ideal scenario would be to detect attacks as early in the kill-chain as possible, either predicting an attack or detecting it in the early stages, since this minimizes costs and actual exposure. Before we can hope to achieve this, we need to know what sort of attacks can be performed, how we can detect them and what measures we can implement to prevent their future use.

Prior to identifying migration as a key source of exposure to attack, analysis need to be carried out for both low-technology and high-technology methods of attacking a VM without being detected. An obvious method of obtaining a virtual machine's data would be to make a duplicate, then boot that duplicate either on the same hypervisor or on a hypervisor in a remote location. Making duplicates using the hypervisor however, will leave audit trail entries on the hypervisor making it easy to discover that a VM has been compromised.

One could copy the data store of a VM using the data store Browser in VMWare. This would have the effect of looking like a backup was made and therefore might fit in with the established modus operandi of the organization. The actions carried out using the data store browser are all logged and this log can be monitored in real time using scripts to detect such events. This form of attack would be detectable as it happens making it highly overt. One can access the datastore server directly, for example by iSCSI or USB connections to the Network-Attached Storage (NAS) storage location. Connections via iSCSI and USB are logged by the NAS and often this facility for USB must be manually enabled, making this an easily detectable action in a well-run network.

If USB access to the NAS is not needed, this can be manually unplugged by systems administrators, further reducing the opportunity for unauthorized access. Additionally, the NAS could be assigned a VLAN which is available to a limited number of computers, which are configured based on specific need. By restricting the access to the NAS to the servers that specifically require it, the opportunity for this type of attack is minimized.

If one were to copy a Virtual Machine's configuration files and datastore on a Windows machine, *link* (.lnk) files are made detailing the duplication with its source and destination. In addition to the link files, Windows records serial numbers of external memory attached to the operating system. With these two techniques, a file that is copied can be traced back to the originating machine, user account and destination media together with the date and time that it was copied. Although a user can delete these link files, there will still be a forensic trail, potentially for years. As the link files are generally under 10KB, they are unlikely to be overwritten at the byte level for some time after deletion.

4.2Higher-Technology Attacks

By their very nature, Virtual Machines require flexible storage and adaptive physical locations. A VM may require additional disk space or memory, which it may not be possible to be facilitate on the physical machine on which it currently resides. To accommodate this, the VM's datastore or host files must be moved to a new physical machine. An attacker with a higher level of technological knowledge may use the network to attack a VM while it is in transit, thus removing the possibility of host-based detection. In this section, we introduce three classes of threats to live virtual machine migration and describe several attacks applicable to each.

Control Plane-The communication mechanisms employed by the VMM to initiate and manage live virtual machine migrations must be authenticated and resistant to tampering. In addition, the protocols used in the control plane must be protected against spoofing and replay attacks. A lack of proper access control may allow an attacker to arbitrarily initiate VM migrations.

i. ***Incoming Migration Control:*** By initiating unauthorized incoming migrations, an attacker may cause guest VMs to be live migrated to the attacker's machine and gain full control over guest VMs.

ii. ***Outgoing Migration Control:*** Similarly, by initiating outgoing migrations, an attacker may migrate a large number of guest VMs to a legitimate victim VMM, overloading it and causing disruptions or a denial of service.

iii. ***False Resource Advertising*****:** In an environment where live migrations are initiated automatically to distribute load across a number of servers, an attacker may be able to falsely advertise available resources via the control plane. By pretending to have a large number of spare CPU cycles, the attacker may be able to influence the control plane to migrate a VM to a compromised VMM.

As most existing VM products rely on manual intervention to initiate a migration, their access control mechanisms for the control plane are simplistic. However, as automatic migrations for load-balancing between many machines becomes more common, potentially across multiple administrative domains and between unpredictable host addresses, mechanisms for policing the control plane must be introduced and maintained.

Data Plane- The data plane across which VM migrations occur must also be secured and protected against snooping and tampering in order to protect the VM's state. An attacker may be able to logically position himself in the migration transit path using a number of techniques such as ARP spoofing, DNS poisoning, and route hijacking. With such a position, an attacker can conduct a man-in-the-middle attack.

i. ***Passive Snooping:*** Passive attacks against the data plane may result in leakage of sensitive information. By monitoring the migration transit path and associated network stream, an attacker can extract information from the memory of the migrating VM such as passwords, keys, application data, and other protected resources.

ii. ***Active Manipulation:*** One of the most severe attacks, an inline attacker may manipulate the memory of a VM as it is migrated across the network. Such a man-in-the-middle attack may result in a complete and covert compromise of the guest OS.

Migration Module-The VMM component that implements live migration functionality must also be resilient to attacks. As reported by (Oberheide et al., 2008) the migration module provides a network service over which a VM is transferred, common software vulnerabilities such as stack, heap, and integer overflows can be exploited by a remote attacker to subvert the VMM. Given that VM migration may not commonly be viewed as a publicly exposed service, the code of the migration module may not be scrutinized as thoroughly as other code. While such

attacks are common across all types of software, special attention should be focused on the security of a VMM's migration module. As the VMM controls all the guest operating systems running within it, the severity of a VMM vulnerability is much greater than most normal software. If an attacker is able to compromise a VMM through its migration module, the integrity of any guest VMs running within the VMM, and any VMs that are migrated to that VMM in the future, may also become compromised.

4.3 Design Model for Secure Live VM Migration Data

Objectives two and three and were achieved by performing these activities. The associated research questions were also addressed in the subsequent sub-heading.

Objective 2- To design model to secure the live migration data.

Objective 3- To implement the design model

4.3.1 Physical Design

The security of a virtual machine migration hinges on the migration data plane, or the network transit path of which the migration occurs. If this data plane and network is insecure or un-authentication, an attacker on the network may gain access to the migration, allowing access to the full state of the virtual machine including the operating system kernel, applications and services run within the operating system, and the sensitive data currently being used by those applications.

The lab environment used the following hardware and software;

Software

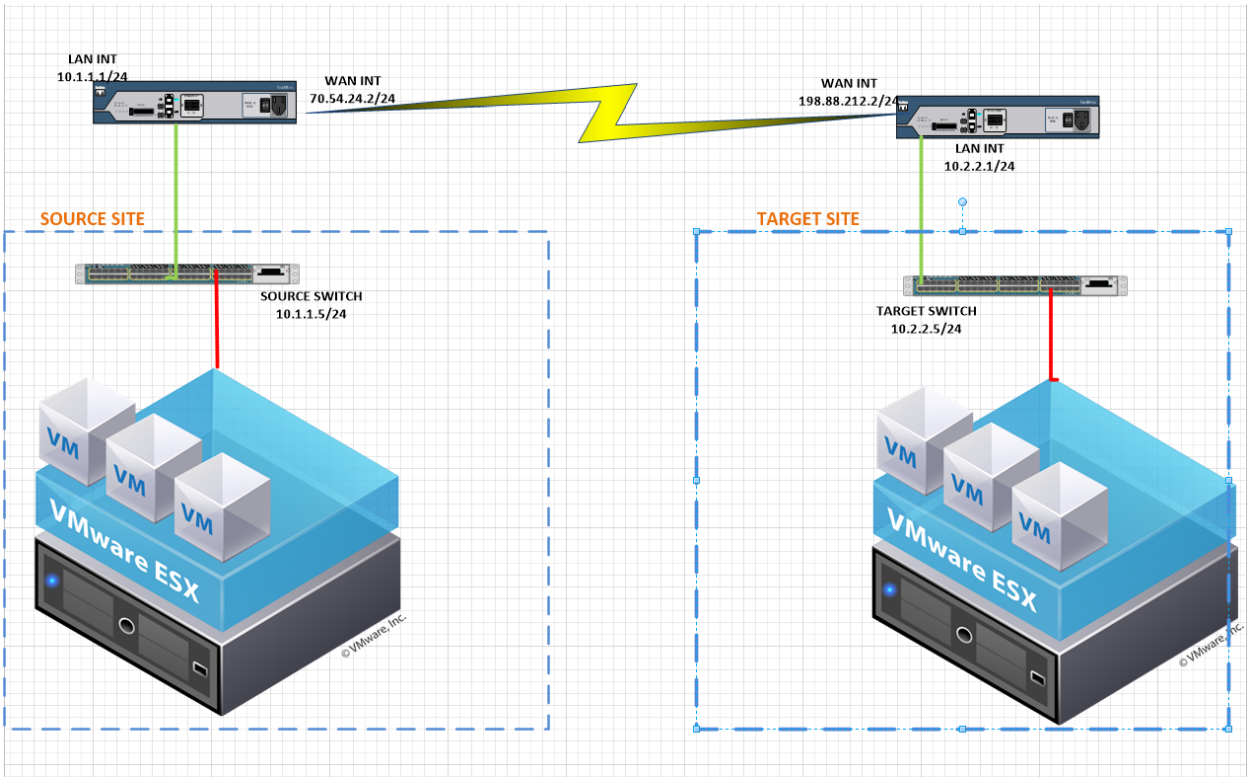
Software	**Version**
Windows 7 Professional	X64
Windows 10 Professional	X64
Wireshark	W64 1.6.6
ESXi Server	5.0.0 Build 469512
VCentre Server	5.0.0.3324 Build 472350
VSphere Client	5.0.0 Build623373
VMotion	as above

Hardware
Cisco Catalyst 3560 switch.
Cisco 2911 Router
HP ProLiant DL380p G8

Source host/target machine specification
Model- HP ProLiant DL380p G8
Operating System- VMware 6.0
CPU- Intel(R) Xeon(R) CPU E5-2670 0 @ 2.60GHz 2.59 Ghz
RAM-12.0 GB

Virtual machine specification
Operating System- Windows 7/10
CPU- QEMU Virtual CPU version 1.5.0 @ 2.40GHz
RAM-1024MB

Figure 6: Physical Layout



4.3.2 Logical Design

The migration of virtual machines has three options. One can migrate the host files, the vmdk file (the datastore), or the two together. This last option is simply a combination of the former two, but requires the OS to be powered off: the former two can be done while the machine is powered

up and being accessed. To properly simulate our environment, the study will used the second option better known as live migration. The benefit to this type of migration is that the user does not experience any down-time. For us to be able to achieve the results of the experiment we will need to perform;

Pre-experiment lab-The VM will transverse the network when no security measure has been put into place. This is live Migration with no security

Post-experiment lab-The VM will be migrated when IPsec security measure has been configured and installed. This is Secure Live Migration

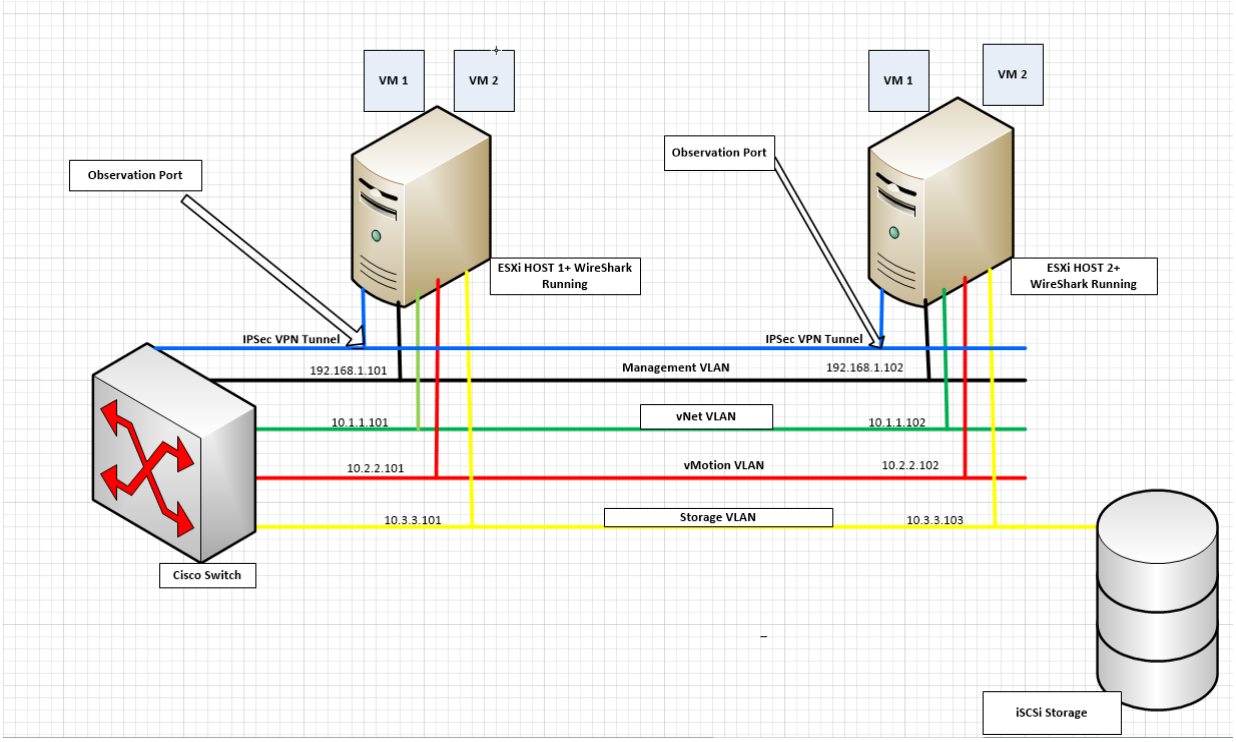
TABLE 3: SOURCE/TARGET IP SETTINGS

	Source Host	Target Host
IP	10.1.1.101	10.1.1.102
Subnet mast	255.255.255.0	255.255.255.0
Gateway	10.1.1.1	10.1.1.1

TABLE 4: CLIENT VMs IP SETTINGS

Category	Windows 7	Windows 10
IP	172.24.103.70	172.24.103.71
Subnet mast	255.255.255.0	255.255.255.0
Gateway	172.24.103.1	172.24.103.1

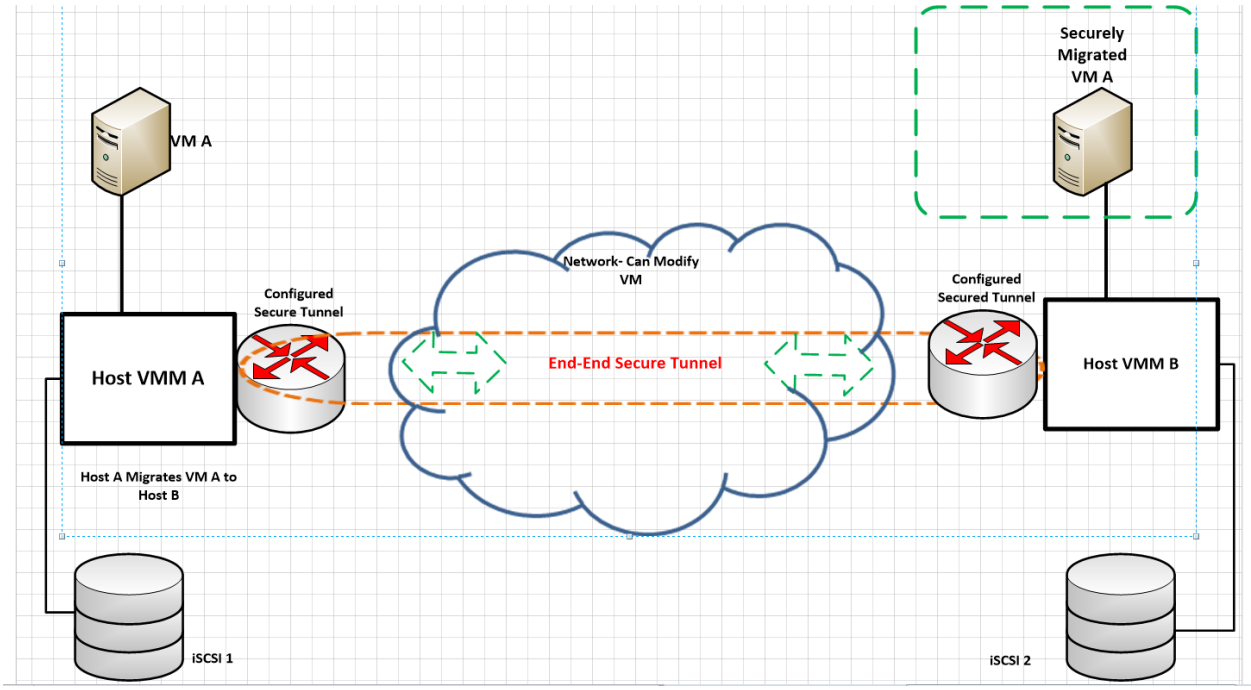
Figure 7: Logical Layout



4.4 Implementation of the Secure Model

The test setup used the latest versions of VMWare software available at the time. The two HP ProLiant DL380p G8 were installed with VMware 6.0 bare-metal. Live-migration facilitates the moving of the guest OS without it being powered down. The user of the VM can continue to use the VM as if migration is not taking place. The hypervisor creates a cache of events that occur, which is then used to update the VM in its new location.

Figure 8: Implementation Design



4.4.1 Pre-experiment analysis-Live Migration with no security

Research Question 3-Is the data-on-transit clear text during live migration?

Research Question 4-Can you intercept traffic data during live migration?

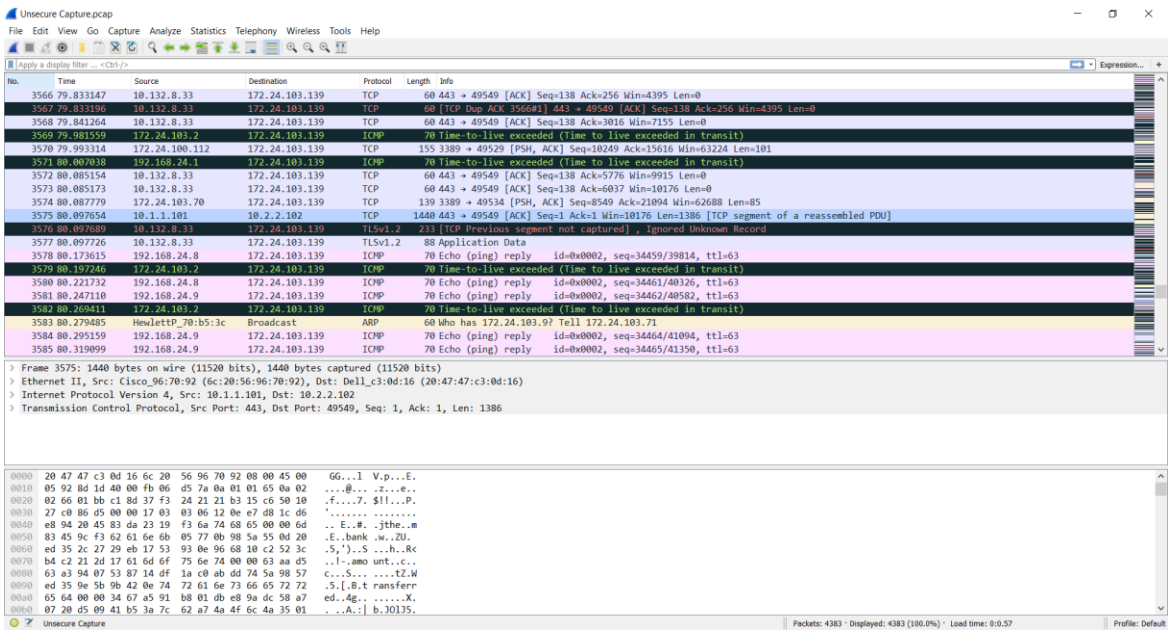
Using the VSphere web client, the Windows 7 and Windows 10 virtual machines were migrated from 101 to 102 and back again with the traffic being analyzed by Wireshark each time looking for information within the packet capture files that will be of value to malicious insiders. To aid detection of specific information during transit, a file was created on the Windows 7/ 10 desktop with the following contents:

The bank amount transferred.

The file is called *bankdetails dot txt* and is saved in the root of C

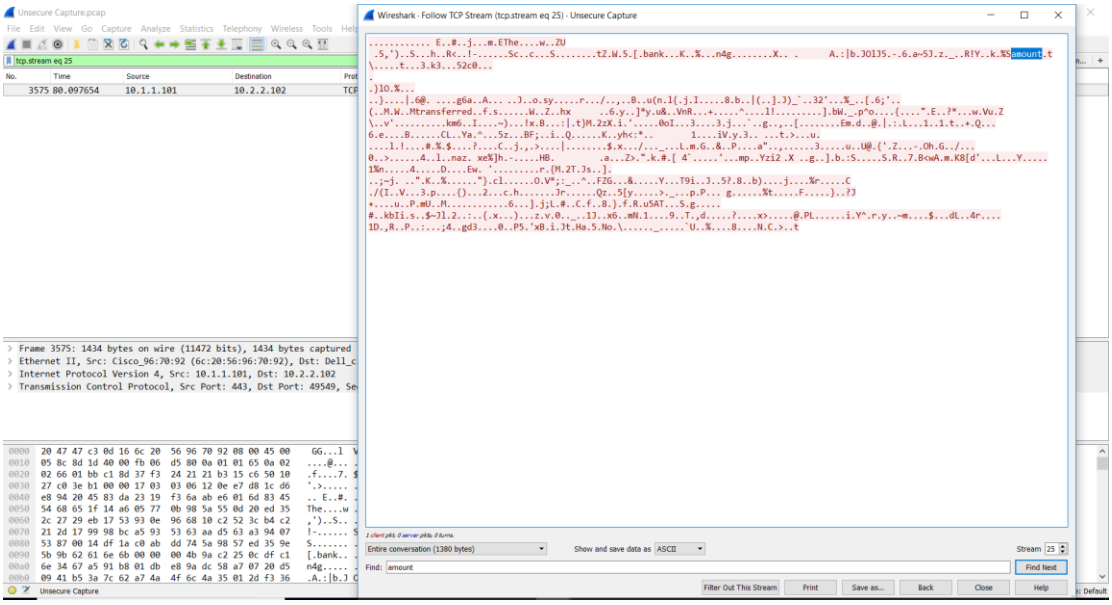
The contents of the notepad were seen in the captured file. Although the contents were separated by hexadecimal character.

Illustration 1: Wireshark Unsecure Capture



Therefore by default, live migration does not encrypt the migration data, since the migration data appears as clear text over the network, and this makes it susceptible to active and passive attacks. It is also possible to intercept migration traffic by use of Wireshark software and enabling promiscuous mode on the port where migration is taking place.

Illustration 2: Wireshark Unsecure Capture TCP Stream Follow



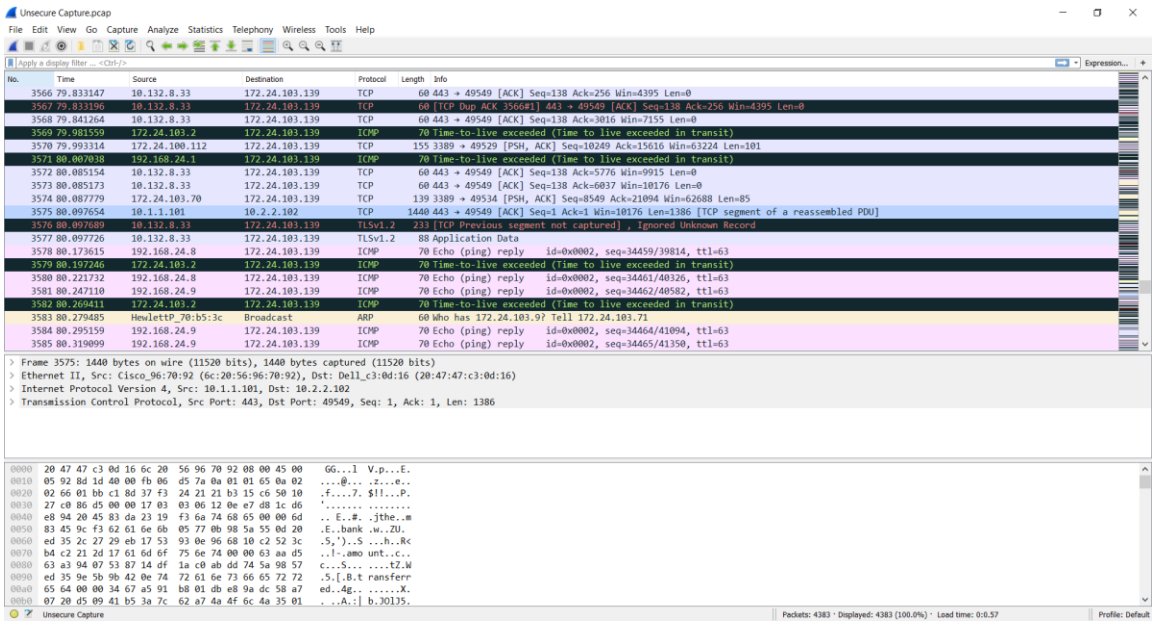
4.4.2 Post-experiment analysis -Secure Live Migration

Research Question 5- Can data-on-transit be encrypted during live migration?

Research Question 6- How do we ensure security of data-on-transit?

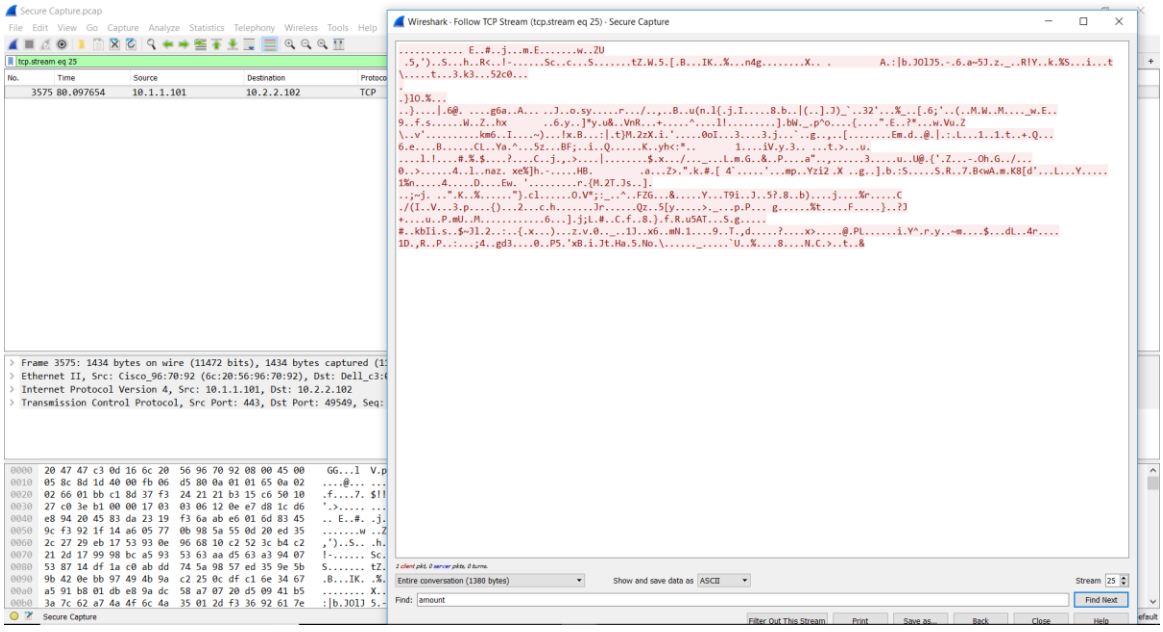
After configuring a VPN tunnel between the Host and Target, the Windows 7 and 10 VMs were live migrated to and from the Host. Wireshark sniffing tool was used to capture the streams of data.

Illustration 3: Wireshark Secure Capture



The captured packet showed the header and the raw data could not be followed and it was unreadable which is encrypted and secure. This shows that the migration data is encrypted and processed by IPsec VPN tunnel. This further shows that the data integrity is kept intact and confidentiality preserved.

Illustration 4: Wireshark TCP Secure Capture TCP Stream Follow



4.5Discussion

Migration is executed in two scenarios for evaluation purpose. The first scenario is where the VM is migrated from the source host to the target host and the second scenario is where the migrated VM is migrated back to the original source host. This is to simulate the situation in real scenarios where a VM is temporarily moved to another host for maintenance and then the same VM is then migrated back to the original host to continue its service.

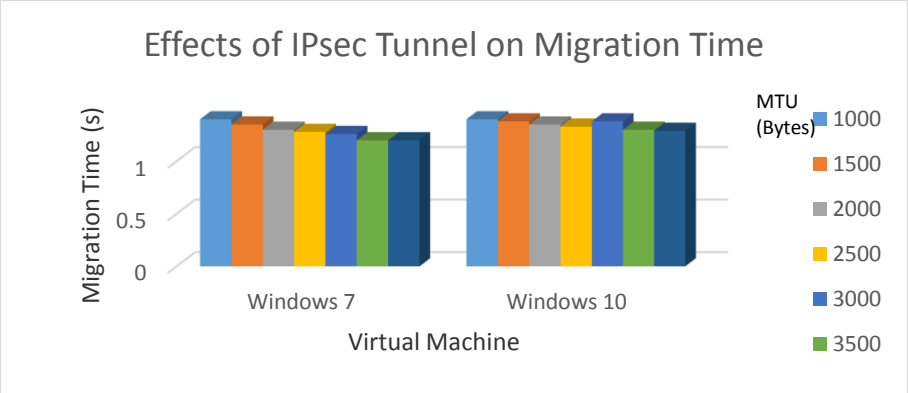
Evaluation of performance is done for both scenarios. Performance evaluation is divided into two categories. The first one is live migration without VPN tunnel and the second one is live migration with VPN tunnel. For each of the category, the migration time, packet transmission and the amount of CPU usage of a migration performance are measured and how their relationship is with the changing variables of value of MTU.

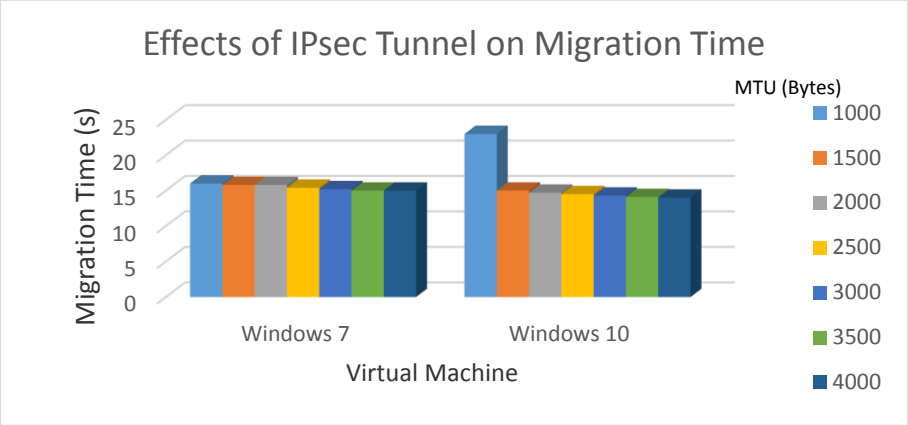
Further, a ping test and iperf test is conducted to evaluate the overhead due to IPsec implementation and the result showed that IPsec implementation resulted in a decreased network throughput. Meanwhile, the ping test showed that the latency also doubled by implementing IPsec on the transmission channel.

4.5.1 Migration Time

The results of migration time capture by the network analyzer, Wireshark for executing a direct live migration without VPN is shown below. The figure shows live migration time decreases as the value of configured MTU increases.

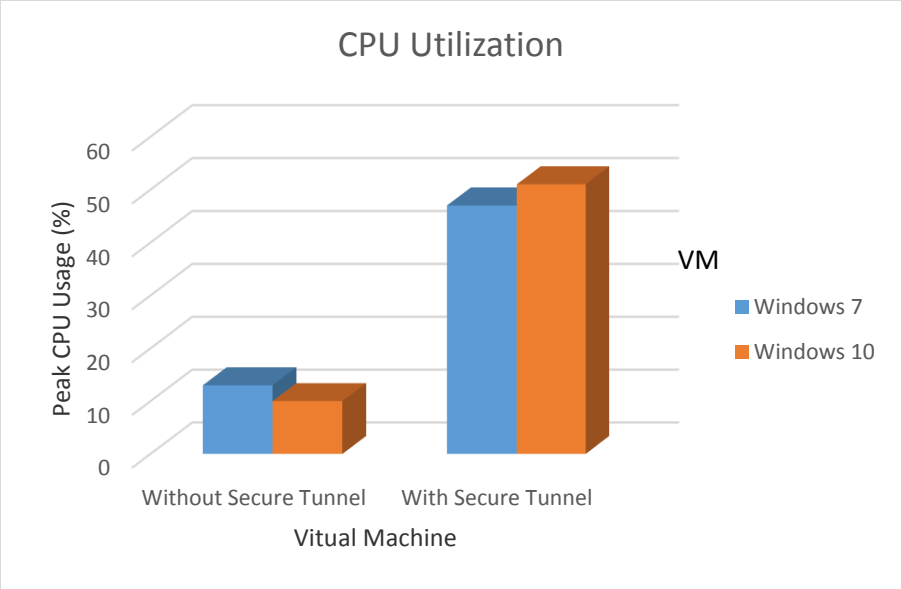
The opposite is true for live migration when IPsec is configured. IPsec affects live migration time by increasing it.





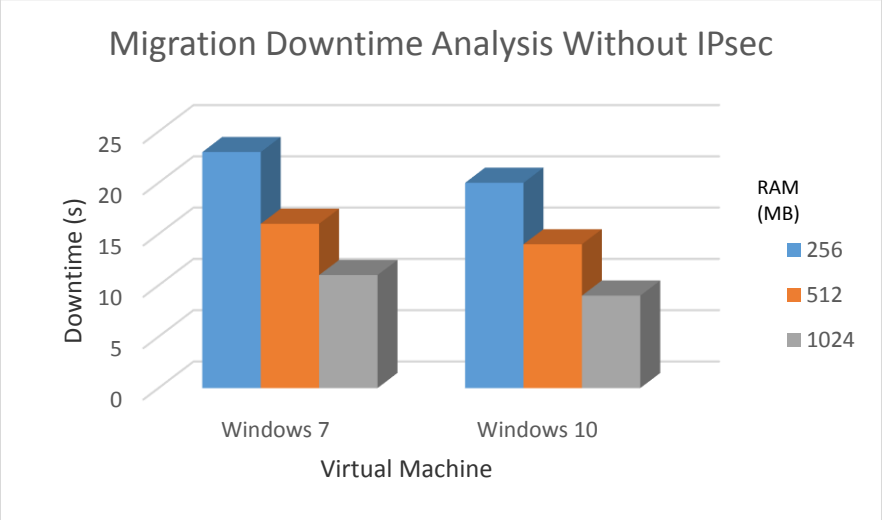
4.5.2 CPU Usage

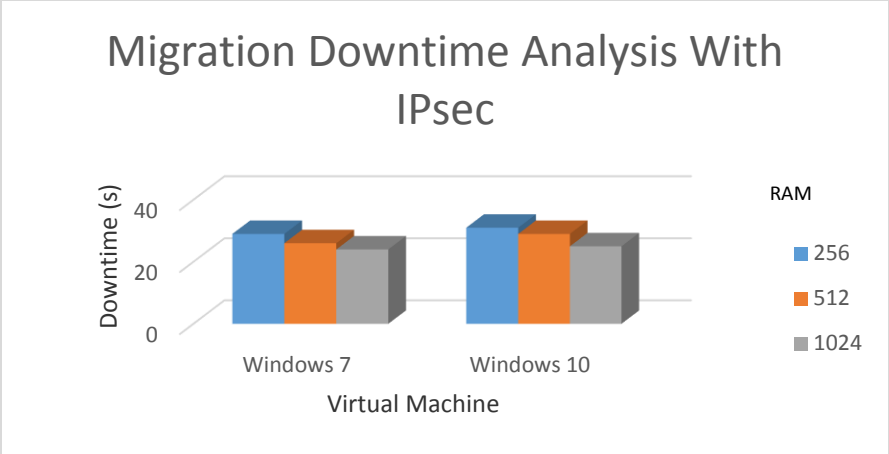
By observing the task manager of the host machines, it was noted that the VPN tunnel has an effect on CPU usage. It is observed that the highest CPU usage result when live migration is performed under VPN tunnel.



4.5.3 Migration downtime

The time for which the VM is unavailable during the migration process. It depends on the load on VM and migration network. It was measured using pink packets i.e. the VM is ping before live migration is started and continued during the live migration from host to target machines. Number of packets lost during migration are counted.





4.6 Comparison of Secure Live Migration with other models

There is a consistent finding derived from this study as compared to others on the same area of study. Different researchers have all alluded to the fact that live migration has security issues that need to be addressed. A lot of research has focused on improving the hypervisor performance and enhance its usability in virtualized environment neglecting the security vulnerability of the live migration feature that encompasses a hypervisor. In order for live migration to be accepted and be adopted on sensitive application, best practices in terms of data confidentiality, integrity and authentication need to be reviewed.

The data on transit during live migration must be protected from any interception or manipulation. Measures should be put in place to secure the live migration process. A secure VPN, a fortress hypervisor, or third party application should be considered

TABLE 4: TABULAR ANALYSIS OF LIVE MIGRATION MODELS

Security Requirements	**vTPM**	**Isolate Migration network VLAN**	**Network Security Engine-Hypervisor/PALM**	**VPN Secure Model**
Migration Time	22	14	19	15Sec
Migration Downtime	26	33	35	29 Sec
Peak CPU Usage	62	51	63	45%
Confidentiality and Integrity	□	□	□	□
Implementation	Nested	Switched/Router	Nested	Switched/Router
Live Migration	□	□	□	□

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.0 Introduction

The goal of the implemented secure live migration model is to secure live migration data. It protects against unauthorized access to live migration data, which could breach confidentiality and integrity of migration data during migration.

5.1 Contribution

Like all complex systems, virtualization platforms are imperfect and can have exploitable security flaws. Successful organizations will treat virtualization as part of a multi-layer partitioning approach, and seek to limit the impact of a security failure in any one piece. This study has reviewed different categories of live migration security systems and the differing levels of security isolation they provide. The proposed secure live migration framework is vital when implementing live migration. The secure tunnel configuration helps to encapsulate and secure the data on transit during live migration. By conducting experiments on this configuration and implementing the framework, data on transit is secured during live migration process.

The research has also contributed to existing studies in live migrations. A lot of research focused on intra-site live migration, commonly regarded as VM-VM migration within a hypervisor. This study ventured into a different niche, the inter-site live migration that involved migration across the WAN.

5.2 Limitations

There are few limitations that are associated with this study. They include the following;

a) Virtualization is a rapidly evolving sector of the industry, and standards for best-practices have not yet been developed for many issues.

b) Virtualization software and hardware were not cheap to acquire. This made the research to use simulation to mimic the real environment.

c) Virtualization software are proprietary and cannot be customized without flouting the antipiracy measures. Therefore it was difficult to dig –in and analyze the code for the software.

d) The study focused on secure live migration using a secure tunnel. Therefore there is need to evaluate secure live migration using other security measures, for instance, socket layer and certificate based security.

5.3 Recommendation

Despite the myriad issues identified above, virtualization isn't inherently unsecure, but the way it's being deployed may be through an unsecure nature. Immature security policies and processes, as well as lack of training, may be the bigger cause of issues and vulnerabilities which may lead to greater risk.

Performing a live migration under a VPN tunnel is indeed secure because the confidentiality and integrity of the migration data remains intact but as a penalty, a great amount of CPU overhead is produced due to encryption and packets processing and an increased migration time which decreases the migration performance.

The future work would be to consider enhancing the performance of live migration performed under a VPN tunnel so that it's practical enough to be applied at commercial data centers or cloud services. Further research need to be conducted and proposal made on the possibility of reducing the penalty to minimal.

There is also need to study the stages of the data transfer during a live migration to further understand the mechanism which could be applied to lower the overhead due to IPsec implementation to further increase the migration performance.

REFERENCES

Benzidane, K., Khoudali, S., & Sekkaki, A. (2013). Secured architecture for inter-VM traffic in a Cloud environment. *2nd IEEE Latin American Conference on Cloud Computing and Communications, LatinCloud 2013*, 23–28. https://doi.org/10.1109/LatinCloud.2013.6842218

Bhaskar Prasad Rimal, Eunmi Choi, I. L. (2009). 2009 Fifth International Joint Conference on INC , IMS and IDC (pp. 44–51). https://doi.org/10.1109/NCM.2009.218

Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., … Warfield, A. (2005). Live Migration of Virtual Machines. In *Live migration of virtual machines* (pp. 273–286).

Duncan, A., Creese, S., Goldsmith, M., & Quinton, J. S. (2013). Cloud computing: Insider attacks on virtual machines during migration. *Proceedings - 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2013*, 493–500. https://doi.org/10.1109/TrustCom.2013.62

Noorafiza, M., Maeda, H., Uda, R., Kinoshita, T., & Shiratori, M. (2015). Vulnerability Analysis using Network Timestamps in Full Virtualization Virtual Machine. *Icissp*, 83–89. Retrieved from http://dblp.uni-trier.de/db/conf/icissp/icissp2015.html#NoorafizaMUKS15

Nyaga, S. (2016). *UNIVERSITY OF NAIROBI SCHOOL OF COMPUTING AND INFORMATICS AMBIENT LEARNING MODEL FOR RESEARCH PROJECT SUPERVISION SUPPORT :*

Oberheide, J., Cooke, E., & Jahanian, F. (2008). Empirical exploitation of live virtual machine migration. *Proc. of BlackHat DC …*, (Vmm). https://doi.org/10.1.1.173.2903

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, *24*(3), 45–77. Retrieved from http://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=28843849&site=ehost-live&scope=site

Schiffman, J., Berger, S., Sailer, R., & Goldman, K. a. (2006). vTPM : Virtualizing the Trusted Platform Module It could happen to you …. *USENIX Security'06*, *15*, 1–19.

Sheinidashtegol, P., & Galloway, M. (2017). Performance Impact of DDoS Attacks on Three Virtual Machine Hypervisors. In *2017 IEEE International Conference on Cloud Engineering*. https://doi.org/10.1109/IC2E.2017.18

Shetty, J. (2012). A Survey on Techniques of Secure Live Migration of Virtual Machine. *International Journal of Computer Applications*, *39*(12), 34–39.

Study, P. (2012). vMotion Architecture , Performance and Best Practices.

Wan, X., Zhang, X., Chen, L., & Zhu, J. (2012). An Improved vTPM Migration Protocol Based Trusted Channel. In *An Improved vTPM Migration Protocol Based Trusted Channel* (pp. 870–875).

Zhang, F., Huang, Y., Wang, H., Chen, H., & Zang, B. (2008). PALM: Security preserving VM live migration for systems with VMM-enforced protection. In *Proceedings - 3rd Asia-Pacific Trusted Infrastructure Technologies Conference, APTC 2008* (pp. 9–18). https://doi.org/10.1109/APTC.2008.15

APPENDIX 1: IPsec CONFIGURATION

SOURCE ROUTER

SOURCE_ROUTER#exit

SOURCE_ROUTER con0 is now available

Press RETURN to get started.

SOURCE_ROUTER#
SOURCE_ROUTER#show run
Building configuration...

Current configuration : 1996 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname SOURCE_ROUTER
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
!

ip tcp synwait-time 5
!
!
!--- Create an ISAKMP policy for Phase 1

```
!--- negotiations for the L2L tunnels.
!
crypto isakmp policy 101
 hash md5
 authentication pre-share
!
!--- Specify the pre-shared key and the remote peer address
!--- to match for the L2L tunnel.
!----  6  Specifies an ENCRYPTED password will follow
!
!
crypto isakmp key 6 livemigration address 198.88.212.2
!
!
!
!--- Create the Phase 2 policy for actual data encryption.
!
!
crypto ipsec transform-set migration esp-des esp-md5-hmac
!
!--- Create the actual crypto map. Specify
!--- the peer IP address, transform
!--- set, and an access control list (ACL) for the split tunneling.
!
!
crypto map migrationmap 101 ipsec-isakmp
 set peer 198.88.212.2
 set transform-set migration
 match address 100
!
!
!
!
interface FastEthernet0/0
 ip address 10.1.1.1 255.255.255.0
 speed auto
 half-duplex
!
!
!--- Apply the crypto map on the outside interface.
!
!
interface Serial0/0
 ip address 70.54.24.2 255.255.255.0
 clock rate 2000000
 crypto map migrationmap
```

```
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2
 no ip address
 shutdown
 clock rate 2000000
!
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 70.54.24.1
!
!
no ip http server
no ip http secure-server
!

!--- Create an ACL for the traffic to
!--- be encrypted. In this example,
!--- the traffic from 10.1.1.0/24 to 10.2.2.0/24
!--- is encrypted. The traffic which does not match the access list
!--- is unencrypted for the Internet.
access-list 100 permit ip 10.1.1.0 0.0.0.255 198.88.212.0 0.0.0.255
!
!
!
control-plane
!
!
!

line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
```

```
 logging synchronous
line vty 0 4
 login
!
!
end

SOURCE_ROUTER#
```

TARGET ROUTER

```
TARGET_ROUTER#show run
Building configuration...

Current configuration : 1994 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname TARGET_ROUTER
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
memory-size iomem 5
no ip icmp rate-limit unreachable
ip cef
!
!
!
!
no ip domain lookup
ip auth-proxy max-nodata-conns 3
ip admission max-nodata-conns 3
!
!
!
ip tcp synwait-time 5
!
!
!--- Create an ISAKMP policy for Phase 1
!--- negotiations for the L2L tunnels.
```

```
!
!
crypto isakmp policy 101
 hash md5
 authentication pre-share
!
!
!--- Specify the pre-shared key and the remote peer address
!--- to match for the L2L tunnel.
!----  6  Specifies an ENCRYPTED password will follow
!
!
!
crypto isakmp key 6 livemigration address 70.54.24.2
!
!
!
!--- Create the Phase 2 policy for actual data encryption.
!
!
crypto ipsec transform-set migration esp-des esp-md5-hmac
!
!--- Create the actual crypto map. Specify
!--- the peer IP address, transform
!--- set, and an access control list (ACL) for the split tunneling.
!
!
crypto map migrationmap 101 ipsec-isakmp
 set peer 70.54.24.2
 set transform-set migration
 match address 100
!
!
!
!
interface FastEthernet0/0
 ip address 10.2.2.1 255.255.255.0
 speed auto
 half-duplex
!
!
!
!--- Apply the crypto map on the outside interface.
!
!
!
```

```
interface Serial0/0
 ip address 198.88.212.2 255.255.255.0
 clock rate 2000000
 crypto map migrationmap
!
interface FastEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/1
 no ip address
 shutdown
 clock rate 2000000
!
interface Serial0/2
 no ip address
 shutdown
 clock rate 2000000
!
!
ip forward-protocol nd
ip route 0.0.0.0 0.0.0.0 198.88.212.1
!
!
no ip http server
no ip http secure-server
!--- Create an ACL for the traffic to
!--- be encrypted. In this example,
!--- the traffic from 10.1.1.0/24 to 10.2.2.0/24
!--- is encrypted. The traffic which does not match the access list
!--- is unencrypted for the Internet.
access-list 100 permit ip 10.2.2.0 0.0.0.255 70.54.24.0 0.0.0.255
!
!
!
control-plane
!
!
!
!
!
!
!
!
```

```
!
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
!
end

TARGET_ROUTER#
```

APPENDIX 2: BUDGET

#	Item	Qty	Unit Cost (Ksh)	Total Cost (Ksh)
1.	Vmware	1	100,000.00	100,000.00
3.	Training on using tools (Wireshark,Vmware,Iperf)		100,000.00	100,000.00
4.	Printing (concept paper, proposal and final report)		50,000.00	50,000.00
5.	Binding (proposal and final report)		20,000.00	20,000.00
6.	Transport and Miscellaneous		50,000.00	50,000.00
7.	Communication (including internet access)		20,000.00	20,000.00
8.	Conferences and publications		100,000.00	100,000.00
	TOTAL			440,000.00

APPENDIX 3: WORKPLAN

Illustration 4: Work plan

	1	2	3	4	5	6	7	8
Project proposal and defense								
Project proposal corrections								
Work in Progress 1 Submission								
Work in Progress 2 Submission								
Final report presentation								
Final report corrections								
Submission of report for examination								
Submission of final project report								
Publication								

Key

1- July 2017
2-3 -Aug-Sept 2017
4-5- Oct-Nov 2017
5-6- Jan-Feb 2018
6- March 2018
7- April 2018
8- June 2018