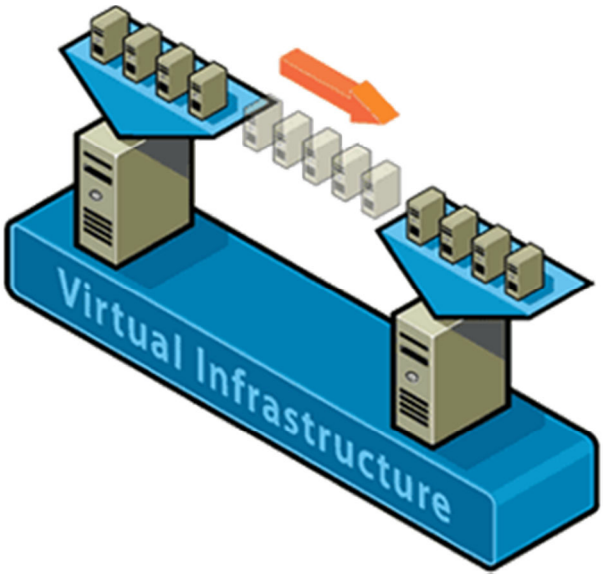
2011

Dynamic load balancing based on live migration of virtual machines: Security threats and effects

Melvin Ver



Dynamic Load Balancing based on Live Migration of Virtual Machines: Security Threats and Effects

By

Melvin Ver

This thesis report is a partial fulfillment of the requirements for the degree of Masters in Networking and System Administration

Supervised by

Chair	**: Prof. Charles Border**
Reader	**: Prof. Luther Troell**
Observer	**: Prof. Bo Yuan**

Rochester Institute of Technology
B. Thomas Golisano College of Computing and Information Sciences (GCCIS)
Rochester, NY, U.S.A.
January 2011

THESIS REPORT RELEASE PERMISSION FORM

Rochester Institute of Technology
B. Thomas Golisano College of Computing and Information Sciences

Title: Dynamic Load Balancing based on Live Migration of Virtual Machines: Security Threats and Effects

I, Melvin Ver, hereby grant permission to the Wallace Memorial Library reproduce my thesis in whole or part.

_____
Melvin Ver

_____
Date

The thesis "Dynamic Load Balancing based on Live Migration of Virtual Machines: Security Threats and Effects" by Melvin Ver has been examined and approved by the following Examination Committee:

_____
Prof. Charles Border

_____
Prof. Luther Troell

_____
Prof. Bo Yuan

Acknowledgements

First and foremost, I would like to thank my supervisor Assoc. Professor Charlie Border for his support throughout this thesis. His feedback has been of great help and is highly appreciated. I extend my gratitude towards Prof. Luther Troell and Assoc. Professor Bo Yuan, for accepting my proposal to work on this topic and being a part of my Thesis Advisory committee. Thank you, to all the people involved in this endeavor, for the countless valuable discussions, helpful tips and enthusiasm.

I would also like to thank my friends and my family for the constant support in this time and for showing interest in my work and activities at all times.

Melvin Ver

Table of Contents

List of Figures & Graphs

Chapter 1

Introduction

1. Background

History of Virtualization – When virtualization was first conceived in 1960s, it was known to programmers and researchers as time sharing. It was Christopher Strachey, Professor of Computation at Oxford University who coined the term in his paper "*Time Sharing in Large Fast Computers*" wherein he was referring to what he called multi-programming. According to this technique, while one programmer is developing a program on his console and another programmer debugging his, there would be no usual wait for peripherals. Multi-programming and similar ideas began to drive innovation which has resulted in several computers that have been brought to existence like the Atlas and IBM's M44/44X.

Atlas computer was one of the first supercomputers of the early 1960s that used concepts such as time sharing, multi-programming, as well as shared peripheral control. Atlas was one of the fastest computers of its time partially due to a separation of OS processes from the executing user programs. The component called the supervisor managed the computer's processing time, and was passed extracodes, thus helping in the management of the user program's instructions. This was considered as the birth of the hypervisor or virtual machine monitor (VMM).

IBM spearheaded the M44/44X Project at the IBM Thomas J. Watson Research Center. The architecture which was similar to that of Atlas computer led to coining of the term *virtual machines*. The IBM 7044 (M44) scientific computer was supported by several simulated 7044 virtual machines, using hardware and software, virtual memory, and multi-programming.

What is Virtualization? - Virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments. In other words, virtualization is a framework or methodology of dividing the resources of a computer into multiple execution environments, by applying one or more concepts or technologies such as hardware and software partitioning, time-sharing, partial or complete machine simulation, emulation, quality of service, and many others.

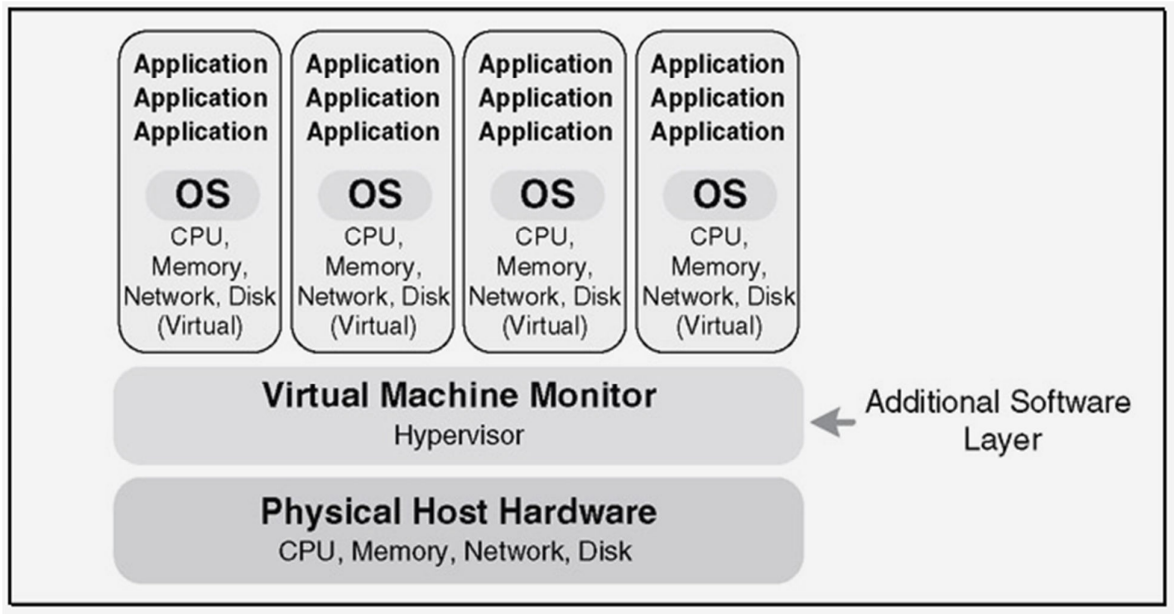


Figure 1. Simple representation of virtualize system

Why is Virtualization important? - Live migration has many advantages. It gives the user flexibility and options to take down a working server during the day, rather than at night or on weekends, upgrade the operating system, apply patches, etc., then bring it back up again during regular working hours. This is a very useful concept, for instance, operations managers in data centers look at where they have heavy workloads and move virtual machines around so that the cooling system isn't working excessively hard trying to keep just part of the data center at the right temperature.

Following are some representative reasons for and benefits of virtualization:

- Virtual machines can be used to consolidate the workloads of several underutilized servers to fewer machines, perhaps a single machine (server consolidation).
- Related benefits are savings on hardware, environmental costs, management, and administration of the server infrastructure.
- The need to run legacy applications is served well by virtual machines.
- Virtual machines can be used to provide secure, isolated sandboxes for running non-trusted applications. Virtualization is an important concept in building secure computing platforms.

- Virtual machines can be used to create operating systems, or execution environments with resource limits, and given the right schedulers, resource guarantees.
- Virtual machines can provide the illusion of hardware, or hardware configuration that you do not have (such as SCSI devices, multiple processors, etc)
- Virtual machines can be used to run multiple operating systems simultaneously: different versions, or even entirely different systems, which can be on hot standby.
- Virtual machines allow for powerful debugging and performance monitoring.
- Virtual machines can isolate what they run, so they provide fault and error containment. Virtual machines make software easier to migrate, thus aiding application and system mobility.
- Virtual machines are great tools for research and academic experiments.
- Virtualization can enable existing operating systems to run on shared memory multiprocessors.
- Virtual machines can be used to create arbitrary test scenarios, and can lead to some very imaginative, effective quality assurance.
- Virtualization can make tasks such as system migration, backup, and recovery easier and more manageable.
- Virtualization can be an effective means of providing binary compatibility.
- Virtualization is fun.

Types of Virtual Machines:

1. Process VMs
2. System VMs

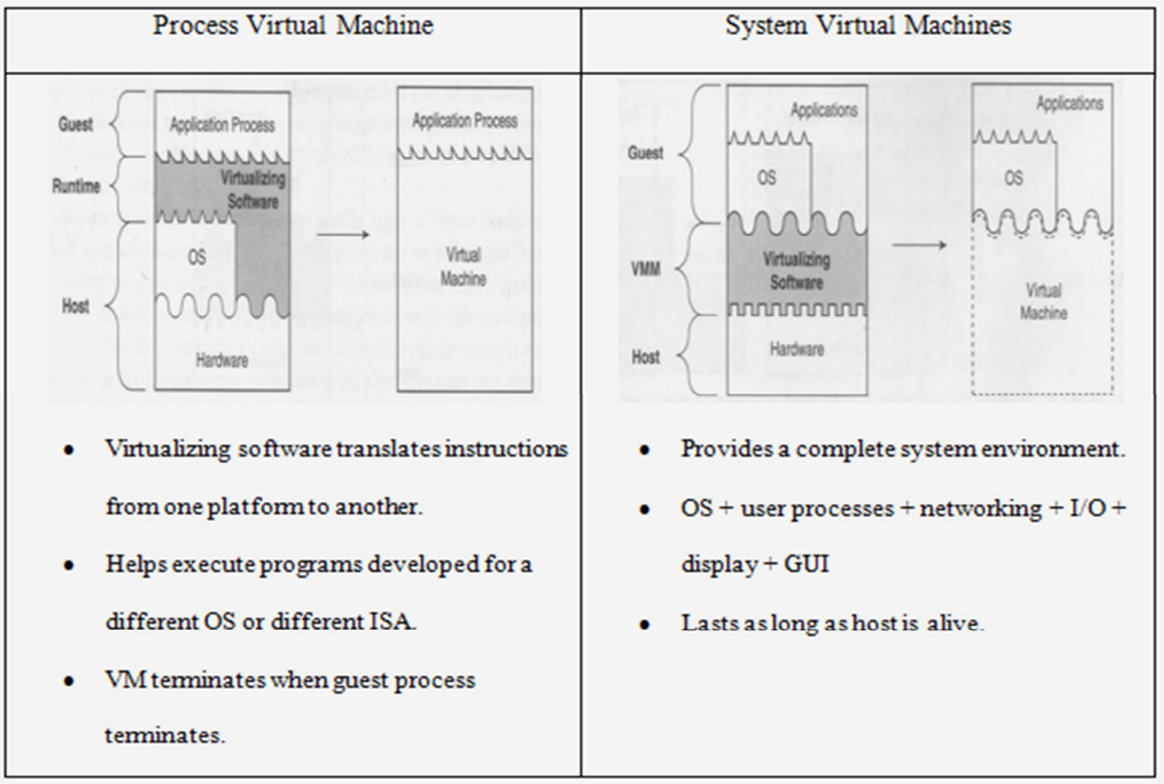


Figure 2. Types of Virtual Machines

Virtual Machine Applications:

1. Emulation – This allows mix and match cross-platform portability.
2. Optimization – This provides platform-specific performance improvement. It is usually done with emulation.
3. Replication – This allows having multiple virtual machines on a single platform.
4. Composition – Similar to replication but forms more complex but flexible systems.

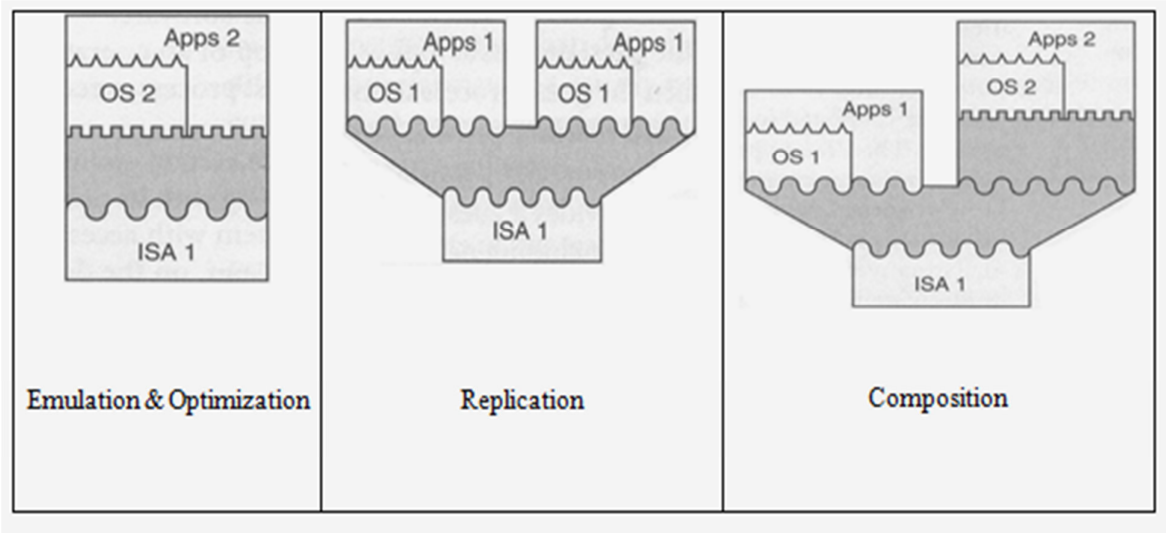


Figure 3. Virtual Machine Applications

Performance Goals in Migration:

- Migration of Virtual Machines helps minimize 'Downtime'.
- One of the major performance goals in migration is reduction of total migration time of virtual machines.
- It helps if there is no interference with normal system activity which is an important goal of migration.
- To minimize network activity is also a performance goal in migration.

Business Challenge:

Initially virtualization was embraced as a method to consolidate multiple server applications onto fewer servers so as to increase utilization and reduce energy requirements. Today, IT organizations are looking beyond just server consolidation. Today virtualization is being used to create more dynamic data centers.

Live migration dynamically allocates and balances computing resources and thus helps us align the infrastructure with enterprise-level business goals. Live migration performs migrations without actually interrupting the services the Virtual Machines are providing and thereby reducing the additional work usually involved with migrating Virtual Machines. Traditionally, the extra work involves first shutting down the applications, then moving the Virtual Machines to new servers, and finally restarting the Virtual

Machines and the applications that were shutdown. So by eliminating this additional work, live migration helps improve flexibility and efficiency in managing data center resources.

Advanced data centers are benefitted by live migration by use of following capabilities:

1. Dynamic Load balancing
2. No VM downtime during maintenance

Dynamic Load Balancing:

Dynamic load balancing will change the mapping of jobs to resources at any point, but there may be a cost associated with the changes. There are two main types of dynamic load balancing algorithms: preemptive and non-preemptive. In a preemptive algorithm, the mapping of jobs to physical resources may change while a job is running whereas with the non-preemptive algorithms, once a job has started the physical resources assigned to that job cannot change. Dynamic load balancing redistributes tasks during execution which typically moves tasks from heavily loaded processors to lightly loaded processors. Also the dynamic load balancing algorithm can be centralized or distributed. In centralized scheme, all information is sent to a single decision-making agent that decides when load balancing will occur and which tasks will be moved. In distributed dynamic load balancing, load information is not shared globally and all processors take part in deciding when load-balancing occurs and which tasks will be migrated. One major advantage of dynamic load balancing is that the run-time behavior of the system does not need to be known in advance. While these algorithms are very good at maximizing utilization of all resources, there is a run time overhead associated with gathering information and transferring tasks to different processors.

No VM downtime during maintenance:

When enterprise-level servers need maintenance and if they are hardware-based physical servers, they need to be shutdown and hence due to this downtime there may be loss of business and/or inconvenience to users and clients. But by using live migration, enterprises can perform server maintenance like software upgrades without experiencing VM downtime, as the VMs can be moved from the primary host server to secondary server before shutdown or reboot.

2. Motivation

Live migration of virtual machines (VMs) is the process of transitioning a VM from one virtual machine monitor (VMM) to another without halting the guest operating system, often between distinct physical machines, has opened new opportunities in computing. It allows a clean separation between hardware and software, and facilitates fault management, load balancing, and low-level system maintenance. Implemented by several existing virtualization products, live migration also aids in aspects such as high availability services, transparent mobility and consolidated management.

While virtualization and live migration enable important new functionality, the combination introduces novel security challenges. A virtual machine monitor that incorporates a vulnerable implementation of live migration functionality may expose both the guest and host operating system to attack and result in a compromise of integrity.

Given the large and increasing market for virtualization technology, a comprehensive understanding of virtual machine migration security is essential. So the motivation for my setup is to create a test environment that is suitable for this experiment and analyze the security implications in case of exploitation of Live Migration of Virtual Machines.

3. Objective

Applying published dynamic load balancing algorithms to virtual machines and the experimental study of security implications by exploiting Live Migration of Virtual Machines, is the central idea of my thesis.

I am primarily interested in answering these following questions:

- "What is the overhead of load balancing using specified policy engine?"
- "What happens when Live Migration of Virtual Machines is exploited? "
- "What is the effect of these security implications on the various Performance factors on which Live Migration depends?"
- "How does exploiting the security affect migration-enabled dynamic load balancing?"
- "Can there be effective Resource Isolation in case of security lapse?"
- "What exactly conspires when the process of Live Migration is exploited?"

4. Scope

The scope of my thesis research is essentially to use Live VM migration for dynamic load balancing or scheduling. This experimentation will determine workload hotspots in physical environment and through use of effective Live Migration process; it will try to carry out resource profiling. By carrying out effective profiling, this thesis research will be able to determine how much of each resource needs to be allocated to a VM. To understand exactly why process migration would not work in such scenarios and better understand Live VM Migration, this thesis will try to provide requisite incites as to which model is most appropriate for automatic load balancing for virtual machine infrastructure based on resource consumption. The security implications of exploiting the process of migration may end in unexpected results or results that are not noticeable. The scope of my thesis research is identifying these results and the causes for them.

Chapter 2

Literature Review

Virtual Machine Monitors were developed to better utilize the expensive mainframe hardware and its resources so that multiple applications and servers could coexist on the same physical host. The motivation for research in this field today is increasing dramatically as more hardware is developed with built in native virtualization support and more software solutions appear.

Reliability and redundancy are key features in modern virtualization technology. Mission critical applications do not tolerate downtime and corporations can quickly lose money if their availability is affected by maintenance, hardware- or software failure or malicious activity. Virtual machine technology deals with this challenge with just simply moving the virtual machine(s) from the physical host that needs maintenance onto other server hardware while the virtual machine is running and maintaining service availability. With the virtual machines running elsewhere there is no harm in doing a server shutdown in order to do maintenance quick and undisturbed.

A virtual infrastructure can utilize live server migration to move running production servers to other network hosts in order to do, for instance scheduled maintenance or replace faulty hardware. This makes the network setup very fault-tolerant and eliminates any maintenance or management downtime. There are several ways of presenting a virtual hardware layer, and the many Virtual Machine Monitors solve the problems in different ways. As stated by Christopher Clark, et al, migrating an entire OS and all of its applications as one unit allows us to avoid many of the difficulties faced by process-level migration approaches. In particular the narrow interface between a virtualized OS and the virtual machine monitor (VMM) makes it easy avoid the problem of ‘residual dependencies’. Live migration of virtual machines allows a separation of concerns between the users and operator of a data center or cluster. Christopher Clark, et al, have addressed the importance of using live OS migration [1] as a powerful tool for cluster administrators, allowing separation of hardware and software considerations, and consolidating clustered hardware into a single coherent management domain. The idea of migrating OS instances including the applications to alternative

machine(s), freeing the original machine for maintenance, is central to their implementation of high-performance migration support for XEN. Their implementation is very significant and suggests an efficient way to enable rapid movement of interactive workloads within clusters and data centers. It is also important to know that resources do also migrate which may depend on resource to process binding – by identifiers like specific website, ftp server; or by type like printers, local devices. Resources can also migrate depending on type of attachments – fixed resources like local devices, communication end points; or resources moved only at high costs like databases. This current research has helped understand the implementation by Clark, et al by furthering the research with support of Resource Allocation but using VMWare technology which is vMotion.

The practice of dedicating one server to each service or application is costly and precious IT-resources are stretched thin procuring, provisioning and maintaining a growing number of under-utilized servers. Business continuity requires continuous server uptime and meeting this demand is costly in terms of redundancy, management and maintenance. The solution could be a virtual infrastructure. In the research paper by Wesley Emeneker, et al; they implement a system of virtual machines [2] to increase cluster utilization by enabling job forwarding and spanning, that flexibly allow software environment changes, and effectively sandbox users and processes from each other and the system. The authors here have created an initial implementation of DVC model that has reduced the queue wait and turnaround time, and increased job throughput. They have implemented a model that performs forwarding and spanning with VMs which can improve cluster utilization, increase throughput, and decrease turnaround time. Similar to the DVC model, a model that also improves cluster utilization by reducing downtime time through implementation of dynamic cluster reservation supported by live migration of VMs would facilitate efficient vacation of resources using scheduling algorithm (load balancing algorithm) so as to fully utilize resources. This further research which was based on the idea of utilizing one such published scheduling algorithm provided by VMWare and as known in virtual world as Distributed Resource Scheduler (DRS) that automatically detects workload hotspots and schedules resource allocation across VMs and migrates them to underutilized servers was successful in determining when workloads occur and when and where the virtual machines are migrated.

It is important to detect workload hotspots which arise if the demand exceeds allocated capacity. An efficient way to overcome this is to build a framework for dynamic allocation of resources that can be adjusted based on workload. This framework must be secure and not allowed to be compromised. Virtualization which is one of the hottest concepts these days can be utilized to make sure that resources are used more efficiently as it has the capability to adjust resource provisioning while maintaining isolation. Live VM Migration is a virtualization concept that can be used to create such a model.

Furthermore, as described in the paper by Marvin McNett, et al (2007); they describe the implementation of Usher [3] as a cluster management system. The system is designed to substantially reduce the administrative burden of managing cluster resources while simultaneously improving the ability of users to request, control, and customize their resources and computing environment, enabling administrators to choose how their VM environment will be configured and the policies under which they will be managed. Usher multiplexes individual VMs on available physical machine hardware and allows creating and use of machines according to the needs rather than according to assigned physical resources by decoupling logical machine resources from physical machines. Now to have a similar model that does dynamic allocation of resources and a model that performs automatic computation to determine which physical resources are overloaded and then by allocating resources to relevant VMs, perform VM scheduling to migrate VMs to appropriate underutilized server, would be an approach that administrators would look forward to implementing as a model for their VM migration policy and so that idea was experimented in my research using the concept of DRS.

When we discussing the resource management model, we need to consider which portions of resource slots can be bound either to virtual resources or to overhead associated with the creation or maintenance of those virtual resources. Borja Sotomayor, et al; in their implementation of Virtual Resource Management Model [4] (2007) have implemented a model that manages the overhead of using VMs to minimize the negative effects on performance and leverage virtualization features that could increase utilization of physical resources. They have suggested use of multiple techniques like Draining, Backfilling and Suspend-Resume to evaluate scheduling accuracy and efficiency. While their implementation was focused on deployment of VM images, the current research based on Live Migration of VMs evaluates these factors but by simply

measuring the utilization of resources and allocating them to appropriate VMs. Sotomayor, et al, have tested their model against real as well artificial workloads which will be done for the current study too, not based on user requests for virtual workspace but for performance testing of the physical servers that host these resources. Virtual Resource Management model was more or less based on models conceived earlier like "Virtuoso and VSched" [5] (2005) and "Shirako" [6] (2006). These models are quite similar to the Virtual Resource Management Model wherein the authors created a virtual workspace from scratch but with certain assumptions like no preparation overhead during resource leasing in case of virtual clusters were being deployed on multiple physicals clusters and no deployment overhead while co-scheduling interactive and batch workloads on individuals machines. While these models missed out or deliberately not consider overheads (preparation overhead as in case of Virtuoso and deployment overhead as in case of Shirako), the current research considers overheads due to Live Migration of VMs while resource reservation takes place based on utilization of physical servers like for instance variation of memory load and network overload factor which was studied and researched by carrying out tests that show results that are used during dynamic load balancing.

While physical machine state is protected by hardware, virtual machine state is protected by VMM/hypervisor and software attacks are perpetrated due to weak VMM isolation. Full VM state is exposed to the network during migration as there is no encryption involved. Security concerns range from authentication to confidentiality to isolation. In the paper [8] by Jon Oberheide et al, they have discussed about the different classes of threat to the Live Migration process. They have given a detailed report of their investigation of possible attack classes during the live migration process. A compromised system inside a network employing live migrations can facilitate un-trusted access to migrating Virtual Machines. While they have discussed different attack classes, they have not elaborated on the effects of these attacks or security implications over the migration process and (specifically) the concept of dynamic load balancing facilitated by Live Migration of VMs. This current thesis research identifies the effects of the security lapse on the migration process and specially, the effect on the performance factors of virtual machines. I have used some of the attack classes coupled with VMWare technology and tried to exploit and stress test

vMotion using varying memory values and network load. This research has used MiTM as the major attack class along with host of other attacks like sniffing or gathering data; fingerprinting VMWare products and brute force attack to retrieve hypervisor login information.

Resource utilization in expensive servers is often at 25% average. As mentioned earlier the practice of assigning one application or service to its own physical server can be costly. A virtual infrastructure improves resource utilization and high priced servers can be justified. Administrators can look at their servers as a big pool of resources and map virtual machines with running services to any part of this resource pool without having to think about available resources on a specific host. This in turn also prevents the proliferation of physical servers. Espen Braastad (2006) describes the need for open source tools capable of providing migration of VMs as a method for virtualization and repair. [7] He describes the use of 'Heartbeat' as high availability software that seamlessly migrates VMs between physical nodes in the cluster. This paper evaluates migration of VMs using Heartbeat based on impact on TCP and UDP services and also the users of high availability services. The core discussion presented here is the results related to failures in case of migration and effect on network performance. While network performance is very important for administrators, it would be a significant measurement to test the migration traffic. Evaluating this migration traffic was a part of the current research so as to determine the most appropriate model for resources to be migrated using VMs across the network which was done using VMWare measurement tools provided by Veeam coupled with VMWare's inbuilt virtual infrastructure components. The effect of exploiting the migration traffic can be evaluated and to determine what transpires in the network pipe during the migration process is an integral part of this current thesis research which showed how the total migration time is affected and what is the change in migration delay as compared with varying values of memory and migration delay comparison study between migration of VMs in sequence against parallel.

A literature study by Andre van Cleeff, et al. [19] has details and charts of security impact per feature group that includes hardware, VM, VMM, VMMM, emergent. These feature groups have been weighed against various exploits and vulnerabilities which affect one or more properties like confidentiality, integrity, availability, non-repudiation and authenticity. My research uses some of these features to test and validate

the vulnerabilities like for instance data integrity during vMotion when data which may be financial records or otherwise are seen in cleartext just by sniffing or MiTM attack.

Furthermore, Ming Zhao, et al. [21] in their "Experimental study of virtual Machine migration in support of reservation of Cluster Resources", has provided us with an experimental model which can predict performance of VM migration process. While their experimental model was not based on VMWare vMotion, mine was. I was able see similar results using scenarios from their experimental study but for VMWare vMotion while theirs was based on stop-copy-resume migration strategy.

Administrators who work in virtualized domain [20] have tried and tested exploits that compromise such systems like using MiTM attacks, stealing guest VMs, credential harvester attack, taking control of management servers and console. So I have used these instances and methodologies as described in the world of hacker or for an ethical tester – penetration tester, to understand the logs and performance parameters of a breach. My results show graphical representations of the timeframe during the vMotion breach or virtualization exploit. While the scope of their might be larger as compared to mine, I have tried to encapsulate various factors related to VM migration and the threats related to it.

Exploits related to resources when network is flooded is a case wherein I tried to chalk out a result that tells me how the live migration is affected even if there is no attempt to steal or corrupt data. My research is an attempt to get a behind-the-scenes glimpse of live migration exploit or stress testing.

The above reviewed literature presents different oversights to resource allocation and virtualization concepts along with possible security loopholes. Each of the them have their significant problem statements that the authors have tried to overcome and present a model or infrastructure to create an efficient methodology. The current research idea will be based on some of these concepts and support or prove them in an alternative approach. The current research approach will use some of the ideas from these models and act as a guide for working towards defining a methodology that overcomes the problems encountered by these previous approaches.

Chapter 3

Live Virtual Machine Migration

What is Live Virtual Machine Migration?

To put the answer in most simplest terms would be “transfer of VM/s from one physical machine to another with minimal or no service downtime”.

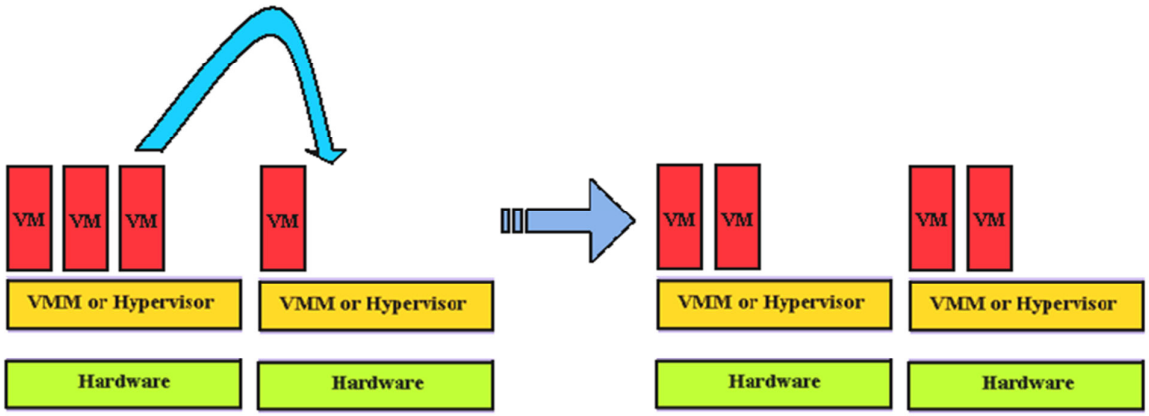


Figure 4. Virtual Machine Migration

Migration Types - There are generally two basic types of migration which may be categorized as pre-copy and post-copy.

Pre-copy

Method
- First transfer all memory pages and then copy pages just modified during the last round iteratively
- Only memory and CPU status needs to be transferred
- VM service downtime is expected to be minimal by iterative copy operations

Problem
- A great amount of transferred data

Post-copy

Method
- Memory transfer phase starts only after the VM’s CPU state has already been transferred to the target and resumed there
- All memory pages are transferred only once during the whole migration
- The baseline total migration time is achieved

Shortage
- Downtime is prolonged due to the latency of fetching pages from the source node before VM can be resumed on the target

Following are some specific migration techniques [9]:

Stop-and-Copy Migration - This type of migration is non-live migration. This type of migration results in VM downtime. This works well for VMs under maintenance. The only positive aspect of this type of migration is that it provides a baseline to compare the total number of pages transferred and the total migration time.

Following are the details regarding stop-and-copy migration:

- The source VM (one that has to be migrated) is stopped (shutdown).
- All the pages are copied over the network.
- Finally the destination VM is started.
- This migration type has longest service downtime.
- This migration type has the shortest migration duration.

Demand Migration – This type of migration is based on post-copy based live VM migration.

Following are the details regarding demand migration:

- Initially all the critical and essential OS structures are copied over the network.
- Then the destination VM is started.
- Every page fault triggers copy of those pages over the network.
- This migration type has shortest service downtime.
- This migration type has the longest migration duration.

Iterative Pre-Copy Migration – This type of migration combines pre-copy migration with a bounded iterative push phase. This migration type also involves a very short stop-and-copy phase.

Following are the details regarding iterative pre-copy migration:

- Iteratively copies pages over the network from source to destination.
- Keeps copying pages until a particular threshold is reached, then stops source VM and further copies all the remaining pages, before finally starting the destination VM.
- This kind of migration equally balances migration duration and service downtime.

<table>
<tr><th></th><th colspan="2">Categories</th><th>Characteristics</th></tr>
<tr><td rowspan="3">Live Migration</td><td rowspan="2">Pre-copy</td><td>Memory</td><td>Transfer memory data in pre-copy phase</td></tr>
<tr><td>Trace</td><td>Transfer checkpoint and execution trace files in pre-copy phase</td></tr>
<tr><td colspan="2">Post-copy</td><td>Memory transfer is deferred after CPU status transfer phase</td></tr>
<tr><td>Non-Live Migration</td><td colspan="2"></td><td>VM is suspended during whole migration process</td></tr>
</table>

Figure 5. Migration Techniques

Chapter 4

Security Risks – Holistic View

Companies which network their computers for cost-saving purposes by using live virtual machine migration are vulnerable to hackers/attackers. Since live virtual machine migration is relatively new, allowing virtual machines to migrate between servers with little or no service downtime and thus help load balance across several servers; the security of live virtual machine migration has not been researched upon extensively.

It is possible that hackers/attackers compromise the integrity of a virtual machine's operating system during live migration because even though virtualization-based companies have created software that makes it easy for reducing server downtime and equalize loads when there is fluctuations across the servers, these software do not encrypt the data (virtual machines in this case) as it migrates from one server to other server.

While a short-term fix would be to isolate network used for migration from other network traffic or have the data encrypted by using some reputed hardware encryption software, there is an urgent need to educate companies and raise awareness about this vulnerability.

To understand how the security of virtual environment can be compromised, it was essential to think like an attacker. So this chapter provides an overview of types of attacks and how they work and gives the point view of a Penetration Tester, hacker or disgruntled employee.

Common Types of Attacks

1. Buffer Overflows

This class of attack involves data being written into a buffer which is greater than the allocated space of that buffer, resulting in corruption of memory inside a running process. The reason behind such an attack could be to either halt that running process or call an injected malicious code as a result of buffer overflow.

2. Heap Overflows

This attack class occurs in heap data area and is a type of buffer overflow. As in case with buffer overflow, attacker supplies the application with data that is larger than the size of the chunk (data blocks in heap)

which causes the overwriting of metadata of the succeeding chunk. Again, the attacker data may contain malicious code.

3. Web-based Attacks
    - Fake Certificate Injection

An attacker acts as a man-in the middle between the website and the computer trying to access that website. By using this type of attack, the attacker, much like a proxy server, after taking requests from an unsuspecting user, makes some (possibly) malicious changes and forwards the changed request to the target. But since the query has been tampered with, the user gets back a new location that the attacker wants him/her to go to.

    - Cross-Site Scripting (XSS)

Using this type of computer security vulnerability, an attacker (malicious web user) can inject malicious code into the web applications. Such vulnerability is usually used by the attacker to bypass access controls, for phishing attacks and browser exploits. Type 1 XSS or non-persistent is a reflected vulnerability which shows up when server-side scripts instantly use the data provided by web clients without validating the received data which can allow insertion of client-side code into dynamic web pages. Type 2 XSS or persistent is a second order attack. This type of vulnerability occurs when the data provided by the user to the web application is stored persistently (in database or file systems), and is displayed on to the web page without being encoded using HTML.

4. SQL Injection

When an input provided to a web application in interpreted as a SQL input and the application replies back with a failure message, the attacker is able to realize that the target application is vulnerable to SQL injection. In this type of attack, the attacker tries to get around the filters by testing out the weaknesses of the filters by querying the filters persistently. So basically, SQL injection takes place when data is inputted to the SQL query engine that is not expected by the web user.

5. Layer 2 Attacks
    - Content Addressable Memory (CAM) Table or MAC Flooding

CAM table is simply Dynamic Content Addressable Memory on an Ethernet switch and is responsible for correctly echoing out the frames out of relevant port or else the switch is no different from a hub. If an attacker gets control of any device that is connected to the Ethernet switch, he/she can attack the CAM table by MAC flooding which is vulnerability in the design of the switch when the switch has no more space to record MAC address to Port mapping during its learning stage. So as a result, switch loses its identity and begins forwarding any received frames out of all ports, like a hub.

- Double Encapsulation Attacks

Such attacks are common in VLAN environments and so they are also referred to as VLAN hopping attacks. The transmitted frames are tagged with a "VTag" identifier so as to forward these frames to wrong VLAN.

- Multicast Brute Force Attacks

This attack uses the switch's vulnerability to a storm of multicast frames. When the switch receives such a storm of multicast frames rapidly, it will try to constrain the traffic to the original VLAN but failing to do so results in frame leaks to other VLANs which may potentially be malicious.

- Spanning Tree Attacks

Spanning Tree Protocol (STP) is responsible in avoiding switching or bridging loops that cause broadcast storms, which can bring the L2 networks to its knees. In STP, path redundancy is enabled to prevent network loops by assigning ports in forwarding or blocked state, all of which is done by use of election based on MAC address value. So in spanning tree attacks, attacker can force an alternate election which may be rigged by making its MAC value higher than root bridge value and hence compromising the entire network.

- Random Frame Attacks

This attack involves attacker randomly varying the fields of a packet without changing the source and destination addresses. So the attacker is able to send out unwanted malicious traffic to Private VLANs from untrustworthy devices.

6. Layer 3 Non-router Attacks

Attacker predominantly uses ARP cache poisoning so a mode of attack for Layer 3 non-router. During ARP broadcast, the protocol believes in anything that it is told even though it may not have asked for that information. The attacker can send out ARP cache update with malicious information by forging the contents causing the victim's machine to expire its own cache. The attacker is able to sniff, modify and drop packets by poisoning the default gateway and victim's machine by making them believe that his is the requested machine. So basically the attack is directed towards TCP/IP V4 protocol stack which might require replacement.

7. DNS Attacks
    - DNS Cache Poisoning Attack

This attack involves changing the values in the cache that contain previous lookup requests. So by controlling the name resolutions of the sites, the attacker could send unsuspecting users to fake sites without the users knowing the difference.

- Pharming Using the Host File

While this is similar to DNS poisoning, the attacker makes changes to the host file rather than the DNS server. Since TCP/IP protocol consults the host file for resolving DNS names to IP addresses, by any changes to this file, the user may get redirected to fake site even though he/she is typing the correct URL.

8. Layer 3 Routing Attacks
    - Route Table Poisoning

Route table poisoning is inserting fake routes in victim router's routing table by using outdated routing protocols like RIPv1 or IGRP that do not require authentication.

- Source Routed Packets

In this type of attack, the attacker changes the information in the source routing table of the packet itself which causes the packet to select its own path to destination. So basically, the attacker modifies or poisons each router.

9. Man in the Middle Attack (MiTM)

If the integrity of the channel used for communication between two devices or parties is not ensured along with the verification of the identity of the parties at both ends of this channel, then MiTM attack can easily be put into action. MiTM examples include standard MiTM Arp cache Poison attack, SSL attack and iSCSI MiTM. Such an attack is used to accomplish the following:

Impersonate – Attacker tries to impersonate the victim for intercepting messages and send fake certificate to sender.

Eavesdrop – Attacker tries to intercept and listen in on traffic by forcing it pass by a hodgepodge enabled NIC.

Modify – Attacker can modify the message if he is able to decrypt the message to plain text.

Replay of Messages – Similar to network injection attacks, the attacker who is able to listen in on the traffic can change the sequence of the packets making up the message by doctoring the sequence and hence causing the replay of packets due to incorrect sequencing.

Prevent Clock Synchronization – Attacker may prevent synchronization of sender and receiver clocks so as to listen in on the traffic as the synchronized clocks make it difficult to snoop around.

Organizations, either big or small, never want their live systems to be interrupted if they are using virtualization technologies as that would defeat the whole purpose of high-availability features. But such organizations do fail to understand the security risks involved in live migration of virtual machines.

While any and all of the above mentioned vulnerabilities and attack classes would work in creating operational hazards with respect to the virtualized environment, there are certain known virtualization system attacks which have been documented. [10]

1. Management Server attacks – The management console can be exploited to gain login information and hence eventually access to the management server. Also these attacks can allow attacker to change privileges and gain elevated privileges.

2. Admin VM attacks – These attacks result in Denial of Service resulting in crashing admin VM. Passwords stored in cleartext can be obtained easily and which can lead to attackers bypassing authentication.
3. Guest VM attacks – Similar to admin VM attack, this attack can include gaining increased privileges, crashing VM, executing rogue code with admin permissions.
4. Hypervisor attacks – Hypervisor can crash and attackers can easily jump between guests VMs.

Chapter 5

Design and Implementation

The goal of this study was to test the impact security exploits has on a virtualized environment within a typical (simulated) network environment by using consistent, mixed workload simulation.

To accomplish the goals stated above, I created a simulated datacenter consisting of Windows Server 2003 Standard Edition which was the domain controller, two ESX hosts running at at least two virtual machines (VM) each and a virtual machine for VMware vCenter Management Server hosted on Windows Server 2003 Standard Edition. All these servers were the upper level Virtual machines as shown in the generic view below. Each ESX host piggy-backed multiple VMs which were eventually nested VMs.

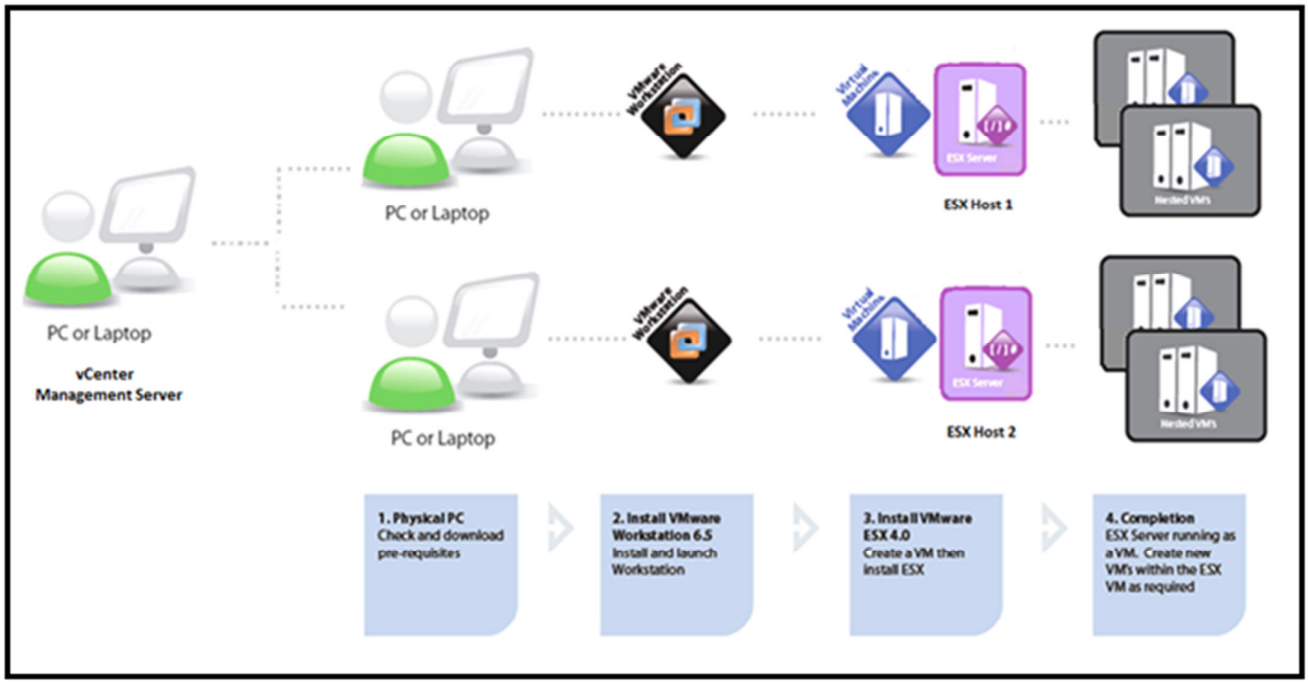


Figure 6. Design Architecture

Being able to run VMware ESX 4.0 and higher under VMware Workstation as a VM, gives IT professional great flexibility. This enables them to test and experiment with products. While the approach is similar to installing and running ESX server in standalone environment, the differences lie in hardware requirements.

Hardware Used

I used three laptops for my testing purpose. Following are the configurations for those laptops:

Hewlett Packard (HP) Pavilion v6000 laptop computer had the following specification:

- Intel Core 3i M370 2.40GHz CPU
- 4GB RAM
- Windows 7 Home Premium 32-bit

Lenovo R400 laptop computer had the following specification:

- Intel Mobile Core 2 Duo P8700 2.53GHz CPU
- 6GB RAM
- Windows 7 Professional 64-bit

Dell Inspiron 8600 laptop computer had the following specification:

- Intel Pentium M Processor 1.60GHz CPU
- 1GB RAM
- Windows Server 2003 R2 Standard Edition

Software Used

- VMWare ESXi 4.0 Releasebuild-171294 [16]
- VMware Workstation 6.5.1 (Build 126130) [15]
- OpenFiler NSA ver 2.3 (Virtual SAN for shared storage - free) [11]
- BackTrack 4 (Penetration testing CD) and Wireshark (Sniffer and Analyzer) [12]
- Veeam Monitor and Reporter(Management tools for VMWare infrastructure – free) [13][14]
- VMware Workstation 6.5.1 (Build 126130) [15]

Figure 7. ESXi Host 1 - Rochester (10.0.0.84)

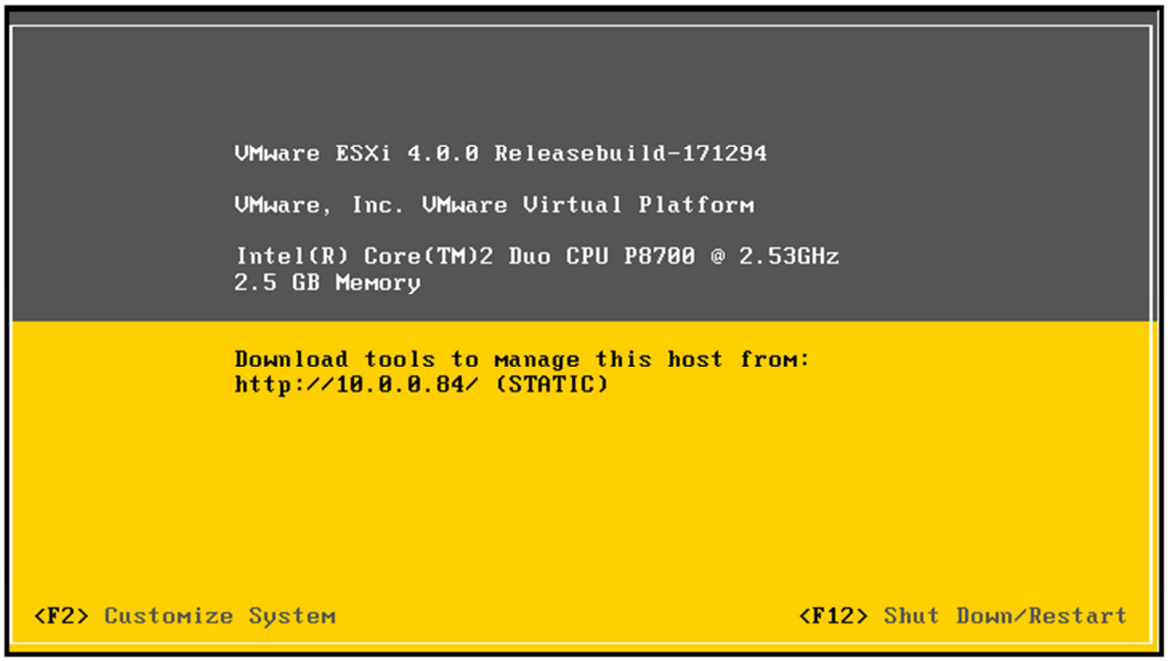


Figure 8. ESXi Host 2 - Milan (10.0.0.85)

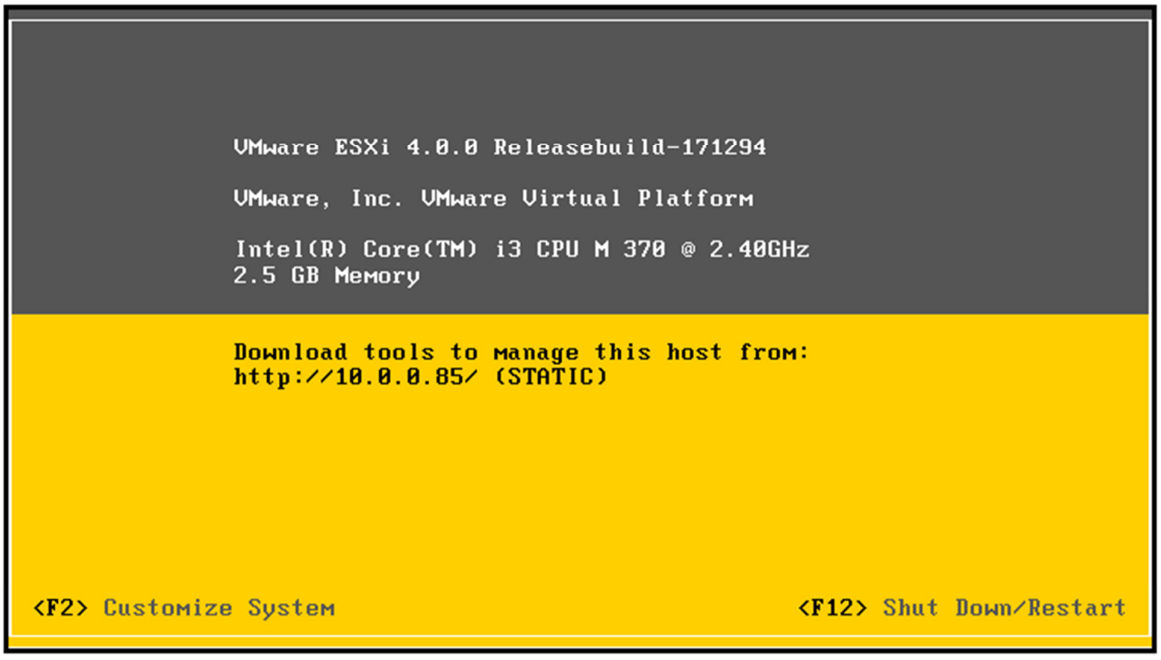


Figure 9. Shared Storage – VSAN (10.0.0.88)

Figure 10. vSphere Server – VCENTER (10.0.0.10)

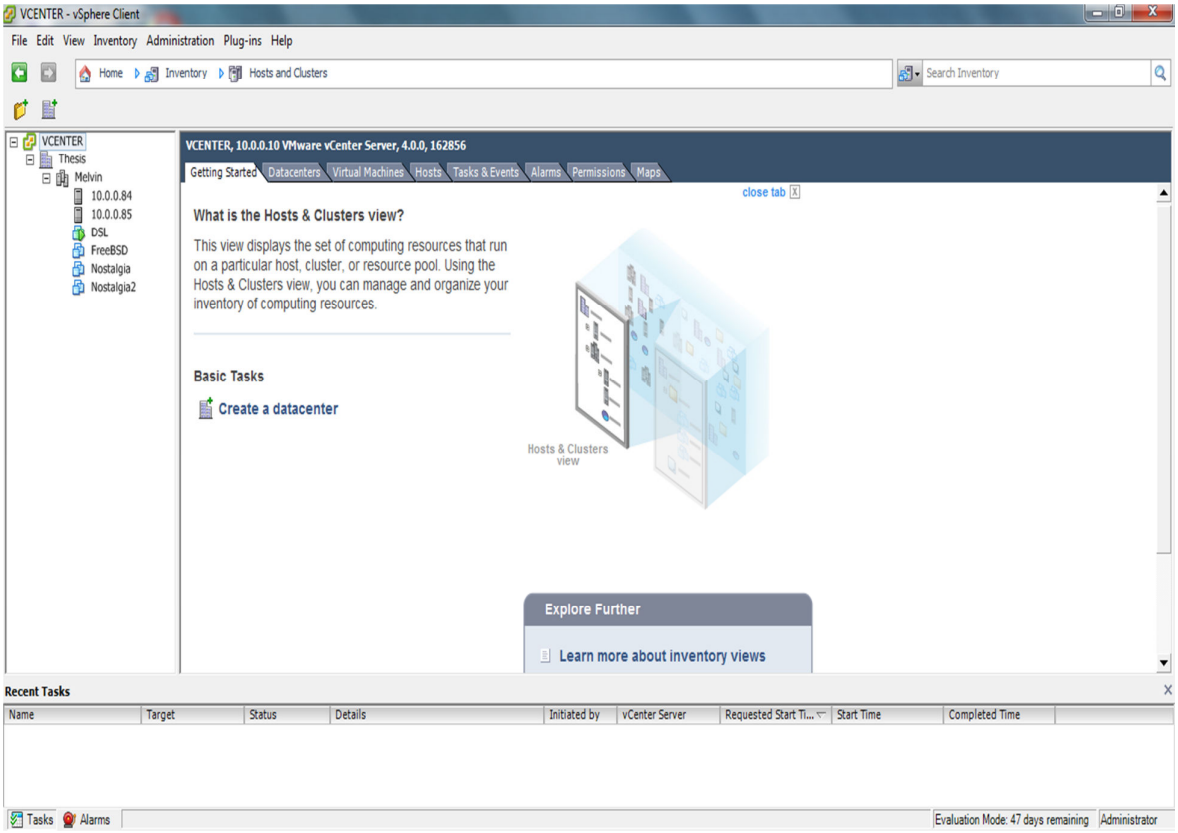


Figure 11. Attacker – Backtrack (10.0.0.15)

Chapter 6

Results – Research and Testing

Starting off with testing the security loopholes in live migration of VMs, I chose the VMWare technology and hence the exploitation of Vmotion which is a patented VMWare technology. Using the design architecture that I mentioned in the Chapter 3, I setup two ESXi hosts which were version 4.0 (Build 171294). I setup a third computer which acted as the vCenter server (management server). I proceeded by creating VMs on each of these hosts which were primarily unix/linux based. Below shown is the layout of the vcenter server which has been aptly named as "VCENTER". There is a datacenter named "Thesis" with a cluster (Melvin) of ESXi Hosts, VMs and templates. Also the Datastore view shows the two local storage called 'Local1' belonging to "10.0.0.84" and 'Local2' belonging to "10.0.0.85". VSAN is the virtual shared storage that has the storage limit of 200 Gb which seemed to serve well for this research.

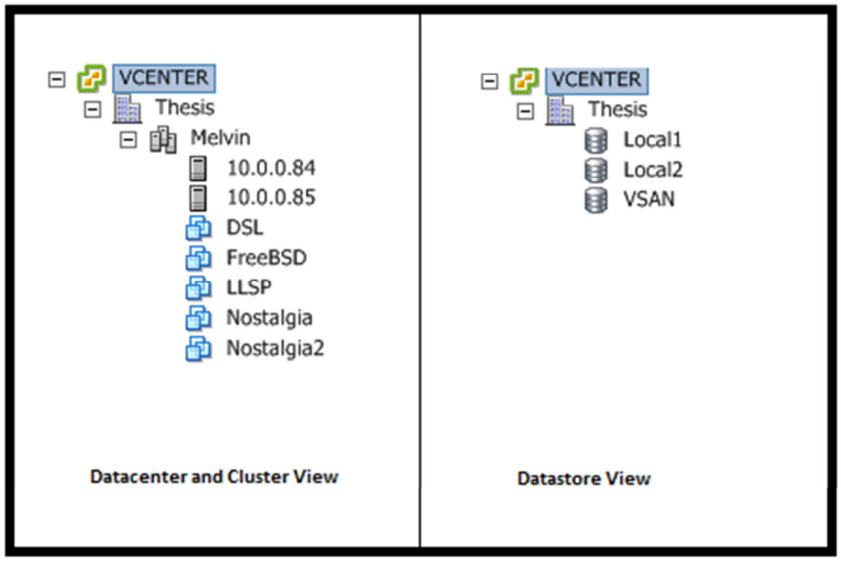


Figure 12. vCenter View

Speaking of security, by default, the VMotion encryption is disabled and most system administrators fail to realize the consequence of that setting assuming that organization firewall would protect against all external threats. Based on the default vCenter settings, I left the encryption disabled.

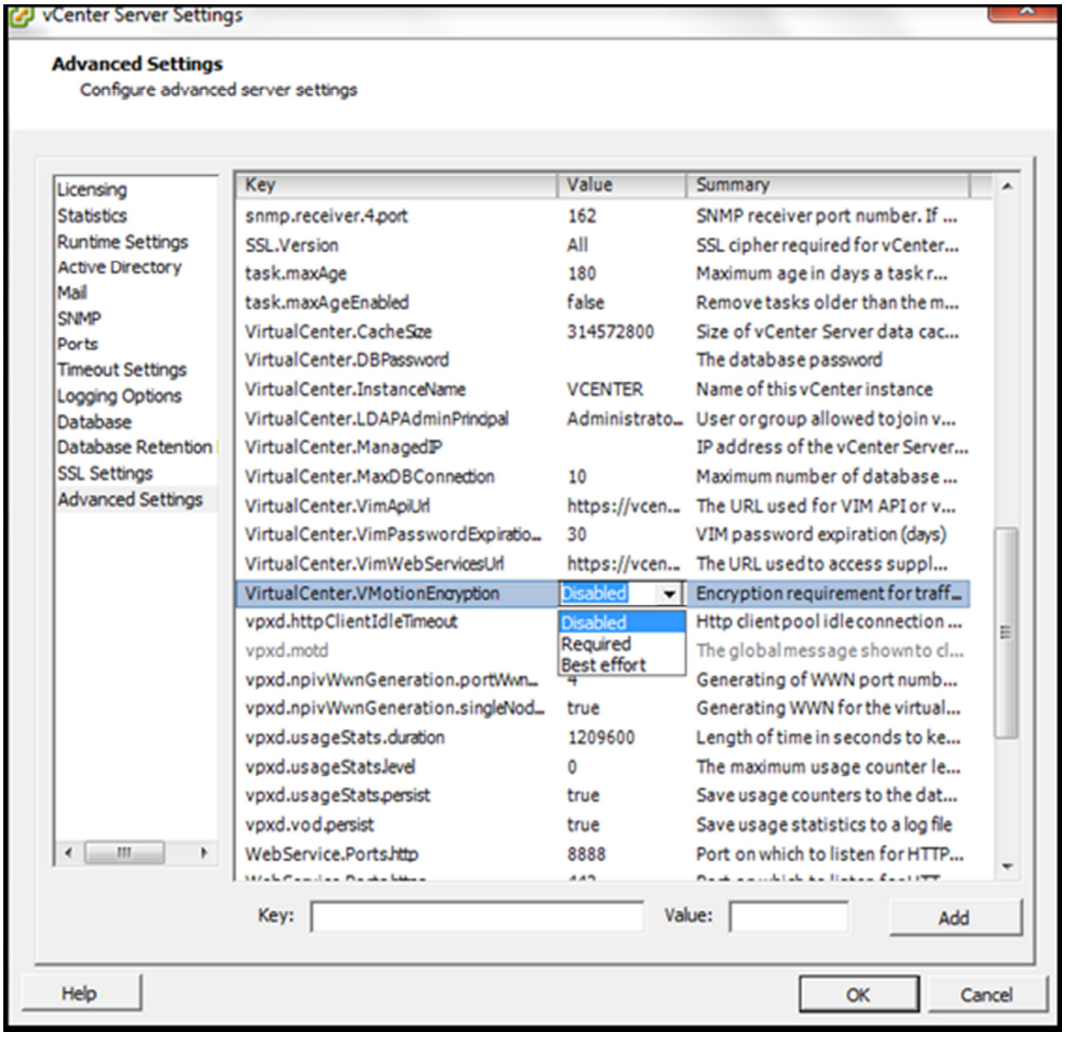


Figure 13. vCenter Server settings window

Technically speaking, vMotion traffic should be isolated which means that it is recommended that we use a separate physical switch or isolate vMotion traffic with a VLAN. The greatest drawback of vMotion is that any related traffic (memory contents of a VM) is transmitted in clear text. There is always a concern as to someone could sniff the network and eventually get access to the memory contents of the VM.

Non-Live Virtual Machine Migration

Initially, exploitation was carried out with power-off VMs so as to compare results with results of exploits with powered-on VMs. It was discovered that if VMs are powered-off, any exploits that were carried out were futile with no loss to application data or virtual machine except in case wherein exploits against VMs involving stealing of guests entirely which meant copying VM files and changing them and pushing them back to the shared store.

Following exploits carried out using the VASTO plugin in Metasploit 3 from BackTrack 4 are related to gaining host based information like and login credentials.

Figure 14. Metasploit Shell console

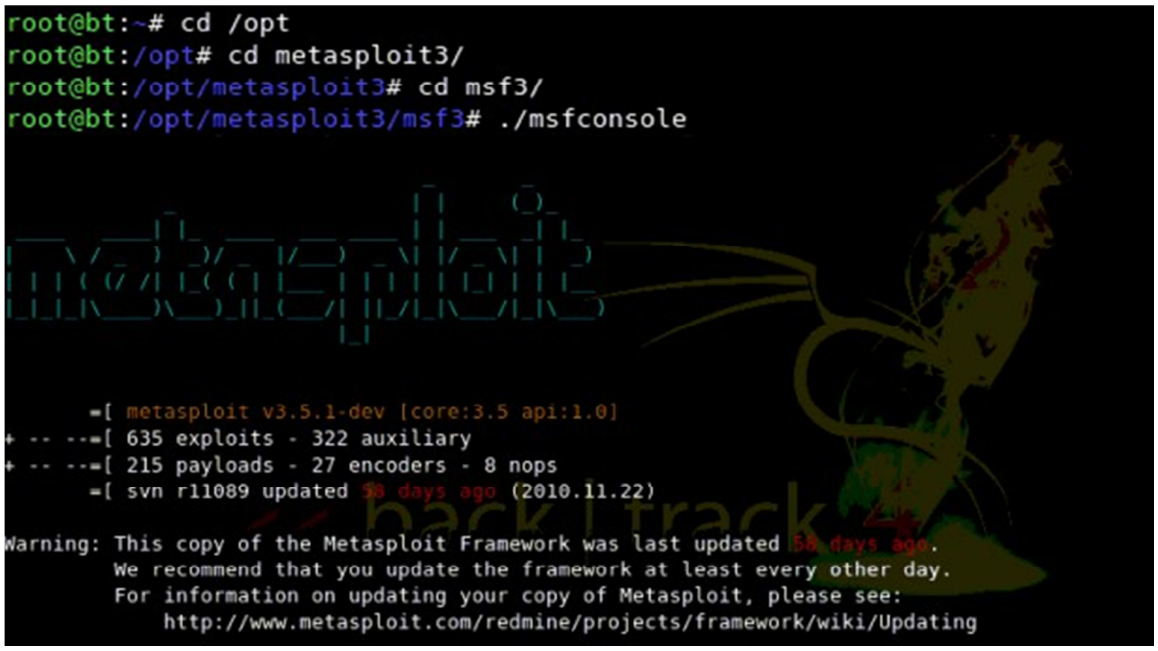
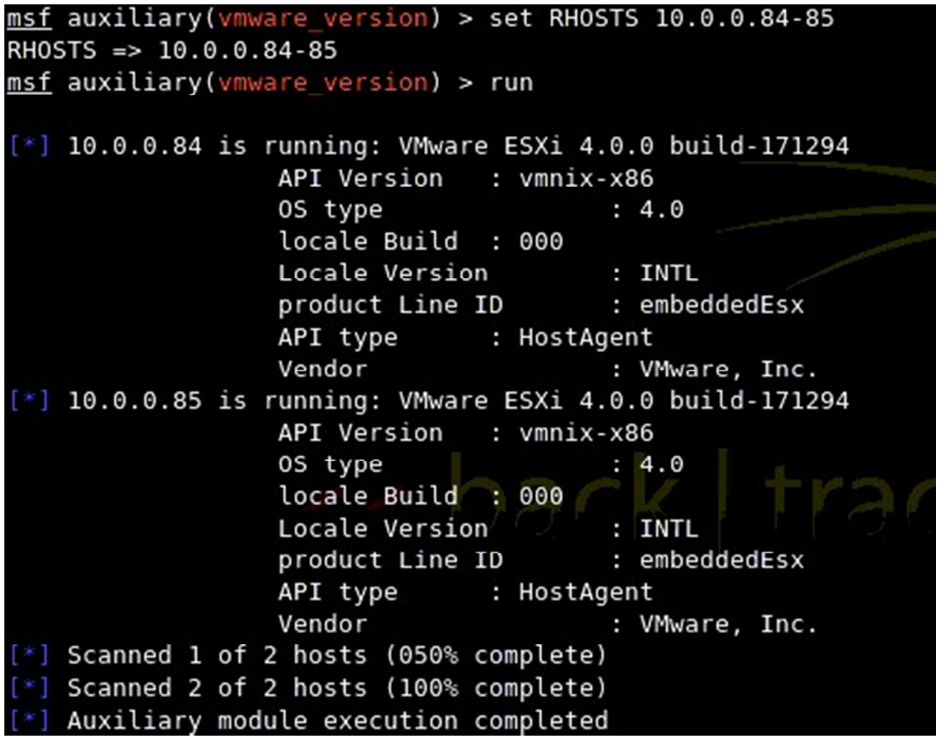
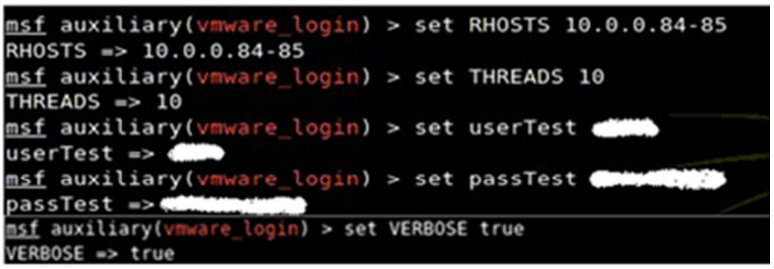


Figure 15. 'VMWARE_VERSION' exploit:



As we see that using this plugin, I was able to extract the versions and build of the ESXi hosts. Using this plugin, I should be able to fingerprint most of the VMware products.

Figure 16a. 'VMWARE_LOGIN' exploit:



Using this plugin, I was able to brute force login information from the ESXi hosts. Like get the username and password information which have been blanked out as shown below.

Figure 16b. 'VMWARE_LOGIN' exploit:

Live Virtual Machine Migration

Now to illustrate the important part of this thesis, that is based on Live Migration of Virtual machines. Trying to exploit and observe this type of migration process is the main purpose of this thesis paper.

Test Method 1 (using Wireshark)

During testing, I created a scenario in where I wanted to sniff and analyze VM guest memory during the transition phase known as VMotion. Taking the advantage of the default encryption mode (=disabled) of VMotion, I initially used only Wireshark (formerly Ethereal) which is an sniffing tool, to capture and analyze packets created on the network due the transfer of VM from one ESXi host to another. The VM that I used was a linux-based Operating System called as "Damn Small Linux" aptly named as DSL in the vCenter. I setup the VM Portgroup to accept promiscuous mode where the Wireshark is running.

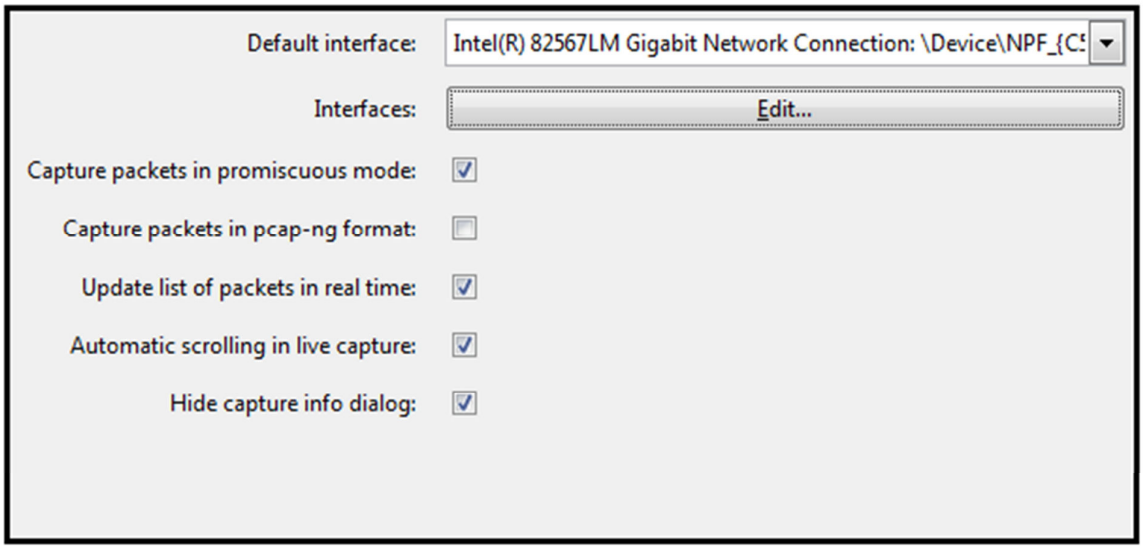


Figure 17. Wireshark settings

In DSL, I opened up a terminal session and simply wrote onto a file (saved it as a text document) using the 'CAT' command. This was done while sniffer was doing its job. I "VMotioned" the guest VM (DSL) from one ESXi host to other (10.0.0.84 → 10.0.0.85). When Vmotion was complete, I stopped the sniffer and analyzed the sniff wherein I noticed that I was able to see my entire operation including the commands that I inputted to create and write a file in the guest VM.

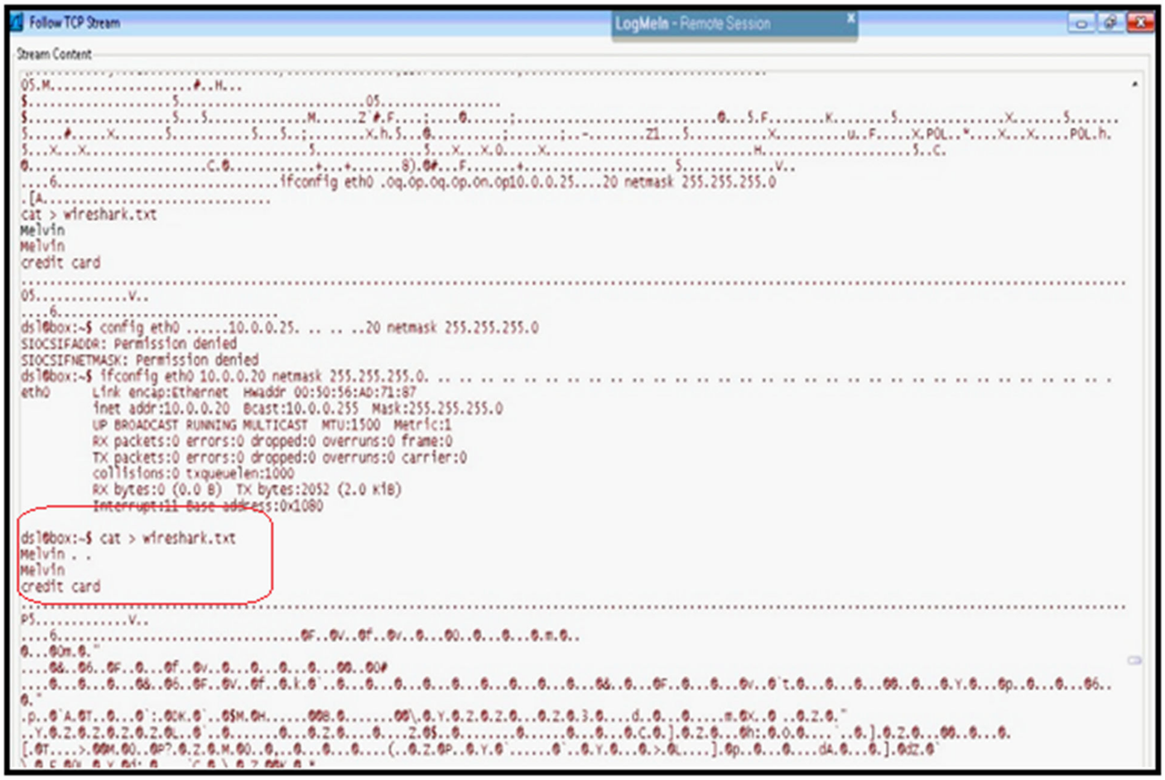


Figure 18a. Wireshark sniffing results

So continuing on with the test, I enabled VMotion Encryption and selected “Required” over “Best Effort”. From the research and reading that I have done so far, I am optimistic that the result is going to be the same and I was correct. Wireshark has still been able to sniff out the TCP data allowing me to see plain text that’s crossing the network. How can this be? As with any system administrator, thinking that restarting services of the ESXi host machines and checking if “vCenter.VMotionEncryption” setting was applied correctly, I did the same but still VMotion remains unencrypted. Is this a bug? Probably so, but after such instances being reported over past two years VMWare would have been expected to provide a patch. But again not to sound critical, VMWare has come up with application like vShield to protect VMotion.

While wireshark displayed plain text data passing over the network during VMotion, I was also able to view other clear text system properties using the TCP stream:

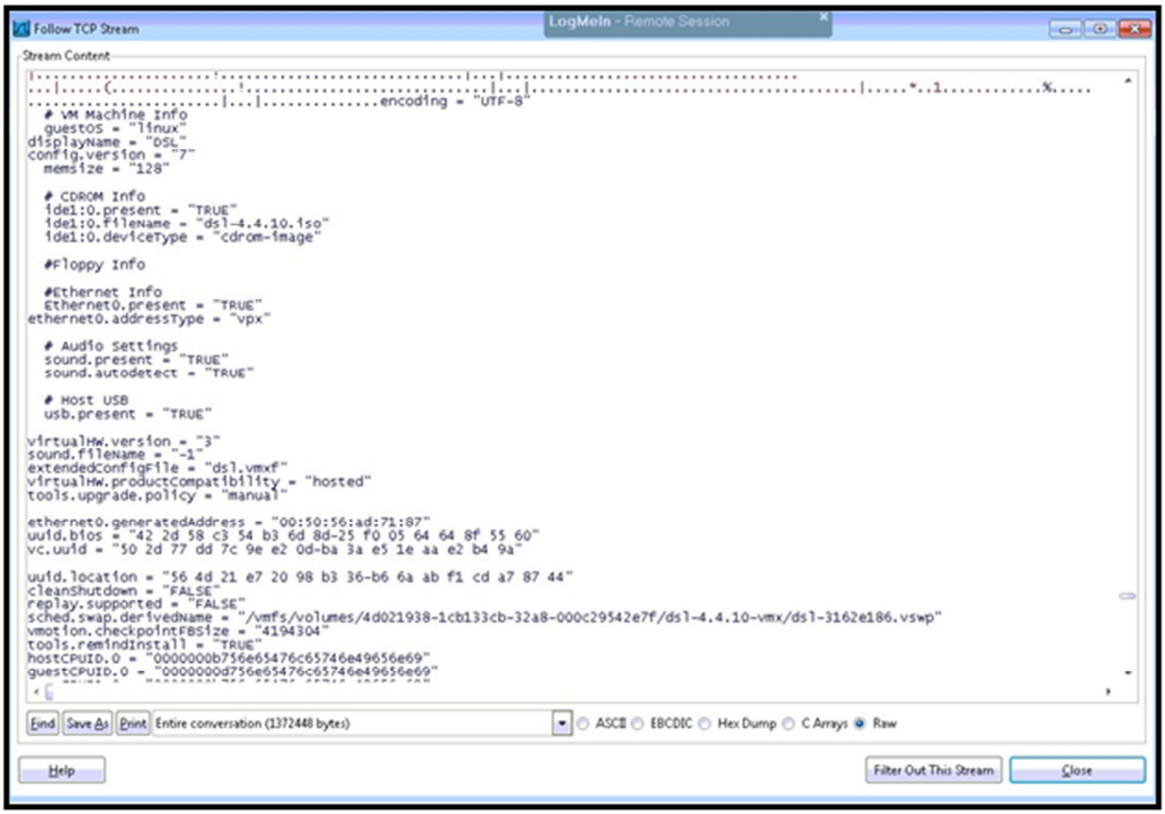


Figure 18b. Wireshark sniffing results

Test Method 2 (using Backtrack – ARP Poisoning MiTM)

Let me elaborate the exploits that I carried out in such a scenario. Using one of most powerful penetration testing and hacking utilities provided by BackTrack, certain exploits like gaining ESXi host-based information like root logins, versions, builds, number of VMs residing on each host, and so forth, was collected, which is an enormous security risk. Along with the above information that was easily acquired, I was also able to access VM file structure and hence unleash an exploit called as “Guest Stealer”.

Having conducted these exploits, gave me certain insights to possible loopholes in virtualization security and so I took the next step as to exploit powered-on VMs while in transit. Since this research was concentrated towards VMWare appliances, I proceeded to exploit “VMOTION”.

One of the most well-known attacker-victim based scenario is Man-In-The-Middle (MITM) attack which I incorporated using ARP poisoning methodology.

MITM attacks are most often attacks against data confidentiality and these attacks run at layer 2. It is easily possible to execute a MITM in a switched ethernet environment. There are two types of MITM attacks –

Active and Passive. While active attack involves modifying data packets as they are intercepted, passive attack is more concentrated towards stealing information in transit without modifying it. Passive attack is more common MITM as it is easy to execute. I will be demonstrating a passive MITM attack on Vmotion.

I started off by executing an ARP poison on the attacker – 10.0.0.15, using ETTERCAP. The attack wrote the output to a text file. The victim machines are the two ESXi hosts – 10.0.0.84 and 10.0.0.85.

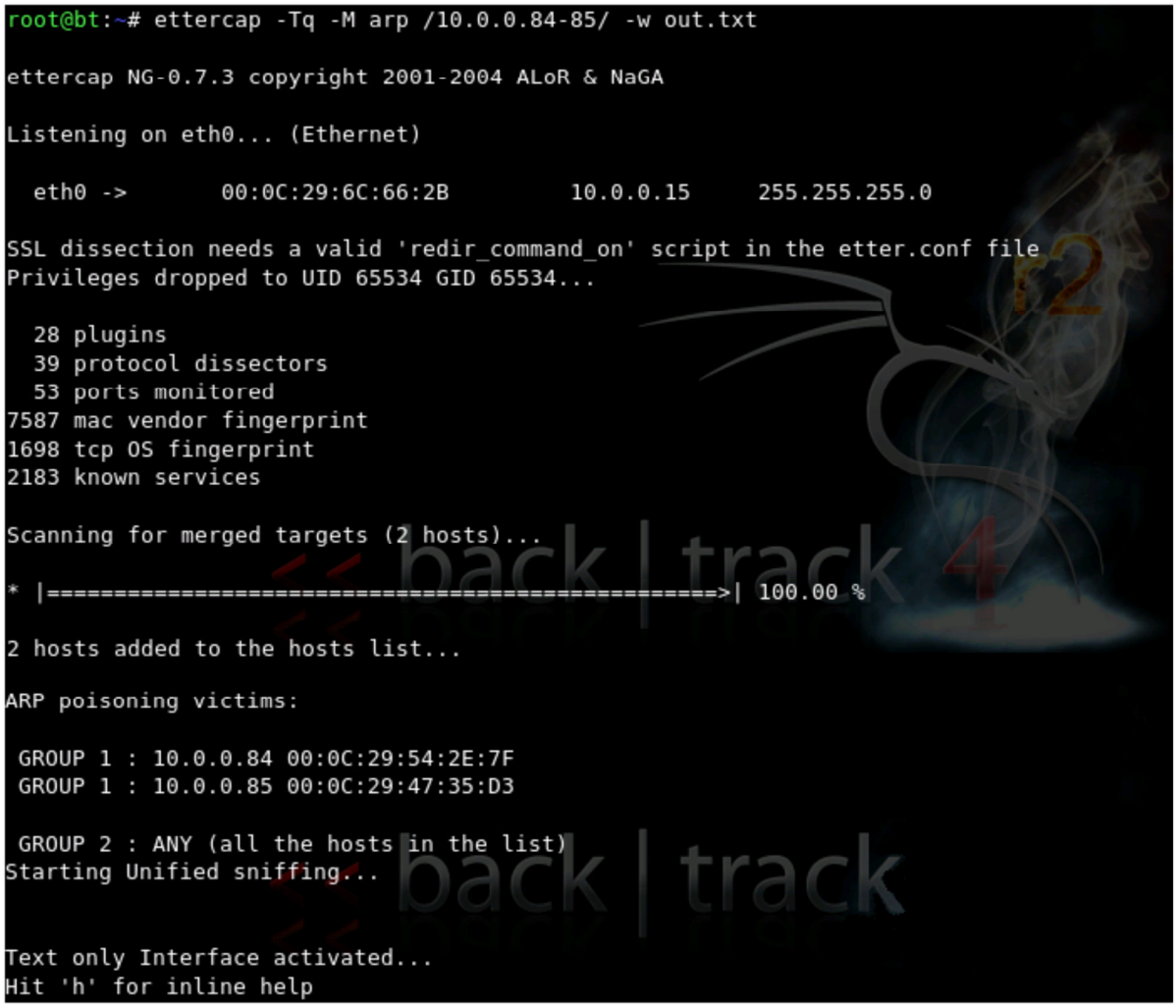


Figure 19. Metasploit – Ettercap (ARP Poisoning - Start)

The attacker started sniffing on the 10.0.0.0/24 network but specifically for data exchange between the ESXi hosts Rochester (10.0.0.84) and Milan (10.0.0.85). I proceeded to write data onto a file on the VM that I chose to migrate in live state.

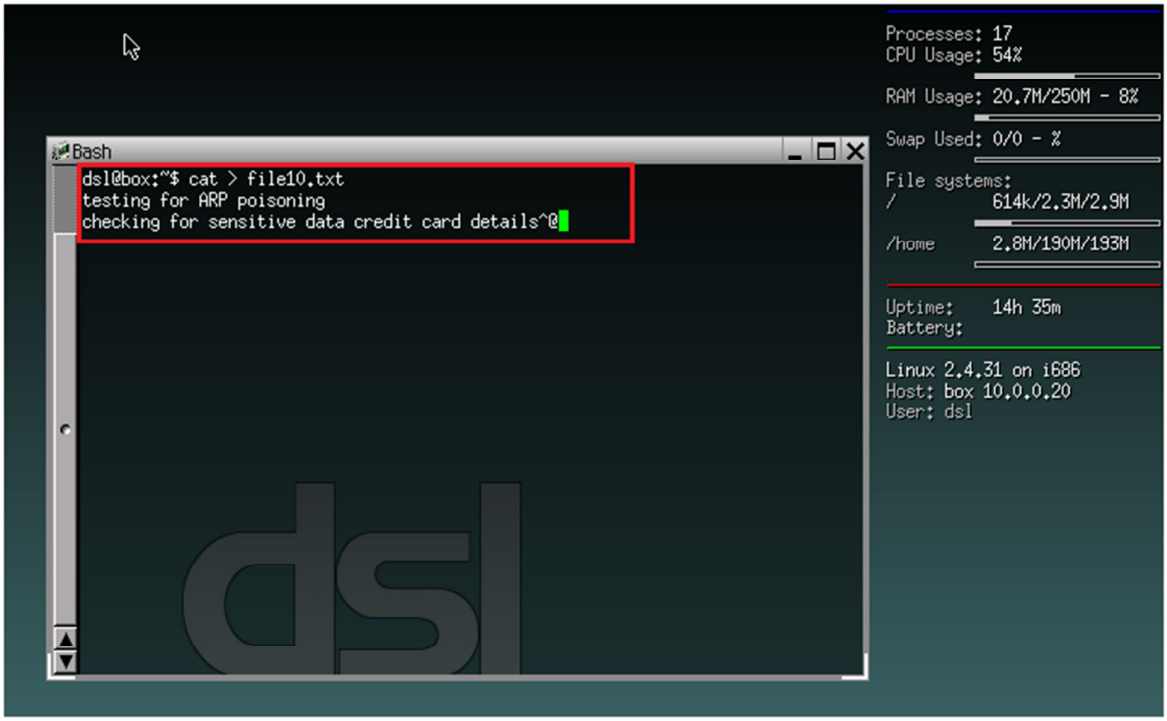


Figure 20. VM data preparation before Live Migration

I saved this data to the file using 'cat' command. Then I proceeded to perform live migration of the VM (DSL) from 10.0.0.84 to 10.0.0.85 using Vmotion as shown below.

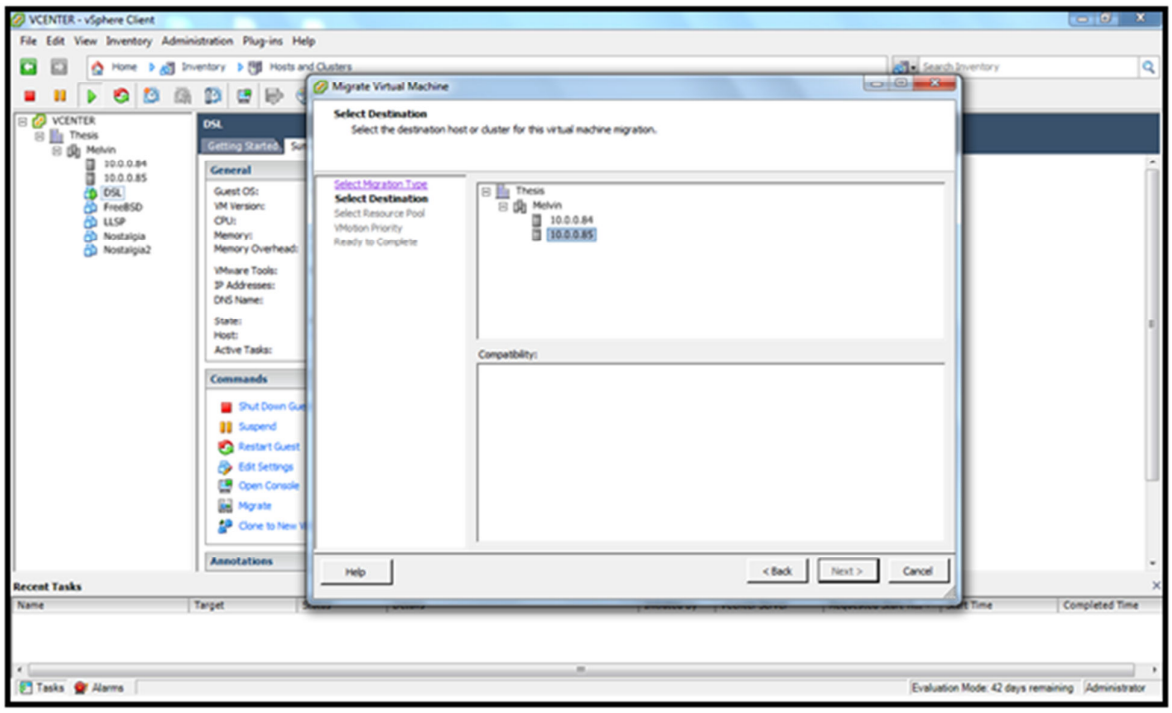


Figure 21. Live VM Migration

The process of live migration took less than about a minute to complete which was extended making me believe that the MITM attack had something to do with it. I was able to observe this as I compared the first method using wireshark with this current attempt by using ARP poisoning. While the data travelled directly from source to destination in first case and wireshark sniffed packets on the network; using MITM, TCP packets were first intercepted by attacker (10.0.0.15) and then forwarded to destination. Following graphics show the prolonged time taken by VMotion during MITM as observed based on various machine properties.

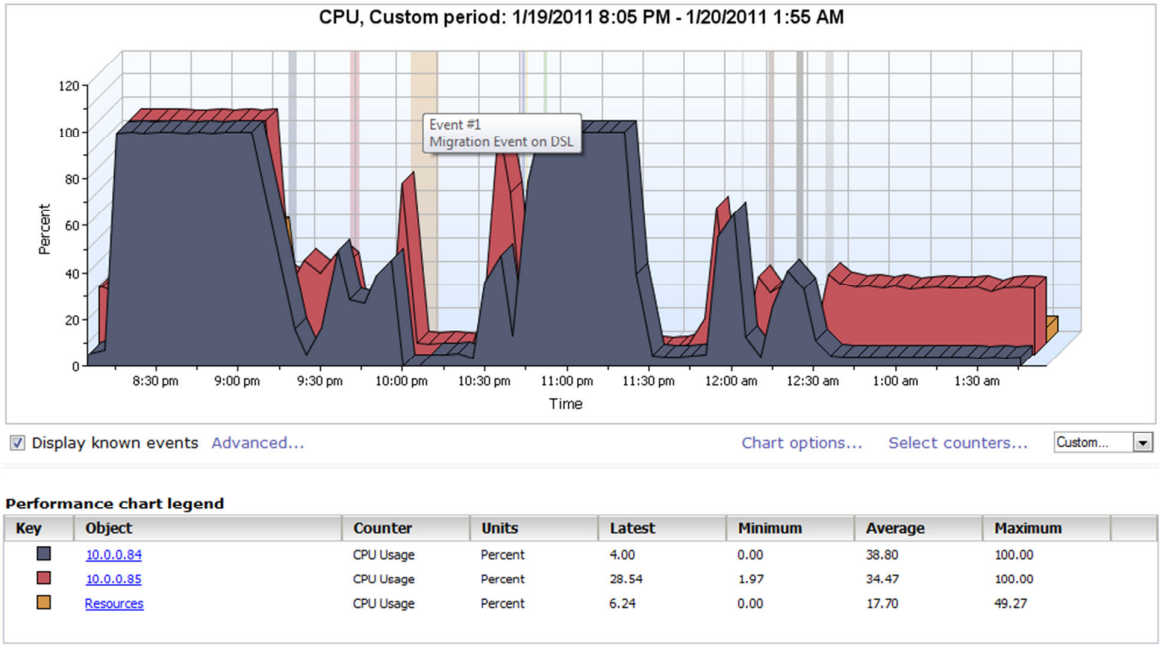
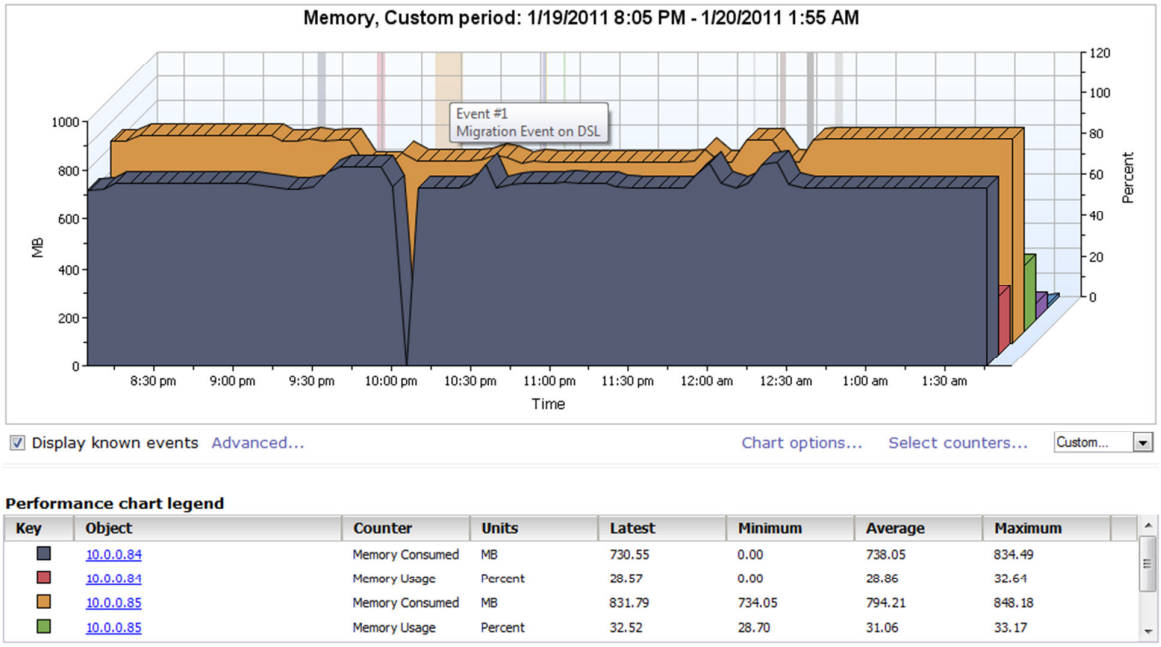


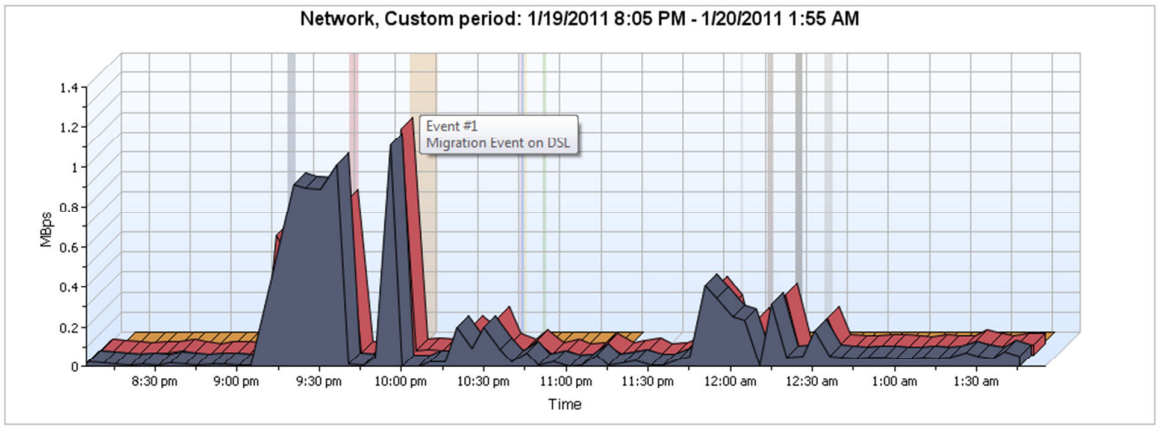
Figure 22. CPU readings during Migration of DSL



Performance chart legend

Key	Object	Counter	Units	Latest	Minimum	Average	Maximum
	10.0.0.84	Memory Consumed	MB	730.55	0.00	738.05	834.49
	10.0.0.84	Memory Usage	Percent	28.57	0.00	28.86	32.64
	10.0.0.85	Memory Consumed	MB	831.79	734.05	794.21	848.18
	10.0.0.85	Memory Usage	Percent	32.52	28.70	31.06	33.17

Figure 23. Memory readings during Migration of DSL



Performance chart legend

Key	Object	Counter	Units	Latest	Minimum	Average	Maximum
	10.0.0.84	Network Usage	MBps	0.05	0.00	0.13	1.11
	10.0.0.85	Network Usage	MBps	0.05	0.00	0.10	1.13
	Resources	Network Usage	KBps	0.00	0.00	0.00	0.00

Figure 24. Network readings during Migration of DSL

I will discuss more about the above observed facts as I proceed. So once, the live migration was completed successfully, I went back to the attacker and stopped the ARP Poison attack. As you can notice below, the attacker was kind enough to re-arp the victims back to original state.

Figure 25. Metasploit – Ettercap (ARP Poisoning - Stop)

Now that attacker had sniffed the packets while the Vmotion was in progress; the attacker machine (10.0.0.15) needed to analyze the raw packets. While there are multiple ways to do that like simply open the output text file that was recording the network data packets and search for the string or numeric data or conduct a TCP dump from the file based on word count and/or string which would mean scanning each page or filtering out using a search parameter. I chose to simply use the 'find' option to look for the string that I knew was written on to the file in the VM. Below shown is the complete data that was entered in the VM and sniffed out during the MITM attack.

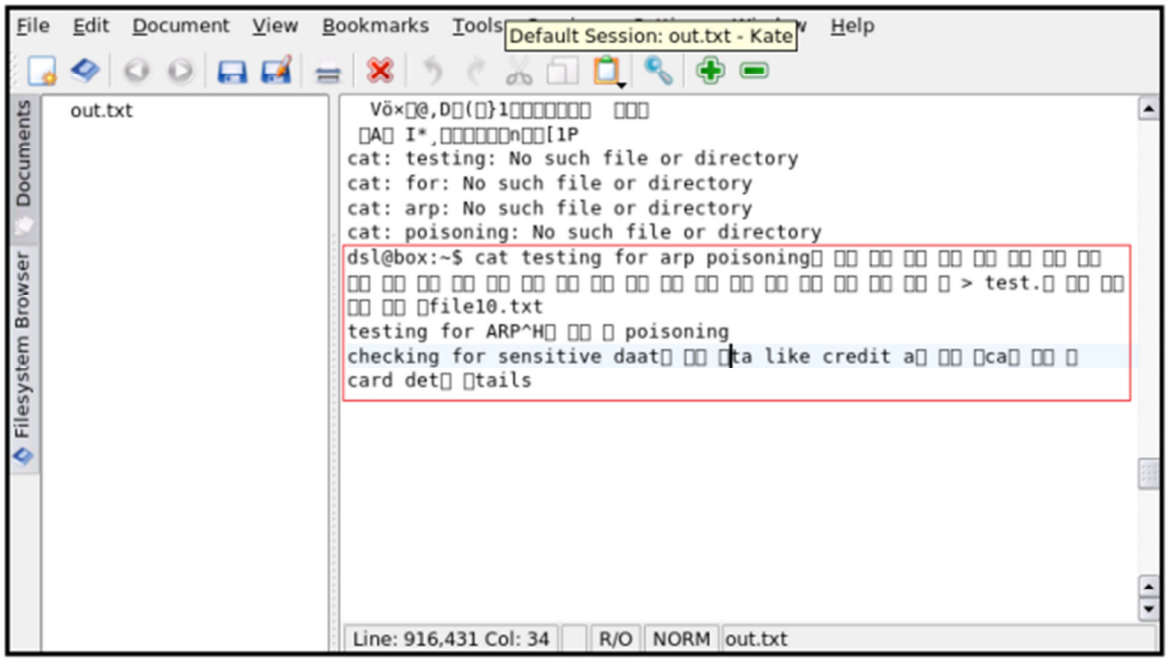


Figure 26. ARP Poisoning – Data compromise

Workload disturbance

In this part of the experiment, I created an artificial workload which would serve as the network traffic generator from a range of IP ranges on the same network as the ESXi host machine. I generated about 100 billion TCP packets and targeted it towards one of the ESXi host (10.0.0.85).

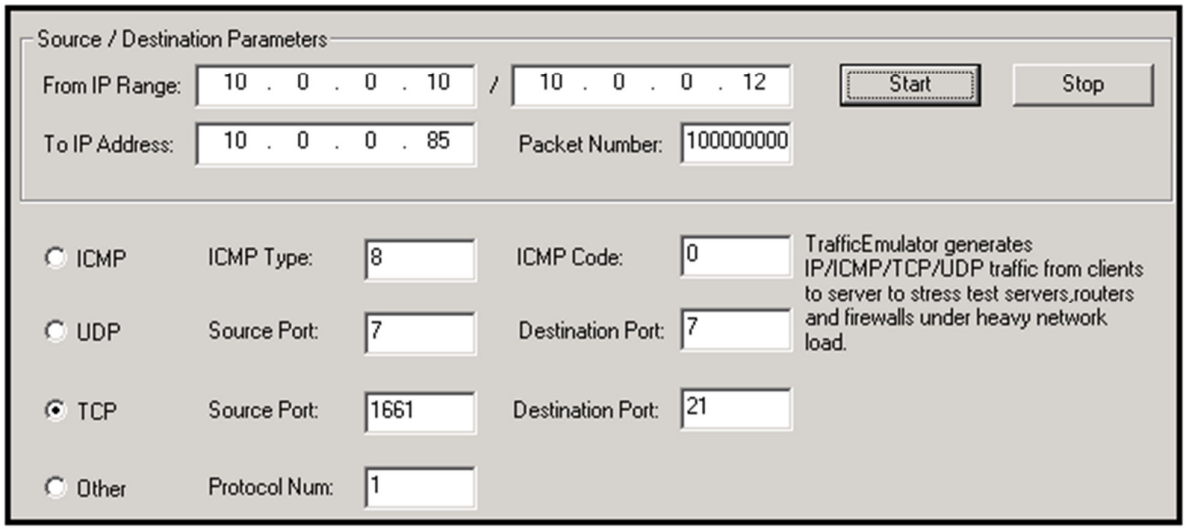


Figure 27. Network Packet Generator - TCP

I proceeded to check the vCenter and wanted to carry out a migration from that EXi host to other host. But, this extreme bombardment of network traffic towards the ESXi host caused the host to disconnect from the vCenter. I had to abandon the traffic generation midway so as to restore connection back between ESXi host and vCenter.

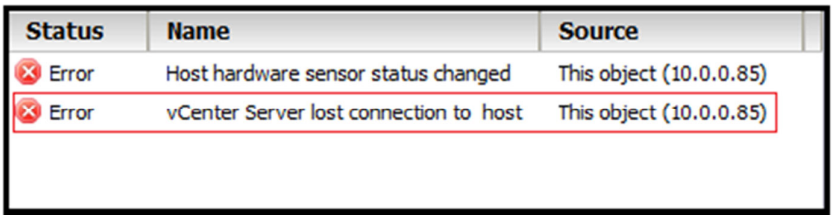
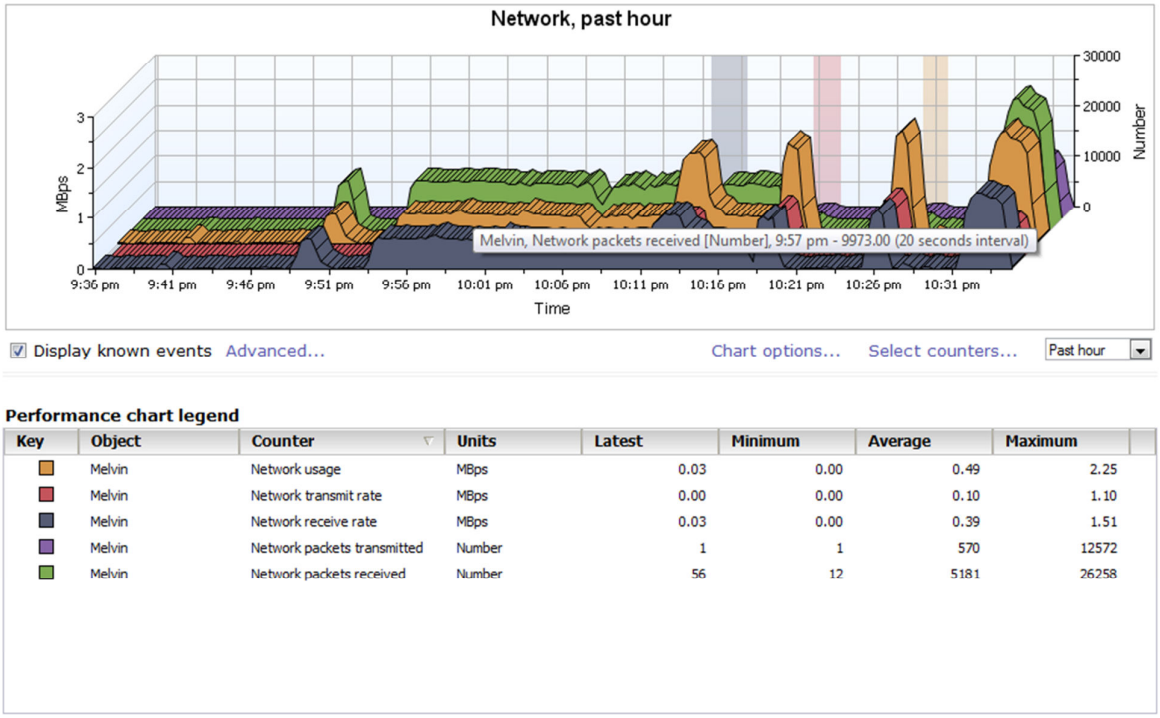


Figure 28. Result of Artificial network traffic

Let me also mention that before I bombarded the ESXi host with network traffic, I tested to see if that similar traffic could be tested against the vCenter server. But apart from spikes in network counters like network usage, transmit rate, receive rate, packets transmitted and packets received, the live migration of VM was not affected with a migration time ranging between 1-2 minutes.



Performance chart legend

Key	Object	Counter	Units	Latest	Minimum	Average	Maximum
	Melvin	Network usage	MBps	0.03	0.00	0.49	2.25
	Melvin	Network transmit rate	MBps	0.00	0.00	0.10	1.10
	Melvin	Network receive rate	MBps	0.03	0.00	0.39	1.51
	Melvin	Network packets transmitted	Number	1	1	570	12572
	Melvin	Network packets received	Number	56	12	5181	26258

Figure 29. Graphical representation for Network traffic conjestion

Moving ahead, I created network traffic while the vMotion was in progress, the same scenario popped up and the vCenter lost connection with the ESXi host and eventually all the VMs associated with the host, I waited for about 15minutes for the reaction of the migration process. Hoping for the worst, as I stopped the artificial traffic towards the ESXi host, the live migration failed as expected but the VM was unharmed. The VM was still in live state and I was able to access the VM. Performance counters within the vSphere client for vCenter showed elevated readings for CPU, System and Network. There was no trace of any activity for Memory and Disk performance counters which points to lost connectivity with storage and no heartbeat detected otherwise.

Status	Name	Source
Error	Heartbeat is missing for VM	DSL
Error	Host hardware sensor status changed	10.0.0.84
Error	Host hardware sensor status changed	10.0.0.85
Error	Storage connection failure	10.0.0.84
Error	vCenter Server lost connection to host	10.0.0.85

Figure 30. vCenter log –VM migration crash due to Host connection loss

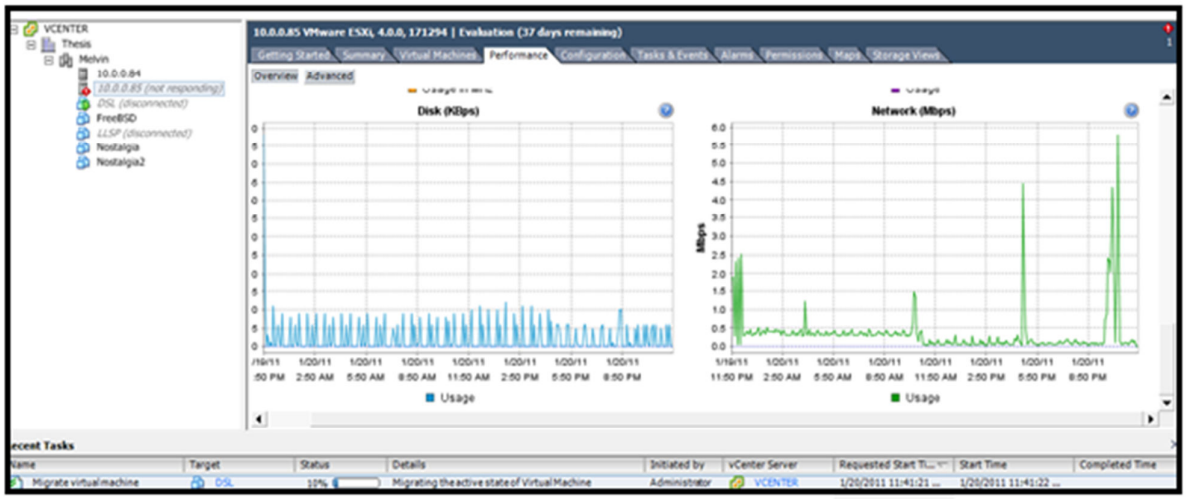


Figure 31. Host connection lost (vCenter graphs)

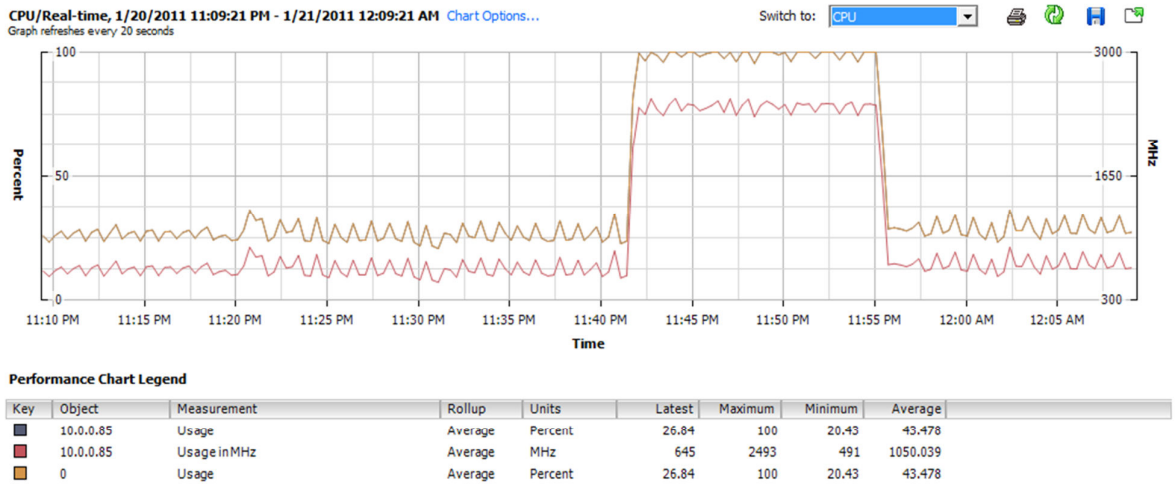


Figure 32. CPU readings during hung state

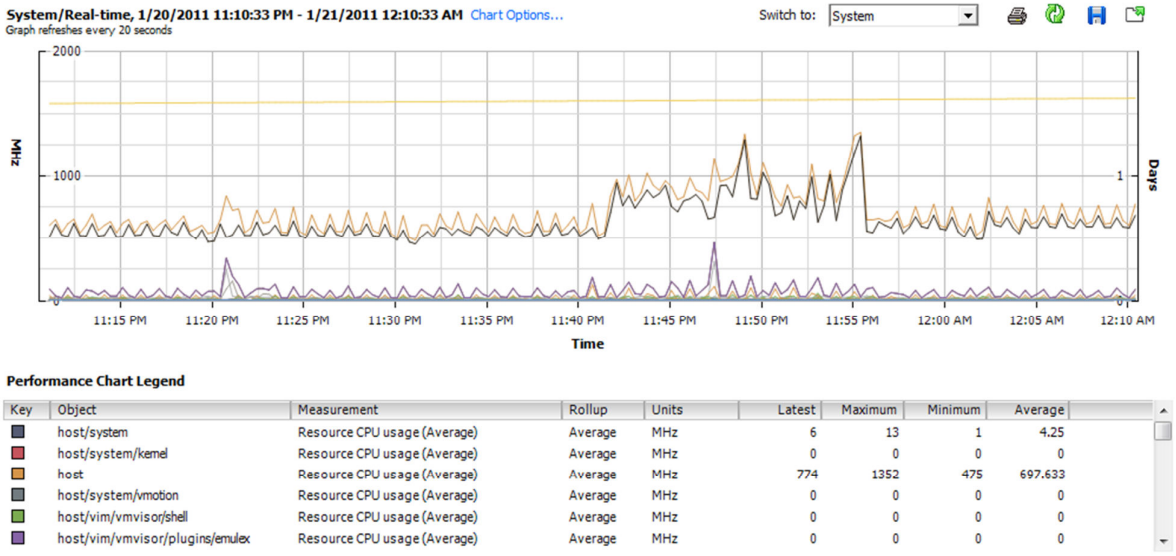
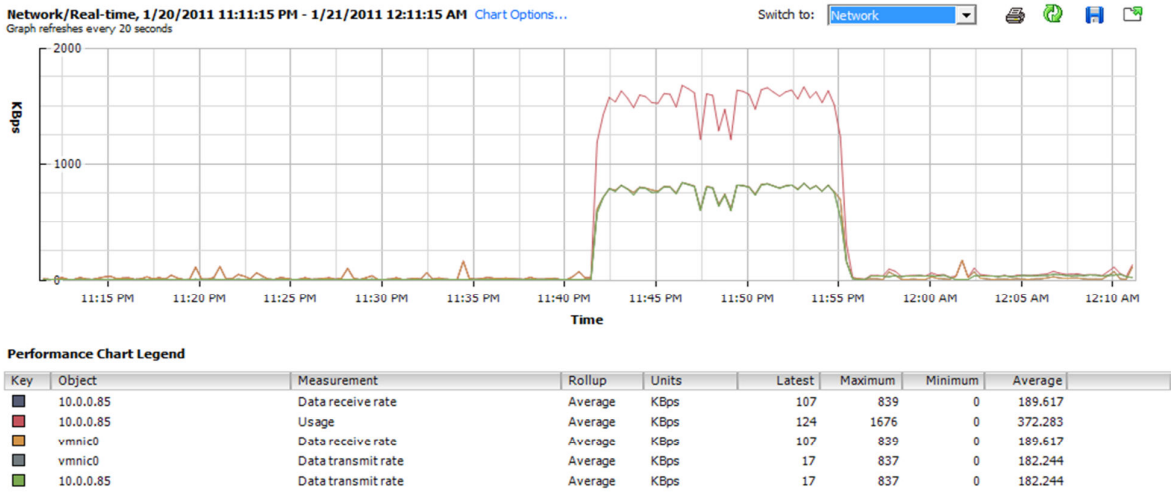


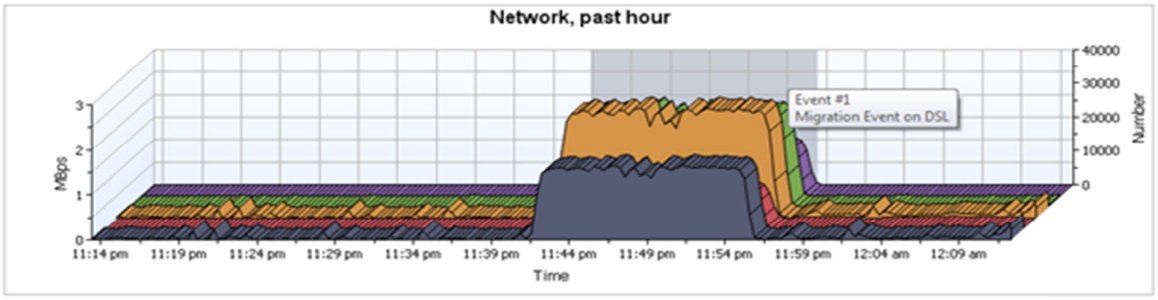
Figure 33. System readings during hung state



Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
	10.0.0.85	Data receive rate	Average	KBps	107	839	0	189.617
	10.0.0.85	Usage	Average	KBps	124	1676	0	372.283
	vmnic0	Data receive rate	Average	KBps	107	839	0	189.617
	vmnic0	Data transmit rate	Average	KBps	17	837	0	182.244
	10.0.0.85	Data transmit rate	Average	KBps	17	837	0	182.244

Figure 34a. Network readings during hung state

When the ESXi host machine detected about 2.5 Mbps of traffic through its network interface, the process of live migration of the VM (DSL) was sent in halted state which we can see from the graphical representation below, was extended for about 15 minutes. After the artificial traffic was ceased, the VM failed to recognize the storage adaptor and the tunnel to the destination ESXi host was disconnected. The process of vMotion failed but VM was unharmed and subsequent vMotion worked correctly. From this output we can assume that creating workload hotspots does create issues with vMotion. This is an security risk as even though the extended state of vMotion did not harm the VM in this case, but if there were workloads that were severe enough to cut the vMotion in half creating a split image, then the VM could be thrown in unstable state and hence non-recoverable from this crash.



Key	Object	Counter	Units	Latest	Minimum	Average	Maximum
	Melvin	Network receive rate	MBps	0.22	0.00	0.40	1.64
	Melvin	Network transmit rate	KBps	6.00	0.00	184.66	824.00
	Melvin	Network usage	MBps	0.22	0.00	0.58	2.45
	Melvin	Network packets received	Number	338	9	6312	29084
	Melvin	Network packets transmitted	Number	12	1	3019	14620

Figure 34b. Network Readings during the state

Workload distribution for virtual machines

The goal for this test scenario was to research and test an efficient virtual machine based resource reservation in a clustered environment. This had to be done by making sure that resources are vacated in time without underutilizing the resources allocated to it. The model that I worked on has been already tested and my idea for test analysis is solely to test the security implication of the exploits at the time of this distribution. [17]

Using Distribution Resource Scheduler (DRS), which allows concurrent migrations by allowing cluster resources to be, divided into smaller resource pools for the VMs. Since the major concept of DRS is to enable automatic assignment of VMs to hosts managed through vCenter so as to load balance resources, I was intrigued as how this would react if there was security threat that indicated performance counter suggesting DRS be activated.

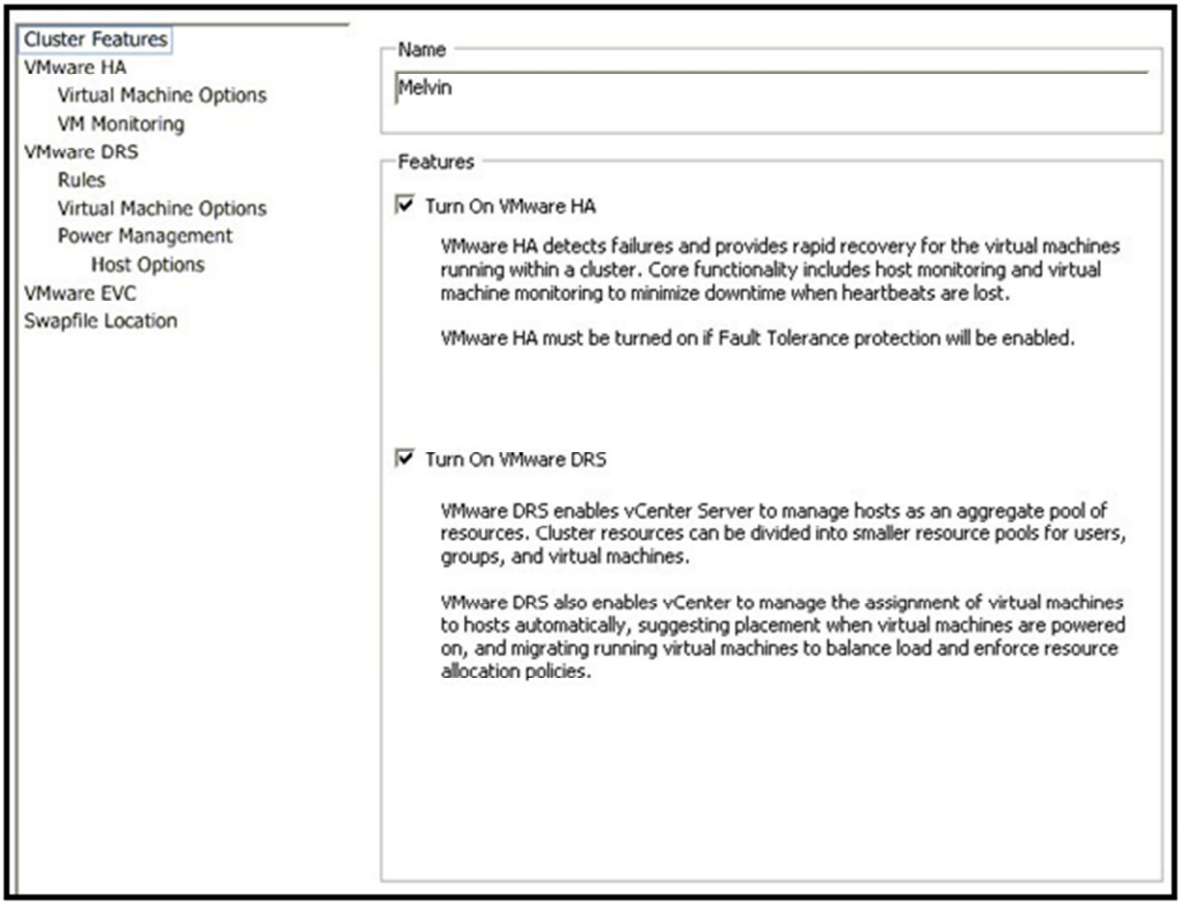


Figure 35. vCenter DRS settings

So as I went about testing Live VM migration, I tested based on VM memory. First test was done while migrating single VM (DSL) based on varying memory values ranging from 64 Mb to 512 Mb and for each value, I was able to observe that there was an exponential increase in the migration delay (in seconds). This delay can be attributed to the suspension of VM at the source rather than resume phase at the destination. The memory range was created using a rogue memory injection program which was basically a stress test tool available as freeware from Google. This stress test tool prolonged the suspend phase at source during vMotion which is due to extra time needed for synchronization.

Not being too familiar with as how DRS would react in such a scenario, wherein memory injection needing more resource from the Host, I failed to notice the change. I could not stress test beyond 512 Mb every time I tried doing that my ESXi host machine would go in non-responsive state with 2 out of 3 VMs alive during this state. So DRS would not kick in when it was supposed to leading to this failed state.

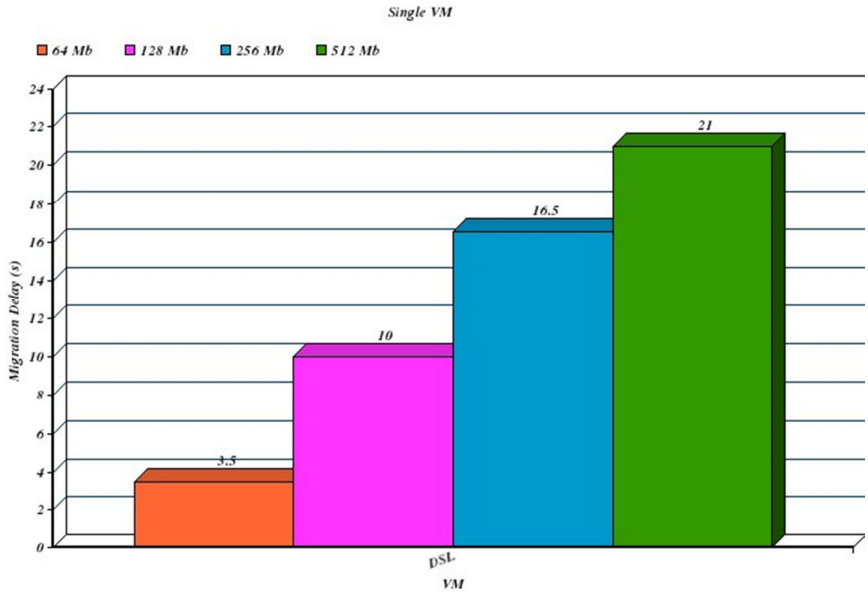


Figure 36. Single VM – Migration delay

Since DRS failed to act upon the load balancing state, I proceeded to test using migrating sequence of VMs which were similar in performance. Let me mention this as a reminder that migration is a memory intensive process and it was necessary to account for performance degradation. When live VMs were migrated in sequence the migration delays seemed nominal as Host based resource pool had enough time to finish vacating resources from one VM before working on the next one. In a different case when CPU stressing

workload was iteratively run so that 100% CPU was consumed, the applications or processes that were running on VMs take longer to complete.

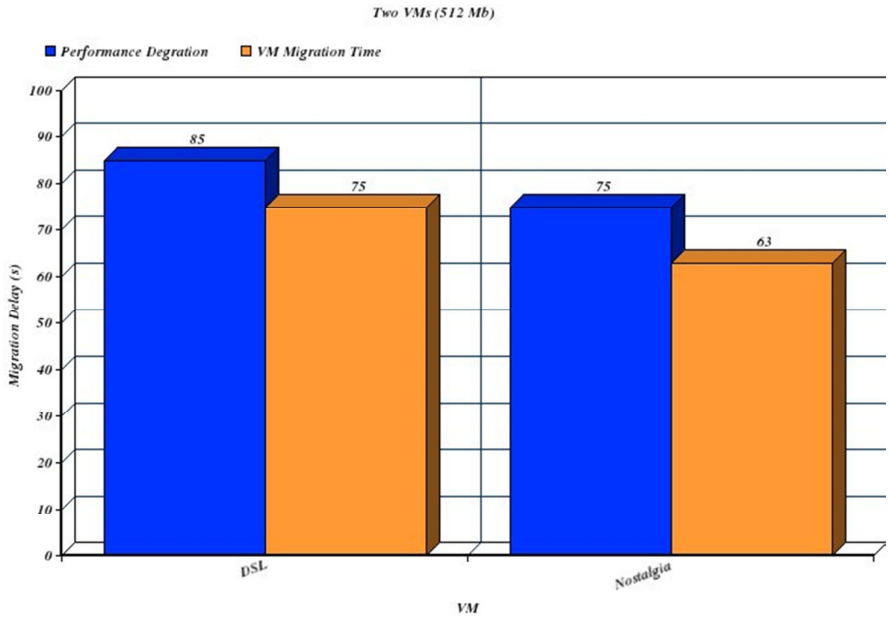


Figure 37. Two VMs – Migration delay

Finally when I worked on migrating VMs in parallel, I had to compare it with the results as seen in sequential process. Since all this resource handling is done on the Host machine, the migration time is obviously going to vary based on the fact whether it is sequential or parallel. Parallel migration results in resources on the Host being freed up sooner and hence the total migration time is shorter as compared to sequential process which takes longer overall migration time but shorter time per-VM and lesser impact at the application level.

I performed the same test for workload disturbance and distribution so as to achieve DRS and this time I coupled memory variation test with network load stress. At about 2.5 Mbps of bandwidth being utilized for the network, varying memory load between 64 Mb and 512 Mb, I was able to observe that DRS kicked in, the VM in transit was forced back to source host in case migration state between 25% and 75%. This was observed when destination host was bombarded with TCP packets. Above 75%, there was a crash in migration state and the VM became inactive.

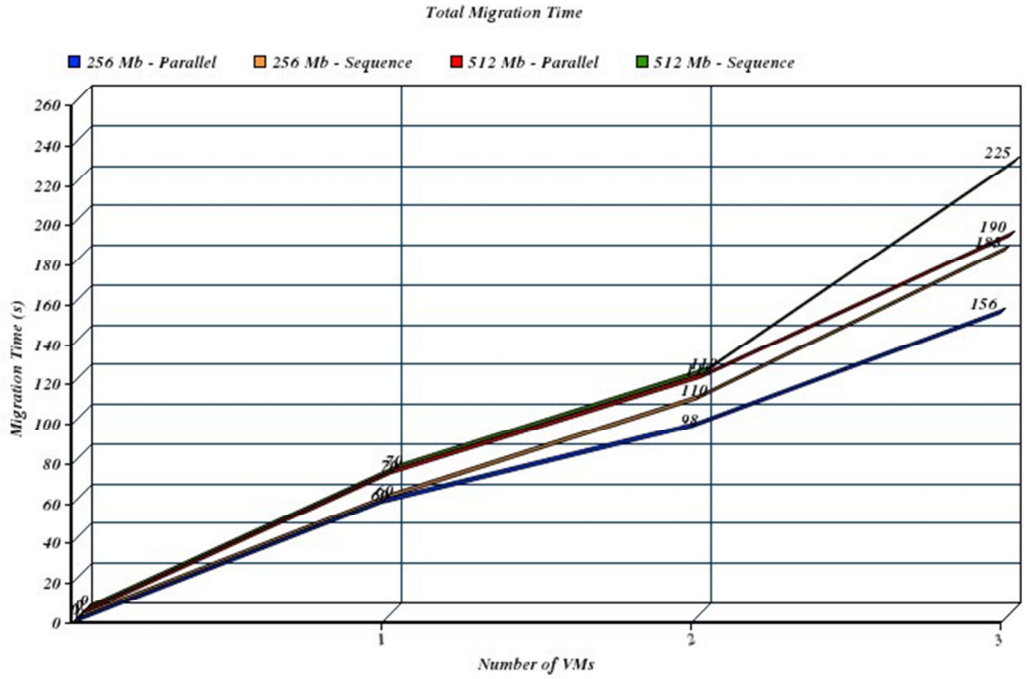


Figure 38 – Comparison graph – Sequential vs Parallel Migration time

Letting VMs juggle between ESXi hosts depending on resource pool load on the Hosts is what DRS does. But since we are looking at security implications of this concept, so if DRS is smart enough to relocate VMs if the resource pool is overworked for one Host and under-worked for other, it most certainly does not understand that if this VM is overworked may it may CPU or Memory, is that due to natural cause or artificial reasons like rogue memory stress tool. If the memory stress tool manages to extend the VMs resources which are acquired from the Host itself, then if DRS kicks in and performs Live migration, then it pre-medicated by the attacker who is waiting on the network to sniffed out the data transfer due to vMotion and/or exploit virtual machine files by injecting unsuspecting code like botnet, etc.

As an end of the testing for dynamic allocation, I exploited the live migration using MiTM attack. The results were similar except that due to increased migration time, the range for DRS being activated was narrowed down and took place early in the cycle. One of the other instance in which I started the network packet generation late in the cycle after migration began, DRS failed and VM went in inactive state which told me that vMotion failed due to extreme network conditions. The destination hypervisor which was bombarded with TCP packets lost connection with vCenter and VM migration which was at 80% migration state, could not recover using DRS at this stage.

Chapter 7

Real-World Analysis

From the beginning of the 21st century, more companies and businesses have started switching over to virtualization technology but also since the time the modern virtualization system as existed, vulnerabilities have been around. As per the statistics, 40% of the reported vulnerabilities and threats to the virtualized environments have high severity because these setups have been fairly easy to exploit and resulting in open-ended attacks that allow attackers to gain complete control. The result of all the reported vulnerabilities to virtualized systems over past 10 years indicates a growing trend in attacks to servers as well as workstations. Most of these attacks have been accounted for in production systems which run on hypervisors while workstations which are on top of host OS.

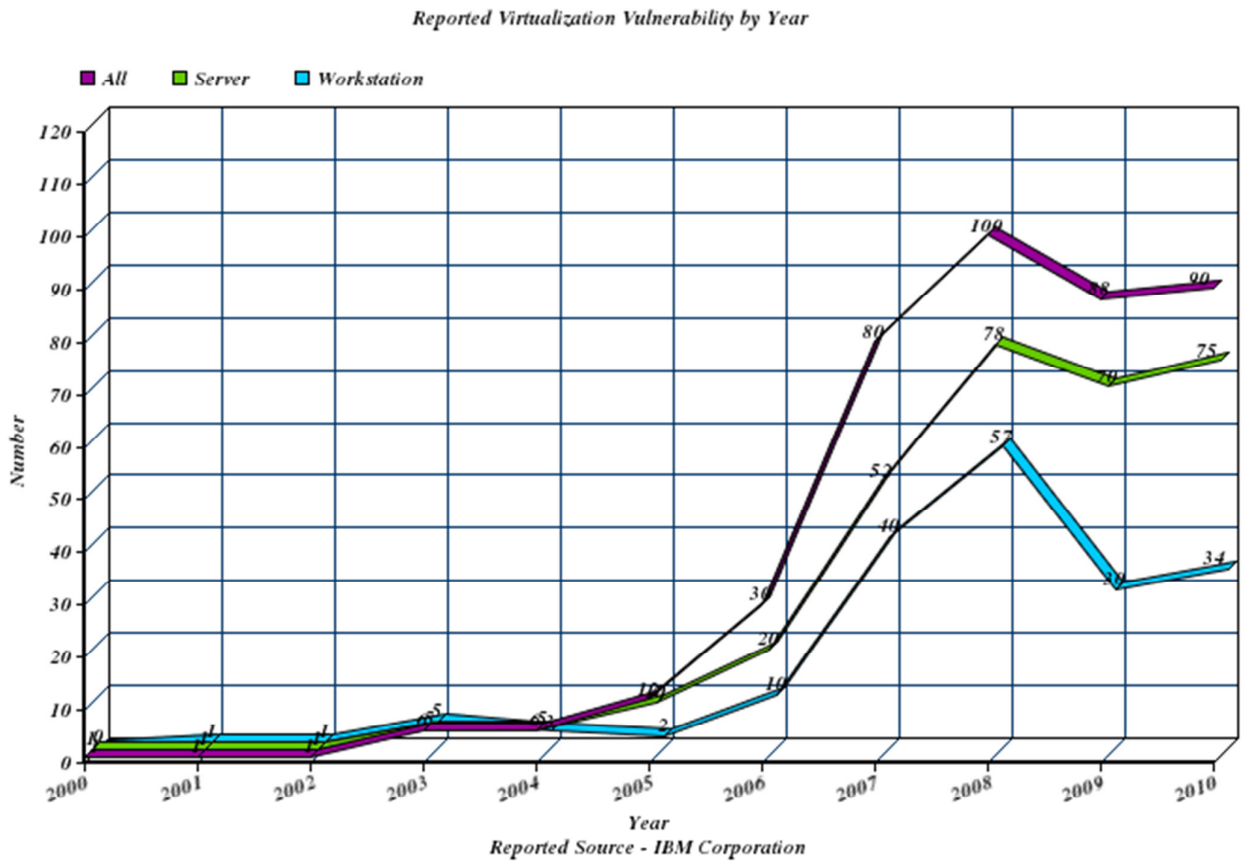


Figure 39. Reported Virtualization Vulnerability by Year (2000-2010)

Unlike non-virtualized systems in which vulnerabilities affect users, hardware and operating system, vulnerabilities in virtualized systems are more widespread and hence categorized in different classes.

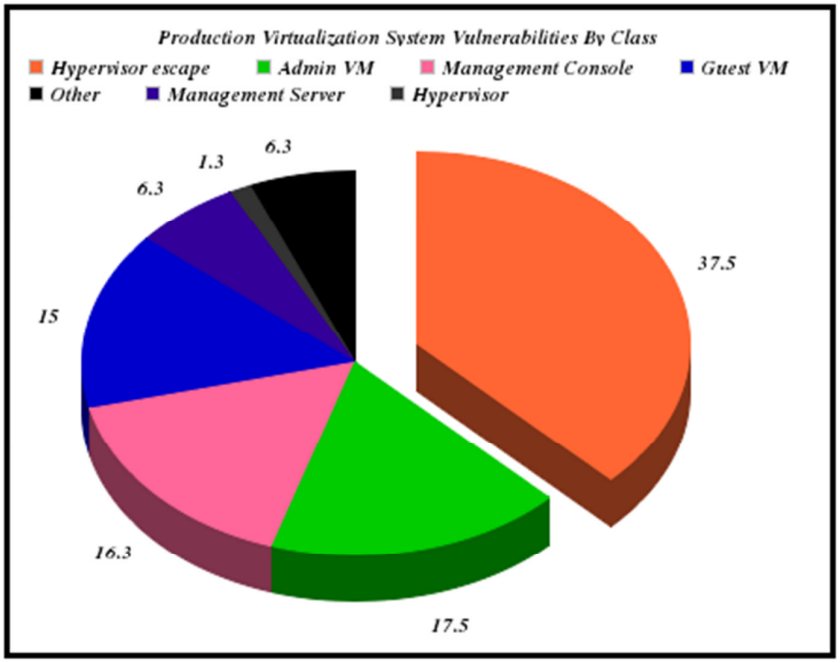


Figure 40. Production Virtualization System Vulnerabilities by Class

Management console vulnerabilities
- Affect the management console host
- Can provide platform or information allowing attac k of management server
- Can occur in custom consoles or web applications

Management server vulnerabilities
- Potential to compromise virtualization system configuration
- Can provide platform from which to attack administrative VM

Administrative VM vulnerabilities
- Compromises system configuration
- In some systems (like Xen), equivalent to a hypervisor vulnerability in that all guest VMs may be compromised
- Can provide platform from which to attack hypervisor and guest VMs

Guest VM vulnerabilities
- Affect a single VM
- Can provide platform from which to attack administrative VM, hypervisor, and other guest VMs

Hypervisor vulnerabilities
- Compromise all guest VMs
- Cannot be exploited from guest VMs

Hypervisor escape vulnerabilities
- A type of hypervisor vulnerability
- Classified separately because of their importance
- Allow a guest VM user to "escape" from own VM to attack other VMs or hypervisor
- Violate assumption of isolation of guest VMs

The above results have been derived from research conducted by IBM Corporation. There are newer evolving virtualization system-specific attacks like VM jumping (using hypervisor escape attacker is able to jump from one victim VM to other), attacks on VM during deployment which lead to control of those VMs, rogue code injection into the VMs file structure and Live VM migration attack using MiTM attack class, as described and experimented for this paper. Lately, there is a new virtualization attack class called as Hyperjacking which involves deployment of a rogue hypervisor resulting in attacker assuming control of the Host OS. There are tools that have developed for Hyperjacking but used only in testing purposes.

Chapter 8

Conclusion

This paper has demonstrated how one of the most popular and widely deployed VMM - VMware is vulnerable to practical attacks targeting their live migration functionality. These threats are cause for concern and require that appropriate solutions be applied to each class of live migration threats. This paper is in no way a criticism of virtualization technology especially VMWare. I personally am a follower and user of VMWare technologies and this paper is research in progress of security around virtual technologies which should not be taken for granted.

In order to make sure that live migration of virtual machines is secure, there needs to be an authentication algorithm that protects the communication pane not only between the source and destination VMMs but also the management servers and agents. The administrator should have access to security policies that controls effective migration of privileges that are assigned to various players involved during the course of migration. The so-called tunnel used by migration must be secure and have policies setup so as to detect sniffing and manipulation of the data or migration state during the migration phase. This can be done by making sure the vMotion parameter is encrypted correctly which at the moment seems to be in a state of testing or needs extensive add-ons and monitoring.

One important thing that can be established is setting up of separate virtual switches for vMotion that is kept aloof from other network objects and tasks. Also vMotion takes place in cleartext by default and VMWare encryption for vMotion is not full-proof as evident from the tests conducted; administrators need to test vMotion before the production servers go online. Administrators can also try to incorporate tested encryption algorithms for vMotion to work securely. A timely audit of these parameters is necessary so as to reduce chances of breach that can occur during the migration.

Having mentioned earlier in my report, this research and analysis is no way a criticism or deterrent to use vMotion in production environment. Over the past decade, VMWare has invested considerably towards their hypervisor security with introduction of VMSafe and VShield that provides APIs for partnering companies (mainly security firms) to develop patches for the virtualized system. These patches include monitors for VM's memory, virtual disk and/or drivers.

Since I have been able to successfully replicate the exploits as performed by experts and understand the performance-related results, where do we stand after this and what should be done for future?

Management of virtual infrastructure should be done with a strict code of conduct. Security policies should be re-visited. There should a close monitor for any changes to the virtualized systems as while it is easy to execute changes, it is difficult to manage them. Any exploit like stolen guest VM can be used someplace else and if for instance it a database server, the nothing can stop the thief from stealing data. Furthermore, it is possible to look into limiting VM intervention by focusing on availability such that the application running on the VM is not affected due to intervention and data is not compromised. So basically to avoid integrity and confidentiality loss, we can focus on loss of availability.

As any administrator can understand that any vulnerability to network security could result in breach of data and intellectual property but when it comes to VM migration that involves full OS, any compromise to the network can also result in breach of VMM integrity. So to sum it up, the approach towards securing a virtualized network needs certain add-ons to access control and a methodology that assures complete isolation from other network objects.

References

[1] Clark, Christopher; Fraser, Keir; Hand, Steven; Hansen, Jacob; Jul, Eric; Limpach, Christian; Pratt, Ian; Warfield; Andrew. (2005). Live Migration of Virtual Machines. In *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2.* http://portal.acm.org/citation.cfm?id=1251223

[2] Emenekar, Wesley, Stanzione, D. (2007). Dynamic Virtual Clustering. In *Proceedings of Cluster Computing, 2007 IEEE International Conference.* http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4629220

[3] McNett, Marvin; Gupta, Diwaker; Vahdat, Amin; Voelker, Geoffrey (2007). An Extensible Framework for Managing Clusters of Virtual Machines. In *Proceedings of the 21st Large Installation System Administration Conference (LISA '07)*. http:// http://portal.acm.org/citation.cfm?id=1349426.1349440

[4] Sotomayor, Borja; Keahey, Kate; Foster, Ian (2006). Overhead Matters: A Model for Virtual Resource Management. In *Proceedings of the 2nd International Workshop on Virtualization Technology in Distributed Computing IEEE Computer Society Washington, DC, USA ©2006.* http://portal.acm.org/citation.cfm?id=1308175.1308350

[5] Dinda, Bin Lin. (2005, November). Mixing Batch and Interactive Virtual Machines Using Periodic real-time Scheduling. In *Proceedings of the ACM/IEEE SC 2005 Conference*. http://ieeexplore.ieee.org/Xplore/login.jsp?url=http://ieeexplore.ieee.org/iel5/10435/33129/01559960.pdf?arnumber=1559960&authDecision=-203

[6] David, Irwin; Chase, Jeff; Grit, Laura; Yumerefendi, Aydan; Becker, David; Yocum, Ken (2006). Sharing networked resources with brokered leases. In *Proceedings ATEC '06 Proceedings of the annual conference on USENIX '06 Annual Technical Conference USENIX Association Berkeley, CA, USA ©2006.* http://portal.acm.org/citation.cfm?id=1267377

[7] Braastad, Espen. (2006, May 22). Management of high availability services using virtualization. http://www.citeulike.org/user/baurm/article/848349

[8] Oberheide, Jon; Cooke, Evan; Jahanian, Farnam (2008, March 24). Empirical Exploitation of Live Virtual Machine Migration. http://www.eecs.umich.edu/fjgroup/pubs/blackhat08-migration.pdf

[9] Venkatesha, Sharath; Sadhu, Shatrugna; Kintali, Sridhar (2009). Survey of Virtual Machine Migration Techniques. http://cs.ucsb.edu/~ssadhu/papers/vmmt.pdf

[10] Williams, Bryan and Cross, Thomas (2010). Virtualization System Security. http://blogs.iss.net/archive/papers/VirtualizationSecurity.pdf

[11] (2010). Openfiler 2.3 [Computer]. https://www.openfiler.com/

[12] (2010, November 22). Backtrack 4 R2 [Computer]. http://www.backtrack-linux.org/

[13] (2010). Veeam Reporter 4.0 [Computer]. http://www.veeam.com/

[14] (2010). Veeam Monitor 5.0 [Computer]. http://www.veeam.com/

[15] (2008). VMWare Workstation 6.5.1 (Build 126130)[Computer]. http://www.vmware.com/

[16] (2008). VMWare vSphere 4.0 [Computer]. http://www.vmware.com/

[17] Wood, Timothy; Cherkasova, Ludmila; Ozonat, Kivanc and Shenoy, Prashant. Profiling and Modeling Resource Usage of Virtualized Applications. In *Proceeding Middleware '08 Proceedings of the ACM/IFIP/USENIX 9th International Middleware Conference.* http://lass.cs.umass.edu/papers/pdf/middleware08.pdf

[18] Rutkowska, Joanna. (2008, April 8). Security Challenges in Virtualized Environments. Invisible Things Lab. *RSA Conference, San Francisco, USA (Joanna Rutkowska).* http://www.invisiblethingslab.com/itl/Welcome.html

[19] Cleeff, Andre van.; Pieters, Wolter; Wieringa, Roel. (2009) Security Implications of Virtualization: A Literature Study. In *Proceedings of 2009 International Conference on Computational Science and Engineering*. http://doc.utwente.nl/67484/1/Security_Implications_of_Virtualization.pdf

[20] Social-Engineer.org: Security through Education. http://www.social-engineer.org

[21] Zhao, Ming & Figueiredo, Renato J.(2007). Experimental study of virtual machine migration in support of reservation of cluster resources. *VTDC '07 Proceedings of the 2nd international workshop on Virtualization technology in distributed computing ACM New York, NY, USA ©2007.* http://portal.acm.org/citation.cfm?id=1408659

Appendix - Report

Description

An inventory of VI3 objects with their properties and relations.

Summary

Object	Total
Shared Datastores	1
Datastores	2
Hosts	2
Virtual Machines	5

Operating System	Running	Total
FreeBSD (32-bit)	0	1
Other (32-bit)	1	3
Ubuntu Linux (32-bit)	0	1

1. Datastore to Host

Datastore	Host
Local1	10.0.0.84
Local2	10.0.0.85
VSAN	10.0.0.84; 10.0.0.85

2. Datastore to LUN

Datastore	LUN
Local1	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
Local2	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
VSAN	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435

3. VM to Datastore

VM	Datastore
DSL	VSAN
FreeBSD	VSAN
LLSP	VSAN
Nostalgia	VSAN
Nostalgia2	VSAN

4. VM to LUN

VM	LUN
DSL	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
FreeBSD	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
LLSP	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
Nostalgia	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
Nostalgia2	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435

5. VM to Host

VM	Host
DSL	10.0.0.84
FreeBSD	10.0.0.84
LLSP	10.0.0.85
Nostalgia	10.0.0.84
Nostalgia2	10.0.0.84

6. iSCSI Adapters

Name	vmhba33
Current Speed (MB)	0
Driver	iscsi_vmk
ISCSI Name	iqn.2006-01.com.openfiler:tsn.e5cba8c98c8c
Key	key-vim.host.InternetScsiHba-vmhba33
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vmhba33
Max Speed (MB)	0
Model	iSCSI Software Adapter
Name	vmhba33
PCI	UNKNOWN - NULL PCI DEV IN VMKCTL
Status	unknown
Target Address : Target Port 1	10.0.0.88 : 3260
Type	HostInternetScsiHba

Name	vmhba33
Current Speed (MB)	0
Driver	iscsi_vmk
ISCSI Name	iqn.1998-01.com.vmware:milan-1a41a0cf
Key	key-vim.host.InternetScsiHba-vmhba33
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vmhba33
Max Speed (MB)	0
Model	iSCSI Software Adapter
Name	vmhba33
PCI	UNKNOWN - NULL PCI DEV IN VMKCTL
Status	unknown
Target Address : Target Port 1	10.0.0.88 : 3260
Type	HostInternetScsiHba

7. SCSI Adapters

Name	vmhba1
Bus	0
Driver	mptspi
Key	key-vim.host.ParallelScsiHba-vmhba1
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vmhba1
Model	53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI
Name	vmhba1
PCI	00:10.0
Status	unknown
Type	HostParallelScsiHba

Name	vmhba1
Bus	0
Driver	mptspi
Key	key-vim.host.ParallelScsiHba-vmhba1
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vmhba1
Model	53c1030 PCI-X Fusion-MPT Dual Ultra320 SCSI
Name	vmhba1
PCI	00:10.0
Status	unknown
Type	HostParallelScsiHba

8. LUNs

Name	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
Block	419430400
Block Size	512
Capacity	200 GB
Device Name	/vmfs/devices/disks/mpx.vmhba1:C0:T0:L0
Device Type	disk
Host	10.0.0.84
Key	key-vim.host.ScsiDisk-0000000000766d686261313a303a30
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>mpx.vmhba1:C0:T0:L0
LUN Type	disk
Model	VMware Virtual S
Name	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
Revision	1.0
SCSI Level	2
SCSI Level Specified	True
Serial Number	unavailable
Type	VimApi.HostScsiDisk
Uuid	0000000000766d686261313a303a30
Vendor	VMware,

Name	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
Block	410779648
Block Size	512
Capacity	195.875 GB
Device Name	/vmfs/devices/disks/t10.F405E46494C45400371363B66337D235A46676D2D4748435
Device Type	disk
Host	10.0.0.84
Key	key-vim.host.ScsiDisk-0100000007331366b36732d534a66672d4d474853564952545541
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>t10.F405E46494C45400371363B66337D235A46676D2D4748435
LUN Type	disk
Model	VIRTUAL-DISK
Name	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
Revision	0
SCSI Level	4
SCSI Level Specified	True

Serial Number	unavailable
Type	VimApi.HostScsiDisk
Uuid	01000000007331366b36732d534a66672d4d474853564952545541
Vendor	OPNFILER

Name	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
Block	419430400
Block Size	512
Capacity	200 GB
Device Name	/vmfs/devices/disks/mpx.vmhba1:C0:T0:L0
Device Type	disk
Host	10.0.0.85
Key	key-vim.host.ScsiDisk-0000000000766d686261313a303a30
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>mpx.vmhba1:C0:T0:L0
LUN Type	disk
Model	VMware Virtual S
Name	VMware,:VMware Virtual S:mpx.vmhba1:C0:T0:L0
Revision	1.0
SCSI Level	2
SCSI Level Specified	True
Serial Number	unavailable
Type	VimApi.HostScsiDisk
Uuid	0000000000766d686261313a303a30
Vendor	VMware,

Name	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2D4748435
Block	410779648
Block Size	512
Capacity	195.875 GB
Device Name	/vmfs/devices/disks/t10.F405E46494C45400371363B66337D235A46676D2D4748435
Device Type	disk
Host	10.0.0.85
Key	key-vim.host.ScsiDisk-01000000007331366b36732d534a66672d4d474853564952545541
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>t10.F405E46494C45400371363B66337D235A46676D2D4748435
LUN Type	disk
Model	VIRTUAL-DISK
Name	OPNFILER:VIRTUAL-DISK:t10.F405E46494C45400371363B66337D235A46676D2

	D4748435
Revision	0
SCSI Level	4
SCSI Level Specified	True
Serial Number	unavailable
Type	VimApi.HostScsiDisk
Uuid	01000000007331366b36732d534a66672d4d474853564952545541
Vendor	OPNFILER

9. Shared Datastores

Name	VSAN
Accessible	True
Capacity	195.75 GB
Capacity (GB) numeric	195.75
Free Space	163.054 GB
Free Space (GB) numeric	163.054
FS block size	1 MB
FS type	VMFS
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>VSAN
Managed Object Reference	datastore-213
Multiple Host Access	true
Name	VSAN
Summary	sanfs://vmfs_uuid:4d021938-1cb133cb-32a8-000c29542e7f/
Type	Datastore
VMCount	5

10. Datastores

Name	Local1
Accessible	True
Capacity	195 GB
Capacity (GB) numeric	195
Free Space	194.451 GB
Free Space (GB) numeric	194.451
FS block size	1 MB
FS type	VMFS
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>Local1
Managed Object Reference	datastore-208
Multiple Host Access	false
Name	Local1

Summary	sanfs://vmfs_uuid:4d00dc67-399ae1b8-fb9d-000c29542e7f/
Type	Datastore

Name	Local2
Accessible	True
Capacity	195 GB
Capacity (GB) numeric	195
Free Space	194.451 GB
Free Space (GB) numeric	194.451
FS block size	1 MB
FS type	VMFS
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>Local2
Managed Object Reference	datastore-220
Multiple Host Access	false
Name	Local2
Summary	sanfs://vmfs_uuid:4d00d7f6-3c89bcfc-8617-000c294735d3/
Type	Datastore

11. Clusters

Name	Melvin
Current balance	-1000
Current failover level	-1
DRS	disabled
Effective cpu (in MHz)	2712
Effective memory (in MB)	1306
HA	disabled
Location Path	>10.0.0.10>Thesis>Melvin
Managed Object Reference	domain-c141
Name	Melvin
Number of cpu cores	2
Number of cpu threads	2
Number of effective hosts	2
Number of hosts	2
Parent	Thesis
Status	green
Target DRS balance	-1000
Total cpu (in MHz)	4940
Total memory (in bytes)	5361614848
Type	ClusterComputeResource
Vmotions number	41

12. Hosts

Name	10.0.0.84
Cluster	Melvin
Connection State	connected
Cpu core frequency (in MHz)	2537
CPU overall usage	1224
CPU Sockets	1
Datacenter	Thesis
Hyperthreading	False
License expiration date	3/10/2011 2:08:27 AM
License feature 1 Name	vmotion
License feature 2 Name	esxHost
License used units	1
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84
Managed Object Reference	host-205
Manufacturer	VMware, Inc.
Memory overall usage	820
Memory Size	2.497 GB
Memory Size, GB	2.497
Model	VMware Virtual Platform
Name	10.0.0.84
Parent	Melvin
Power State	poweredOn
Processor cores	1
Processor type	Intel(R) Core(TM)2 Duo CPU P8700 @ 2.53GHz
Processors	1
Product	VMware ESXi 4.0.0 build-171294
Runtime Info boot time	1/20/2011 2:16:20 AM
Type	HostSystem

Name	10.0.0.85
Cluster	Melvin
Connection State	connected
Cpu core frequency (in MHz)	2403
CPU overall usage	141
CPU Sockets	1
Datacenter	Thesis
Hyperthreading	False
License expiration date	2/26/2011 6:40:59 PM
License feature 1 Name	vmotion
License feature 2 Name	esxHost
License used units	1
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85

Managed Object Reference	host-217
Manufacturer	VMware, Inc.
Memory overall usage	744
Memory Size	2.497 GB
Memory Size, GB	2.497
Model	VMware Virtual Platform
Name	10.0.0.85
Parent	Melvin
Power State	poweredOn
Processor cores	1
Processor type	Intel(R) Core(TM) i3 CPU M 370 @ 2.40GHz
Processors	1
Product	VMware ESXi 4.0.0 build-171294
Runtime Info boot time	1/19/2011 2:11:33 PM
Type	HostSystem

13. Resource Pools

Name	Resources
ConfiguredMemoryMB	1207959552
Cpu expandable	True
Cpu level	normal
Cpu limit	2712
Cpu reservation	2712
Cpu shares	4000
LastModified	Not set
Location Path	>10.0.0.10>Thesis>Melvin>Resources
Managed Object Reference	resgroup-143
Memory expandable	True
Memory level	normal
Memory limit	1318
Memory reservation	1318
Memory shares	163840
Name	Resources
OverallStatus	green
Parent	Melvin
SummaryName	Not set
Type	ResourcePool

14. Virtual Switches

Name	vSwitch0

Active Adapters	vmnic0
Forget Transmits	True
Host	10.0.0.84
Link discovery protocol	cdp
Link discovery protocol operation	listen
Load Balancing	Route based on the source of the port ID
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vSwitch0
MAC Address Changes	True
Name	vSwitch0
Network Failure Detection	False
Notify Switches	True
Number of ports	56
Promiscuous Mode	False
Type	HostVirtualSwitch

Name	vSwitch0
Active Adapters	vmnic0
Forget Transmits	True
Host	10.0.0.85
Link discovery protocol	cdp
Link discovery protocol operation	listen
Load Balancing	Route based on the source of the port ID
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vSwitch0
MAC Address Changes	True
Name	vSwitch0
Network Failure Detection	False
Notify Switches	True
Number of ports	56
Promiscuous Mode	False
Type	HostVirtualSwitch

15. Physical Adapters

Name	vmnic0
Connection State	Connected
DHCP	False
Driver	e1000
Duplex	full
Host	10.0.0.84
IP Hints	;vlan: 0, subnet: 10.0.0.8-10.0.0.15
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vmnic0
Name	vmnic0

SpeedMb	1000
Type	PhysicalNic

Name	vmnic0
Connection State	Connected
DHCP	False
Driver	e1000
Duplex	full
Host	10.0.0.85
IP Hints	;vlan: 0, subnet: 10.0.0.8-10.0.0.15
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vmnic0
Name	vmnic0
SpeedMb	1000
Type	PhysicalNic

16. VM Kernels

Name	vmk0
Host	10.0.0.84
IP Address	10.0.0.84
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vmk0
MAC Address	00:0c:29:54:2e:7f
Name	vmk0
Subnet Mask	255.255.255.0
Type	HostVirtualNic

Name	vmk1
Host	10.0.0.84
IP Address	10.0.0.86
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>vmk1
MAC Address	00:50:56:75:c2:2d
Name	vmk1
Subnet Mask	255.255.255.0
Type	HostVirtualNic

Name	vmk0
Host	10.0.0.85
IP Address	10.0.0.85

Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vmk0
MAC Address	00:0c:29:47:35:d3
Name	vmk0
Subnet Mask	255.255.255.0
Type	HostVirtualNic

Name	vmk1
Host	10.0.0.85
IP Address	10.0.0.87
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>vmk1
MAC Address	00:50:56:7f:cd:88
Name	vmk1
Subnet Mask	255.255.255.0
Type	HostVirtualNic

17. Networks

Name	VM Network
Host	10.0.0.84
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>VM Network
Managed Object Reference	network-124
Name	VM Network
Type	HostPortGroup
VLANID	0

Name	VMkernel
Host	10.0.0.84
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>VMkernel
Name	VMkernel
Type	HostPortGroup
VLANID	0

Name	Management Network
Active Adapters	vmnic0
Host	10.0.0.84
Load Balancing	Route based on the source of the port ID
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>Management Network
Name	Management Network
Network Failure Detection	False

Notify Switches	True
Type	HostPortGroup
VLANID	0

Name	VM Network
Host	10.0.0.85
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>VM Network
Managed Object Reference	network-124
Name	VM Network
Type	HostPortGroup
VLANID	0

Name	VMkernel
Host	10.0.0.85
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>VMkernel
Name	VMkernel
Type	HostPortGroup
VLANID	0

Name	Management Network
Active Adapters	vmnic0
Host	10.0.0.85
Load Balancing	Route based on the source of the port ID
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>Management Network
Name	Management Network
Network Failure Detection	False
Notify Switches	True
Type	HostPortGroup
VLANID	0

18. Virtual Machines

Name	FreeBSD
Alternate guest name	FreeBSD (32-bit)
Annotation	Not set
Boot delay	0
Capability cpu feature mask supported	True
Capability is boot options supported	True
Capability is disabling snapshots supported	False
Capability is disk shares supported	True

Capability is lock snapshots supported	False
Capability is memory snapshots supported	True
Capability is multiple snapshots supported	True
Capability is npiv wwn on non rdm vm supported	True
Capability is powered off snapshots supported	True
Capability is quiesced snapshots supported	True
Capability is record replay supported	True
Capability is revert to snapshot supported	True
Capability is s1Acpi management supported	True
Capability is setting display topology supported	False
Capability is setting screen resolution supported	False
Capability is setting video ram size supported	True
Capability is snapshot config supported	True
Capability is snapshot operations supported	True
Capability is swap placement supported	True
Capability is tools auto update supported	False
Capability is tools sync time supported	True
Capability is virtual mmu usage supported	True
Capability is vm npiv wwn disable supported	True
Capability is vm npiv wwn supported	True
Capability is vm npiv wwn update supported	True
CD/DVD drive 1 label	CD/DVD Drive 1
CD/DVD drive 1 summary	ISO [VSAN] ISOs/FreeBSD-7.3-RELEASE-i386-livefs.iso
CDROM Mounted	False
Change Tracking Enabled	False
ChangeVersion	1/19/2011 8:53:29 PM
Cluster	Melvin
Committed 1	8590277234
Connection State	connected
CPU	1
Cpu expandable	False
Cpu level	normal
Cpu limit	Not set
Cpu reservation	Not set
Cpu shares	1000
Default power off operation type	soft
Default reset type	soft
Default suspend type	hard
Disable acceleration	False
Disk uuid enabled	False
Enable logging	True
Guest id	Not set
Guest OS	FreeBSD (32-bit)
Hardware virtualization for execution usage	hvAuto
Has snapshot(s)	No

Host	10.0.0.84
Hostname	Not set
Hot plug memory increment size	0
Hot plug memory limit	0
Ht sharing	any
Instance UUID	526ffa43-7f9f-1027-735a-96b61e99b490
Ip address	Not set
Is cpu hot add enabled	False
Is cpu hot remove enabled	False
Is question raised	Not set
Last modified	1/1/1970 12:00:00 AM
List of datastore name and urls	name: VSAN, url: /vmfs/volumes/4d021938-1cb133cb-32a8-000c29542e7f
Location	564d73ca-ce99-3472-b49d-da4ee80d5904
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>FreeBSD
Log directory	[VSAN] FreeBSD/
Managed Object Reference	vm-277
Memory (MB)	256
Memory expandable	False
Memory hot add enabled	False
Memory level	normal
Memory limit	Not set
Memory reservation	Not set
Memory shares	2560
Monitor type	release
Name	FreeBSD
Network Adapter 1 Label	Network adapter 1
Network Adapter 1 MAC	00:50:56:ad:56:71
Network Adapter 1 Summary	VM Network
Notes	Not set
NPIV desired node WWNs	0
NPIV desired port Wwns	0
NPIV is temporary disabled	True
NPIV node World Wide Name	Not set
NPIV on non rdm disks	False
NPIV port world wide name	Not set
NPIV world wide name type	Not set
Number of cpu	1
Number of ethernet cards	1
Number of virtual disks	1
OverallStatus	green
Parent	Not set
Power off operation type	preset
Power State	poweredOff
Record replay enabled	False

Record replay state	inactive
Reset type	preset
Run with debug info	False
Runtime Info boot time	Not set
Runtime Info clean power off	False
Runtime Info fault tolerance state	notConfigured
Runtime info max cpu usage	1393
Runtime info max memory usage	256
Runtime Info memory overhead	114311168
Runtime Info need secondary reason	Not set
Runtime Info number of active MKS connections	0
SCSI bus sharing 1	noSharing
SCSI controller 1	LSI Logic
SCSI controller name 1	SCSI controller 0
Snapshot directory	[VSAN] FreeBSD/
Snapshot power off behavior	powerOff
Snapshot tree locked	False
Snapshots disabled	False
Standby action	powerOnSuspend
Suspend directory	[VSAN] FreeBSD/
Suspend interval	0
Suspend operation type	preset
Suspend time	Not set
Swap placement	inherit
Template	False
Tools installer mounted	False
Tools status	guestToolsNotRunning
Tools version status	guestToolsNotInstalled
Type	VirtualMachine
Uncommitted 1	268435456
Unshared 1	8590277234
Use TCP/IP Offloading	False
UUID	422d9e50-0500-44f6-8202-99643638e2bb
Version	vmx-07
Virtual Disk 1 Disk Mode	persistent
Virtual Disk 1 Is split	No
Virtual Disk 1 Is thin provisioned	No
Virtual Disk 1 Is write through	No
Virtual Disk 1 Label	Hard disk 1
Virtual Disk 1 Summary	8388608 KB
Virtual Disk 1 Summary, GB	8
Virtual MMU usage	automatic
Vm config file	[VSAN] FreeBSD/FreeBSD.vmx
VMCapability is change tracking supported	True
VMCapability is console preferences supported	False

Vmdk File 1	[VSAN] FreeBSD/FreeBSD.vmdk
VMTools after power on	True
VMTools after resume	True
VMTools before guest reboot	False
VMTools before guest shutdown	True
VMTools before guest standby	True
VMTools pending customization	Not set
VMTools sync time with host	False
VMTools tools version	0
VMtools upgrade policy	manual
VMWare Tools	tools not installed
Vmx File	[VSAN] FreeBSD/FreeBSD.vmx
Vmx File Short	FreeBSD.vmx
Will enter BIOS setup during next boot	False

Name	Nostalgia2
Alternate guest name	Not set
Annotation	Nostalgia contains a great collection of ancient DOS Games, ready to play! Now you can begin to waste your time once again.
Boot delay	0
Capability cpu feature mask supported	True
Capability is boot options supported	True
Capability is disabling snapshots supported	False
Capability is disk shares supported	True
Capability is lock snapshots supported	False
Capability is memory snapshots supported	True
Capability is multiple snapshots supported	True
Capability is npiv wwn on non rdm vm supported	False
Capability is powered off snapshots supported	True
Capability is quiesced snapshots supported	True
Capability is record replay supported	True
Capability is revert to snapshot supported	True
Capability is s1Acpi management supported	True
Capability is setting display topology supported	False
Capability is setting screen resolution supported	False
Capability is setting video ram size supported	True
Capability is snapshot config supported	True
Capability is snapshot operations supported	True
Capability is swap placement supported	True
Capability is tools auto update supported	False
Capability is tools sync time supported	True
Capability is virtual mmu usage supported	True

Capability is vm npiv wwn disable supported	False
Capability is vm npiv wwn supported	True
Capability is vm npiv wwn update supported	False
CD/DVD drive 1 label	CD/DVD Drive 1
CD/DVD drive 1 summary	ATAPI CDROM 0
CDROM Mounted	False
Change Tracking Enabled	False
ChangeVersion	1/19/2011 8:53:40 PM
Cluster	Melvin
Committed 1	107656118
Connection State	connected
CPU	1
Cpu expandable	False
Cpu level	normal
Cpu limit	Not set
Cpu reservation	Not set
Cpu shares	1000
Default power off operation type	soft
Default reset type	soft
Default suspend type	hard
Disable acceleration	False
Disk uuid enabled	False
Enable logging	True
Guest id	Not set
Guest OS	Other (32-bit)
Hardware virtualization for execution usage	hvAuto
Has snapshot(s)	No
Host	10.0.0.84
Hostname	Not set
Hot plug memory increment size	0
Hot plug memory limit	0
Ht sharing	any
Instance UUID	502d6bb1-0bfe-ceff-3531-d5432fcfb2a6
Ip address	Not set
Is cpu hot add enabled	False
Is cpu hot remove enabled	False
Is question raised	Not set
Last modified	1/1/1970 12:00:00 AM
List of datastore name and urls	name: VSAN, url: /vmfs/volumes/4d021938-1cb133cb-32a8-000c29542e7f
Location	564d69bb-f70f-2652-da91-30f661e6efca
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>Nostalgia2
Log directory	[VSAN] Nostalgia2/
Managed Object Reference	vm-294
Memory (MB)	64

Memory expandable	False
Memory hot add enabled	False
Memory level	normal
Memory limit	Not set
Memory reservation	Not set
Memory shares	640
Monitor type	release
Name	Nostalgia2
Network Adapter 1 Label	Network adapter 1
Network Adapter 1 MAC	00:50:56:ad:76:c8
Network Adapter 1 Summary	VM Network
Notes	Nostalgia contains a great collection of ancient DOS Games, ready to play! Now you can begin to waste your time once again.
NPIV desired node WWNs	0
NPIV desired port Wwns	0
NPIV is temporary disabled	False
NPIV node World Wide Name	Not set
NPIV on non rdm disks	False
NPIV port world wide name	Not set
NPIV world wide name type	Not set
Number of cpu	1
Number of ethernet cards	1
Number of virtual disks	1
OverallStatus	green
Parent	Not set
Power off operation type	soft
Power State	poweredOff
Record replay enabled	False
Record replay state	inactive
Reset type	soft
Run with debug info	False
Runtime Info boot time	Not set
Runtime Info clean power off	False
Runtime Info fault tolerance state	notConfigured
Runtime info max cpu usage	1393
Runtime info max memory usage	64
Runtime Info memory overhead	111149056
Runtime Info need secondary reason	Not set
Runtime Info number of active MKS connections	0
SCSI bus sharing 1	noSharing
SCSI controller 1	BusLogic
SCSI controller name 1	SCSI controller 0
Snapshot directory	[VSAN] Nostalgia2/

Snapshot power off behavior	powerOff
Snapshot tree locked	False
Snapshots disabled	False
Standby action	checkpoint
Suspend directory	[VSAN] Nostalgia2/
Suspend interval	0
Suspend operation type	hard
Suspend time	Not set
Swap placement	inherit
Template	False
Tools installer mounted	False
Tools status	guestToolsNotRunning
Tools version status	guestToolsNotInstalled
Type	VirtualMachine
Uncommitted 1	67108864
Unshared 1	107656118
Use TCP/IP Offloading	False
UUID	564d69bb-f70f-2652-da91-30f661e6efca
Version	vmx-04
Virtual Disk 1 Disk Mode	persistent
Virtual Disk 1 Is split	No
Virtual Disk 1 Is thin provisioned	No
Virtual Disk 1 Is write through	No
Virtual Disk 1 Label	Hard disk 1
Virtual Disk 1 Summary	104858 KB
Virtual Disk 1 Summary, GB	0.1
Virtual MMU usage	automatic
Vm config file	[VSAN] Nostalgia2/Nostalgia.vmx
VMCapability is change tracking supported	False
VMCapability is console preferences supported	False
Vmdk File 1	[VSAN] Nostalgia2/Nostalgia.vmdk
VMTools after power on	True
VMTools after resume	True
VMTools before guest reboot	False
VMTools before guest shutdown	True
VMTools before guest standby	True
VMTools pending customization	Not set
VMTools sync time with host	False
VMTools tools version	0
VMtools upgrade policy	manual
VMWare Tools	tools not installed
Vmx File	[VSAN] Nostalgia2/Nostalgia.vmx
Vmx File Short	Nostalgia.vmx
Will enter BIOS setup during next boot	False

Name	Nostalgia
Alternate guest name	Not set
Annotation	Nostalgia contains a great collection of ancient DOS Games, ready to play! Now you can begin to waste your time once again.
Boot delay	0
Capability cpu feature mask supported	True
Capability is boot options supported	True
Capability is disabling snapshots supported	False
Capability is disk shares supported	True
Capability is lock snapshots supported	False
Capability is memory snapshots supported	True
Capability is multiple snapshots supported	True
Capability is npiv wwn on non rdm vm supported	False
Capability is powered off snapshots supported	True
Capability is quiesced snapshots supported	True
Capability is record replay supported	True
Capability is revert to snapshot supported	True
Capability is s1Acpi management supported	True
Capability is setting display topology supported	False
Capability is setting screen resolution supported	False
Capability is setting video ram size supported	True
Capability is snapshot config supported	True
Capability is snapshot operations supported	True
Capability is swap placement supported	True
Capability is tools auto update supported	False
Capability is tools sync time supported	True
Capability is virtual mmu usage supported	True
Capability is vm npiv wwn disable supported	False
Capability is vm npiv wwn supported	True
Capability is vm npiv wwn update supported	False
CD/DVD drive 1 label	CD/DVD Drive 1
CD/DVD drive 1 summary	ATAPI CDROM 0
CDROM Mounted	False
Change Tracking Enabled	False
ChangeVersion	1/19/2011 10:14:20 PM
Cluster	Melvin
Committed 1	107618581
Connection State	connected
CPU	1
Cpu expandable	False
Cpu level	normal
Cpu limit	Not set
Cpu reservation	Not set

Cpu shares	1000
Default power off operation type	soft
Default reset type	soft
Default suspend type	hard
Disable acceleration	False
Disk uuid enabled	False
Enable logging	True
Guest id	Not set
Guest OS	Other (32-bit)
Hardware virtualization for execution usage	hvAuto
Has snapshot(s)	No
Host	10.0.0.84
Hostname	Not set
Hot plug memory increment size	0
Hot plug memory limit	0
Ht sharing	any
Instance UUID	526ed5f0-0059-c0ac-0de0-5695547d63b6
Ip address	Not set
Is cpu hot add enabled	False
Is cpu hot remove enabled	False
Is question raised	Not set
Last modified	1/1/1970 12:00:00 AM
List of datastore name and urls	name: VSAN, url: /vmfs/volumes/4d021938-1cb133cb-32a8-000c29542e7f
Location	564dc808-69cc-a534-400a-ed9edd1d93d0
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>Nostalgia
Log directory	[VSAN] Nostalgia/
Managed Object Reference	vm-303
Memory (MB)	64
Memory expandable	False
Memory hot add enabled	False
Memory level	normal
Memory limit	Not set
Memory reservation	Not set
Memory shares	640
Monitor type	release
Name	Nostalgia
Network Adapter 1 Label	Network adapter 1
Network Adapter 1 MAC	00:50:56:ad:10:3c
Network Adapter 1 Summary	VM Network
Notes	Nostalgia contains a great collection of ancient DOS Games, ready to play! Now you can begin to waste your time once again.
NPIV desired node WWNs	0
NPIV desired port Wwns	0

NPIV is temporary disabled	False
NPIV node World Wide Name	Not set
NPIV on non rdm disks	False
NPIV port world wide name	Not set
NPIV world wide name type	Not set
Number of cpu	1
Number of ethernet cards	1
Number of virtual disks	1
OverallStatus	green
Parent	Not set
Power off operation type	soft
Power State	poweredOff
Record replay enabled	False
Record replay state	inactive
Reset type	soft
Run with debug info	False
Runtime Info boot time	Not set
Runtime Info clean power off	False
Runtime Info fault tolerance state	notConfigured
Runtime info max cpu usage	1319
Runtime info max memory usage	64
Runtime Info memory overhead	111149056
Runtime Info need secondary reason	Not set
Runtime Info number of active MKS connections	0
SCSI bus sharing 1	noSharing
SCSI controller 1	BusLogic
SCSI controller name 1	SCSI controller 0
Snapshot directory	[VSAN] Nostalgia/
Snapshot power off behavior	powerOff
Snapshot tree locked	False
Snapshots disabled	False
Standby action	checkpoint
Suspend directory	[VSAN] Nostalgia/
Suspend interval	0
Suspend operation type	hard
Suspend time	Not set
Swap placement	inherit
Template	False
Tools installer mounted	False
Tools status	guestToolsNotRunning
Tools version status	guestToolsNotInstalled
Type	VirtualMachine
Uncommitted 1	67108864
Unshared 1	107618581
Use TCP/IP Offloading	False

UUID	564d7779-6174-6a0c-e594-9c84443f9d51
Version	vmx-04
Virtual Disk 1 Disk Mode	persistent
Virtual Disk 1 Is split	No
Virtual Disk 1 Is thin provisioned	No
Virtual Disk 1 Is write through	No
Virtual Disk 1 Label	Hard disk 1
Virtual Disk 1 Summary	104858 KB
Virtual Disk 1 Summary, GB	0.1
Virtual MMU usage	automatic
Vm config file	[VSAN] Nostalgia/Nostalgia.vmx
VMCapability is change tracking supported	False
VMCapability is console preferences supported	False
Vmdk File 1	[VSAN] Nostalgia/Nostalgia.vmdk
VMTools after power on	True
VMTools after resume	True
VMTools before guest reboot	False
VMTools before guest shutdown	True
VMTools before guest standby	True
VMTools pending customization	Not set
VMTools sync time with host	False
VMTools tools version	0
VMtools upgrade policy	manual
VMWare Tools	tools not installed
Vmx File	[VSAN] Nostalgia/Nostalgia.vmx
Vmx File Short	Nostalgia.vmx
Will enter BIOS setup during next boot	False

Name	LLSP
Alternate guest name	Not set
Annotation	Not set
Boot delay	0
Capability cpu feature mask supported	True
Capability is boot options supported	True
Capability is disabling snapshots supported	False
Capability is disk shares supported	True
Capability is lock snapshots supported	False
Capability is memory snapshots supported	True
Capability is multiple snapshots supported	True
Capability is npiv wwn on non rdm vm supported	True
Capability is powered off snapshots supported	True
Capability is quiesced snapshots supported	True
Capability is record replay supported	True

Capability is revert to snapshot supported	True
Capability is s1Acpi management supported	True
Capability is setting display topology supported	False
Capability is setting screen resolution supported	False
Capability is setting video ram size supported	True
Capability is snapshot config supported	True
Capability is snapshot operations supported	True
Capability is swap placement supported	True
Capability is tools auto update supported	False
Capability is tools sync time supported	True
Capability is virtual mmu usage supported	True
Capability is vm npiv wwn disable supported	True
Capability is vm npiv wwn supported	True
Capability is vm npiv wwn update supported	True
CD/DVD drive 1 label	CD/DVD Drive 1
CD/DVD drive 1 summary	ATAPI ide0:0
CDROM Mounted	False
Change Tracking Enabled	False
ChangeVersion	1/19/2011 8:42:43 AM
Cluster	Melvin
Committed 1	1073801554
Connection State	connected
CPU	1
Cpu expandable	False
Cpu level	normal
Cpu limit	Not set
Cpu reservation	Not set
Cpu shares	1000
Default power off operation type	soft
Default reset type	soft
Default suspend type	hard
Disable acceleration	False
Disk uuid enabled	False
Enable logging	True
Eula	Not set
Guest id	other26xLinuxGuest
Guest OS	Ubuntu Linux (32-bit)
Hardware virtualization for execution usage	hvAuto
Has snapshot(s)	No
Host	10.0.0.85
Hostname	ubuntu.local
Hot plug memory increment size	0
Hot plug memory limit	0
Ht sharing	any
InstallBootStopDelay	0

Instance UUID	52eb61eb-9f96-4dcd-6c32-348d33aa2a84
Ip address	10.0.0.22
IpAllocationPolicy	fixedPolicy
IpProtocol	IPv4
Is cpu hot add enabled	False
Is cpu hot remove enabled	False
Is question raised	Not set
Last modified	1/1/1970 12:00:00 AM
List of datastore name and urls	name: VSAN, url: /vmfs/volumes/4d021938-1cb133cb-32a8-000c29542e7f
Location	564d8a44-8d32-f1f5-2209-e4eb5c3ddfdf
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.85>LLSP
Log directory	[VSAN] LLSP/
Managed Object Reference	vm-404
Memory (MB)	512
Memory expandable	False
Memory hot add enabled	False
Memory level	normal
Memory limit	Not set
Memory reservation	Not set
Memory shares	5120
Monitor type	release
Name	LLSP
Network Adapter 1 IP	10.0.0.22
Network Adapter 1 Label	Network adapter 1
Network Adapter 1 MAC	00:50:56:ad:01:6c
Network Adapter 1 Summary	VM Network
Notes	Not set
NPIV desired node WWNs	0
NPIV desired port Wwns	0
NPIV is temporary disabled	True
NPIV node World Wide Name	Not set
NPIV on non rdm disks	False
NPIV port world wide name	Not set
NPIV world wide name type	Not set
Number of cpu	1
Number of ethernet cards	1
Number of virtual disks	1
OverallStatus	green
OvfEnvironmentTransport	Not set
Parent	Not set
Power off operation type	soft
Power State	poweredOff
Record replay enabled	False
Record replay state	inactive

Reset type	soft
Run with debug info	False
Runtime Info boot time	Not set
Runtime Info clean power off	False
Runtime Info fault tolerance state	notConfigured
Runtime info max cpu usage	1319
Runtime info max memory usage	512
Runtime Info memory overhead	118517760
Runtime Info need secondary reason	Not set
Runtime Info number of active MKS connections	0
SCSI bus sharing 1	noSharing
SCSI controller 1	LSI Logic
SCSI controller name 1	SCSI controller 0
Snapshot directory	[VSAN] LLSP/
Snapshot power off behavior	powerOff
Snapshot tree locked	False
Snapshots disabled	False
Standby action	checkpoint
SupportedAllocationScheme	Not set
SupportedIpProtocol	IPv4
Suspend directory	[VSAN] LLSP/
Suspend interval	0
Suspend operation type	hard
Suspend time	Not set
Swap placement	inherit
Template	False
Tools installer mounted	False
Tools status	guestToolsNotRunning
Tools version status	guestToolsCurrent
Type	VirtualMachine
Uncommitted 1	536870912
Unshared 1	1073801554
Use TCP/IP Offloading	False
UUID	422d2e0c-2fd5-ad39-1836-e18ce991bcc2
Version	vmx-07
Virtual Disk 1 Disk Mode	persistent
Virtual Disk 1 Is split	No
Virtual Disk 1 Is thin provisioned	No
Virtual Disk 1 Is write through	No
Virtual Disk 1 Label	Hard disk 1
Virtual Disk 1 Summary	1048576 KB
Virtual Disk 1 Summary, GB	1
Virtual MMU usage	automatic
Vm config file	[VSAN] LLSP/LLSP.vmx
VMCapability is change tracking supported	True

VMCapability is console preferences supported	False
Vmdk File 1	[VSAN] LLSP/LLSP.vmdk
VMTools after power on	True
VMTools after resume	True
VMTools before guest reboot	False
VMTools before guest shutdown	True
VMTools before guest standby	True
VMTools pending customization	Not set
VMTools sync time with host	False
VMTools tools version	8322
VMtools upgrade policy	manual
VMWare Tools	not running
Vmx File	[VSAN] LLSP/LLSP.vmx
Vmx File Short	LLSP.vmx
Will enter BIOS setup during next boot	False

Name	DSL
Alternate guest name	Other (32-bit)
Annotation	Not set
Boot delay	0
Capability cpu feature mask supported	True
Capability is boot options supported	True
Capability is disabling snapshots supported	False
Capability is disk shares supported	True
Capability is lock snapshots supported	False
Capability is memory snapshots supported	True
Capability is multiple snapshots supported	True
Capability is npiv wwn on non rdm vm supported	True
Capability is powered off snapshots supported	True
Capability is quiesced snapshots supported	True
Capability is record replay supported	True
Capability is revert to snapshot supported	True
Capability is s1Acpi management supported	True
Capability is setting display topology supported	False
Capability is setting screen resolution supported	False
Capability is setting video ram size supported	True
Capability is snapshot config supported	True
Capability is snapshot operations supported	True
Capability is swap placement supported	True
Capability is tools auto update supported	False
Capability is tools sync time supported	True
Capability is virtual mmu usage supported	True
Capability is vm npiv wwn disable supported	True

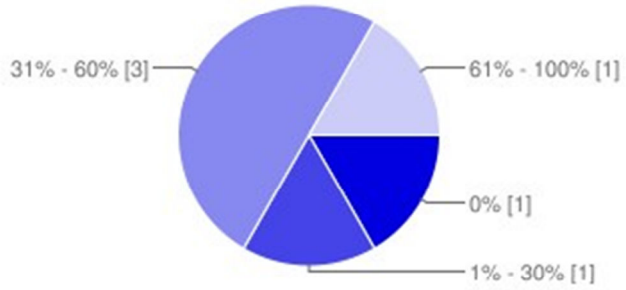
Capability is vm npiv wwn supported	True
Capability is vm npiv wwn update supported	True
CD/DVD drive 1 label	CD/DVD Drive 1
CD/DVD drive 1 summary	ISO [VSAN] ISOs/dsl-4.4.10.iso
CDROM Mounted	True
Change Tracking Enabled	False
ChangeVersion	1/20/2011 10:08:15 AM
Cluster	Melvin
Committed 1	16374876340
Connection State	connected
CPU	1
Cpu expandable	False
Cpu level	normal
Cpu limit	Not set
Cpu reservation	Not set
Cpu shares	1000
Default power off operation type	soft
Default reset type	soft
Default suspend type	hard
Disable acceleration	False
Disk uuid enabled	False
Enable logging	True
Guest id	Not set
Guest OS	Other (32-bit)
Hardware virtualization for execution usage	hvAuto
Has snapshot(s)	No
Host	10.0.0.84
Hostname	Not set
Hot plug memory increment size	0
Hot plug memory limit	256
Ht sharing	any
Instance UUID	528a2ef3-5113-c216-74c3-98a86e65cf3b
Ip address	Not set
Is cpu hot add enabled	False
Is cpu hot remove enabled	False
Is question raised	Not set
Last modified	1/1/1970 12:00:00 AM
List of datastore name and urls	name: VSAN, url: /vmfs/volumes/4d021938-1cb133cb-32a8-000c29542e7f
Location	564d2765-23c1-35cb-0889-f9a6f2c2a6ff
Location Path	>10.0.0.10>Thesis>Melvin>10.0.0.84>DSL
Log directory	[VSAN] DSL/
Managed Object Reference	vm-410
Memory (MB)	256
Memory expandable	False

Memory hot add enabled	False
Memory level	normal
Memory limit	Not set
Memory reservation	Not set
Memory shares	2560
Monitor type	release
Name	DSL
Network Adapter 1 Label	Network adapter 1
Network Adapter 1 MAC	00:50:56:ad:61:c2
Network Adapter 1 Summary	VM Network
Notes	Not set
NPIV desired node WWNs	0
NPIV desired port Wwns	0
NPIV is temporary disabled	True
NPIV node World Wide Name	Not set
NPIV on non rdm disks	False
NPIV port world wide name	Not set
NPIV world wide name type	Not set
Number of cpu	1
Number of ethernet cards	1
Number of virtual disks	1
OverallStatus	green
Parent	Not set
Power off operation type	preset
Power State	poweredOn
Record replay enabled	False
Record replay state	inactive
Reset type	preset
Run with debug info	False
Runtime Info boot time	Not set
Runtime Info clean power off	False
Runtime Info fault tolerance state	notConfigured
Runtime info max cpu usage	1393
Runtime info max memory usage	256
Runtime Info memory overhead	75235328
Runtime Info need secondary reason	Not set
Runtime Info number of active MKS connections	1
SCSI bus sharing 1	noSharing
SCSI controller 1	LSI Logic
SCSI controller name 1	SCSI controller 0
Snapshot directory	[VSAN] DSL/
Snapshot power off behavior	powerOff
Snapshot tree locked	False
Snapshots disabled	False
Standby action	powerOnSuspend

Suspend directory	[VSAN] DSL/
Suspend interval	0
Suspend operation type	preset
Suspend time	Not set
Swap placement	inherit
Template	False
Tools installer mounted	False
Tools status	guestToolsNotRunning
Tools version status	Not set
Type	VirtualMachine
Uncommitted 1	0
Unshared 1	16374876340
Use TCP/IP Offloading	False
UUID	422dd6a1-38e8-1594-c77c-fd4fc5eed7e9
Version	vmx-07
Virtual Disk 1 Disk Mode	persistent
Virtual Disk 1 Is split	No
Virtual Disk 1 Is thin provisioned	No
Virtual Disk 1 Is write through	No
Virtual Disk 1 Label	Hard disk 1
Virtual Disk 1 Summary	15728640 KB
Virtual Disk 1 Summary, GB	15
Virtual MMU usage	automatic
Vm config file	[VSAN] DSL/DSL.vmx
VMCapability is change tracking supported	True
VMCapability is console preferences supported	False
Vmdk File 1	[VSAN] DSL/DSL.vmdk
VMTools after power on	True
VMTools after resume	True
VMTools before guest reboot	False
VMTools before guest shutdown	True
VMTools before guest standby	True
VMTools pending customization	Not set
VMTools sync time with host	False
VMTools tools version	0
VMtools upgrade policy	manual
VMWare Tools	tools not installed
Vmx File	[VSAN] DSL/DSL.vmx
Vmx File Short	DSL.vmx
Will enter BIOS setup during next boot	False

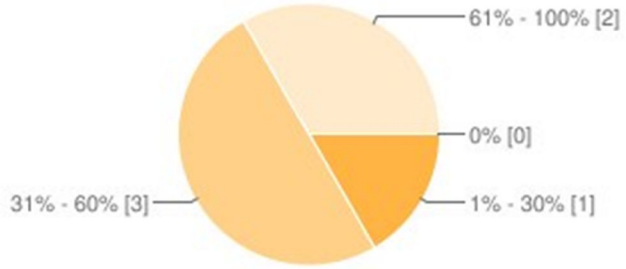
Virtualization Security Survey

What percentage of your production servers is currently virtualized?



0%	1	17%
1% - 30%	1	17%
31% - 60%	3	50%
61% - 100%	1	17%

What percentage of your production servers do you expect to virtualize by the end of 2011?

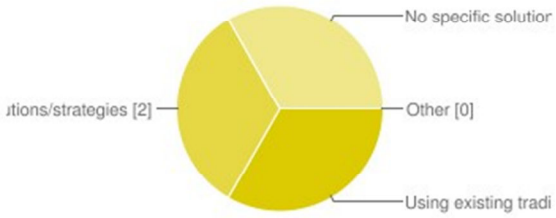


0%	0	0%
1% - 30%	1	17%
31% - 60%	3	50%
61% - 100%	2	33%

How important is it for you to secure your virtual environment?

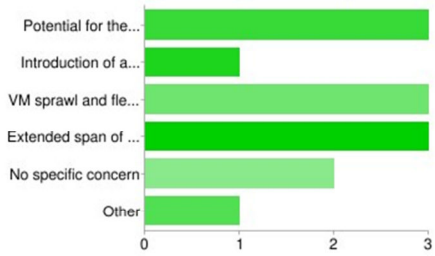
As important as securing the rest of my IT architecture	6	100%
More important than securing the rest of my IT archiecture	0	0%
Less important than securing the rest of my architecture	0	0%
Not important	0	0%

How are you securing your virtual environment?



Using existing traditional security solutions/strategies	2	33%
Using virtual-environment specific security soutions/strategies	2	33%
No specific solutions/strategies in place	2	33%
Other	0	0%

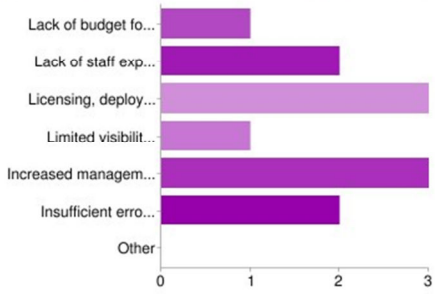
Which of the following are security concerns for you when it comes to virtualization? (Multiple selections allowed)



Potential for the Hypervisor to create a single- point of entry into multiple machines	3	50%
Introduction of a new layer that can be attacked	1	17%
VM sprawl and flexible deployment capabilities leading to unmonitored/inivisble machines	3	50%
Extended span of control available to admins leading to greater potential for insider abuse	3	50%
No specific concern	2	33%
Other	1	17%

People may select more than one checkbox, so percentages may add up to more than 100%.

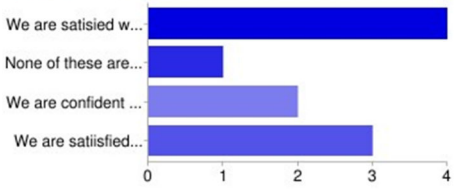
What are the primary inhibitors to securing your virtual environment? (Multiple selections allowed)



Lack of budget for virtual-environment specific security solutions	1	17%
Lack of staff expertise	2	33%
Licensing, deployment and support models of security vendors not optimized for virtualized systems	3	50%
Limited visibility provided by traditional security management consoles into the virtualization	1	17%
Increased management complexity	3	50%
Insufficient error/logging information provided by the virtualization layer	2	33%
Other	0	0%

People may select more than one checkbox, so percentages may add up to more than 100%.

Which of these apply or are true for your organization? (Multiple selections allowed)



We are satisied with the logging and auditing capabilities for our non-virtual environment	4	67%
None of these are true for my organization	1	17%
We are confident our virtual environment is as secure as the rest of our IT architecture	2	33%
We are satiisfied with the logging and auditing capabilities for our virtual environment	3	50%

People may select more than one checkbox, so percentages may add up to more than 100%.