Implementing Live Migration Security

Darren O'Neill

Submitted as part of the requirements for the degree
of MSc in Cloud Computing
at the School of Computing,
National College of Ireland
Dublin, Ireland.

August 2015

Supervisor Mikhail Timofeev

Abstract

This dissertation was carried out in order to investigate the current methods for securing live virtual machine migration and to contribute to the current frameworks that currently exist. The Live migration of virtual machines is fundamental component of cloud computing. In recent years, researchers have discussed frameworks and methodologies to secure the process of live virtual machine migration. However the majority of research that is carried out to date has focused on the performance and not on security. The research that we have undertaken focuses on the security of virtual machine during migration. To fulfill this we investigated two popular open source hypervisors XEN and KVM. We found security flaws in each system and have proposed securing the live migration process by using encryption, intrusion detection systems, role based access control and firewall rules. We recommend that XEN is the more secure of the two systems available.

Acknowledgements

First and foremost I would like to thank my supervisor Mikhail Timofeev, who has supported me throughout this dissertation with his skill and knowledge. This dissertation would not have been completed without his support, quite simply I could not wish for a better supervisor. Besides for my supervisor I would like to thank Robert Duncan, who freely and graciously offered advice on multiple occasions throughout my studies. I would like thank my parents and grandparents, who from an early age instilled the value of education in me. Finally I would like to thank Emily for her patience and understanding while I undertook this research.

Contents

List of Figures

Listings

Chapter 1

Introduction

Live Migration (LM) is a maintenance process in Cloud Computing, which is extensively used in load balancing operations and supports computational continuity, however, security concerns are causing many organizations to be hesitant to adopt the technology. LM refers to the process of transferring a running virtual machine from one physical host to another. A virtual machine, is a duplicate of a physical machine which does not need its own physical hardware to operate.. This allows for multiple virtual machines to reside on one physical host, and for the virtual machines to be transfered or migrated from one machine to another.

The majority of research that has been undertaken to date has focused on the performance of LM with little research being undertaking in the area of security. The definitive piece of research on LM security was carried out by Oberheide et al. (2008). Multiple authors have built on this work throughout the years however LM security is not an extensively researched area.

Of the research that does exist, the majority of it focuses on three areas:

- The data plane
- The control plane
- The migration module

The migration module is the software component of the hypervisor which initiates LM. The control plane is the management interface, to which an administrator can initiate migrations. The data plane is the network, which the migration data transverses. Within the literature many authors note that the data plane is the most crucial to secure, and thus, the majority of our research will look at this area. Based on Oberheide

et al. (2008) work many authors have proposed frameworks and security measures to ensure that LM is secure. Many of these frameworks remain untested. This research will look at multiple frameworks used in LM research and investigate it's effectiveness of securing two systems: KVM and XEN. The goal of this research is to implement security features recommended in these frameworks and to contribute to the area of LM security.

Chapter 2

Background

Virtual machine Live Migration (LM) is a new and evolving technology that is touted as the foremost advantage of Cloud Computing and Virtualization. Recent articles state that the advantages of LM are many, but security concerns are making industries hesitant to adopt it (Shetty et al., 2012; Upadhyay and Lakkadwala, 2014; YamunaDevi et al., 2011; Zhang et al., 2008). To compound this issue the majority of research on LM has focused on the performance of LM and not the security of it.

This literature review begins with an introduction to Cloud Computing, and some of the security concerns which are more prevalent in Cloud Computing today. This is followed by a discussion of Virtualization and the main benefit of Virtualization which is machine migration. From there the security concerns of LM are investigated followed by what is currently being done to address these concerns.

2.1 Cloud Computing

Computing is being transformed to a model similar to that of utilities such as water, electricity and telephony. In this model services are available on demand, over a network (Buyya et al., 2009; Devi and San, 2014). It is a form of computing that allows for ubiquitous, convenient , on demand access to a pool of computing resources (Mell and Grance, 2011; Dinh, 2011; Ahmad et al., 2013; Upadhyay and Lakkadwala, 2014). These computing resources are provisioned quickly and with little effort (Ahmad et al., 2013). The Cloud allows users to access applications through a web browser over an Internet connection, where the information is stored in a remote location (Armbrust et al., 2010; Upadhyay and Lakkadwala, 2014). The rate of adoption of Cloud Computing has been increasing steadily for the past several years and it is allowing users to reduce costs and

increase both flexibility and scalability (Sabahi, 2011; Garfinkel and Rosenblum, 2005; Kushwah and Saxena, 2013). This is achieved by allowing users to switch to a pay-per-use model for applications, development environments and computing infrastructure (Bojanova et al., 2013). To the user cloud gives the appearance of unlimited computing resources and eliminates the need for up front costs (Mell and Grance, 2011; Dinh, 2011). In order to facilitate this Cloud Computing is built on a concept known as Virtualization.

2.2 Virtualization

Virtualization is a key enabler of Cloud Computing, which is achieved by converting traditional hardware components such as hard drives into software based components (Ahmad et al., 2013). When this conversion is complete the software component functions the same as a traditional physical component would. This virtualization of traditional hardware components makes it possible to virtualize entire machines. A physical machine that has been virtualized is known as a virtual machine (VM). The term virtual machine was first coined by Popek and Goldberg in 1974 (Popek and Goldberg, 1974). Virtual machines are built on virtual hardware, known as hardware virtualization. Hardware virtualization, is the adding of an additional layer between the physical hardware and the operating system. This in turn allows the separation of the operating system from the underlying physical hardware of the machine (Aiash et al., 2014). In doing this the operating system is decoupled from the physical machine. This separation allows for virtual machines to be moved from one physical machine to another in a process known as machine migration (König and Steinmetz, 2011).

Since virtual machines are now decoupled from the physical hardware it also allows for multiple virtual machines to reside on one physical machine as virtualization adds a layer of abstraction between the virtual machine and the underlying hardware (Biedermann et al., 2013). The resources that exist on the physical machine are pooled and can be distributed among the virtual machines (Aiash et al., 2014).

A component known as a hypervisor facilitates this sharing of resources between the physical hardware and the virtual machines by allocating resources to virtual machines (YamunaDevi et al., 2011; Biedermann et al., 2013). It ensures that isolation exists between virtual machines, in order to prevent one virtual machine from influencing another (Perez-Botero et al., 2013). The hypervisor prevents itself from been influenced by the virtual machines it achieves this by running at a privilege level above the virtual machines (Colp et al., 2011).

There are two types of hypervisor, type one and type two. Type one hypervisors are installed directly onto the system hardware. This can offer better performance and security to the virtual machines as there is no operating system between it and the hardware (YamunaDevi et al., 2011). However this may lead to a more complex installation process for the hypervisor. A typical type one hypervisor installation can be seen in 2.1. This shows a total of three layers.

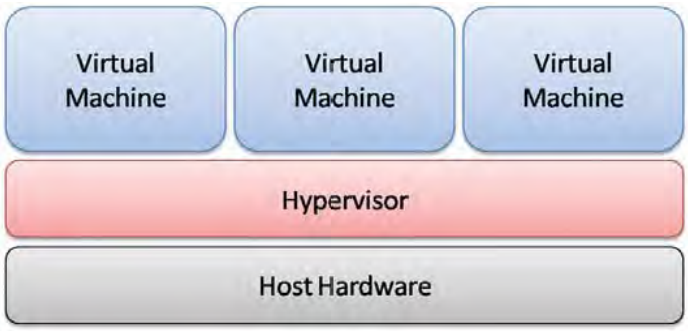

Figure 2.1: A Type 1 Hypervisor which is installed directly onto the hardware (Daniels, 2009)

Type two hypervisors are installed on an already installed operation system (host operating system). Type two hypervisors offer worse performance than type one as there is an extra layer for instructions to travel through, i.e the host operating system. However type two hypervisors can be less complicated to install.It is possible to install and configure many type two operating systems within minutes (Perez-Botero et al., 2013). Figure 2.2 shows the four layers that typically present in a type two hypervisor.

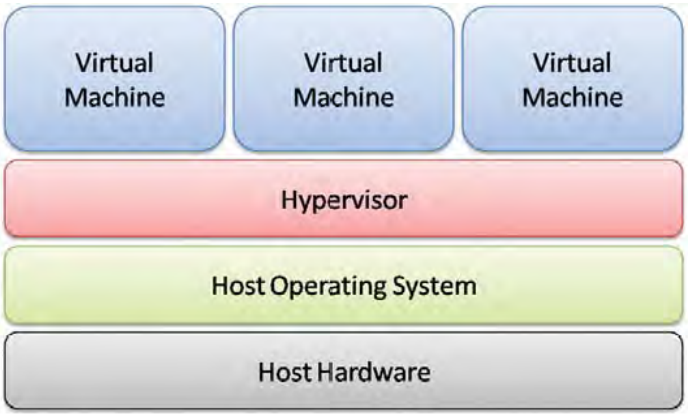

Figure 2.2: A Type 2 Hypervisor which is installed on an Operating system which is in turn is installed directly onto the hardware (Daniels, 2009)

There are many advantages of virtualization: Multiple virtual machines can reside on one physical server, thus the sever is utilized efficiently and costs are reduced (Rashmi Rao and Pawan Prakash, 2013). Other advantages of virtualization include better performance, high availability and low boot time of virtual machines. However, the utmost advantage of virtualization is the ability to transfer virtual machines to different physical hosts, this process is know as machine migration (Aiash et al., 2014; YamunaDevi et al., 2011). System virtualization improves the overall security of the system as the hypervisor runs in a privilege level above that of the operating system of the guest virtual machines (Zhang et al., 2008).

However virtualization also causes security concerns. Issues can arise from the multi tenet nature of virtualization. Multi tenancy is a configuration in which multiple virtual machines with different owners use the same physical hardware. Disk space can now be shared by multiple users which can lead to data security concerns where one user may be able to illegally access another user's data (Perez-Botero et al., 2013; Fornaeus, 2010).

Virtualization suites and hypervisors are extremely complex and thus are likely to have many vulnerabilities that have yet to be discovered, thus meaning that many virtualization infrastructures may be open to attacks(McDaniel and Nance, 2013b; Pearce et al., 2013). The hypervisor has full control over all virtual machines that reside within it. This makes the hypervisor a key part of virtualization technology, as if the hyper- visor is compromised all virtual machines are compromised (Perez-Botero et al., 2013; McDaniel and Nance, 2013a).

2.3 Deployment Models

There are four deployment models in Cloud Computing(Mell and Grance, 2011):

- Public Cloud

  When a cloud is made available to the general public on a pay-as-you-go model and is provided by an organization that is promoting cloud services it is know as public cloud (Devi and San, 2014). Amazon AWS and Microsoft Azure are public cloud offerings.

- Private Cloud

  When data centres are large enough to benefit from the features of Cloud Computing but are not made available to the general public, are kept behind a firewall

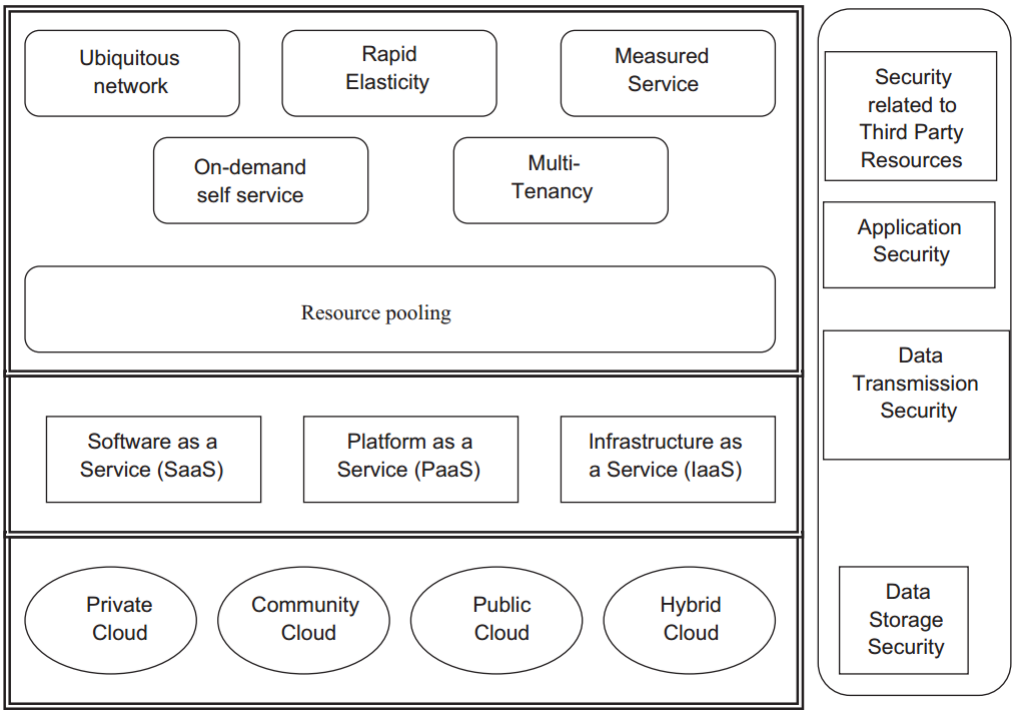

Figure 2.3: (Subashini and Kavitha, 2011)

and used solely by one business, they are known as private clouds(Armbrust et al., 2010).

- Community Cloud

  Community Cloud is a Cloud that attempts to achieve the cost benefits of public cloud, while having the control similar to private cloud, This is achieved by organizations with similar agendas pooling their resources together to build a cloud for their own use (Dinh, 2011).

- Hybrid Cloud

  Hybrid cloud is a cloud that is a combination of two or more of the other deployment models.

According to the National Institute of Standards and Technology (NIST) a cloud should have specific characteristics to be considered as one, regardless of its type or a deployment model. These characteristics are known as the five essential characteristics of Cloud Computing (Mell and Grance, 2011):

- On-Demand Self Service

  Allows the user to provision computing resources as needed.

- Broad Network Access

  Allows the user to access these resources on multiple devices such as phones and tablets

- Resource Pooling

  Allows resources to be shared in a multi-tenet model.

- Rapid Elasticity

  Allows for the resources to be scaled up or down quickly to meet demand

- Measured Service

  Allows the cloud provider to measure the use of the service by the customer and bill them accordingly

The five characteristics of cloud computing can be applied to the three different service models (Mell and Grance, 2011; Armbrust et al., 2010; Ahmad et al., 2013; Kushwah and Saxena, 2013; Devi and San, 2014):

- Software as a Service (SaaS)

  In SaaS the user accesses services, as web applications, usually via a browser. The user does not control the hardware, network infrastructure or the operating system (Upadhyay and Lakkadwala, 2014; Kushwah and Saxena, 2013)

- Platform as a Service (PaaS)

  In PaaS an application development environment is provided by the cloud administrator. This allows the users to develop and deploy applications without the complexity of buying and managing the underlying hardware and software (Upadhyay and Lakkadwala, 2014; Kushwah and Saxena, 2013; Devi and San, 2014)

- Infrastructure as a Service (IaaS)

  In IaaS the user can run full operating systems and applications. The user leases computing resources such as network, storage and processing power. Both PaaS and Saas rely on IaaS for their services, this means that IaaS is the most critical to secure (Ahmad et al., 2013).

  The advantages that are associated with Cloud Computing are many, however, disadvantages also exist, the most notable being security. This has led to Cloud Computing security being a major research topic in computing(Shen et al., 2011).

Cloud Computing security is not just a concern for researchers it is a concern for IT managers with between 74 and 76 percent of IT managers believing that security is the main obstacle that is preventing them from using Cloud Computing (Chen and Zhao, 2012; Rashmi Rao and Pawan Prakash, 2013).

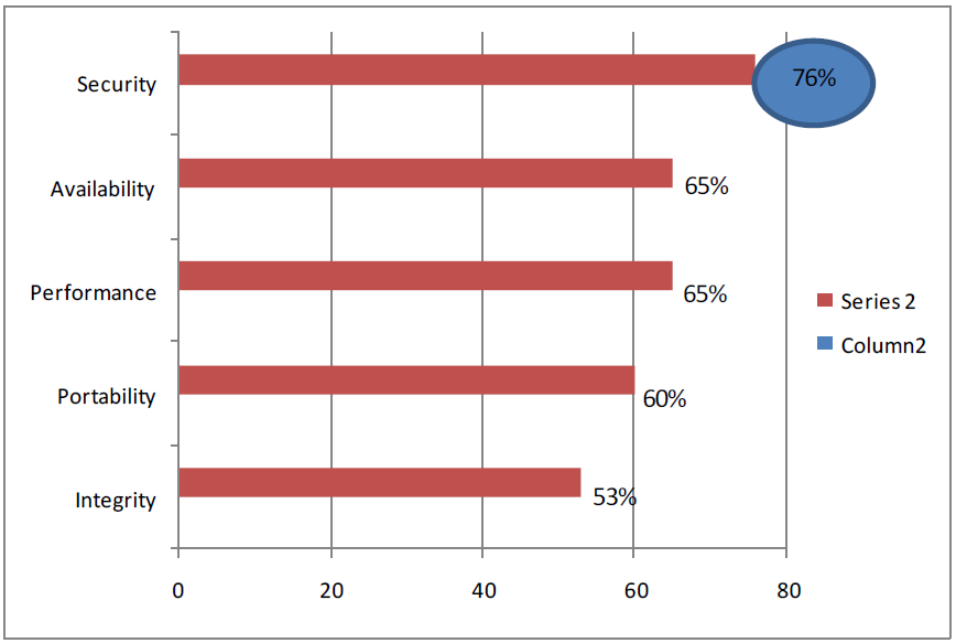

Figure 2.4: Security is a Major Concern for IT Managers (Rashmi Rao and Pawan Prakash, 2013)

The main cause of concern for IT managers, is that data now exists in remote locations, outside the confines of their organisation and thus no longer under their control (Sabahi, 2011). To compound these concerns, vulnerabilities are constantly being found in Cloud Computing technologies, which causes managers to be even more sceptical of using the technology (Chen and Zhao, 2012). Given these issues managers may choose to shy away from the technology altogether as cloud security requirements are also more difficult than traditional security requirements. Traditional steps to secure systems such as operating system patching are still required in the cloud, with the organization also having to take the additional cloud securing measure which did not exist in the traditional data centres(Chen and Zhao, 2012).

2.4 Migration

Migration allows full virtual machines to be moved from one physical host to another. Organisations should select a migration strategy based on their needs. There are two kinds of migration that can fulfil this, these are live migration and non-live migration also known as cold migration (Hu et al., 2013; Kuno et al., 2011; Celesti et al., 2010; Buyya et al., 2011). The difference between these two migration methods is that, in LM the virtual machine is running while it is migrated and there is little to no interruption to the VM. (Upadhyay and Lakkadwala, 2014) However since there may be no noticeable interruption to the virtual machine the users may be unable to know if their virtual machines are being live migrated (Biedermann et al., 2013). Cold migration differs from LM in that the virtual machine is powered off while it is migrated. In all forms of migration, be it live or cold, it is important to consider how storage is implemented within the system (Hu et al., 2013). If the virtual machines are using shared storage such as a network file share, LM is possible, however if they are not, cold migration must to be used (Buyya et al., 2011). In LM, memory is transferred from one physical host to another, however disk drives are not transferred.

In order to enable LM physical hosts are configured in a cluster, with access to shared storage. The VM's memory and state exist on one physical host, while it's virtual disk exists on the shared storage. During LM only the VM's memory is transferred to the new host (Perez-Botero, 2011). VMs can be cold migrated in shared storage environment, when this is done ownership and bios settings are transferred from one physical host to another. (Buyya et al., 2011). If shared storage is not used the entire virtual machine disk has to be transferred from one physical host to another which can take a long time (Hu et al., 2013). The advantage of non-LM is that it is not as difficult to implement as shared storage is not required.

LM of virtual machines is powerful as it allows:

- System maintenance
- Load balancing
- Power saving(Jin et al., 2009; Shetty, 2013)
- Workload balancing
- Fault tolerance

– Consolidation of VMs (Shetty et al., 2012; Upadhyay and Lakkadwala, 2014; YamunaDevi et al., 2011; Zhang et al., 2008)

LM allows for the transfer of a running virtual machine to a new physical machine with little to no noticeable interruption. Many cloud providers use LM to facilitate hardware consolidation by moving virtual machines onto physical machines and then switching off physical machines which are then not in use.(Aiash et al., 2014) LM can be used for network load balancing, performance optimization and incident response (Biedermann et al., 2013). LM procedures can differ, but in general there are four phases of LM. Which are: The set up stage, the memory transfer stage, the storage transfer stage and the network clean up stage (Aiash et al., 2014).

The process for LM usually takes the following steps:

– Push phase

  The source VM continues running, pages are pushed across the network to the new destination, to be consistent any pages modified during this stage must be resent.

– Stop and copy phase

  The source VM is stopped, pages are copied across to the destination VM and the new VM is started.

– Pull phase

  The new VM starts its execution and if it accesses a page that has not yet been copied it is faulted in across the network from the source VM (Perez-Botero, 2011; Clark et al.).

Advantages of machine migration include: physical hosts can be decommissioned more easily, VM state can be transferred from one physical host to another and since the operating system is completely abstracted from the hardware, the user does not have to share any details about his or her operating system for it to be migrated (Clark et al., 2005).LM moves a virtual machine from one physical host to another, the user should notice very little performance drop off and may not even be aware that the virtual machine has been migrated. (Aiash et al., 2014)

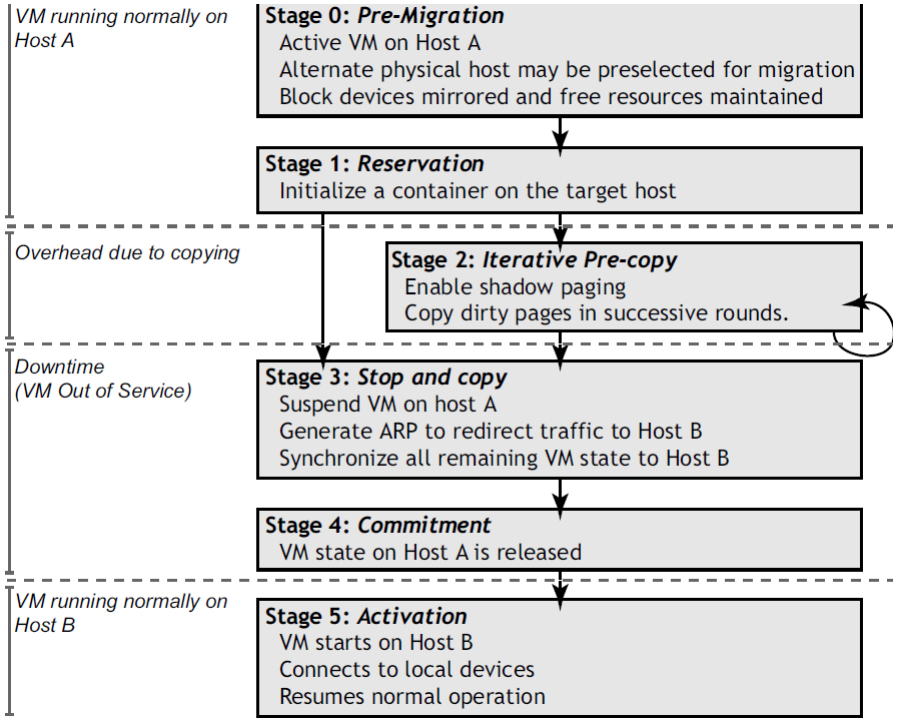

Figure 2.5: LM of a VM from one physical host to another (Clark et al.)

2.5 LM Security

Live virtual machine migration has been adopted more as time has gone on, however LM security is still in its early stage and there are many known security issues with LM (Biedermann et al., 2013; Upadhyay and Lakkadwala, 2014; Ahmad et al., 2013). This has led to many companies such as health care to be cautious of using LM (Shetty, 2013; Aiash et al., 2014; Upadhyay and Lakkadwala, 2014; Ahmad et al., 2013).

Much of the research that has been completed focuses on the performance of LM and not security (Ahmad et al., 2013). This has led to a need for more research in the area of LM security (Ahmad et al., 2013). The virtual machine is also at its most vulnerable while it is being migrated (Chen and Zhao, 2012).

2.5.1 The Attacks

There are three main areas of attack that can happen in LM, these areas are the control plane the data plane and the migration module (Oberheide et al., 2008) (YamunaDevi et al., 2011) (Perez-Botero, 2011) (Ahmad et al., 2013). The data

plane is the network which the VM migration traffic transverses, the migration module is the component of the hypervisor that implements the LM functionality, while the control plane is the communication mechanism to instruct the hypervisor to implement a LM (Perez-Botero, 2011) (Ahmad et al., 2013). Vulnerabilities on these three areas are caused by inappropriate access control policies, unprotected transmission channels and loopholes in the migration module (Upadhyay and Lakkadwala, 2014). Attacks that then happen because of these vulnerabilities are man in the middle attacks, denial of service attacks and stack overflow attacks (Aiash et al., 2014; Shetty, 2013; Ahmad et al., 2013).

2.5.2 Migration Module Attacks

The security risks involved in migration include migrating a VM to an untrusted platform, authentication and authorization of the management interface and bugs in hypervisor code (Upadhyay and Lakkadwala, 2014). For example the migration functionality that is implemented by many hypervisors exposes the entire machine state of a VM to the device module which listens to the incoming LM requests from remote platforms (YamunaDevi et al., 2011). An attacker may be able to hijack the device module process or hypervisor where these attacks occur. If the process is hijacked information of the migrated virtual machine including states of operation system kernel, applications and services, sensitive data and even keystrokes are accessible to the hackers (YamunaDevi et al., 2011).

2.5.3 Data Plane Attacks

The information that is sent during LM can be easily tampered with if the channel is not encrypted. Many LM protocols do not encrypt the the migration data by default and thus migration traffic can be completely vulnerable during the migration procedure as it is sent in plain text. The data in transfer can be susceptible to such attacks as man in the middle attacks, where an attacker can either actively manipulate data or secretly sniff data from the machine that is in migration (Shetty, 2013; Ahmad et al., 2013; Upadhyay and Lakkadwala, 2014; Aiash et al., 2014; Biedermann et al., 2013).

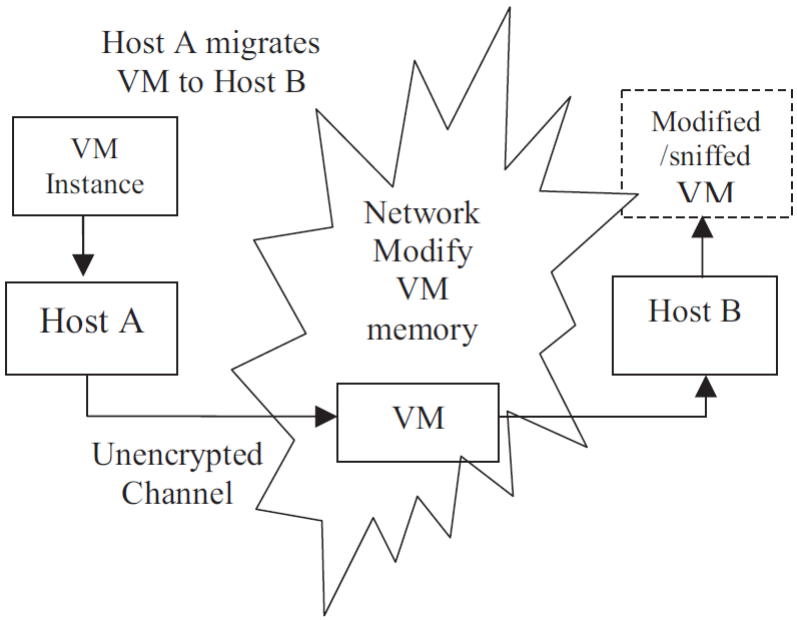

Figure 2.6: (Shetty, 2013)

2.5.4 Control Plane

The control plane is the area in which the administrator can initiate LM. Because of this it is important to have clearly defined administrator roles, in which certain administrators can only do certain things (Shetty, 2013). Attacks that happen on the control plane are:

- Denial of service attacks i.e. making a VM unavailable
- VM hopping: migrating a VM when there is no reason to migrate it which reduces its performance.
- False resource advertisement: Advertising free resources on a host that do not exist. Which can intice VMs to LM toi the host, thus over loading it (Shetty, 2013).

2.6 Categorised LM Attacks

In order to achieve a more comprehensive view of the attacks that exist, the details of the attacks, what plane they attack and the research authors, we created a table of the attacks. This table can be seen in table 2.1

Threat	Details	Plane	Author
Incoming Migration Control	An attacker is able to initiate incoming virtual machine migrations to a physical host, to over load the host and cause a denial of service	Control Plane	(Oberheide et al., 2008) (Aiash et al., 2014) (Perez-Botero, 2011) (Upadhyay and Lakkadwala, 2014) (Shetty, 2013) (Ahmad et al., 2013)
Outgoing Migration Control	An attacker is able to initiate outgoing virtual machine migrations to a physical host, to over load the host and cause a denial of service	Control Plane	(Oberheide et al., 2008) (Aiash et al., 2014) (Perez-Botero, 2011) (Upadhyay and Lakkadwala, 2014) (Shetty, 2013) (Ahmad et al., 2013)
False resource advertising	In automatically LM environments, one where load balancing takes place, an attacker can advertise resources on a host , in order to entice machines to migrate to the host, thus over loading the host and	Control Plane	(Oberheide et al., 2008) (Perez-Botero, 2011) (Upadhyay and Lakkadwala, 2014) (Shetty, 2013) (Ahmad et al., 2013)
Stack/ Heap overflow	Bugs in the migration module and poor coding practices can lead to attacks exploit vulnerabilities	Migration module	(Oberheide et al., 2008) (Aiash et al., 2014) (Ahmad et al., 2013) (Perez-Botero, 2011) (Upadhyay and Lakkadwala, 2014)
VM hopping	An attacker can repeatedly migrate a VM in order to reduce the performance of the VM	Control plane	(Shetty, 2013) (Ahmad et al., 2013)
Attack on transmission channel	Attacks on the transmission channel can Passive or Active, Passive attacks, The attacker listens for information the the network where the migration traffic is traversing. The attacker can gain information such as passwords and encryption keys,The attacker can gain information about the virtual machine, Active attacks, The attacker actively manipulates migration traffic as it is traversing the network in order to make changes to VM memory, so that they can gain access to applications such as sshd	Data Plane	(Shetty, 2013) (Aiash et al., 2014) (Ahmad et al., 2013) (Perez-Botero, 2011) (Upadhyay and Lakkadwala, 2014) (Oberheide et al., 2008)

Table 2.1: Table of attacks in Literature

2.7 Security precautions

Virtual machines that are in migration are more open to attacks than virtual machines that are not. Thus if an attacker wants to target a VM they need to know when it's live migrated. This can be achieved by pinging the machine. A spike in time taken for the ping to return indicates that the machine is in the process of being migrated (König and Steinmetz, 2011). Issues can also exist when a cloud provider wishes to migrate a virtual machine, this is because a virtual machine can be migrated without the knowledge of the users (Biedermann et al., 2013).

Multiple ideas have been researched on how to secure LM, we will now discuss the more prevalent ideas that exist in the literature.

Survey on secure live virtual machine (VM) migration in cloud (Ahmad et al., 2013)

This paper proposed that for live machine migration to be secured effectively the following precautions should be implemented:

Integrity and verification of the platform would have to be assured, which to be achieved, the destination platform must *cryptographically identify* itself to the source platform. Once they have successfully identified to one another the hosts will have to *authenticate*. Once this stage is achieved the next stage is *authorization (Access control)*. This ensures that users who initiate the LM process must be authorized to do so, Unauthorized LMs must be prevented. Once the migration has started the *confidentiality and integrity* of the virtual machine must be insured, this is achieved by using an encrypted channel between the two hosts, thus preventing man in the middle attacks. Once the migration is complete the hosts must be *Replay Resistant* If packets are somehow captured during the migration process, they can be replayed back, therefore the migration process should be resistant to replay attacks. *Source non-repudiation* source host cannot deny that it initiated a vm migration, can be achieved using public key certificate.

Secured and reliable VM migration in personal cloud (Wang et al., 2010)

The paper proposed securing the LM process by using a trusted platform module in conjunction with Intel VPro hardware. In order to achieve a secure LM both host and source must have the trusted platform module which Intel VPro hardware allows, thus each host can communicate securely to one another, and if

both have the acquired security level for LM they are bale to migrate the virtual machine. This is achieved by using a series of modules. the first module is the *attestation service* which enables a hypervisor to cryptographically identify itself to a remote hypervisor. The *seal storage* ensures that data is encrypted and also hashed, this insures that only the operating system which is the owner of the data can access it. The policy service module manages role policies thus insuring only he or she, who has the right to migrate virtual machines can. The *Migration service* module is responsible for the migration, it uses the attestation service to insure that the target machine meets the security requirements for migration to take place. Finally there is *the secure hypervisor module* which protects the processes of the guest OS while it's being migrated by using run-time memory measurement (Wang et al., 2010).

A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective (Perez-Botero et al., 2013)

This paper proposed using a trusted platform module for secure LM so that all systems within the migration process can have their integrity verified. They also stated that the initial trust in the hardware model.

PALM: Security preserving VM live migration for systems with VMM-enforced protection (Zhang et al., 2008)

This paper proposed a framework for security known as PALM in this framework memory pages of secure process are kept inaccessible to prevent an attacker gaining access to them during the LM procedure, this is achieved by using encryption.

Seamless virtual machine live migration on network security enhanced hypervisor (Xianqin et al., 2009)

This paper proposed a network security enhanced hypervisor, one which provide on demand network protection to the virtual machines which are running within it. This network enhanced hypervisor can exam the packets that are sent to virtual machines, and if that does not meet set rules they can be dropped and they do not reach the virtual machine.

Secure Live Migration of VM s in Cloud Computing : A Survey(Upadhyay and Lakkadwala, 2014)

This paper agrees with the research carried out by Xianqin et al. (2009), they state that to secure the LM process a secure network hypervisor can be used. They also state that isolating the migration traffic on it's own VLAN to insure

to ensure isolation will secure the process further.

Improving security of virtual machines during live migrations (Biedermann et al., 2013)

This paper proposes the Live Migration Defence Framework (LMDF). In this framework they detect LM by pinging the virtual machine. If the returned ping time takes an unusually long time to return it can be an indication that LM is about to start. On noticing this the VM owner can constantly dirty memory pages in order to slow down the LM. Once they have achieved this the LMDF will start to remove sensitive information such as keys from memory to insure they are not in memory and thus not transferred across the data plane.

Empirical exploitation of live virtual machine migration (Oberheide et al., 2008)

This paper proposes that in order to secure LM the following techniques need to be used. Migration traffic needs to be encrypted in order to secure data when it is transversing the data plane. Correct access controls need to be implemented on the control plane in order to ensure that administrators can only implement changes that they are allowed to. Proper programming techniques need to be implemented on the migration module to ensure that it is resistant to tampering.

A Survey on Techniques of Secure Live Migration of Virtual Machine(Shetty et al., 2012)

This paper builds on the work carried out by Oberheide et al. (2008). It discusses the need to secure the LM process and some techniques that are currently being used to secure the vulnerabilities that exist on the data plane, the control plane and the migration module. These include isolating migration traffic , implementing a secure network hypervisor and using intrusion detection systems

A Framework for Secure Live Migration of Virtual Machines(Shetty, 2013)

This paper again builds on the work carried out by Oberheide et al. (2008). Shetty (2013) armed with their 2012 research, now, propose their own framework, called Secure Live Virtual Machine Migration (LMVM) In this framework they propose the following precautions must be in place to ensure that migration is secure.

- Attestation or platform integrity verification ensures that migrating source and destination are trusted.

– Access control policies, allow the administrator to implement role based access control to enable certain administrators to do certain things and restrict them from carrying out unwanted actions.

– Digital signatures/check-sums to ensure that data is not. interfered with during migration.

– Host system firewall, allows the administrator to limit traffic to the host.

– Intrusion detection system reports suspicious intrusion attempts to the administrator

– Individual VM security, which includes VM firewalls and anti virus.

2.8 Conclusion

Cloud Computing is a complex technology which relies on other technologies such as virtualization and LM. LM is facilitating cloud providers to use dynamic load balancing and to make cost saving benefits. Plenty of research has been undertaken in the area of LM performance, but little has been carried out into the area of LM security. Security remains a a concern for IT managers to use cloud services (Rashmi Rao and Pawan Prakash, 2013; Chen and Zhao, 2012) This concern exists for LM, with some industries being hesitate to implement the technology (Shetty, 2013; Aiash et al., 2014; Upadhyay and Lakkadwala, 2014; Ahmad et al., 2013).

The aforementioned papers provide a multitude of approaches to securing LM. However little research has been carried out in approving the methods proposed. The attacks that can be carried out on LM are many, and in order to see them easily and what plane they effect, we compiled a table of attacks, and the authors who noted the attack. This table will act as a quick reference for the researcher throughout this paper.

From examining the literature it can be ascertained that LM is not an extensively researched area. LM is a rapidly evolving technology, however the amount of security research being carried out does not coincide with the evolving technology. Only a few research papers in the area of LM security are released each year. For this reason this research aims to contribute to the literature on LM security.

From the research that does exist, many frameworks and security precautions

have been recommended. Although all recommendations are different, some security precautions are recommended multiple times.

* Encrypt the data channel (Ahmad et al., 2013; Oberheide et al., 2008; Shetty, 2013)
* Implement access control (Ahmad et al., 2013; Wang et al., 2010; Upadhyay and Lakkadwala, 2014)
* Detect LM from Ping (Biedermann et al., 2013; König and Steinmetz, 2011)

This dissertation proposes that an objective implementation of these security measures be carried out on two common hypervisor systems. This will allow a best practice recommendation to be contributed back to the community.

Chapter 3

Design

The goals of this research are to design an experiment that will help investigate the security of virtual machine live migration. This chapter will detail the frameworks, methodologies and technologies that will assist us in achieving this goal.To aid us in our research we turn to the literature and in particular the work carried out by Oberheide et al. (2008), which is the definitive work in live virtual machine security. Their paper, Empirical exploitation of live virtual machine migration has a citation index of 59. The framework that Shetty (2013) proposed based on the research that was carried out byOberheide et al. (2008) will also be explored. This framework discusses attacks that can be implemented on virtual machine migration. In conjunction with these two papers, the research that was carried out by König and Steinmetz (2011) will also be explored. König and Steinmetz (2011) research discusses the ability to detect live migration by sending ping requests.

3.1 Framework

The chosen framework Secure Live Virtual Machine Migration (SLVM) discussed by Shetty (2013) as a guide for this research. This framework was chosen over other frameworks as it builds on the work that has already been implemented by Oberheide et al. (2008).

Shetty (2013) states that the following security measures must be in place to ensure secure LM:

Common Security modules:

* Attestation or platform integrity verification

  Access control policies, such as Role Based Access Control to allow users to only do what their permissions allow them to.

  Digital signature/MAC or check-sum to insure that data in transit is not tamper with.

  Encryption or decryption to ensure that data in transit is secure.

  Host system firewall, a firewall on the host to ensure that unwanted traffic is not allowed through to the system

  Intrusion detection system (IDS), a system that will alert an administrator if there is unual activity.

* Per VM firewall, a firewall installed on each VM

* Per VM anti virus; Anti-virus and anti-malware installed on each VM

In order to implement their framework Shetty (2013) implements the following security features:

* Role Based Access Control

* Firewall

* Reactive Intrusion Detection System

* Secure encrypted channel

As both Shetty (2013) and Oberheide et al. (2008) discuss the data plane as being the most crucial to protect during LM, and as such our research will focus on mostly on the data-plane. To ensure that we undertake LM experiments in a correct manner we will next investigate a methodology that can be implemented to ensure that our experiments are implemented correctly.

3.2 Methodology

In order to successfully undertake experiments we will implement the methodology discussed by Edwards et al. (2015) in their paper Creating

Repeatable Computer Science and Networking Experiments on Shared, Public Testbeds. This methodology discusses designing repeatable experiments within an IT environment. This closely coincides with our goal and for that reason it was deemed appropriate for this study. In order to complete successful testing we will formulate a clear plan for our experiments. In order to do this Edwards et al. (2015) state that a series of questions must be answered which we will now address.

There are two series of questions that need to be answered; the first are experimental design questions that will detail exactly what it is that we want to do. The second series of questions are experiment implementation and deployment questions; these questions state how we will implement our experiments.

3.2.1 Experiment Design

The questions below are the questions that we, as the researchers must answer in order to know what exactly it is, we wish to do.

* What are you trying to answer?
* What is the system you want to test?
* What question about the system are you trying to answer?

What are we trying to answer?

We are trying to answer if the framework proposed by Shetty (2013) is feasible to implement on different virtualization systems. We also want to know has the environment changed since the paper was published 2013, and new hypervisor systems are now be available. For example, we will investigate virtual machines in LM to discover if they are still unencrypted, or do migration protocols now encrypt by default. We also want to answer, what can be contribute back to the academic community, and how can we improve the SLVM framework. We also want to know if the work carried out by König and Steinmetz (2011) is correct, and that it is possible to detect LM via pinging the virtual machine. In addition to these two points we also wish to give a recommendation to the reader, as to what is the more secure vitalization system for them to implement

What is the system you want to test?

The two systems that we wish to test are XEN and KVM. Our proposed environment for testing these three systems can be found in figure 3.1 and figure 3.2

What question about the system are you trying to answer?

The questions that we want to answer are:

* Is the migration channel secure?
* Can we encrypt the channel?
* Can we implement Role Based Access Control on our systems?
* Can we contribute back to the academic community?
* Can we detect LM from within the virtual Machine?
* Is one system more secure than another?
* Will one system, out perform the other?

3.2.2 Experiment Implementation

These are the questions that must be answered to ensure that we as the researchers have an understanding of how we are going to answer our research question.

* What resources to you need to run these experiments?
* How many of these resources are needed?
* What parameters should be varied?
* What metrics should be measured?
* How large should the experiment be?
* How many more resources are needed to scale the experiment?

What resources will you need to run the experiment?

In order to facilitate testing in a way that can implement scalability and automaton as discussed by Edwards et al. (2015) we will take advantage of nested virtualization. Nested Virtualization will allow snapshots to be taken before any change is made to the environment. This snapshot will then become a last known working configuration of the environment, which

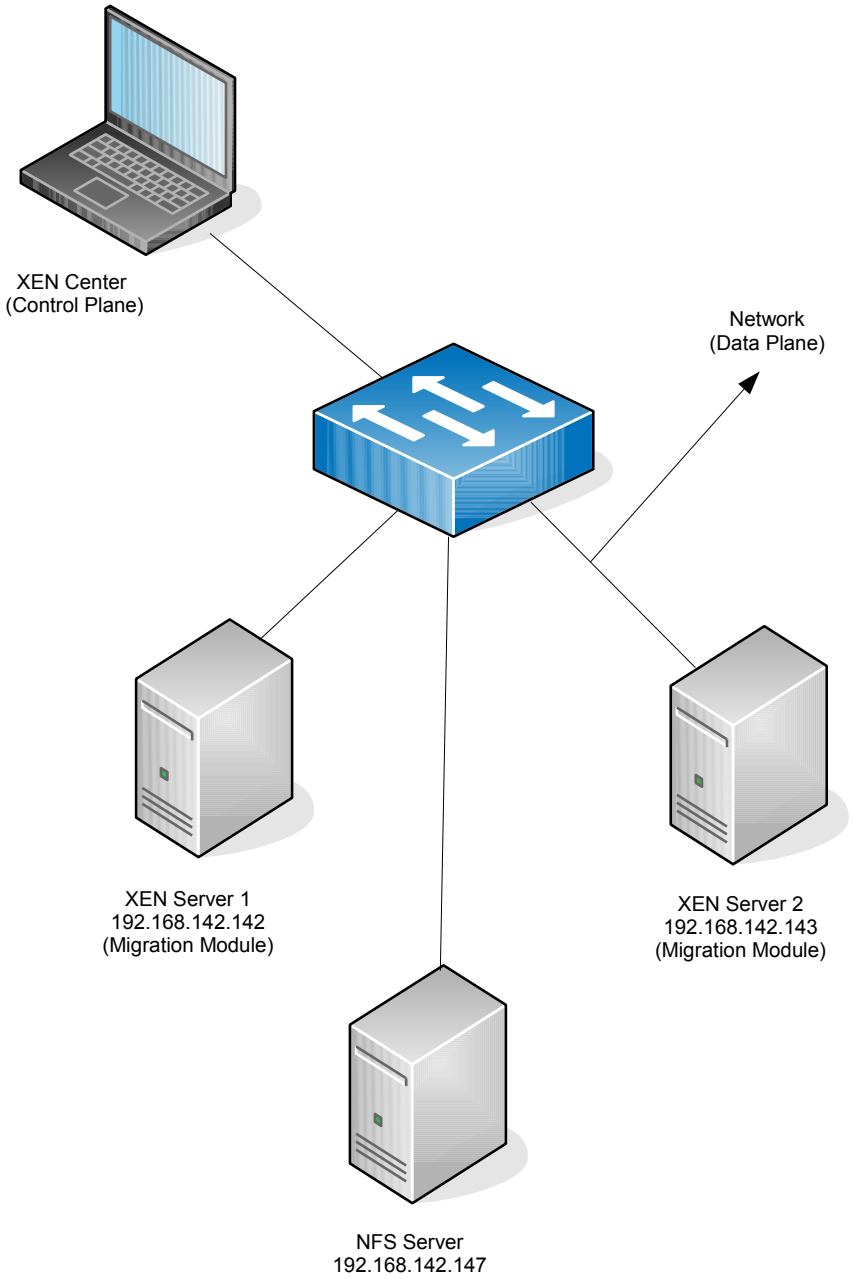

Figure 3.1: Proposed environment for XEN

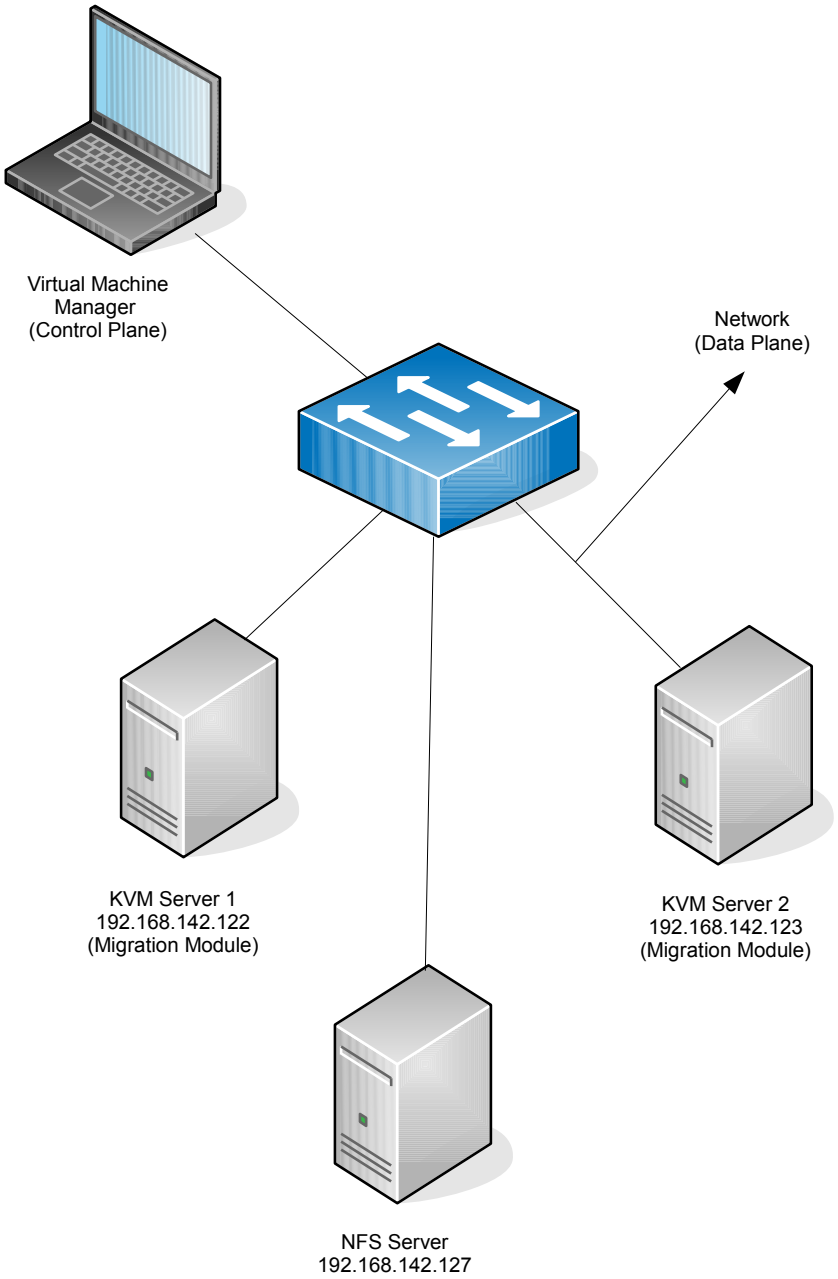

Figure 3.2: Proposed environment for KVM

we will be able to revert to should a configuration change have adverse effects on this system. Scalability of the system will be achieved by cloning virtual machines and adding them into the cluster.

Running a large number of virtual machine requires a powerful machine. For this reason a Dell Latitude E6430 machine was selected the laptop boasts 16GB of Random Access Memory (RAM), an Intel I7 processor and a 500GB hard-drive, thus allowing multiple logical processors for our virtual machines as well as sufficient ram to be allocated to the virtual machines. After the Hardware the next component of the system that needs to be selected is the hypervisor that will run the environments.

How many of those resources will be needed?

It is envisioned that we will need one Dell E6430 laptop, two Citrix XEN Servers with the XEN hypervisor installed, one installation of XEN center to connect to our virtual machines. We will need two Ubuntu servers to act as network file servers for each of our environments, two Centos servers each with KVM installed. We will also need PingHR installed on our Dell host and on each VM. Finally one version of Kali Linux will be needed

What parameters should be varied?

* Hypervisor
* User account
* Configuration settings such as changing user permissions.
* Enable/Disable encryption of the data plane
* Turn off or on RBAC
* Turn off or on IDS

What metrics should be measured?

* Data that is in clear text
* Users ability to do certain activities on the measure plain
* Ease of implementation
* Ping return times
* Emails received

Resource Required	Resource Details
VMWare Workstation 10	VMWare workstation is proprietary software that is released by VMWare, a company that specialises in Virtualization software. It is a type 2 hypervisor and can run on both UNIX and windows based systems. VMware workstation and nested virtualization were chosen as it will allow for the artifacts of the experiments to be shared with other researchers if they so wish. We will be able to provide other researchers with our virtual machine disk drives, with our configuration preloaded on them, for the researchers to do there own work. It will enable us to publish and share all files, compared to if we had run the tests on bespoke hardware.
Citrix XEN Server	Is an open source vitalization platform that is built on the XEN hypervisor. XEN is a type 1 open source hypervisor. It is released by the XEN project. XEN operates using a privileged domain architecture. While the XEN server operating system is built on the red-hat operating system kernel. We propose using XEN Server 6.5 as this is the recommended release that all new installations of XEN server use versions 6.5. Citrix XEN server uses the XEN hypervisor to facilitate virtualization.
XEN	XEN is a type one open source hypervisor. XEN uses a a privileged domain system, were one VM runs at a privilege level above the other virtual machines. The main Virtual machine is known as Dom0 and is responsible for allocating resources to other virtual machines. The other VMs are known as DomU as they are unprivileged machines. The version of XEN which we propose to use is version 4.4, which comes pre-installed on XEN Server 6.5
Citrix XEN Center	Citrix XEN Center is an open source management interface for managing XEN Server clusters and thevirtual machines that exist on these clusters. XEN Center is open source and we propose using XEN Center 6.5.
Ubuntu	Ubuntu is a Debian based Linux operating system. It is free and open source. We propose using Ubuntu as the network file share for shared storage for the XEN and KVM Server cluster.
CentOS	CentOS is a Redhat based Linux operating system. It is free and open source. We propose using CentOS as the operating system to build the KVM cluster on. The version of CentOS we propose using is CentOS 7
KVM	KVM is an open source type one hypervisor that is distributed with Linux.
Qemu	Qemu is a hardware emulator that is distributed with Linux
PingHR	PingHR is a freeware command line tool that providers the user with more in depth information when compared to the default ping operating systems.
Kali	Kali is a open source Linux operating system that is distributed with security penetration tools pre-installed.
Windows 7	Windows 7 is a popular operating system that is distributed by Microsoft.

Table 3.1: Resources that will be needed

* Pings per second during DOS attack
* Live migration Time

How large should the experiment be? The experiment will be large enough to fulfill the goal of this dissertation, it is not foreseen that new resources will be needed.

Other steps that we will implement from Edwards et al. (2015) methodology are steps to avoid human error. In order to do this it is advised that the researcher starts with a simple configuration, and change one thing at a time. the researcher should then test to see what affect this had on the system and either continue or revert the system to it's previous state, thus allowing for version control. It is also advised that all artifacts of the experiment such as configuration files, log files, results are made available to other researchers.

We propose implementing version control using the inbuilt snapshot manager in VMWare workstation. This will be achieved by taking snapshots before configuration changes and if needs be reverting to the snapshot should a configuration change have an adverse effect on the system.This will ensure that we have a last know good configuration to revert to. The methodology also recommends automating as much as possible to reduce human error. This will be achieved by cloning machines when possible.

3.2.3 OWASP Framework for Testing

In order to ensure that our testing is implemented in a successful manner The OWASP testing framework will be implemented.Owasp (2008) This framework is generalised, so some requirements are excluded due to the nature of this dissertation.

The requirements of the framework that will be implemented are:

* Review polices and standards
* Review Security Requirements
* Review Design and Architecture
* Create and Review Threat Models
* Application Penetration Testing

Review polices and standards

There are no standards within Cloud Computing mostly recommendations. XEN server issue best practices for security applications. This advice is not specific to the XEN environment and so, for the purposes of this research, these best practices will also be implemented on the KVM hosts.

* Communication between hosts should be done in SSH

This is the communication between the control plane and the hosts is secured. Passwords and other sensitive information can be sent as hosts authenticate to one another it. Both XEN Server and KVM do this by default.

* Verify each hosts identity

  As this dissertation is a small implementation carried out on a private nested virtualized environment, there will only exist two hosts who can identify to each other.

* Update each host

  Updates will be carried out on each environment after it is installed.

* Ensure that XEN system tools is installed on the guest operating systems.

  XEN tools will be installed on the XEN Server guest OS. Installing XEN system tools is not applicable for the KVM installation

Review Security Requirements

The security requirements that are needed for the system are based on Shetty (2013) and König and Steinmetz (2011) papers. To ensure that the systems are secure. The security requirements for our system is are needed to prevent.

* Compromise of the management console.
* Attack on the transmission channel.
* Attack on host
* Detection of LM

Create and Review Threat Models Based on the security requirements that have been listed the following attacks are seen as threats:

* Comprise of the management interface, either by a rogue administrator or attacker, resulting in VM manipulation, for example VM hopping,i.e. moving it from host to host to reduce performance.
* An attackers machine, that is able to detect LM of a remote machine
* A denial of service attack on a host.
* Compromise of the data plane, resulting in a man in the middle attack.

Vulnerabilities

* Rogue administrator with privilege too great
* Unsecured data plane
* Host or Virtual machine without the ability to detect a possible attack
* VM that is replying the ping requests

3.3 Conclusion

The design of this dissertation will ensure that the implementation is carried out successfully. The SLVM framework will ensure that the tests carried out are relevant to the area of LM security. The Creating Repeatable Computer Science and Networking Experiments methodology will insure that security tests carried out are implemented on an experimental environment that is suited for IT experiments. It will also ensure that the risk of human error is reduced and that automation will be used when possible. Finally the OWASP framework will insure that the environments when built will be configured with default security precautions in place, safeguarding against glaring weaknesses in the design.

Chapter 4

Implementation

The implementation for this dissertation consisted of two nested virtualized environments, one built on the XEN hypervisor and other built on KVM. The XEN hypervisor architecture consisted of two XEN server hosts connected to a network file share (NFS) server. The XEN hosts were built using XEN Server which is, open source, built on the red-hat kernel and distributed by Citrix. The network file server was built using Ubuntu, which is built on the Debian kernel. The second nested virtualized environment consisted of two KVM hosts, which are built on CentOS, which is an open source operating system built on the red hat kernel. We also built a Kali Linux virtual machine. Kali Linux is a distribution of Linux used for security testing. An Intrusion Detection System was also implemented using a Linux bash script and installed on each environment.

4.1 Implementing the XEN Server Cluster

As revealed in the literature, in order to avail of live migration, shared storage must be implemented. Therefore in order to enable live migration we will need to build a minimum of three machines. Two XEN hosts, to allow us to live migrate the virtual machine from one host to another, and a shared storage server that will hold the virtual machines virtual disk.

We took advantage of nested virtualization to build our testing environment. The advantages of this method are several, including the ability to save last working configurations as snapshots, the ability to make artifacts such as virtual disk drives available to other researchers and a reduction in researching

costs as separate physical machines did not have to be purchased.

A powerful physical machine would be needed to run three virtual machines, for this reason a Dell Latitude E6430 was choose. The Specification for this laptop is:

Dell Latitude E6430 Specification
Intel I7 processor
16GB of RAM)
500GB Hard Drive

Table 4.1: Dell Latitude E6430 Specification

16GB of Random Access Memory (RAM), an Intel I7 processor and a 500GB hard drive.

VMWare workstation was chosen to virtualize our hosts and storage. It was chosen as it is available free to all students of the National College of Ireland. To begin we first downloaded XEN server, which is distributed by Citrix and is available to download free from the XEN server website http://xenserver.org/. Once we had downloaded XEN Server, we built a virtual machine in VMware workstation to which we could install it onto. The machine we created had the following specification.

XEN Server Setup
Two Processors with Two Cores
4096 MB of RAM
20GB Hard Drive
NAT Network

Table 4.2: VM Specification for each XEN Server

Once this was done we stepped through the install process of for the Xen server. To ensure that the sever was installed correctly we powered on the virtual machine. In order to automate the creation of the second XEN host, we cloned the first host. We tested that the second host powered on correctly, which it did. From here we moved to the configuring the shared storage for the XEN servers. For simplicity in design we chose a Network File Share (NFS) server. To build a file share, first an operating system had to be chosen, we chose Ubuntu as it is open source and free. We created a virtual machine for our NFS server.The machine we created had the following specification.

In order to ensure that the NFS server would be available we assigned the

Network File Server (NFS Setup
Two Processors with Two Cores
4096 MB of RAM
50GB Hard Drive
NAT Network

Table 4.3: VM Specification for the NFS

following network settings statically.

```
1 auto eth0
2 iface eth0 inbet static
3 address 192.168.142.148
4 netmask 255.255.255.0
5 gateway 192.168.142.2
```

Listing 4.1: Network Settings Applied to the NFS Server

Once networking was configured the next step needed was to configure the machine to run as a network file share. This was done by installing the network file share software.

```
1 sudo apt-get install nfs-kernel-server
```

Listing 4.2: Command to install NFS Service

In order to facilitate shared storage for the XEN hosts, a directory was made available so that each server could mount to it. A directory was created on the root directory and called the folder export.

```
1 sudo mkdir -p /export
```

Listing 4.3: Making a Directory For Export

Since the XEN hosts will be storing virtual machines in this folder, full permissions, i.e. read write and execute had to be granted. This was achieved by running the command:

```
1 sudo chmod 777 /export
```

Listing 4.4: Full Permissions on the NFS Export Directory

Next we had to edit the /etc/exports configuration file. This file is the access control list for file systems which may be exported to nfs clients, it confirms what folders the clients have access to. We exported the /export directory with full read write access.

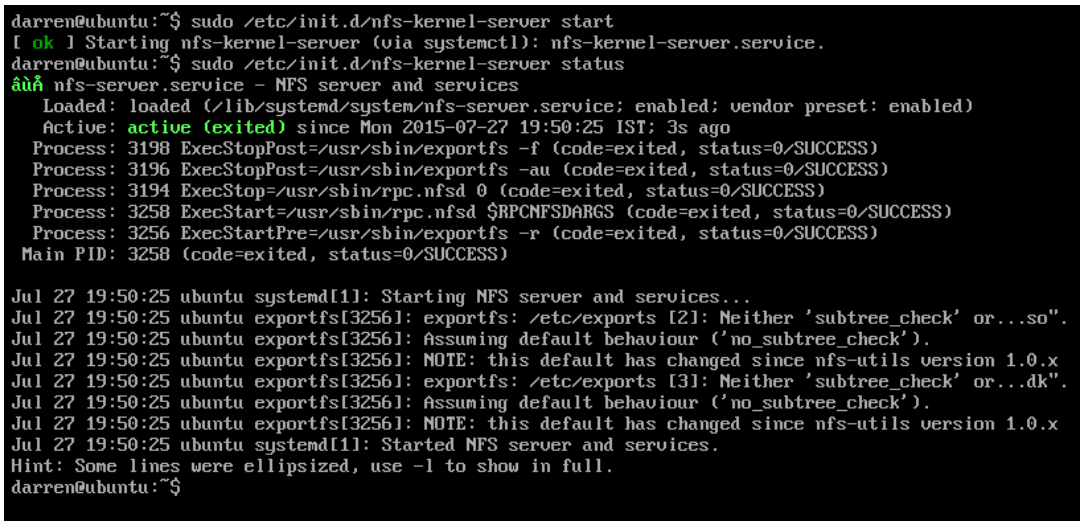

Figure 4.1: NFS server running

```
1 /export *(rw)
```

Listing 4.5: Full Permissions on the NFS Export Directory

The service was started by running the command:

```
1 sudo /etc/init.d/nfs-kernal-server start
```

Listing 4.6: Restarting the NFS Service

In order to ensure that the NFS service was started when the VM was started the following command was entered.

```
1 sudo update-rc.d nfs-kernal-server defaults
```

Listing 4.7: Ensuring the NFS service is started when the server boots

Now that the NFS server was running, and attached to a network, the XEN hosts were now configured so they could attach to the shared storage. The network settings of the XEN hosts were set statically to ensure that network configuration would not change automatically and thus break the cluster. The network settings added to the master sever are:

```
1 IP Address: 192.168.142.142
2 Netmask 255.255.255.0
3 Gateway: 192.168.142.2
```

Listing 4.8: Network Configuration for Master XEN Server

The network settings added to the slave serer in the XEN cluster are :

```
1 IP Address: 192.168.142.143
2 Netmask 255.255.255.0
3 Gateway: 192.168.142.2
```

Listing 4.9: Network Configuration for Slave XEN Server

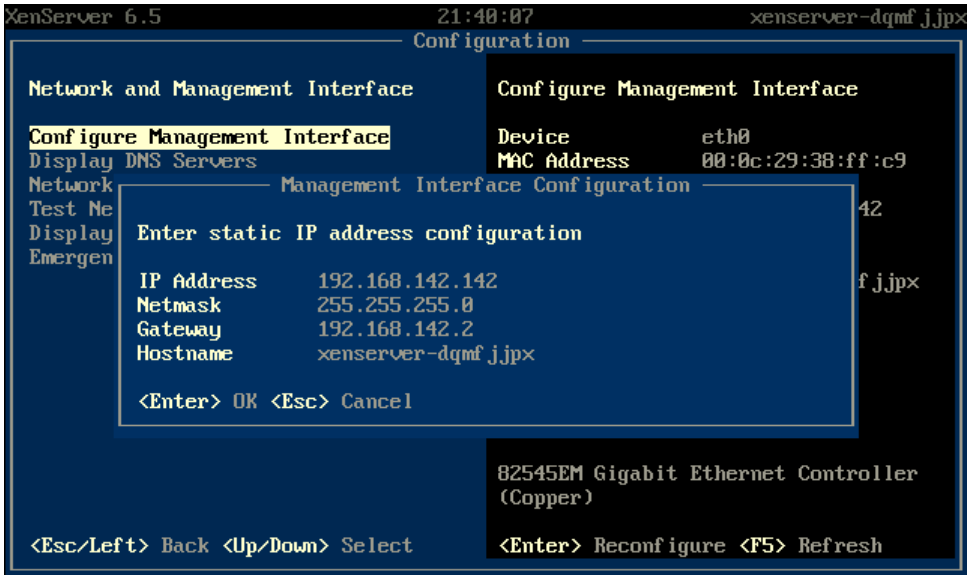

Figure 4.2: Xen Network being set up

Next it was ensured that all machines could see one another on the network. This was achieved by pinging each machine from another machine and confirming that all servers were able to contact one another over the network. Once this was done snapshots were taken of all machines to ensure that there was a last known working configuration of all machines.

Now that all severs could communicate with one another, a management interface that will allow communication to the cluster was needed. To achieve this XEN Center was downloaded and installed . XEN Center is free, open source and available from http://xenserver.org The cluster was configured in XEN Center by first adding a new pool. From within this new pool the two XEN hosts were attached. Finally shared storage was added to the pool. The XEN hypervisor clustered environment was now complete.

No unexpected issues occurred on the installation of the XEN server environment and the installation went as expected. The XEN hosts both connected to the NFS without issue. XEN center connected to both XEN hosts and the NFS without issue. Now that the XEN environment is complete, the

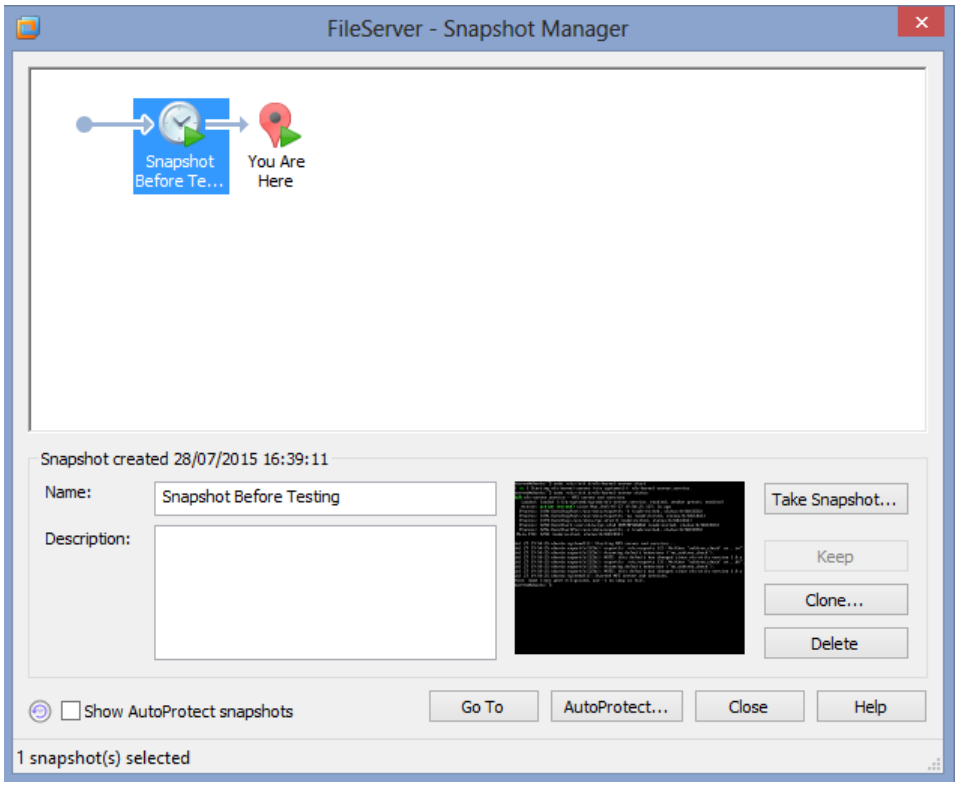

Figure 4.3: File sever snap shot on VMware Workstation

KVM environment will be implemented.

4.2 Building the KVM Cluster

A clone of the NFS was taken, to be used as the NFS for the KVM system. Next a CentOS machine was built. This machine had the following specification:

KVM Server Setup
Two Processors with Two Cores
4096 MB of RAM
20GB Hard Drive
NAT Network

Table 4.4: VM Specification for each KVM Server

During the install of CentOS, the virtualization host options were selected.

This ensures that the server is pre-configured for virtualization. Once CentOS installed, the machine was booted to insure that it was configured correctly. When it was ascertained that it was, a snap shot of the machine was taken. The machine was then cloned, to become the second KVM host.

The first KVM host network settings were configured as follows:

```
IPADDR=192.168.142.122
PREFIX=24
GATEWAY=192.168.142.2
DNS1=192.168.142.2
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
NAME=eno16777736
UUID=b6e4e55b-2b26-4f66-975c-aebb65aafa46
DEVICE=eno16777736
ONBOOT=yes
```

Listing 4.10: Network Configuration for KVM Host One

The second KVM host, network settings were configured with the following settings:

```
IPADDR=192.168.142.123
PREFIX=24
GATEWAY=192.168.142.2
DNS1=192.168.142.2
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
NAME=eno16777736
UUID=b6e4e55b-2b26-4f66-975c-aebb65aafa46
DEVICE=eno16777736
ONBOOT=yes
```

Listing 4.11: Network Configuration for KVM Host Two

The NFS server was given the following network configuration

```
auto eth0
iface eth0 inbet static
```

```
3 address 192.168.142.127
4 netmask 255.255.255.0
5 gateway 192.168.142.1
```

Listing 4.12: Network Configuration for NFS Server

Next it was ensured that the virtualization service was running by issuing the command:

```
1 service libvirtd start
```

Listing 4.13: Command to start Virtualization Service

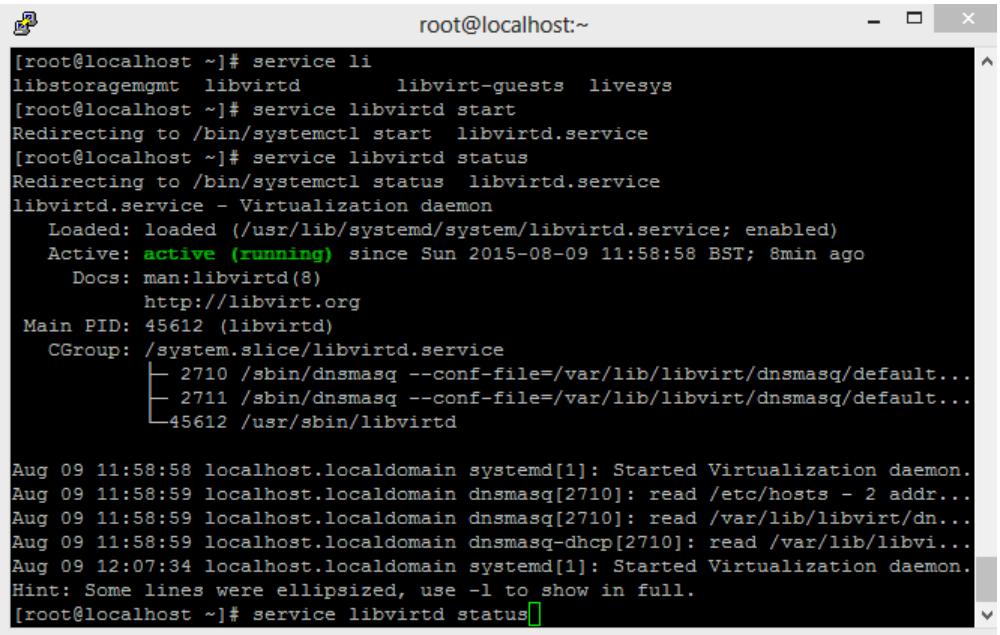

Figure 4.4: The Virtualization Service running on a KVM Host

Next the NFS volume was mounted onto each KVM host by running the following command:

```
1 mount -o wr 192.168.142.127:/export /kvm
```

Listing 4.14: Command to mount NFS volume

Installing CentOS as a virtualization server comes with virtual machine manager pre-installed. Virtual machine manager is a graphical user interface for managing VMs.This is the control plane on the KVM environment. A KVM cluster was created within the virtual machine management interface. The second KVM host was added using the settings:

* Hypervisor: QEMU/KVM
* Method: SSH
* Username: Root
* Password: Password for host
* Hostname: 192.168.142.123

The KVM cluster was now set up successfully.The installation of KVM is also somewhat simplified by the bundling of the virtual resources withing CentOS 7. The installation went well and with little issue. Now that the two environments are set up, testing can start. The first test that will be run against the environments will be an investigation into detecting LM by using pings.

4.3 Detecting Live Migration

Within the literature there exist discussions on detecting virtual machine migration. These discussions focus on the ability to detect virtual machine live migration in order to attack a VM (detecting from outside the VM) or the ability to detect live migration from within the VM in order to be prepared to defend against an attack that may happen during live migration. König and Steinmetz (2011) discuss pinging the virtual machine in order to detect live migration. We attempt to ping out from the virtual machine in order to detect live migration.

In order to test this decided to ping out of the virtual machine to see if there was any noticeable drop off in ping return times while the vm was migrating. In order to facilitate this we downloaded a program called HR ping. HR ping gives a more accurate return time than the default windows ping. A total of four experiments we're implemented. Each of these experiments was carried a total of five times. The four tests that we carried out are:

* Detecting migration from within a virtual machine that is being migrated on KVM
* Detecting migration of a VM from a remote machine on KVM
* Detecting migration from within a virtual machine that is being migrated on XEN

* Detecting migration of a VM from a remote machine on XEN

The process for implementing the test was based from König and Steinmetz (2011) paper in which they ping another node within the network, in this experiment, the default gateway is pinged.

4.3.1 Detecting LM from VM that is being Migrated

This experiment was carried out a total of five times. The configuring for the experiment consisted of starting the a ping -t command from the VM that was about to be migrated to the default gateway. The command to do so is:

```
1 hrping -t 192.168.142.2
```

Once the ping was running, at the 10th ping the migration was started. The return times were monitored and the migration end time noted. The ping command was then halted and the data saved to a log file, which could be interrogated later. This process was implemented on both systems for detecting LM from within the VM that was been migrated.

4.3.2 Detecting LM from outside the VM that is being Migrated

The process that was carried out was similar to that of detecting LM from within the VM that is being migrated. In order to test, the VM was pinged from a machine that was outside the virtualized environment. The command that was issued was:

```
1 hrping -t 192.168.142.142
```

Once ping was running the LM was started again at the 10th ping. The he return times were monitored and the migration end time noted. The ping command was then halted and the data saved to a log file, which could be interrogated later.This process was implemented on both systems for detecting LM of a VM from an external machine.

These tests were successful in detecting a LM from both an external machine and from within a VM that is being migrated. This is the first step in implementing a successful attack. The attacker must know if the machine

is being live migrated. Once the attacker knows this, they can implement attacks on the machine that is being transmitted over unencrypted migration channel.

4.4 Unencrypted Migration Channel

Throughout the literature it can be seen, that live migration data is not encrypted by default when it is being transferred from host to host. This means that any traffic passed between the hosts is in clear text and susceptible to a man in the middle attack. To test this we did the following.

We began our investigation by first installing a virtual machine on XEN cluster. We choose windows 7 as it is the most popular desktop operating system that exists. We then migrated the machine from host to host to ensure that live migration was operating correctly.

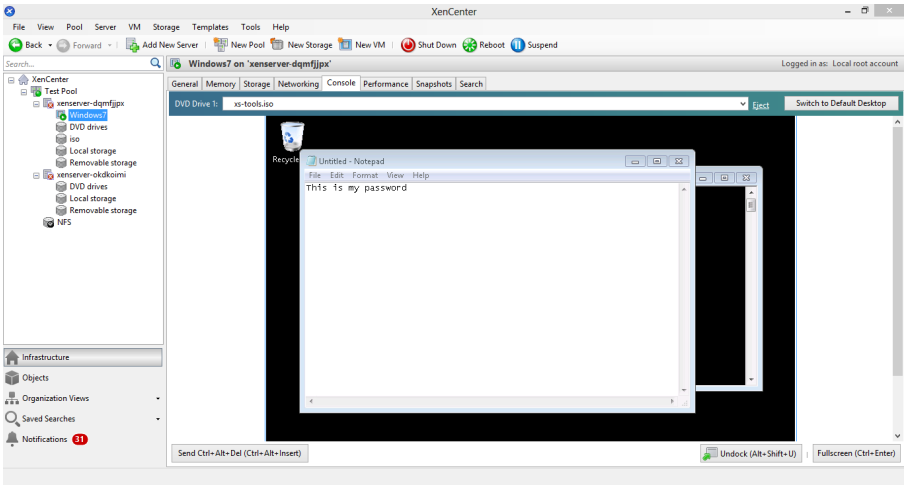

Figure 4.5: Windows Server During Migration

To investigate if live migrated machines memory was sent in clear text over the data plane we first migrated the windows 7 VM from host 192.168.142.142. to host 192.168.142.143. We logged into host 192.168.142.142 and ran the command

```
1 tcpdump - i any -W > file
```

Listing 4.15: Command to Collect Data From any Interface and send output to a File

This command gathers all information that is passed over the network interface into a file that we can then integrate for information after. We then started the live migration of the windows 7 virtual machine. Once the migration was complete we halted the tcpdump command. We then checked the file to see if data was in clear text.

We followed this same implementation on KVM to first check if the migration channel was in clear text. From here we implemented a man in the middle attack.

4.4.1 Implementing Man in the Middle Attack

The man in the middle attack we implement can be seen in figure 4.6 We implemented ARP spoofing, which tricks the hosts into sending their data to another machine, in this case, the Kali Linux machine. We had configured the Kali host to intercept traffic both coming to and from each host. Kali is also configured to pass the traffic on to the correct destination after it had sniffed the traffic, making the hosts completely unaware that there is an issue. We implemented this man in the middle attack on each system, once before we encrypted the data plain and once when we the data plane was not encrypted. We have included a detailed set up on how to create an Arp spoofing man in the middle attack in the appendices.

We now knew that any traffic that was communicated between the two XEN hosts would be intercepted by the Kali Linux Virtual machine. In order to capture any traffic that went betwee the we then opened a third terminal and entered the command

```
1 tcpdump -i any -X > file
```

4.5 Encrypting the Data Plane

Now that we had implemented a man in the middle attack on an unaltered data plane, the next step that we undertook was to encrypt the data plane to see if a difference existed on the kind of information that could be gleaned while the virtual machine was been migrated. We used a service called racoon to encrypt traffic between the two hosts. The settings for the encryption that we used are as follows.

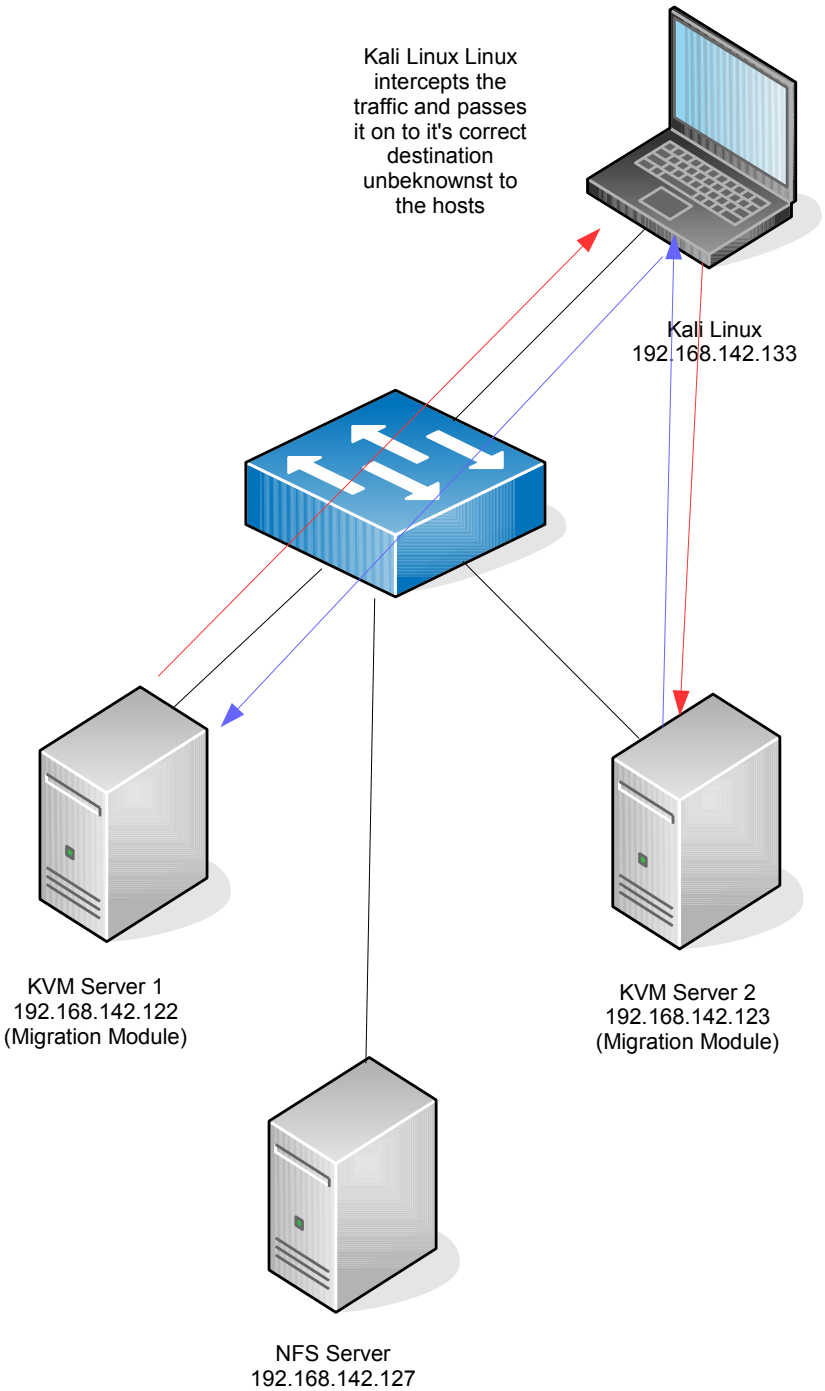

Figure 4.6: Kali Arp spoofing Man in the Middle Attack.

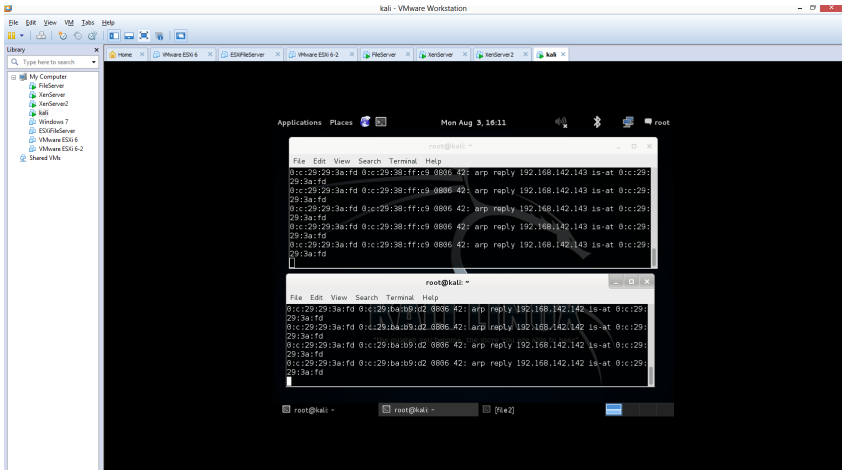

Figure 4.7: Kail Listening for traffic from each host

* encryption algorithm 3des
* hash algorithm sha1
* authentication method pre shared key

We received feedback from each machine that the data plane was now encrypted.

4.6 Protecting the Control Plane with Role Based Access Control (RBAC

The control plane, in XEN server is XEN Center. From here the administrator of the system can initiate live migrations, create virtual machines and delete virtual machines. Shetty (2013) discusses using Role Based Access Control for protecting the virtual machines. Withing KVM the control plane is the Virtual machine manager application.

4.7 Implementing RBAC on XEN

On installation of XEN server the default user is root. This user has full privileges on the machine. The user can access a command prompt as root and run changes on the host. This user can then log into XEN Center and has full permissions to do what they please with the virtual machines. This is far from an ideal situation. To begin implementing RBAC we first created

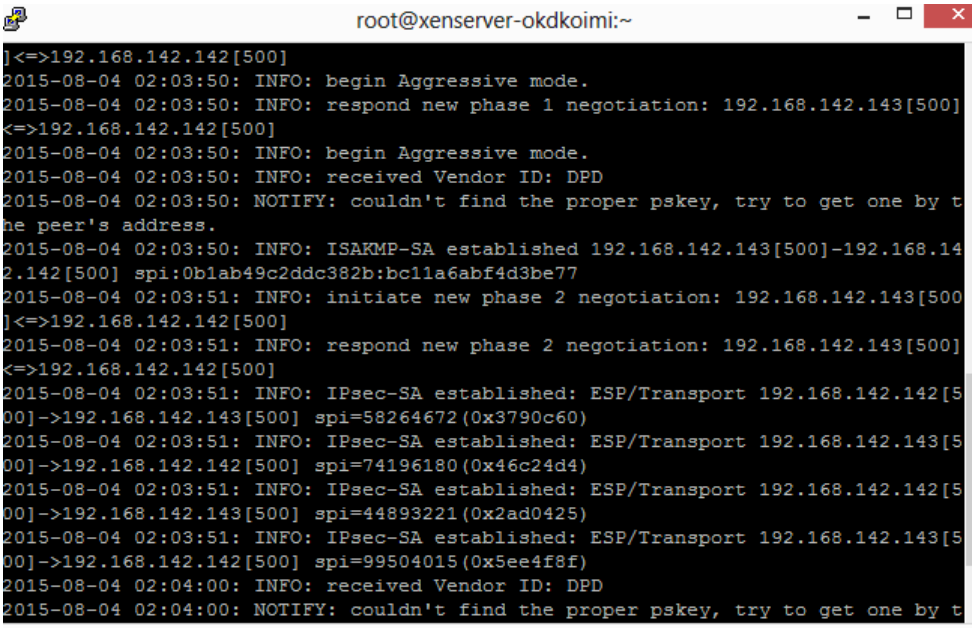

Figure 4.8: Proposed environment for XEN

an alternative user called darren. This was achieved by running the useradd command on our master XEN host (192.168.142.142) Once done we set a password for the user darren by entering the command passwd and entering a password.

However issues exist with XEN Server for implementing RBAC on local accounts.

To allow RBAC on XEN we first needed to enable it for local accounts this was achieved by first enabling PAM on the server.

```
1 xe pool-enable-external-auth auth-type=PAM service-name=test
```

Listing 4.16: Command to enable PAM

This created the service name test, that uses PAM. We then ran the command

```
1 xe subject-add subject-name=darren
```

Listing 4.17: Producing a UUID for the user Darren

This produced a UUID for the account darren. The UUID was 72eddea3-480d-3745-e0b6-f31da81872ac. Next we ran the command

```
1 xe-subject-role-add uuid=72eddea3-480d-3745-e0b6-f31da81872ac role-name= ↩
      read-only.
```

Figure 4.9: The Testuser under it's curreny permissons is unable to access the Vitual machine monitor

```
```

Listing 4.18: Granting User darren Read Only Permissions

Once we had this completed we had two accounts that we evaluated, root with full permission, and darren with read only permissions. Next we attempted to implement RBAC on KVM.

4.8 KVM RBAC

To attempt to implement RBAC on the KVM we first added an new user called testuser on each host. The default permissions on this account are the lowest level of privilege. We then set a password for this user passwd.Next we logged into the testuser account and attempted to access the virtual machine monitor. From here we could see that we could not access it.

We first created a policy kit local authority file by running the following command on each server:

```
1 vi /etc/polkit-1/localauthority/50-local.d/15test.pkla
```

Listing 4.19: Creating Policy Kit File

This created a configuration file with the following information.

```
Identity=unix-group:testuser
Action=org.libvirt.unix.manage
ResultAny=yes
ResultInactive=yes
ResultActive=yes
```

Listing 4.20: Adding Configuration to File

This is giving our test user account full management access on the virtual environment. From here we could see that it was possible to connect to the virtualized environment and we had full control to migrate machines

We next attempted to edit the rights to allow only monitor/read only access. To do this we changed the configuration file on each machine to:

```
Identity=unix-group:testuser
Action=org.libvirt.unix.monitor
ResultAny=yes
ResultInactive=yes
ResultActive=yes
```

Listing 4.21: Changing Test Users Rights to Read Only

Now we have to users on KVM that we could evaluate, test user and root.

4.9 Implementing Intrusion Detection System

In order to detect if the host was under a denial of service attack we implemented a script that would send emails to a designated email address on high network load.In order to achived this we first had to configure both XEN and KVM systems to be able to send emails. This was achieved by installing two packages, mailx and sstmp. To configure sstmp to send mails we entered the following configuration into it;s conf file.

```
AuthUser=darcom51@gmail.com
AuthPass=Password
FromLineOverride=YES
mailhub=smtp.gmail.com:587
UseSTARTTLS=YES
```

Once the system was capable of sending mails, we next installed dstat and implemented the script 4.22. Our script contained a command that would run the dsat command and send network information to a file, from here the

script would look to see if the network value was above a certain amount and if so to send a mail warning the administrator of a possible attack. We positioned the script on both the KVM hosts and both the XEN hosts. We then added a listing to cron that would run the script every minute, thus meaning that the network bandwidth is being monitored every minute of everyday.

```
#!/bin/bash
# Launch script in background
dstat -n > file &
# Get its PID
PID=$!
# Wait for 2 seconds
sleep 3
# Kill it
kill $PID
cat file | awk 'FNR == 4 {print $1}' > testfile
value=$(<testfile)
size=${#value}
#echo $size

if [ "$size" -gt "2" ]
then
echo "WARNING:POSSIBLE DOS ATTACK ON HOST 1" | mail -s "WARNING POSSIBLE ↩
    DOS ATTACK ON XEN HOST 1" darcom51@gmail.com

fi
```

Listing 4.22: IDS Script

In order to test if the script would for a simple DOS attack we will issue the command below. This command sends a ping with 65000 bytes to the hosts. We then tell the script to send an email if a threshold is reached, thus we know if the script has been successful in alerting the administrator of a possible attack

```
ping 192.168.142.122 -t -l 65000
```

4.10 Testing Migration Times

LM times on both environments were measured to see if any discernible difference was observable in LM time when the data plane was encrypted and when it was decrypted. This was achieved by building a windows 7

VM. The VM had 1GB of of RAM and was left under low CPU load during the migration. The machine was live migrated on each system a total of 10 times, fives times when the channel was encrypted and five times when the channel was not. The test was carried out by using a stop watch and starting a LM, observing the LM and noting the time when the migration finished. There was a large difference in encrypted LM times when compared to non-encrypted LM times. The performance of KVM was also superior to that of XEN.

Chapter 5

Evaluation

In the chapter we will evaluate results from the experiments that were conducted on the environments that were built in the implementation chapter of this dissertation. We will know evaluate each of these experiments.

5.1 Unencrypted Migration Channel

An investigation into the security of the data plane on two systems XEN Sever and KVM was undertaken. First a man in the middle attack with no encryption enabled on the data plane was implemented. The tests on both XEN and KVM exported the data to a log file which was then observed. It could be seen that data was sent in clear text in both environments. Details such as machine name could be found and paths that indicated what operating systems was been used were easily visible in the output that was produced. Figure 5.1 shows part of a man in the middle out put file been viewed in the terminal. Multiple authors in the literature review stated that the data would be in clear text. Thus this test yielded a result that was expected.

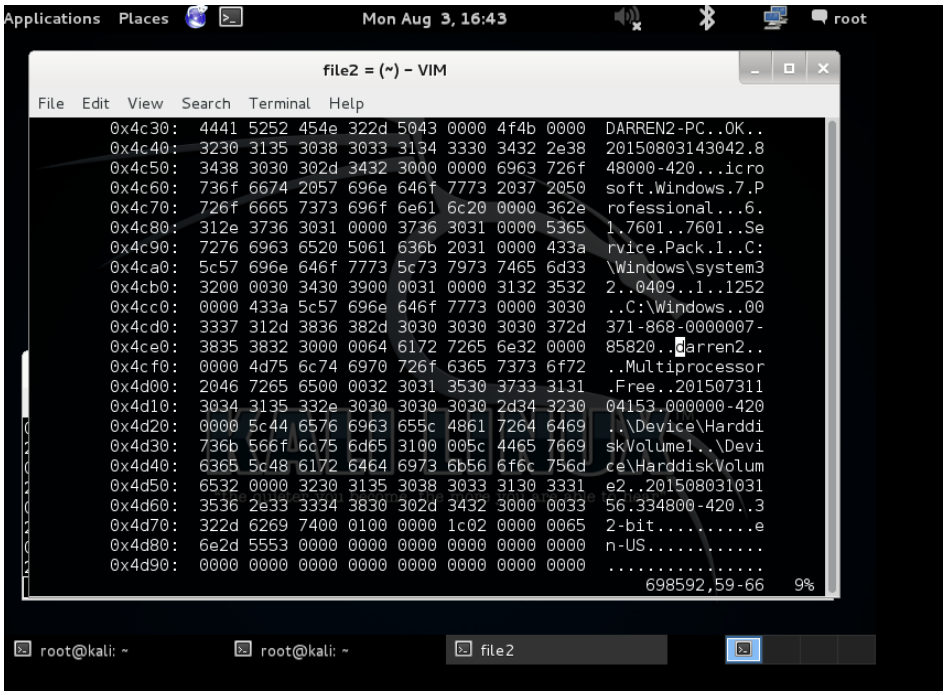

Figure 5.1: Clear text in Kali from Man in the Middle Attack

Next encryption was enabled on both systems and the tests repeated. The data from the attack was again exported to a file. This file was then observed and compared with the file from the previous test. An example of both files from each test can be seen in 5.2 and 5.3. When the data plane was encrypted no discernible data about the VM that was being migrated could be gleaned from the Man in the Middle attack. Thus the protective measure means the VM was protected from a passive snooping attacks

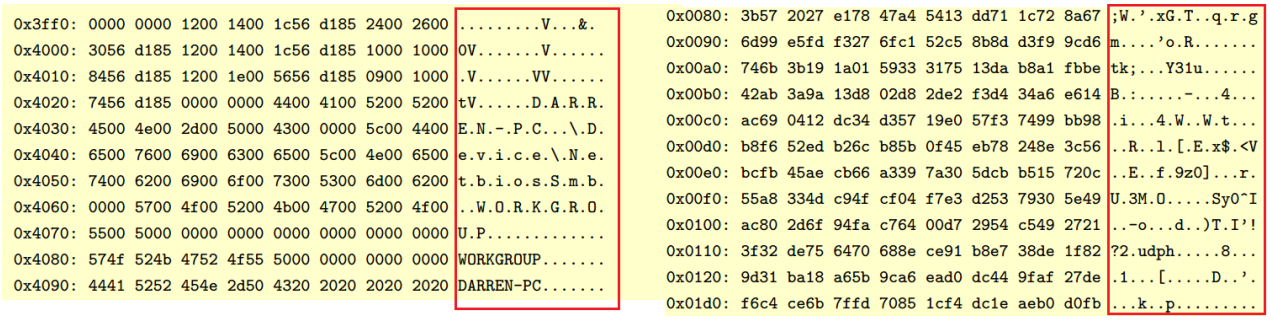

Figure 5.2: Data in XEN is in clear text in the first column, once the channel was encrypted there was no discernible data in clear text

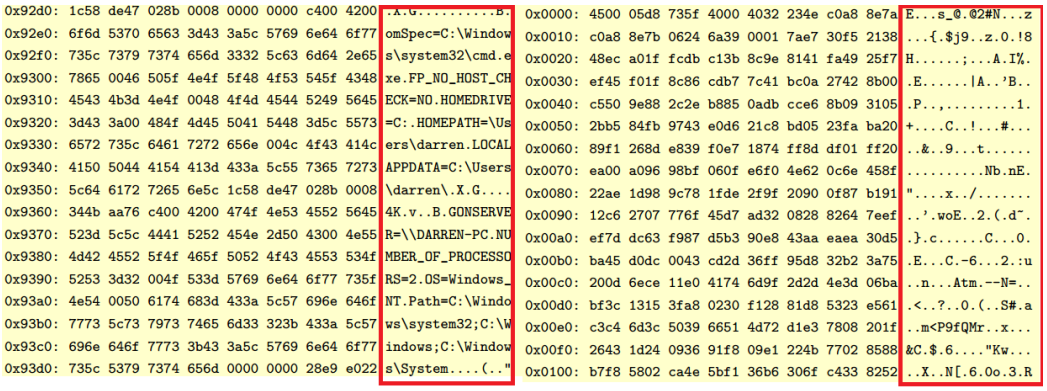

Figure 5.3: Data in KVM is in clear text in the first column, once the channel was encrypted there was no discernible data in clear text

5.2 Evaluating Migration Times

In order to evaluate if there was a significant difference between migration times that occurred when the data plane was not encrypted and when it was, we migrated a VM five times on XEN when the data plane was unencrypted and 5 times when it was. We ran the same tests on KVM. The results if each test, the median time and the mean time for both systems is seen is table 5.1

System	Encryption Status	Total LM Time In Seconds	Total LM Time In Seconds	Total LM Time In Seconds	Total LM Time In Seconds	Total LM Time In Seconds	Mean LM Time In Seconds	Median LM Time In Seconds
XEN	Non Encrypted	24.33	13.81	13.37	12.17	13.51	15.44	13.51
	Encrypted	115.74	106.34	101.14	105.39	105.59	106.84	105.59
KVM	Non Encrypted	3.87	4.16	6.52	4.38	4.76	4.74	4.38
	Encrypted	62.81	62.09	58.21	59.12	60.28	60.52	60.28

Table 5.1: LM times for both XEN and KVM, while unencrypted and encrypted

The results we're noted from the tests and added to this table, to allow ease of readability. The results show that encrypted LMs take considerably longer than non encrypted migrations. This result was not expected, as in the literature would not take significantly longer, which was the case, in our experiments carried out. From the results it can be seen that KVM preformed out preformed XEN in every test. We can see that the mean

non encrypted migration time for KVM migrated the VM 208.45 percent quicker than XEN. While for the the encrypted channel KVM migrated the virtual machine 75.08 percent quicker. The median and mean times for both systems have also been plotted in graphs and can be seen in both 5.4 and 5.5. The results show that the performance of KVM is far superior than XEN.

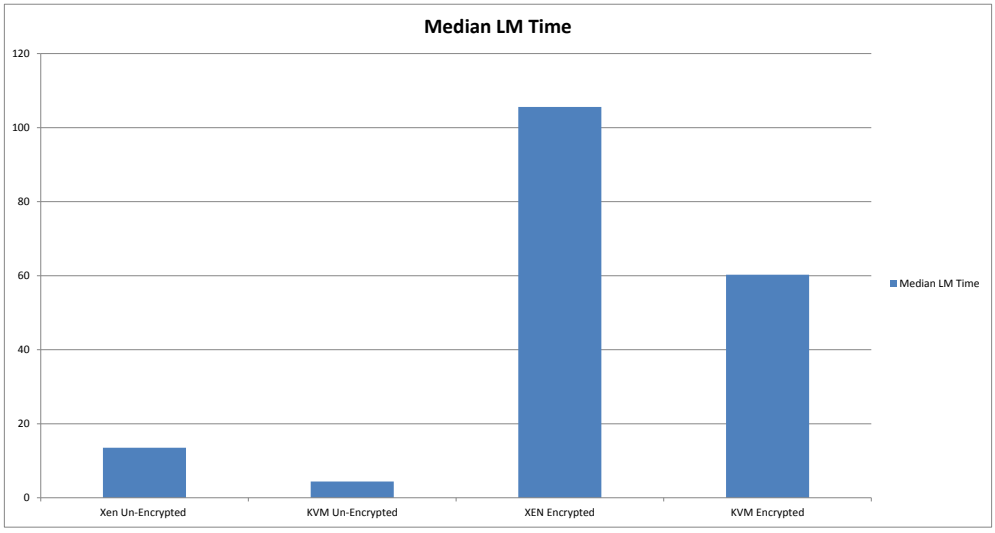

Figure 5.4: Data in KVM is in clear text in the first column, once the channel was encrypted there was no data in clear text

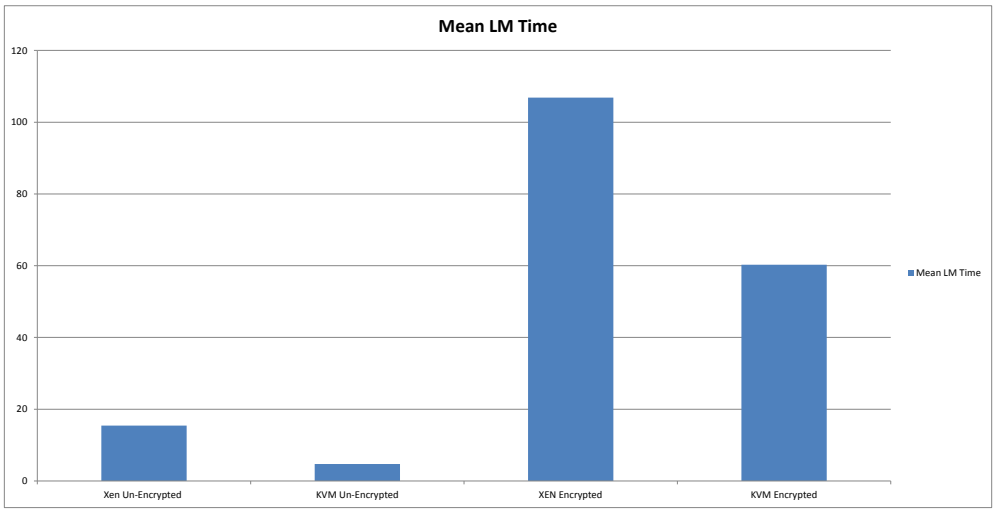

Figure 5.5: Data in KVM is in clear text in the first column, once the channel was encrypted there was no data in clear text

5.3 Evaluating Intrusion Detection System (IDS)

The IDS was evaluated under two tests. One were the it was not enabled on the system and one in which it was enabled. When the system was running without the IDS and a DOS attack launched there were no warnings on either KVM or XEN that an attack was happening. However when the IDS was enabled and the test re-ran on both systems, a email was sent from both systems to the administrator. This email alerted that a possible attack may be happening on the host of the affected system. 5.7 5.6

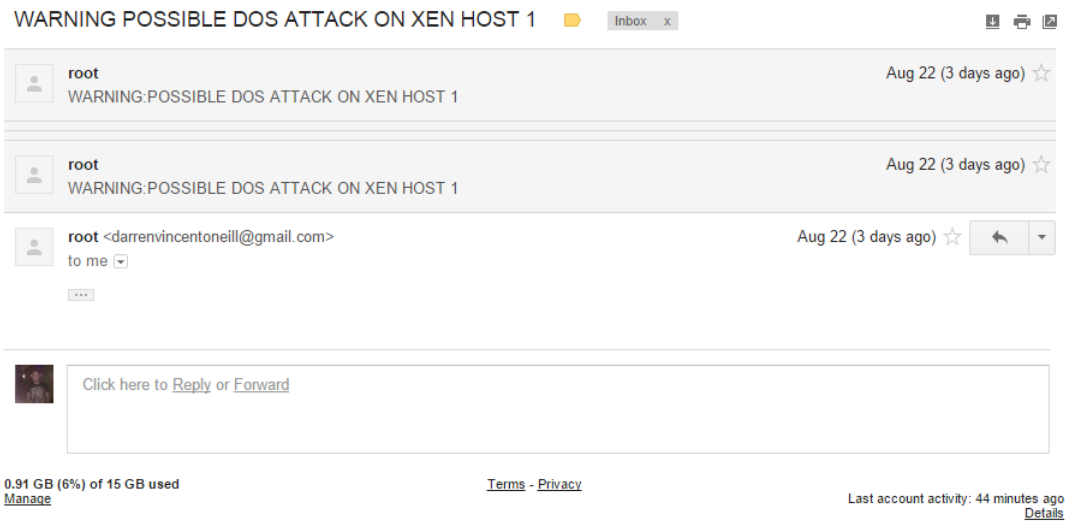

Figure 5.6: Warning mails that were received from XEN host when DOS attack was running

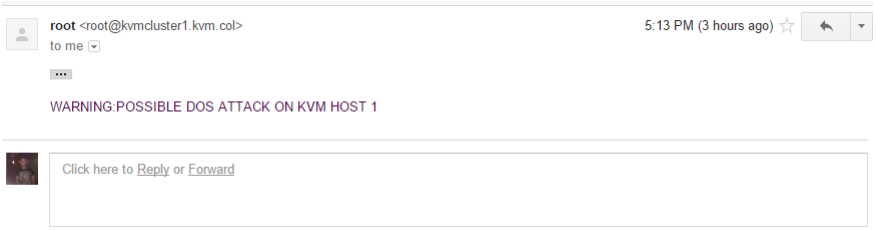

Figure 5.7: Warning from KVM Host One that it may be under attack

5.4 Role Based Access Control

Role Based Access control wad evaluated by seeing what a full administrator user could do a what a read only user could do on each system. We based the evaluation on the control plane attacks that are discussed by Shetty (2013):

* Initiate a Migration
* Overload a host
* Useless migration of VM
* VM hopping
* VM collocation
* Monitor VMs

	Root Access	Read Only Access
Initiate a Migration	Yes	No
Overload a host	No	No
VM collocation	Yes	No
VM hopping	Yes	No
Monitor VMs	Yes	Yes

Table 5.2: Actions possible as Root and as Read Only on XEN Server

In table 5.2 the evaluation of RBAC on XEN Server can be seen. A user with root access can do the majority of tasks. It was not possible to overload a host as XEN Server would prevent a root user from having a host over subscribed. The server would issue a warning stating that it was not possible to run the VMs requested on the host. The read only user, allowed the user to monitor machines while not allowing the user to do any other tasks.

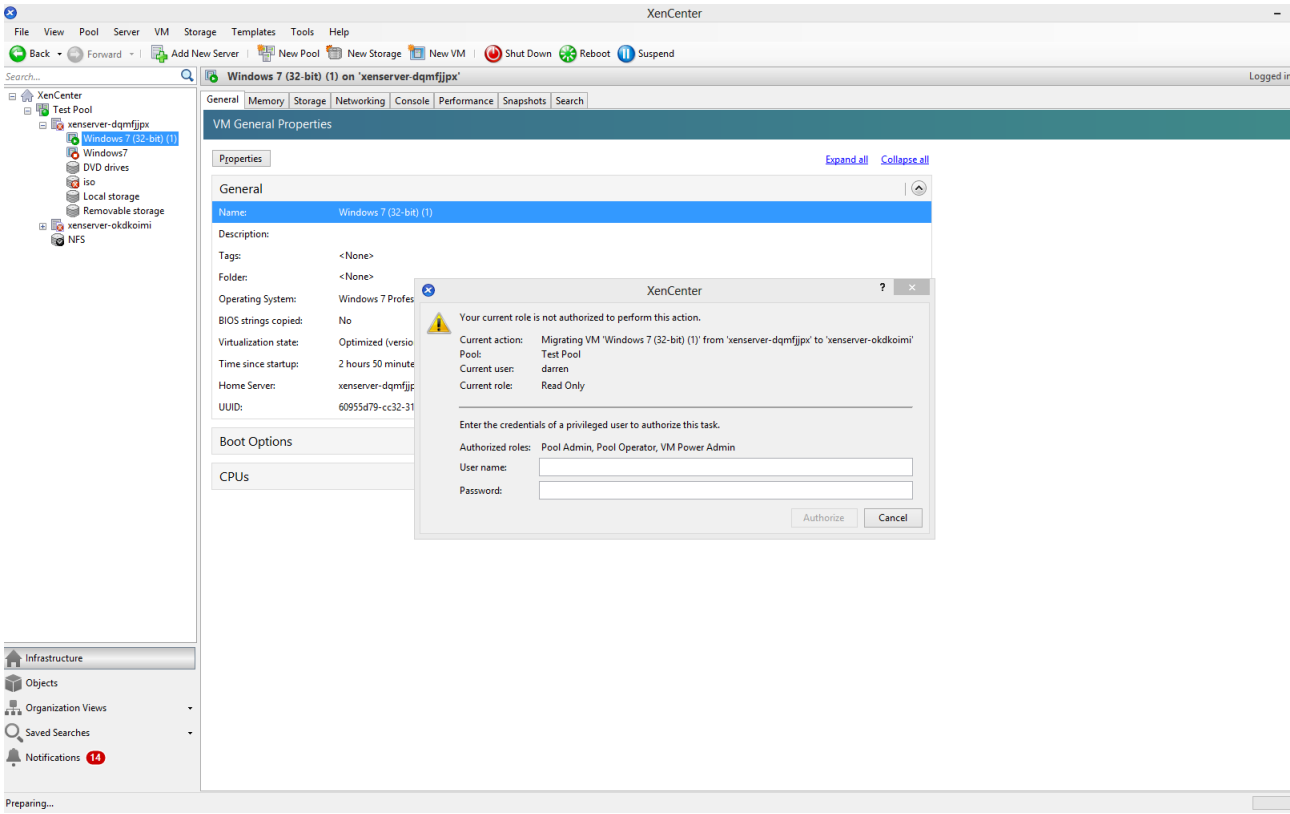

Figure 5.8: Role Based Access Control preventing unauthorised migration on XEN

	Root Access	Read Only Access
Initiate a Migration	Yes	No
Overload a host	No	No
VM collocation	Yes	No
VM hopping	Yes	No
Monitor VMs	Yes	No

Table 5.3: Actions possible as Root and as Read Only on KVM Server

Implementing RBAC on KVM was not successful. An option for a read only user exists, however it is not possible to even monitor the machines with this account, thus making the read only account redundant.

5.5 Detecting Migration Via Ping

A total number of four experiments were ran five times to detect LM:

* Detecting LM from within a VM that is been migrated on KVM.
* Detecting LM of a VM from a remote machine on KVM.
* Detecting LM from within a VM that is been migrated on XEN.
* Detecting LM of a VM from a remote machine on XEN.

The test in figure 5.10 is a test to detect LM from within the VM that is being migrated. In all five tests the same pattern was observed. A sharp rise in ping return time when LM was instated and a sharp rise in ping return time when LM was completing. These results show that a sharp rise in ping return time may indicate that a LM has started on the VM.

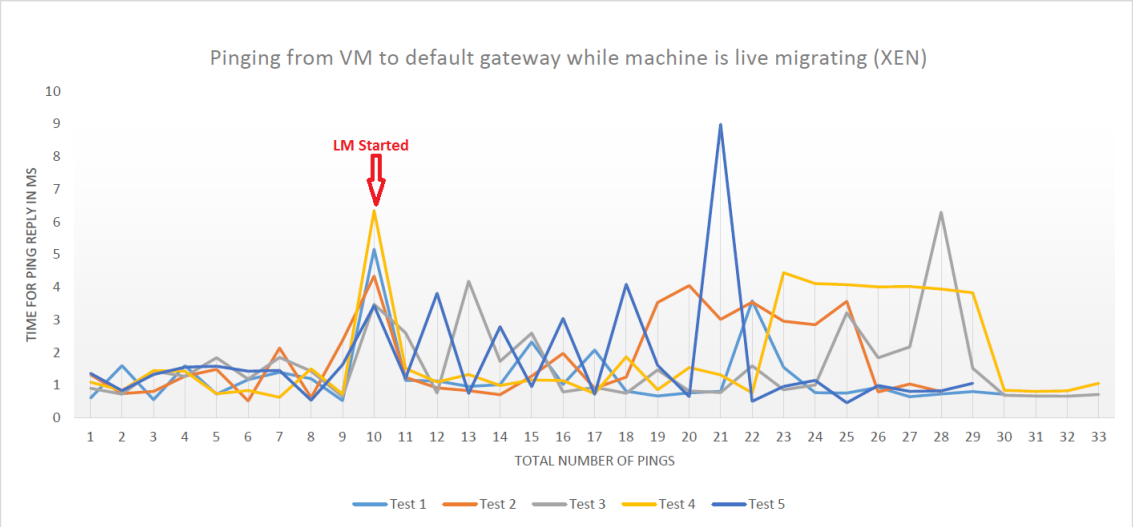

Figure 5.9: Five Xen tests, detecting VM LM from the VM that is being Live Migrated

The next experiment tested was it possible to detect LM of a VM from a remote machine.The test was ran five times. The results can be seen in **??** For ease of readability all migrations started at the forth ping. Similar to the previous test a large spike in ping return times as LMs started was observed. The ping return rate was then erratic during LM. Then as LM began to finish the machine became briefly unreachable, when LM was complete the ping rate returned to a steady low rate. Thus we conclude that with the sharp spike in ping return times that is a indicated of LM being initiated and that an attacker can detect LM.

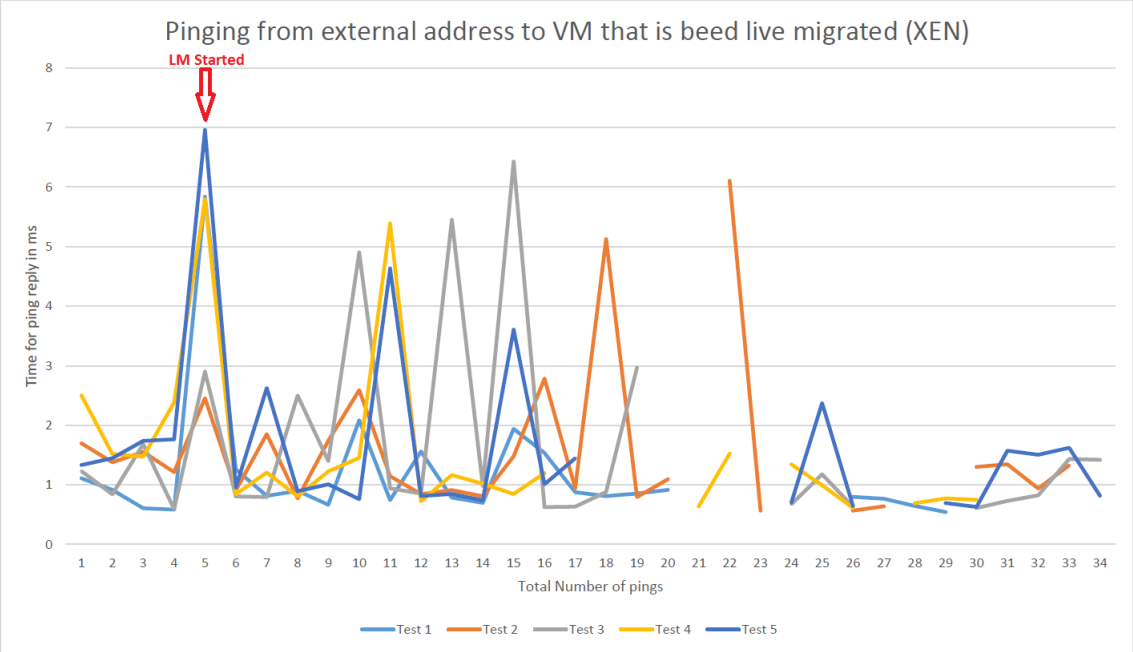

Figure 5.10: Five Xen tests, detecting VM LM from a Remote Machine

Two tests were ran in KVM. The first, attempted to detect LM from within a VM 5.11. It can be seen that 80 percent of the tests experienced a sharp rise in ping return time on LM initiating, with test one been the exception to this rule, experiencing no substantial increase in ping return time. We however still concluded that the user of the VM will be able to detect the vast majority of LMs that occur.

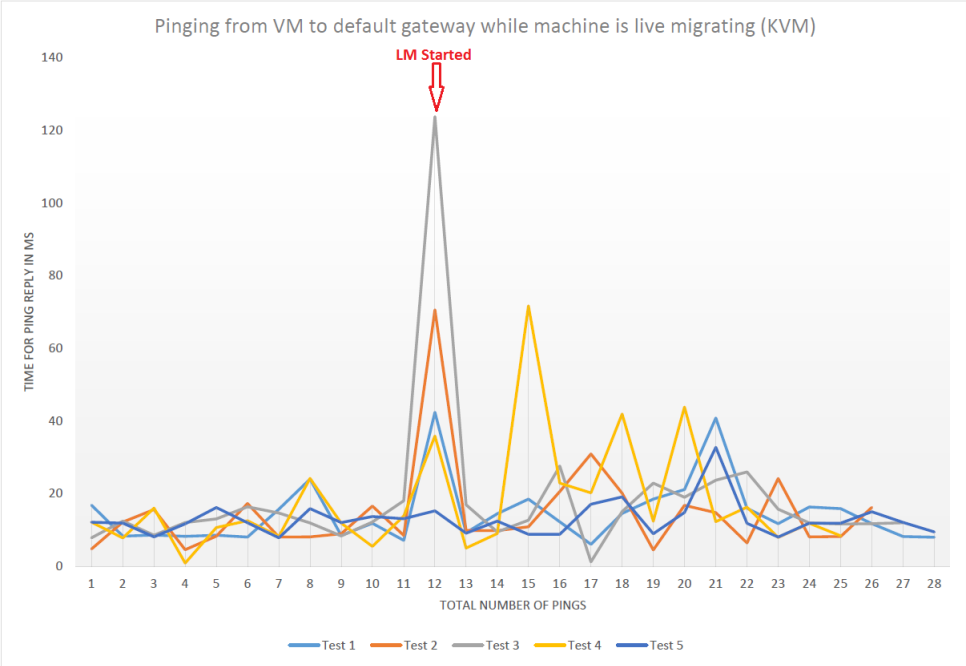

Figure 5.11: Five KVM tests, detecting VM LM from Within VM that is being LM

The second test in KVM attempted to detect the LM of a VM from an external machine 5.12. It can been seen that there is a sharp ping return time when the LM started. During the LM process the ping return times become erratic followed by another sharp increase in return time when LM finishes.

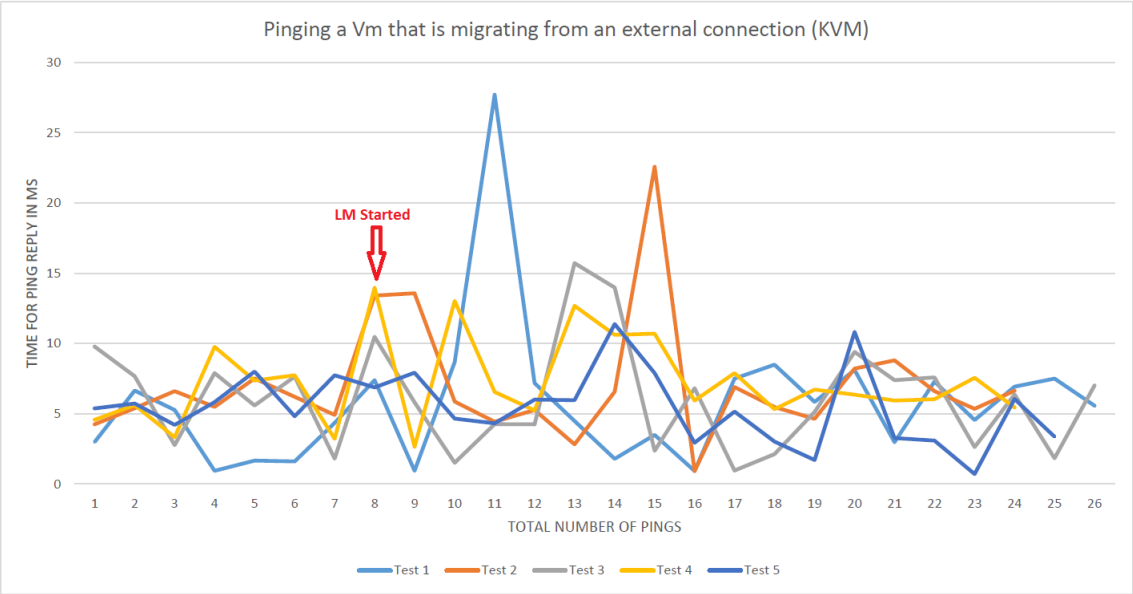

Figure 5.12: Five KVM tests, detecting VM LM from a Remote Machine

Chapter 6

Conclusions

In this dissertation paper our main objective is to find approaches for secure LM. We focused on creating two environments, KVM and XEN that would allow for different strategies for securing LM. Within this dissertation there were four main approaches that were researched. Each of these approaches will now be discussed.

6.1 Detecting LM

An attacker must first detect LM before implementing an LM attack. Attackers can detect an LM of a remote VM by sending ping requests to that VM and monitoring the return time. A sharp rise in return time can indicate that an LM is about to take place. Owners of the VMs should enable a firewall on their VM which will not allow ICMP packets. This will ensure that an attacker will not be able to use ping requests to detect LM.

Owners of VMs, who have rented a VM from a large provider, may not know when their VM is being migrated. To be able to tell if their VM is going to be migrated, it is recommended that they ping the default gateway that exists on their network. The return time taken for the ping request should be monitored. If there is a large rise in the time taken for the return time, it may indicate that an LM is about to take place. The user can then take precautions to defend their machine.

6.2 Encryption of the Data Plane

The findings of this dissertation concluded that data during migration should be encrypted. While this will cause LM to take longer, the user of the VM should notice little difference in interruption from when it is being migrated without encryption as the virtual machine is still running. The configuring of encryption for the migration channel is a reasonable task that IT administrators should have the technical knowledge to implement, however it is surprising that it is not packaged with XEN server and KVM as default. This is leaving less knowledgeable IT administrators vulnerable. The fact that these systems are being produced without this reflects what was seen in the literature that the most LM research is focusing on performance and not security. However, we accept that our research is limited in, that our LM test of two hosts is a small environment and that organisations may have many hosts, meaning that implementing encryption may not be feasible.

In addition to the aims of this study we found that KVM performs LM quicker than XEN. From this we recommend, that if an user wishes to encrypt their migrations but is also worried about the time LM will take, they should implement KVM as their hypervisor. The KVM encrypted LM time did take significantly longer than that the un-encrypted KVM time, but it is also took significant less time that the XEN encrypted migration time.

6.3 Intrusion Detection System

A DOS attack on a host has the capacity to deny access for all VMs that exist on this host. To defend a host against a high network load based attack, a script was implemented. This script checks for high network load. It is recommended that this script be set to what is considered a high network load for an environment. An administrator email should also be added to the script to ensure that he/she receive emails if there is a high network load. This script should then be added to the systems scheduled jobs to run every minute, thus checking network load every minute.

6.4 RBAC

Implementing RBAC on both XEN and KVM is not a trivial task. Issues were found enabling RBAC on XEN server.It was uncovered that XEN server's configuration only allows RBAC with an external authentication type such as LDAP. It was possible to circumvent this by tricking XEN server into believing that an external authentication type was been used when it was not. A series of commands are implemented to achieve this.Once they have been implemented RBAC is possible and administrators can assign read only permissions to users.

It was possible to activate a read only account on KVM. This however was too restrictive. the account could not access the control plane without the need for a root password, and once this was entered the account then had full control to migrate the VMs, which makes the read only account redundant. We propose future work that will investigate more thoroughly the options for RBAC on KVM.

6.5 Overall Best Production Set-Up Recommendation

The production set-up that is recommended is using XEN server. This recommendation is based on the research that was carried out in the implementation and evaluation. XEN had slower LM speeds than KVM but offered greater security. The recommendation for a XEN server set up is:

Problem	Problem Details	Solution on XEN	Soloution On KVM
An Attacker can detect LM of a VM	During this dissertation it was found a LM can be detected remotely by pinging the VM, a sharp rise in ping return time was observed as LM started	Enable a firewall on the VM and ensure that it does not accept ICMP packets	Enable a firewall on the VM and ensure that it does not accept ICMP packets
User cannot detect LM from within a VM	The VM is most at risk during LM, users who's VM exist on a cloud provider may not know when, or if their VM will be live migrated. This dissertation found that by pinging the default gateway (DG) the user will observe a sharp increase in ping return times when the VM is about to be LM	The user can ping the DG to detect LM. The user can then take preventive measures to protect themselves	User can ping the DG to detect LM. The user can then take preventive measures to protect themselves
LM Data sent in clear text	The VM memory that is sent during LM is sent in clear text If an attacker implements a Man in the Middle attack, he/she can glean sensitive information	If the user is an admin on the system, they should encrypt the channel If the user is a customer of a cloud provider, they should give their VMs non descriptive names as the VM name is easily attainable from the output of a MitM attack	If the user is an admin on the system, they should encrypt the channel If the user is a customer of a cloud provider, they should give their VMs non descriptive names as the VM name is easily attainable from the output of a MitM attack
Denial of service on host	If an attacker can implement a DOS attack on a host. All VMs on that host may be made unavailable	A Bash script was developed that will email a selected user if there is high load on on a network interface	A Bash script was developed that will email a selected user if there is high load on on a network interface
Implementing RBAC	Full administrator privileges are give too great control, users should have privileges based on their tasks for example a user should be able to monitor VMs with read only access	Implemented RBAC, based on options that were available for external authentication, edited settings in order to use external RBAC on local accounts	RBAC too restrictive. Read only account could not access the control plane rendering the account useless

Table 6.1: Problems Found and Solutions Offered

6.6 Recommended Configuration for Production Environment

In a two node cluster using shared storage, it is recommended that XEN server be used over KVM. XEN allows for RBAC than KVM. The following

configuration on XEN Server is recommended. Once RBAC is enabled on the configuration users are given permissions based on the tasks they need to do. A separation of duties will help secure the VMs that reside on the environment. In addition to RBAC the data plane which the VM transverse should be encrypted. On each host in the cluster the IDS script should be added. The rate of traffic on an interface that will set of an alarm should be set to meet the needs of the environment. Once this is done, the script should be added to the systems CRON file and set to run every minute. This will ensure that the script will check the load that is on the network and alert if there is a high network load. In addition to security on the hosts, security on the VMS should also be considered. All VMs should have a firewall, and it should be configured to reject ICMP ping requests. This will stop an attacker from gaining information on when the VM is about to LM.

Bibliography

Naveed Ahmad, Ayesha Kanwal, and Muhammad Awais Shibli. Survey on secure live virtual machine (VM) migration in cloud. *Conference Proceedings - 2013 2nd National Conference on Information Assurance, NCIA 2013*, (Vm):101–106, 2013. doi: 10.1109/NCIA.2013.6725332.

Mahdi Aiash, Glenford Mapp, and Orhan Gemikonakli. Secure live virtual machines migration: Issues and solutions. *Proceedings - 2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops, IEEE WAINA 2014*, pages 160–165, 2014. doi: 10.1109/WAINA.2014.35.

Michael Armbrust, Ion Stoica, Matei Zaharia, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, and Ariel Rabkin. A view of cloud computing. *Communications of the ACM*, 53(4):50, 2010. ISSN 00010782. doi: 10.1145/1721654.1721672.

Sebastian Biedermann, Martin Zittel, and Stefan Katzenbeisser. Improving security of virtual machines during live migrations. *2013 11th Annual Conference on Privacy, Security and Trust, PST 2013*, pages 352–357, 2013. doi: 10.1109/PST.2013.6596088.

Irena Bojanova, Jia Zhang, and Jeffrey Voas. Cloud computing. *IT Professional*, 15(2): 12–14, 2013. ISSN 15209202. doi: 10.1109/MITP.2013.26.

Rajkumar Buyya, Rajkumar Buyya, Chee Shin Yeo, Chee Shin Yeo, Srikumar Venugopal, Srikumar Venugopal, James Broberg, James Broberg, Ivona Brandic, and Ivona Brandic. Cloud computing and emerging IT platforms:Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):17, 2009. ISSN 0167-739. doi: 10.1016/j.future.2008.12.001. URL http://portal.acm.org/citation.cfm?id=1528937.1529211.

Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski. *Cloud Computing Principles and Paradigms*. Wiley Publishing, 2011. ISBN 9780470887998.

Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. Improving virtual machine migration in federated cloud environments. *Proceedings - 2nd International Conference on Evolving Internet, Internet 2010, 1st International Conference on Access Networks, Services and Technologies, Access 2010*, pages 61–67, 2010. ISSN 2156-7190. doi: 10.1109/INTERNET.2010.20.

Deyan Chen and Hong Zhao. Data Security and Privacy Protection Issues in Cloud Computing. *2012 International Conference on Computer Science and Electronics Engineering*, 1(973):647–651, 2012. ISSN 1662-8985. doi: 10.1109/ICCSEE.2012.193. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6187862.

C Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live Migration of Virtual Machines. *Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI)*, (Vmm): 273–286, 2005.

Christopher Clark, Keir Fraser, Steven Hand, J.G. Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. *Usenix.Org*, (Vmm):14–26. URL http://www.usenix.org/events/nsdi05/tech/full_papers/clark/clark_html.

Patrick Colp, Mihir Nanavati, Jun Zhu, William Aiello, George Coker, Tim Deegan, Peter Loscocco, and Andrew Warfield. Breaking up is hard to do: Security and functionality in a commodity hypervisor. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 189–202, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0977-6. doi: 10.1145/2043556.2043575. URL http://doi.acm.org/10.1145/2043556.2043575.

Jeff Daniels. Server virtualization architecture and implementation. *Crossroads*, 16(1): 8–12, September 2009. ISSN 1528-4972. doi: 10.1145/1618588.1618592. URL http://doi.acm.org/10.1145/1618588.1618592.

T Devi and R Gane San. Data security frameworks in cloud. In *Science Engineering and Management Research (ICSEMR), 2014 International Conference on*, pages 1–6. IEEE, 2014.

Angela K Dinh. Cloud computing 101. *Journal of AHIMA / American Health Information Management Association*, 82(4):36–37; quiz 44, 2011. ISSN 10605487.

Sarah Edwards, Xuan Liu, and Niky Riga. Creating Repeatable Computer Science and Networking Experiments on Shared, Public Testbeds. *ACM SIGOPS Operating Systems Review*, 49(1):90–99, 2015. ISSN 01635980. doi: 10.1145/2723872.2723884. URL http://dl.acm.org/citation.cfm?doid=2723872.2723884.

Johan Fornaeus. Device hypervisors. In *Proceedings of the 47th Design Automation Conference*, DAC '10, pages 114–119, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0002-5. doi: 10.1145/1837274.1837305. URL http://doi.acm.org/10.1145/1837274.1837305.

Tal Garfinkel and Mendel Rosenblum. When Virtual is Harder Than Real: Security Challenges in Virtual Machine Based Computing Environments. *Proceedings of the 10th Conference on Hot Topics in Operating Systems - Volume 10*, pages 20–25, 2005. URL http://dl.acm.org/citation.cfm?id=1251123.1251143.

Wenjin Hu, Andrew Hicks, Long Zhang, Eli M. Dow, Vinay Soni, Hao Jiang, Ronny Bull, and Jeanna N. Matthews. A quantitative study of virtual machine live migration. *Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference on - CAC*

'13, page 1, 2013. doi: 10.1145/2494621.2494622. URL http://dl.acm.org/citation.cfm?doid=2494621.2494622.

Hai Jin, Deng Li, Song Wu, Xuanhua Shi, and Xiaodong Pan. Live virtual machine migration with adaptive memory compression. *Proceedings - IEEE International Conference on Cluster Computing, ICCC*, 2009. ISSN 15525244. doi: 10.1109/CLUSTR.2009.5289170.

André König and Ralf Steinmetz. Detecting Migration of Virtual Machines. *... of the 10th Würzburg Workshop on IP: ...*, 2011. URL http://www.euroview2011.com/fileadmin/content/euroview2011/abstracts/abstract_koenig.pdf.

Yosuke Kuno, Kenichi Nii, and Saneyasu Yamaguchi. A study on performance of processes in migrating virtual machines. *Proceedings - 2011 10th International Symposium on Autonomous Decentralized Systems, ISADS 2011*, pages 567–572, 2011. doi: 10.1109/ISADS.2011.79.

Virendra Singh Kushwah and Aradhana Saxena. A Security approach for Data Migration in Cloud Computing. 3(5):1–8, 2013.

Lucas McDaniel and Kara Nance. Identifying weaknesses in vm/hypervisor interfaces. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 5089–5095, Jan 2013a. doi: 10.1109/HICSS.2013.255.

Lucas McDaniel and Kara Nance. Identifying weaknesses in vm/hypervisor interfaces. In *System Sciences (HICSS), 2013 46th Hawaii International Conference on*, pages 5089–5095, Jan 2013b. doi: 10.1109/HICSS.2013.255.

Peter Mell and Timothy Grance. The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology, Information Technology Laboratory*, 145:7, 2011. ISSN 1472-0213. doi: 10.1136/emj.2010.096966. URL http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

Jon Oberheide, Evan Cooke, and Farnam Jahanian. Empirical exploitation of live virtual machine migration. *Proc. of BlackHat DC ...*, 2008. URL http://63.236.103.240/presentations/bh-dc-08/Oberheide/Whitepaper/bh-dc-08-oberheide-WP.pdf.

Foundation Owasp. OWASP Testing Guide v3.0. *OWASP Foundation*, page 349, 2008.

Michael Pearce, Sherali Zeadally, and Ray Hunt. Virtualization: Issues, security threats, and solutions. *ACM Comput. Surv.*, 45(2):17:1–17:39, March 2013. ISSN 0360-0300. doi: 10.1145/2431211.2431216. URL http://doi.acm.org/10.1145/2431211.2431216.

D Perez-Botero. A Brief Tutorial on Live Virtual Machine Migration From a Security Perspective. *Cs.Princeton.Edu*, (Vmm), 2011. URL http://www.cs.princeton.edu/~diegop/data/580_midterm_project.pdf.

Diego Perez-Botero, Jakub Szefer, and Ruby B. Lee. Characterizing hypervisor vulnerabilities in cloud computing servers. In *Proceedings of the 2013 International Workshop on Security in Cloud Computing*, Cloud Computing '13, pages 3–10, New York, NY,

USA, 2013. ACM. ISBN 978-1-4503-2067-2. doi: 10.1145/2484402.2484406. URL http://doi.acm.org/10.1145/2484402.2484406.

Gerald J. Popek and Robert P. Goldberg. Formal requirements for virtualizable third generation architectures. *Commun. ACM*, 17(7):412–421, July 1974. ISSN 0001-0782. doi: 10.1145/361011.361073. URL http://doi.acm.org/10.1145/361011.361073.

Rashmi Rao and Pawan Prakash. Improving security for data migration in cloud computing using \nrandomized encryption technique \n. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 11(6):39–42, 2013. URL http://www.iosrjournals.org/iosr-jce/papers/Vol11-issue6/F01163942.pdf.

Farzad Sabahi. Cloud computing security threats and responses. *2011 IEEE 3rd International Conference on Communication Software and Networks*, pages 245–249, 2011. doi: 10.1109/ICCSN.2011.6014715.

Qingni Shen, Lizhe Zhang, Xin Yang, Yahui Yang, Zhonghai Wu, and Ying Zhang. SecDM: Securing Data Migration between cloud storage systems. *Proceedings - IEEE 9th International Conference on Dependable, Autonomic and Secure Computing, DASC 2011*, pages 636–641, 2011. doi: 10.1109/DASC.2011.114.

Jyoti Shetty. A Framework for Secure Live Migration of Virtual Machines. pages 243–248, 2013.

Jyoti Shetty, Anala M R, and Shobha G. A Survey on Techniques of Secure Live Migration of Virtual Machine. *International Journal of Computer Applications*, 39(12):34–39, 2012. ISSN 09758887. doi: 10.5120/4875-7305.

S. Subashini and V. Kavitha. A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1):1–11, 2011. ISSN 10848045. doi: 10.1016/j.jnca.2010.07.006. URL http://dx.doi.org/10.1016/j.jnca.2010.07.006.

Ankit Upadhyay and Prashant Lakkadwala. Secure Live Migration of VM s in Cloud Computing :. pages 0–3, 2014.

Wei Wang, Ya Zhang, Ben Lin, Xiaoxin Wu, and Kai Miao. Secured and reliable VM migration in personal cloud. *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, 1:705–709, 2010. doi: 10.1109/ICCET.2010.5485376.

Chen Xianqin, Wan Han, Wang Sumei, and Long Xiang. Seamless virtual machine live migration on network security enhanced hypervisor. *Proceedings of 2009 2nd IEEE International Conference on Broadband Network and Multimedia Technology, IEEE ICBNMT2009*, pages 847–853, 2009. doi: 10.1109/ICBNMT.2009.5347800.

L. YamunaDevi, P. Aruna, D. Sudha Devi, and N. Priya. Security in virtual machine live migration for KVM. *Proceedings of 2011 International Conference on Process Automation, Control and Computing, PACC 2011*, 2011. doi: 10.1109/PACC.2011.5979008.

Fengzhe Zhang, Yijian Huang, Huihong Wang, Haibo Chen, and Binyu Zang. PALM: Security preserving VM live migration for systems with VMM-enforced protection. *Proceedings - 3rd Asia-Pacific Trusted Infrastructure Technologies Conference, APTC 2008*, pages 9–18, 2008. doi: 10.1109/APTC.2008.15.

Appendix A

Encrypting the Data Plane

```
1 yum install ipsec-tools
```

Listing A.1: Command to Install IP Security Tools

This package installed racoon. We added a configuration file, setkey.conf, to the racoon directory, /etc/raccoon/setkey.conf, on each XEN host.

We added the below configuration the the setkey.conf file on 192.168.142.142

```
1 #!/usr/sbin/setkey -f
2 # Flush the SAD and SPD
3 flush;
4 spdflush;
5 spdadd 192.168.142.142 192.168.142.143 any -P out ipsec
6        esp/transport//require;
7 spdadd 192.168.142.143 192.168.142.142 any -P in ipsec
8        esp/transport//require; ]
```

Listing A.2: The Configuration of setkey.conf on 192.168.142.142

We then went to the 192.168.142.143 machine and reversed the IP addresses, thus, instructed both hosts to secure all incoming and outgoing traffic to and from the other XEN hosts to be encrypted.

```
1 #!/usr/sbin/setkey -f
2 # Flush the SAD and SPD
3 flush;
4 spdflush;
5 spdadd 192.168.142.143 192.168.142.142 any -P out ipsec
6        esp/transport//require;
7 spdadd 192.168.142.142 192.168.142.143 any -P in ipsec
8        esp/transport//require; ]
```

Listing A.3: The Configuration of setkey.conf on 192.168.142.143

Next we added the following configuration to our racoon.conf file on the XEN host 192.168.142.142

```
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
sainfo anonymous {
{
        pfs_group 2;
        lifetime time 1 hour;
        encryption_algorithm 3des, blowfish 448, rijndael;
        authentication_algorithm hmac_sha1, hmac_md5;
        compression_algorithm deflate;
}

remote 192.168.142.143
{
        exchange_mode aggressive, main;
        my_identifier address;
        proposal {
                encryption_algorithm 3des;
                hash_algorithm sha1;
                authentication_method pre_shared_key;
                dh_group 2;
        }
}
```

Listing A.4: The Configuration of racoon.conf on 192.168.142.142

We then made the same configration on the 192.168.142.143 host, and chnaged the remote setting, inputing 192.168.142.142.

```
path pre_shared_key "/etc/racoon/psk.txt";
path certificate "/etc/racoon/certs";
sainfo anonymous {
{
        pfs_group 2;
        lifetime time 1 hour;
        encryption_algorithm 3des, blowfish 448, rijndael;
        authentication_algorithm hmac_sha1, hmac_md5;
        compression_algorithm deflate;
}

remote 192.168.142.142
{
```

```
14        exchange_mode aggressive, main;
15        my_identifier address;
16        proposal {
17              encryption_algorithm 3des;
18              hash_algorithm sha1;
19              authentication_method pre_shared_key;
20              dh_group 2;
21        }
22  }
23
24  \end{lstlising}
25
26  These settings direct the racoon service to the opposite server to which ↩
        it wishes to connect with. The file included the follwoing settings ↩
        to secure the data plane
27
28
29  \begin{itemize}
30  \item tripletriple des algorithm
31  \item SHA1
32  \item pre_shared key
33  \end{itemize}
```

Listing A.5: The Configuration of racoon.conf on 192.168.142.143

We have included a detailed set up of installation and set up of IPSEC/Racoon in the appendices Once we had racoon fully installed we entered the pre-shared key into the file /etc/racoon/psk.txt2 on each server. Next we started the racoon service on each server by running the following command on each.

```
1  setkey -f /etc/setkey.conf
2  racoon -F
```

Listing A.6: Starting the Racoon Service

Appendix B

MitM

To facilitate a man in the middle attack we create a Kali Linux Virtual Machine with the following specification.

* 2GB of Ram
* 20GB harddrive
* 2 processors
* NAT network

The machine was placed on the same network as both the KVM and XEN host. From here we implemented a Man in the middle attack using ARP spoofing. The goal of this is to trick both XEN servers to send there information to the Kali Linux server, thinking that they are sending traffic to one another. We can then sniff the traffic and pass it on to it's destination, meaning that both hosts are unaware that the attack is happening. The first key step in doing this is to implement packet forwarding to ensure that the information is passed on to it's correct destitution. This is achieved by running the following command.

```
1 echo 1 > /proc/sys/net/ipv4/ip_forward
```

Listing B.1: Enabling Packet Forwarding

Now that the packet forwarding is enabled we have to Next We next turned on ARP Spoofing by opening a terminal and issuing the following command

```
1 arpspoof -i eth0 -t 192.168.142.142 192.168.142.143
```

Listing B.2: Enabling ARP Spoofing Between 192.168.142.142 and 192.168.142.143

We then ensured that we would capture traffic that was been sent in the opposite direction

```
1 arpspoof -i eth0 -t 192.168.142.142 192.168.142.143
```

Listing B.3: Enabling ARP Spoofing Between 192.168.142.143 and 192.168.142.142