2011

The Implications Of Virtual Environments In Digital Forensic Investigations

Farrah M. Patterson
University of Central Florida

THE IMPLICATIONS OF VIRTUAL ENVIRONMENTS IN DIGITAL FORENSIC INVESTIGATIONS

by
FARRAH M. PATTERSON
B.S. University of Tampa, 2007

A thesis submitted in partial fulfillment
for the degree of Master of Science in the Department of Digital Forensics
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term

2011

ABSTRACT

This research paper discusses the role of virtual environments in digital forensic investigations. With virtual environments becoming more prevalent as an analysis tool in digital forensic investigations, it’s becoming more important for digital forensic investigators to understand the limitation and strengths of virtual machines. The study aims to expose limitations within commercial closed source virtual machines and open source virtual machines. The study provides a brief overview of history digital forensic investigations and virtual environments, and concludes with an experiment with four common open and closed source virtual machines; the effects of the virtual machines on the host machine as well as the performance of the virtual machine itself. My findings discovered that while the open source tools provided more control and freedom to the operator, the closed source tools were more stable and consistent in their operation. The significance of these findings can be further researched by applying them in the context of exemplifying reliability of forensic techniques when presented as analysis tool used in litigation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER ONE: INTRODUCTION

1.1 History of Computer Forensic Investigations

Computer forensics investigations really began to take shape when congress passed a series of laws starting in the 1980's in response to growing occurrence of computer related crime. Most notably is the Computer Fraud and Abuse Act of 1984 (aka 18 U.S.C. § 1029 & 1030). The act was originally designed to target hackers who accessed computers to steal information or to disrupt or destroy computer functionality, as well as criminals who possessed the capacity to access and control high technology processes vital to our everyday lives. Another key piece of legislation pivotal to the history computer investigations was the Electronic Communications Privacy Act of 1986 (ECPA Pub. L. 99-508, Oct. 21, 1986, 100 Stat. 1848, 18 U.S.C. § 2510.) It was enacted by Congress to extend government restrictions on wiretaps from telephone calls to include transmissions of electronic data by computer. More importantly, the ECPA was an amendment of the Omnibus Crime Control and Safe Streets Act of 1968, which was largely designed to prevent unauthorized government access to private electronic communications. With legislation in place to regulate computer investigation, it became inherently important that forensic investigators tool kits needed to expand and be accurate to present credible evidence in the court of law.

During the early stages of digital forensics investigations law enforcement faced the burden of proving that their data collection methods were "forensically sound". In research paper entitled "Digital Forensics: Civilizing the Cyber Frontier" authored by Ian Chambers, he called this early phase the ‘Adhoc Phase”. The adhoc phase of digital forensic investigations is characterized by “a lack of structure, a lack of clear goals, and a lack of adequate tools, processes and procedures. At this time it was not uncommon to see an organization’s management carefully collect evidence that IT equipment was being used “inappropriately” by an individual, only to find that HR and Corporate Counsel would refuse to act citing the lack of a published appropriate use policy”(2009) .

From then on investigations transitioned from various phases. Starting with having inadequate tools, to better tools, to pioneering how crimes involving technology are investigated. In addition to developing tools to aid forensic investigations, numerous laws have been developed in attempt to be in step with the crimes that are committed with computers. Admittedly the law is not completely in step with technology. One of the most pioneering and useful innovations was the introduction of disk drive duplication on 1991. ICS was the pioneer of this technology. The importance of this invention to digital investigations is that the exact duplicate of the original can be investigated with compromising the original piece of evidence. Another example, of a pioneering and invaluable tool is virtual environments, which will be discussed in more depth in succeeding chapters and the focus of experimentation.

1.2 Introduction of Virtual Environments

Let’s begin with a definition of a virtual machine (VM). A VM is a software program that behaves and executes programs like a physical machine (also known as the host). A virtual machine has an OS just like a physical machine. VM’s mimic CPU usage, access the physical computer’s hard drive, RAM, Ethernet connections and utilizes it processors. In other terms, “a virtual machine or VM “is the construct of a program that behaves so much like a real machine that an OS, or other program written to run alone on a real machine, is fooled into thinking that it is running on a real bare machine by itself” (Gibson, Virtual Machine History & Technology. Security Now!, 2006). IBM is credited with creating the first major system with virtual memory around 1967 with its 360 model 67. Following the CP 67 model was the VM/370, “a direct descendant of CP 67, was used about equally by users of CMS and those who needed to run various alternate operating systems for reasons such as testing” (Gibson, 2006). Virtualization was first implemented by IBM as a way to logically partition mainframe computers into separate virtual machines. The partitions allowed mainframes to “multitask. At the time, mainframes were expensive; resource partitioning allowed organizations to benefit from the investment in such as machine. In a 1974 paper entitled “Formal Requirements for Virtualizable Third generation Architecture” by Popek and Goldberg outlines three essential characteristics of a Virtual Machine. First, it provides an environment that is nearly identical to the original machine. Secondly, programs that are run within the virtual environment will only show minor decreases in speed. Lastly, the virtual machine monitor is in complete control of system resources.

Following the development of x86 servers and client server applications that led to distributed computing, virtualization was more or less abandoned in the 1980’s and 1990’s. Windows and Linux operating systems took ground and became very popular during this time. With that popularity, came many challenges in terms of analyzing data from these systems. VMWARE the leading company in virtualization cited the following reasons for the reemergence of virtual machines (VMWARE, 2010):

- Low Infrastructure Utilization
- Increasing Physical Infrastructure Costs
- Increasing IT Management Costs
- Insufficient Failover and Disaster Protection.
- High Maintenance end-user desktops and the numerous challenges present.

Oracle’s Virtual Box, VMware’s Workstation, Cooperative Linux, and Citrix’s XEN Desktop are just a handful of the virtualization tools that are currently on the market; they will be the focus of the experimental portion of this paper. Each of the before mentioned VMs represent some of the diverse flavors that exist, such as desktop virtualization, cooperative virtualization and the traditional VM concept.

1.2.1 Desktop Virtualization

Desktop virtualization aims to segregate a personal computer desktop from a physical machine using a client-server model in which a central server stores the "virtualized" desktop, in lieu of local storage on the remote client; when users work from their local machine, each and every one of the programs, applications, processes, and data used are kept and run on the central server (see fig. 1). This enables end-users to run operating system and execute applications from a mobile device or thin client, which would normally surpass the hardware available on the client, which it can run.

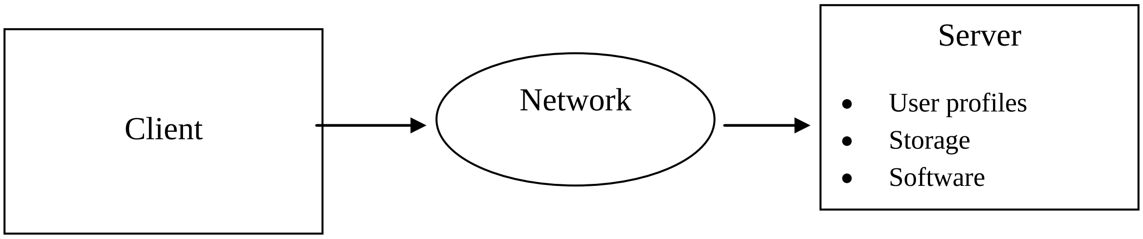


Figure 1: Desktop Virtualization Architecture

1.2.2 Cooperative Virtualization

The premise of cooperative virtualization is that the guest and host share existing resources on the physical machine. The host and the guest OS both share the same resources and run parallel to each other (see fig.2). Via the CVM (Cooperative Virtual Machine) the guest OS can control the host machine. Both the host and guest OS have equal privileges.

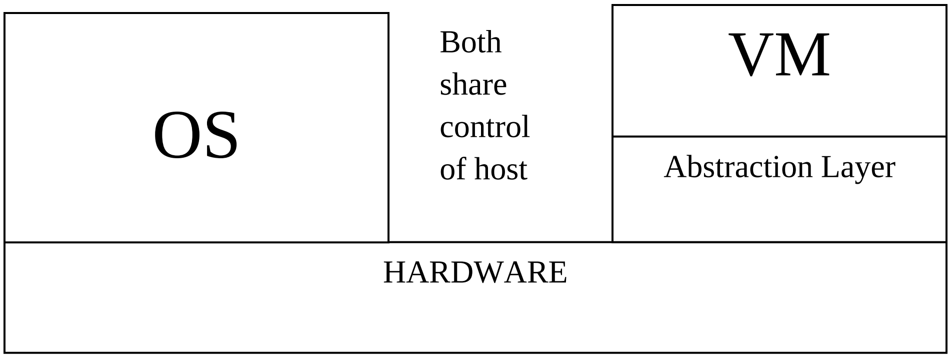


Figure 2: Cooperative Virtualization Architecture

1.2.3 Traditional Virtualization

Within a Traditional VM concept, the resources of the host machine are virtualized for every guest OS. This includes network resources, memory, hard drive size, etc. One physical machine can run multiple operating systems simultaneously. Each instance of a VM can run different operating systems (see fig. 3). The ability to run different OSs is accomplished through use of a virtual appliance. By definition “a virtual appliance is a *pre-integrated, self contained* system that is made by combining a software application (e.g., server software) with just enough operating system for it to run optimally on industry standard hardware or a virtual machine (e.g., VMware, VirtualBox, Xen HVM, KVM)” (What is a virtual appliance?, 2008). The most common format is OVF. Open Virtualization Format (OVF) is an open standard for packaging and distributing virtual appliances or more generally software to be run in virtual machines.

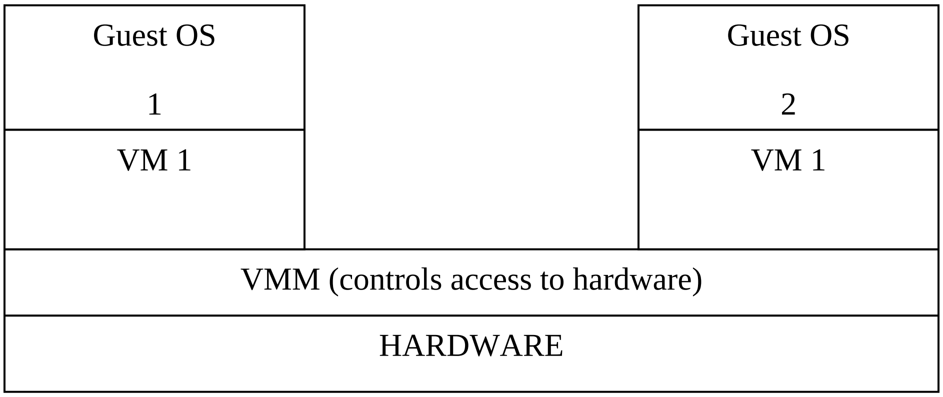


Figure 3: Traditional Virtualization Architecture

1.3 Role of VMs and their Advantages and Disadvantages

Now that the various types of virtual machine have been established we can delve into the advantages and disadvantages. There are a number of advantages that exist in using a VM in the world of computer forensics. VMs are often referred to as sand boxes. This is due to the fact that everything done within the VM is contained within the application and does not cross over into the underlying system or any other applications. This can be useful when investigating the effects of malware or understanding the makeup of a suspect system. Typically a forensically sound copy of suspect system or files would be mounted onto the virtual machine which has the respective operating system installed. At which point, theoretically, the investigator is able to examine the system as if he/she were sitting in front of actual PC, the image came from. A virtual environment gives investigators a type of freedom that cannot be experienced if they were analyzing an original version of evidence. Investigators can make an attempt to mimic suspect environments without compromising original evidence or suspect systems.

As with some advances in technology there can be negative implications associated innovation. One of the more prevalent issues that virtual environments present is that the existence of a VM on a user's PC may indicate, but not necessarily, that steps may have been taken to mask criminal activity or activity that is against corporate policy. Because all of the action within a VM is encapsulated within it, its very existence doesn't constitute foul play, but it certainly has the propensity to hide it. This very fact often damages the credibility of the evidence found in many cases. The investigator must ask several questions during in analysis in these types of cases for example: 1) what remnants are left behind on the host, if any? 2) How will this evidence be received in court? 3) What actions were taken within reasonable certainty on the VM?

Technological advances in virtualization tools essentially make removable media a PC that can be carried around in a pocket or around a neck. Running operating systems and applications within a VM leaves very little trace on the host system. The existence of a VM does not automatically infer the cover up of a crime or wrongdoing but it certainly complicates matters for investigators when analyzing a device that has one present.

CHAPTER TWO: LITERATURE REVIEW

Choosing a virtualization tool has become a contest of which one is more popular or user friendly. But does anyone really know the capabilities of the tools or selection, beyond what GUI is easy to deal with? Or more importantly what tools is more appropriate to accomplish the task at hand and what of kind stress they put on host systems. There is a gap in knowledge about the true capabilities of various open and closed source virtual machines. The aim of this research is to understand the architecture and limitations of various VM's and their implications on the analysis phase of digital forensics investigations. Closed and open source virtual machines are becoming an increasingly popular tool used in the analysis phase of computer forensic investigations. The role of VMs within investigations has expanded in conjunction with the capabilities they have come to posses over their brief history. There is a need to develop a comprehensive understanding of them beyond what is contained in vendor factsheets or online forums for as the findings found from using VM for evidence analysis are used in litigation. The findings hold a degree of accountability on the examiner to fully understand the tools used to analyze important those forensic examiners.

2.1 Literature Review

There have been many articles and research efforts that discuss the evolution of virtual machines and their use as an analysis tool in computer forensic investigations. These pieces are relevant to this case study because they set the framework for exploring the implications virtual machines have on digital forensic investigations. What is missing from these studies is a current assessment of the how these tools operate in terms of forensic applications and known limitations. The framework that is the basis for this study is segmented into three parts:

- Evolution of virtual machines to provide some history
- Role and capabilities of virtual machines in the analysis phase
- Challenges of using them and potential areas for more research

These stages are evidenced by the following sources:

2.1.2. Virtualization and Digital Forensics: A Research and Education Agenda

In a study titled “Virtualization and Digital Forensics: A Research and Education Agenda” some of the issues and capabilities of virtual environments relevant to the time the study were discussed such as scalability and future support. This particular work discusses the issues and capabilities of virtual environments at the time the study was written. A discussion of open source and commercial virtualization environments and their impacts on digital forensics is also a focal point of this article. Lastly this piece of research also develops ideas for new research categories in regards to virtualization. Within those discussions the authors make valid points that virtualization is a continuously changing landscape and needs more focused research to better understand how it can be used on the field.

2.1.3. Forensic Examination of Volatile System Data Using Virtual Introspection

This research paper discusses the sensitive nature of volatile memory and the challenges it presents to forensic investigators in the analysis phase of digital forensic investigations. The author’s correctly argue that “live analysis of target systems to uncover this volatile data, presents significant risks and challenges to forensic investigators as observation techniques are generally intrusive and can affect the system being observed" (Hay & Nance).

The authors investigate one possible resolution to this issue is by suggesting perform live analysis on a target system using virtual introspection. Virtual inspection in the simplest of terms means the target system remains unchanged while being monitored via virtual machine monitor. This work further solidifies the positive aspects of introducing virtual environments to the digital forensic landscape.

2.1.4 Computer Forensic Analysis in a Virtual Environment

In a study entitled "Computer Forensic Analysis in a Virtual Environment" the potential role of virtual environments in the analysis phase of computer forensics investigations is researched. The authors assert that the conclusions they arrive to after some research is that computer forensic analysis in a virtual environment cannot be considered to be a replacement for conventional techniques of computer evidence collection and analysis. Part of research centers around VMware. The authors also attest that "VMware simulates different hardware than the hardware of the original Windows XP installation. We show that the environment created by VMware differs considerably from the original computer system, and because of that VMware by itself is very unlikely to produce court admissible evidence. Although "an additional advantage is that the virtual machine environment makes it easy to demonstrate the findings to a non-technical audience" (Bem & E.H, 2007).

2.1.5. Digital Forensics: Validation and Verification in a Dynamic Work Environment

Beckett and Slay attempt to research the issues surrounding the validation and verification of forensic software tools. I feel that this is an important article as it touches on the fact that many virtual machines are open source, and with the source code continuously changing there should be some sort of standard in regards to verifying the validity of the tool. With the forensic tool landscape always changing there is a growing need for standardization in terms of evaluating the products used by investigators on the field. One very valid a critical point raised by Beckett and Slay is " the other problem faced is that not all tools used by specialists were designed originally with the forensic process in mind, instead developed to meet the needs of particular interest groups, such as file system drivers, operating systems, Indexing engines, etc." This fact cannot be ignored in terms of the use of virtual environments.

2.1.6 Virtualization and Forensics: A Digital Forensic Investigator's Guide to Virtual Environments

Barrett and Kipper investigates virtualized environments and how they have integrated themselves in computer forensics with the work entitled "Virtualization and Forensics: A Digital Forensic Investigator's Guide to Virtual Environments." The book explains the different types of virtualization, and how virtualization is integrated into the forensic process. The book explains how virtualization is used to retrieve artifacts on dead drives, live analysis and identify virtual activities. Barrett and Kipper touch on some issues presented by virtualization such as security, data retention policies, and where virtualization is leading in the future.

2.1.7 Virtual Forensics: A Discussion of Virtual Machines Related to Forensics Analysis by Ben Shavers

This article delves into the concept of the virtual machine, the uses of virtual machines, how to trace a virtual machine, the collection and recovery of deleted or encrypted virtual machines, and the imaging and cloning of virtual machines. Shavers also conduct a case study using VMware one of the more popular virtualization tools. The research exemplifies some of the capabilities at a high level that make virtual environments an optimal tool for investigators and also a burgeoning forensic tool.

2.2 Research Purpose

The purpose of the research is to explore various open and closed source virtual machine tools within a digital forensic investigators toolkit. The aim is to have the forensic investigator look beyond the lure of a particular GUI/command line closed or open source applications and consider what makes a tool truly useful, and the implications said tool may have when used for investigatory purposes. A secondary goal is to provide more knowledge to the users of these tools by conducting an intensive study through application of these tools.

The following research questions are being investigated:

What criteria should investigators employ before integrating these tools within their existing labs? How do the interaction of the host and virtual machine affect each other? What limitations do these tools have when used in the analysis phase of an investigation? Are special lab considerations such as licensing, budget and training when acquiring these tools?

My hypothesis is the open sources tools will be more geared to the needs of an investigator do to the open source nature of the source code, but may prove to be less stable than its closed source counter parts. I think that the closed sources tools may put more stress on the host systems. The experimentation portion of the thesis will be organized in the following manner, an introduction of the virtual machine, data analysis, and discussion. Having laid framework of the research we can now examine the methodology of the experiment.

CHAPTER THREE: METHODOLOGY

3.1 Subjects/Sample

Two open source VM's and two closed VM's were chosen as to have a decent sampling of what may be available to investigators at the present time. Open and closed source applications are prevalent tools within an examiners tool kit so in order to make this study more worthwhile and encompassing of some of the conditions on the field. Guest OS is defined as the operating system installed within the virtual environment. Host is defined as the physical host machine where the virtual machine software is installed.

The following Virtual Machines (VM) where chosen for this study:

Cooperative Linux – aka CoLinux v. 0.7.3

CoLinux is a Cooperative Virtual Machine. In contrast to a traditional VM, a CVM shares the resources that currently exist on the host OS. The CVM gives both OSs complete control of the host machine while the traditional VM sets every guest OS in an unprivileged state to access the real machine. The key feature touted by CoLinux is that each kernel has its own complete CPU condition and allocated address space. Each kernel decides when to give control back to its partner. CoLinux kernel runs in a privileged mode with the host PC switching between host OS state and CoLinux kernel state. CoLinux has full control of the physical memory management unit (MMU). CoLinux is an open source tool. CoLinux runs an actual Linux kernel, and the Linux file system is stored as in image in a standard Windows file, CoLinux also allows access to standard Windows hard drives.

VMware Workstation 7

VMware Workstation 7 is distributed by VMWARE. Considered a traditional VM in which resources of the host machine are virtualized for every guest OS. Some of the key features this tool offers are the ability to pause, copy, clone and create a team of VMs. The "team" capability enables the user to create many different VMs that are able to launch at the same time and form a network. This feature can be valuable when simulating how different OS would interact when on the same network. Additionally this tool supports many OSes and there are exuberant amounts of virtual appliances available for download for free available on the World Wide Web. VMware Workstation 7 is a closed source tool.

VirtualBox aka Oracle VM Virtual Box 4.0

Virtual Box is an x86 virtualization software package developed by Sun Microsystems (which has been acquired by Oracle Corporation), and is completely an open source tool, although there is a closed source version available. VirtualBox much like VMware Workstation also allows for the pausing of virtual machines for resumption of its current state at a later junction in time. Additionally, Virtual Box supports teleporting. Teleporting provides a means of moving a VM over a network from one host to another. Lastly, but not all inclusive of this tool's features is ability to control the host remotely via an unencrypted web service. Virtual Box is also considered a traditional VM. Originally propriety software, in January 2007, VirtualBox

Open Source Edition (OSE) was released as free software, subject to the requirements of the GNU General Public License (GPL), version 2.

XEN Desktop 5.5

Developed by Citrix, XenDesktop is a desktop virtualization tool. It separates a personal computer desktop environment from a physical machine using the client–server model of computing. With XEN Desktop users can access virtual desktops from any PC, thin client, smartphone or tablet. This tool centralizes applications on the server and with its Single sign-on feature users can access all applications they have permissions to all at once. XenDesktop is a closed source tool.

3.2 Instruments/Measures

Two Forensic PCs, for the purposes of this experiment the Forensic PC is defined as the PC's had the necessary software needed to examine the compromised drive and was not the subject of the investigation.   One was a 64bit Windows 7 Laptop, and the other was a 32bit Windows XP Pro Desktop. Both were necessary to accommodate VM's that had particular hardware specifications.

Table 1 highlights the hardware comparison needs among other details of the four virtualization tools and forensic PC's.

Table 1: VM Hardware Needs Comparison Matrix

VM Test Subjects				
	Virtual Box 4.0.8	**VMware Workstation 7.1.4**	**Cooperative Linux 0.7.3**	**XEN Desktop 5.5**
CPU	64bit and 32bit	64bit and 32bit	32bit only	*64 bit*
OS	Compatible with Windows 2000 and higher and *most* Linux distributions. Works with Mac OS server MAC OS 2;	All windows and Linux distributions	CoLinux requires Windows 2000, XP, 7 or Windows Server 2003	All windows distributions Windows XP and higher
RAM	512 MB recommended	256 MB recommended	512 MB recommended	3 GB recommended
Processor	400mhz or faster	400mhz or faster	400mhz or faster	400mhz or faster
VM file formats	Only accepts OVF format appliances. You may download VDI	. log, .VMDK, VMEM, .VMSN, .VMSD, .NVRAM, .VMX, .VMSS, VDI	.FS	XVF, OVA, OVF
Interface	GUI	GUI	Command Line Interface.	GUI
Monetary Cost	Free	30 day free trial. Full version $162 as of 11/21/2010	Free	Free Trial, License cost approx. $175 each

Table 2: Forensic PC Specifications

	Forensic PC 1	**Forensic PC 2**
CPU	64 bit	32 bit
OS	Windows 7	Windows XP
RAM	4GB	1GB
PROCESSOR	Intel I3 core CPU	Intel duo core

Two PC images were created, one of a compromised Linux based system to accommodate the needs of CoLinux and another of a compromised PC running Windows 7 operating system. Compromised PC for the purpose of this experiment is a PC that had various malware and virus on the hard drive. The purpose of using a compromised system, was guided under the assumption that a normally running PC could not reasonably provide enough data to test the subsequent test areas and also analysis of a compromised image is more closely aligned with the use of virtualization tools in the computer forensic landscape. The images of the compromised PC then ran for several days within each of the open and closed VM's. The host system as well as the VM was monitored. The host systems were monitored using Windows Performance Monitor utility. In addition to monitoring the overall performance of the overall system the hardware needs, software dependencies, maintenance need of the host and VM were documented in order to exemplify what a computer forensics lab would need to consider in order to operate these tools.

Image Details

Linux Image

The Linux system image was that of a system operating the Ubuntu distribution. This image was examined solely in the CoLinux virtual environment. The system was infected with the Adore worm, which was discovered to be a call encapsulated in a script within a file called red.tar found on the PC. The typical behavior of this worm is to scan vulnerable hosts from indiscriminate Class B subnets on the victims network. Once a host is found, an attempt is made to download the principal worm from another webserver unto the victim machine. The worm establishes a backdoor. The backdoor triggers when it receives a ping packet with correct identifying information and open a shell in the port 65535. The worm sends sensitive system data, such as the contents of the "/etc/shadow" file emails addresses programmed within the worm.

The aforementioned was discovered during the course of the VM evaluation of the image I would create, as detailed in Section 3. A DD image of the system was created and saved to the desktop of Forensic PC 2. The following commands were executed on the machine to create the DD image:

In order to capture a record of the commands entered on the compromised PC, they were sent to the forensic machine by running the following:

On the forensic works station at the command prompt:

```
nc –v –l –p 2222 > LinuxImageLog.txt
```

This serves as a road map and proof of the actions taken in the live/ behavioral analysis.

The suspect hard drive was piped to the forensic machine using the following commands:

First on the forensic machine:

```
nc –l  -p 8888 > LinuxUbuntuImage.dd (Listening to Victim machine for dd image of hard drive)
```

On the victim machine:

```
dd if=source drive | nc ip_adress_forensic_pc2  8888 –w 3
```

The resultant image is LinuxUbuntuImage.dd

In order to examine the image the following commands were executed with the CoLinux Virtual Machine.

1) The dd image was mounted Virtual Machine onto the forensic machine:

```
sudo mount –t auto –o ro, LinuxUbuntuImage.dd
```

2) Capture current network connections:

```
netstat –an | nc forensic_workstation_ip 2222
```

3) To capture internal routing table---typically used by hackers to reroute traffic

```
netstat –rn | nc forensic_workstation_ip 2222
```

4) A packet capture was also performed with tcpdump to view any changes in network activity that took place, and viewed within Wireshark.

Command to capture packets:

```
        tcpdump –v –w tcpdump.cap –Xx –s 0
```

5) The following ps command displays all running process:

```
   ps aux | less
```

Windows 7 PC

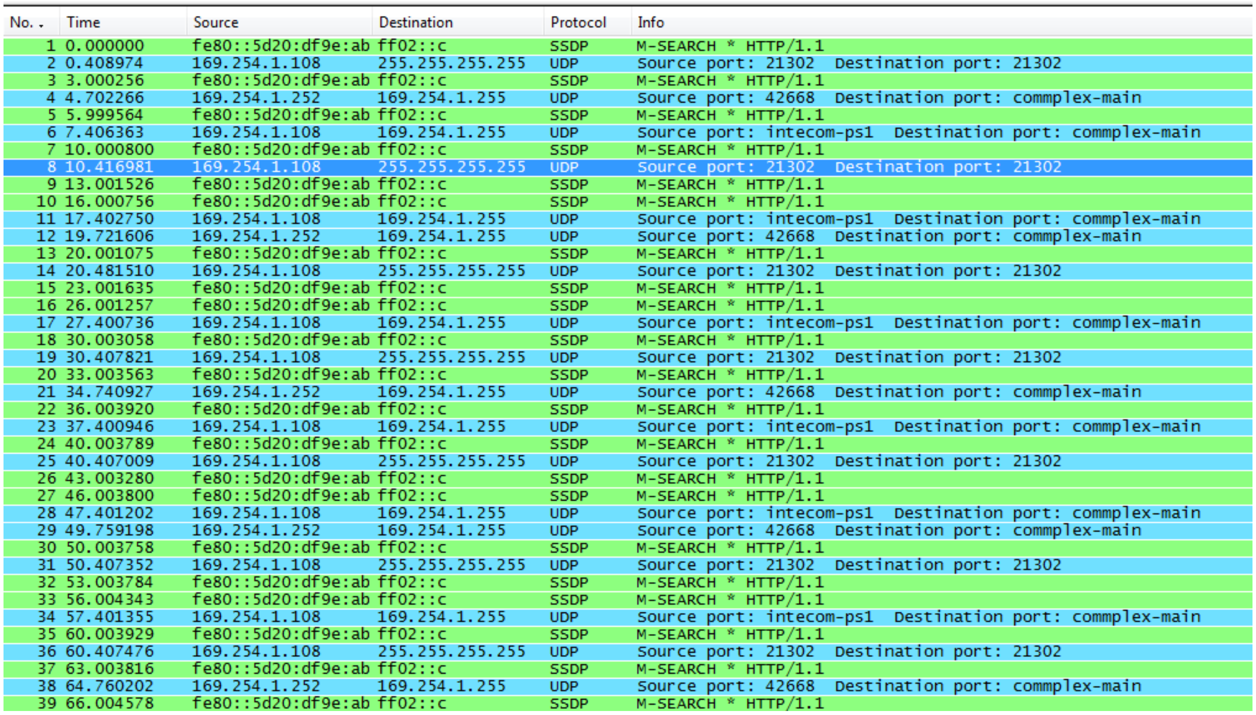
A virus scan was run on the PC after witnessing abnormal behavior such as pop ups while not connected to Internet, slowed performance, and the occasional blue screen. The virus scan returned no results. It was determined further analysis would be needed within a controlled environment. The analysis, discussed in a succeeding section revealed the abnormal behavior to be attributed malware program posing as an AntiVirus Software. The malware program tricks users into installing it by displaying fake messages that appear in their Web browsers claiming that the computer has been compromised. Upon installation, the program reports fabricated or exaggerated system security threats on the users PC.

A dead acquisition of a 80GB hard drive operating Windows 7 was acquired using FTK Imager Version 3.0.1. The image was a physical image capturing all the partitions (partitioned and un-partitioned space) of the hard drive. The acquired image was named WINDOWS7_VM_TEST_80GB.E01 and saved to the desktop of the forensic PC 1.

In order for the image to be mounted into Virtual Box and XenDesktop it needed to be converted into the applicable formats supported by the respective Virtual Machines. Disk image mounting of forensic images for the purposes of booting the image as a disk in a virtual machine environment can be accomplished a number of ways, the tool Mount Image Pro is capable of mounting Encase images, DD images, and SMART images as drive letters, this was the method chosen to mount the image into the previously mentioned environments.

OUTLINE OF COMROMISED PC ANALYSIS

Analysis of the compromised PC images within the VMs was accomplished using the various methods such as analysis of network traffic logs within WireShark, analysis of process logs, registry logs. A packet capture was created with tcpdump to view any changes in network activity that took place was viewed within Wire shark. For example, below is a packet capture of the activity of the Adore Worm. It was observed that there were repeated attempts to search for PC's on the network.



No.	Time	Source	Destination	Protocol	Info
1	0.000000	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
2	0.408974	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
3	3.000256	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
4	4.702266	169.254.1.252	169.254.1.255	UDP	Source port: 42668 Destination port: commplex-main
5	5.999564	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
6	7.406363	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
7	10.000800	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
8	10.416981	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
9	13.001526	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
10	16.000756	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
11	17.402750	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
12	19.721606	169.254.1.252	169.254.1.255	UDP	Source port: 42668 Destination port: commplex-main
13	20.001075	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
14	20.481510	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
15	23.001635	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
16	26.001257	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
17	27.400736	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
18	30.003058	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
19	30.407821	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
20	33.003563	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
21	34.740927	169.254.1.252	169.254.1.255	UDP	Source port: 42668 Destination port: commplex-main
22	36.003920	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
23	37.400946	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
24	40.003789	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
25	40.407009	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
26	43.003280	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
27	46.003800	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
28	47.401202	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
29	49.759198	169.254.1.252	169.254.1.255	UDP	Source port: 42668 Destination port: commplex-main
30	50.003758	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
31	50.407352	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
32	53.003784	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
33	56.004343	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
34	57.401355	169.254.1.108	169.254.1.255	UDP	Source port: intecom-ps1 Destination port: commplex-main
35	60.003929	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
36	60.407476	169.254.1.108	255.255.255.255	UDP	Source port: 21302 Destination port: 21302
37	63.003816	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1
38	64.760202	169.254.1.252	169.254.1.255	UDP	Source port: 42668 Destination port: commplex-main
39	66.004578	fe80::5d20:df9e:ab	ff02::c	SSDP	M-SEARCH * HTTP/1.1

On the Windows 7 PC, the following was noted:

The following executable was found:

%Temp%\avXXX.exe

This executable appeared to have created the following files:

%fpatterson%\Application Data\Microsoft\Internet Explorer\Quick Launch\Antivirus XXX.lnk

%fpatterson%\Desktop\Antivirus XXX.lnk

%fpatterson%\Start Menu\Antivirus XXX\Antivirus XXX.lnk

%fpatterson%\Start Menu\Antivirus XXX\Help.lnk

%fpatterson%\Start Menu\Antivirus XXX\Registration.lnk

Information found on Symantec’s website, indicated the behavior noted on the PC and the particulars of the discovered files are in line with a rouge virus.

3.3 Data Analysis of Virtual Machine

For the data analysis I looked at 5 key areas for each VM:

- Installation process and the VM dependency on other tools/apps.
- Host performance

  The following criteria were used to establish host performance level: 1) Amount of CPU usage on physical machine after launching VM 2) number of open processes after establishing running VM. 3) Virtual memory and physical memory statistics 4) Crashes 5) use of network bandwidth and resources.

- VM Performance evaluation aims to address such questions as, did the tool crash or malfunction under certain conditions? Do the key advertised features function as anticipated? How was the process of updating and managing the tool?
- Vulnerabilities VM presents - Such as the VM being able to execute code on the host.
- Limitations – such as over committing RAM, and inability to simulate certain devices.

Finally based on the above results we determine the implications to forensic investigators.

XenDesktop 5.5

- Installation and dependencies

The installation requires two PCs, one to function as the desktop the other as the server. A zip file with 8 files including an ISO image had to be burned onto a CD in order to download the necessary files. Installation time took approximately 90 minutes using the 17-minute instruction video available on the web from Citrix. Citrix recommends having 3GB memory available to server.

In order for the application to function properly there are many dependencies on a variety of Microsoft applications, suites, and services. XEN Desktop depends on .NET framework version 2 or higher, Microsoft SQL Express 2008 R2, Microsoft Visual C++ 2008 Runtime, Microsoft IIS, Visual J# 2.0 SE, and Java Runtime Environment 1.5.0.15. In addition, the following Citrix tools must be installed as well: The Desktop delivery controller which assigns a desktop to the user upon login and gives access to a pool of applications, XenServer hypervisor provides an environment in which to run the virtualized desktop and lastly XenDesktop must be installed on the client machine. The golden image aka the master image is the desktop image that is going to be streamed from the server to the various VM's when the user logs into the environment via the web.

Citrix maintains the following stance regarding the use of the "golden image": "we have a model within Xen Desktop where all VMs boot off the same OS golden image and all have the same base applications. To deliver a user-specific model, user-specific applications are streamed into the VM based on the user's roaming profile. This approach minimizes the number of OS images and VMs that need to be stored. Anything that's written to disk by an executing VM is cached locally in the VM and never written back to the hard drive, and all changes are discarded on every reboot. For certain classes of users, such as call center operators, this approach works very well" (Creeger, 2010).

There are different methods of establishing a virtual machine image, for instance and ISO image of the image may be used, an established virtual appliance can be downloaded from the web, or a using Xen Convertor on a local drive (C:\, D:\, etc.) can be used to convert into the following formats: Xen Virtual Appliance, OVF package, or provisioning services disk, or import a virtual machine that XenDesktop accepts.

- Host performance

The compromised image ran for several days and using the tracert command I found that latency was around 90ms on my FIOS connection. Which is not terrible considering the intensive back and forth between the client and the server. Average CPU was 57% of what was available on the forensic PC. There was increased network activity in comparison to that of the other Virtual machines evaluated as was expected to the nature of the server-client setup. The running of the XenDesktop established a process for the desktop controller, XenServer, XenDesktop, and XenCenter.

- VM Performance

There is most definitely a dependency on the server being up running so the desktop to can be accessed and utilized. To test the resilience of the server attack packets were sent using the AttackPing application to the PC functioning as the server. It must be noted that the server had no firewall settings or configurations blocking certain port activity. The following behavior was noticed on the client PC: 1) Stalling in the client when executing certain functions such opening applications within the virtual desktop. 2) Locked up sessions 3) Disconnected sessions.

- Vulnerabilities VM presents

The majority of the vulnerabilities presented by the XenDesktop lie heavily on the security and defense mechanisms available on the network the application is being run on. A weak network could provide a gateway for attacker to attack the server containing the image. A possible remedy is to have all the computers accessing the virtual environment protected by a firewall establishing program and port rules. The communications between the client (desktops) should be secured through encryption to also help mitigate the risk of infiltrating the network.

- Limitations

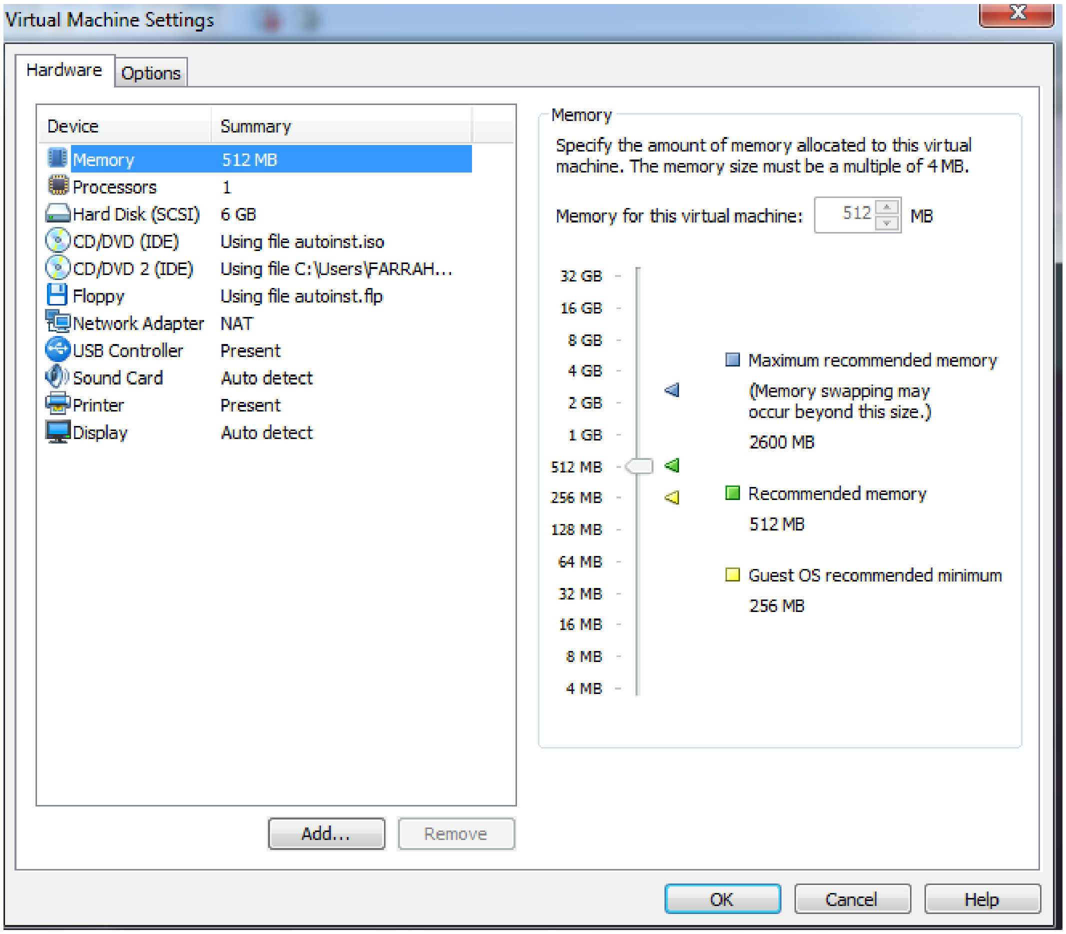
XenDesktop had a heavy dependency on having all network resources up and running. If an instance existed when the server was down or connection was interrupted the VM experienced issues and in terms of performing forensic work this possibly would not be the most optimal implementation of this type of virtual machine configuration.

There is also dependency on Active Directory to authentic and authorize users to access the Desktop image. Active Directory is required for XenDesktop utilization for authentication and authorization while using Active Directory provides some concealment of communication its not an all inclusive deterrent of the threats that exist.

VMware Workstation 7

- Installation process and dependencies

Installation of the tool is fairly simplistic. After downloading and executing the installation wizard the application was up and available for use. Installation took approximately 10 minutes. Many virtual appliances are available on the worldwide web for the plethora of operating systems available to use with this virtual machine.   During the set up of the actual virtual environment a few key prompts appeared, one for RAM allotment (See fig. 4) another for virtual hard disk size. At installation there are no other dependencies on other tools to get the tool up and running. Additionally, establishing either a bridged or NAT network can be accomplished by modifying the settings on the hardware tab.



Device	Summary
Memory	512 MB
Processors	1
Hard Disk (SCSI)	6 GB
CD/DVD (IDE)	Using file autoinst.iso
CD/DVD 2 (IDE)	Using file C:\Users\FARRAH...
Floppy	Using file autoinst.flp
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Memory

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

Memory for this virtual machine: 512 MB

🟦 Maximum recommended memory
(Memory swapping may occur beyond this size.)
2600 MB

🟩 Recommended memory
512 MB

🟨 Guest OS recommended minimum
256 MB

Figure 4: VMware Workstation 7 Hardware Settings

- Host performance

Upon the first start up the tool the vmware.exe process was established. After guest OSes were created within the tool each guest OS also established a process, vmware-vmx.exe. In other words, when a new virtual machine was created, in this instance Ubuntu, Windows 7 and Fedora machines each established a unique process (see fig 5). After monitoring the machine for 48 hours the average total private working memory set of VMware Workstation 7 was 800,000 KB. Each individual set averaged 45,000 KB. The private memory working set is amount of memory an application absolutely needs to operate and cannot share in contrast to work set of memory, which all processes can share. The significance of this figure is that it helps determine the amount of memory solely dedicated to the process.

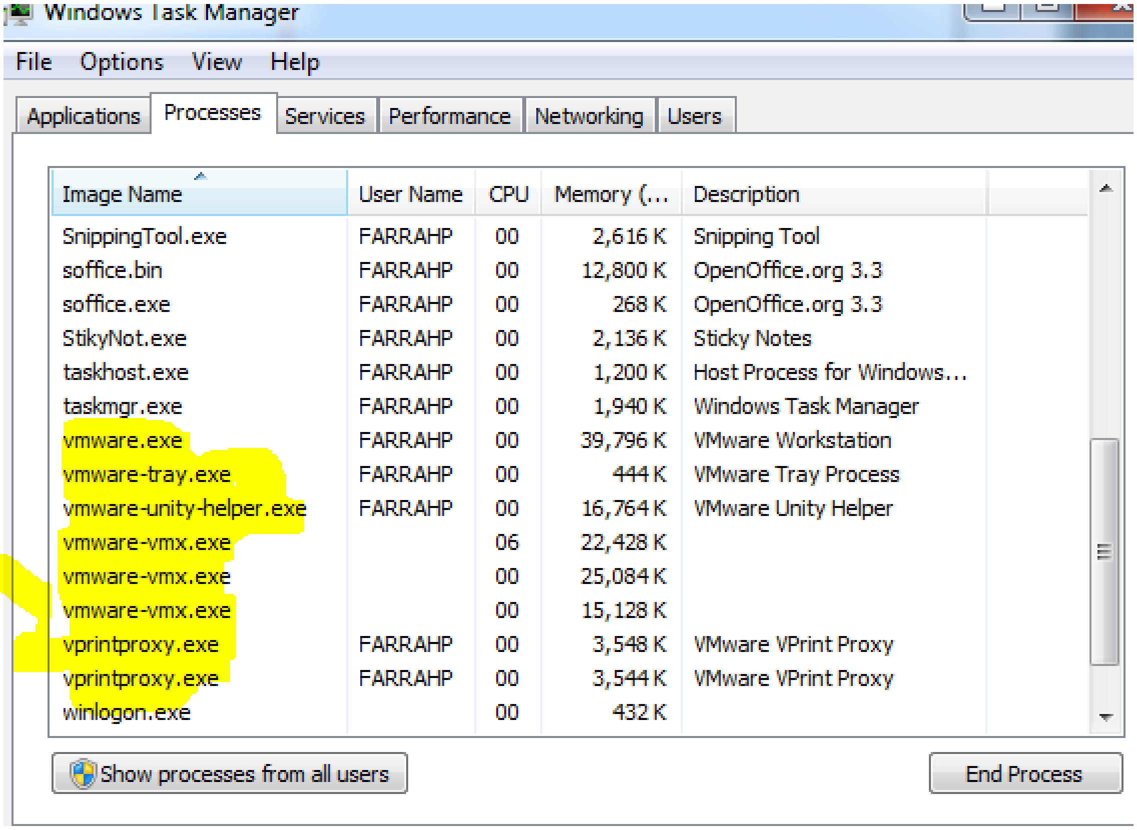


Image Name	User Name	CPU	Memory (...	Description
SnippingTool.exe	FARRAHP	00	2,616 K	Snipping Tool
soffice.bin	FARRAHP	00	12,800 K	OpenOffice.org 3.3
soffice.exe	FARRAHP	00	268 K	OpenOffice.org 3.3
StikyNot.exe	FARRAHP	00	2,136 K	Sticky Notes
taskhost.exe	FARRAHP	00	1,200 K	Host Process for Windows...
taskmgr.exe	FARRAHP	00	1,940 K	Windows Task Manager
vmware.exe	FARRAHP	00	39,796 K	VMware Workstation
vmware-tray.exe	FARRAHP	00	444 K	VMware Tray Process
vmware-unity-helper.exe	FARRAHP	00	16,764 K	VMware Unity Helper
vmware-vmx.exe		06	22,428 K	
vmware-vmx.exe		00	25,084 K	
vmware-vmx.exe		00	15,128 K	
vprintproxy.exe	FARRAHP	00	3,548 K	VMware VPrint Proxy
vprintproxy.exe	FARRAHP	00	3,544 K	VMware VPrint Proxy
winlogon.exe		00	432 K	

Figure 5: VMware Workstation 7 Open Processes upon Startup

When employing the “team” function I set up 3 OSes all 6GB in size, with the compromised images loaded to the respective OSes and started them, it was immediately noticed when toggling back between the host desktop and VM was very slow, and the guest OS took longer than favorable to render when switching between them. Average time was 30 seconds to render the virtual machines desktop. In addition, when application outside of the virtual machine was opened such as Adode Flash and games, there was a lag in rendering images. For example, when Adobe Flash was used to took on average a full minute for the images I was attempting to view to render and become viewable. This was noticed when 3 guest OSes were set up inside the virtual machines, in addition to instances when 6 OSes where within the VM operating within the "team function".

- VM Performance

Under the same conditions noted previously when using the "team" the desktops within the VMs were slow to render when activated within team function. In comparison to the boot up and shut down of a Virtual Machine within Virtual Box. VMWare Workstation 7 took two and half minutes longer to boot and a minute and half to completely shutdown.

In the initial stages of establishing a network connection, difficulty was experienced establishing a bridged network, but establishing NAT connection was not an issue. I presumed the issue could be a firewall, but the host PC did not have one set up. Uninstalling and reinstalling the program seemed to "fix" the issue.

- Vulnerabilities VM presents

I was unable uncover any vulnerabilities during my experimentation. That by no means derives the conclusion that none exists.

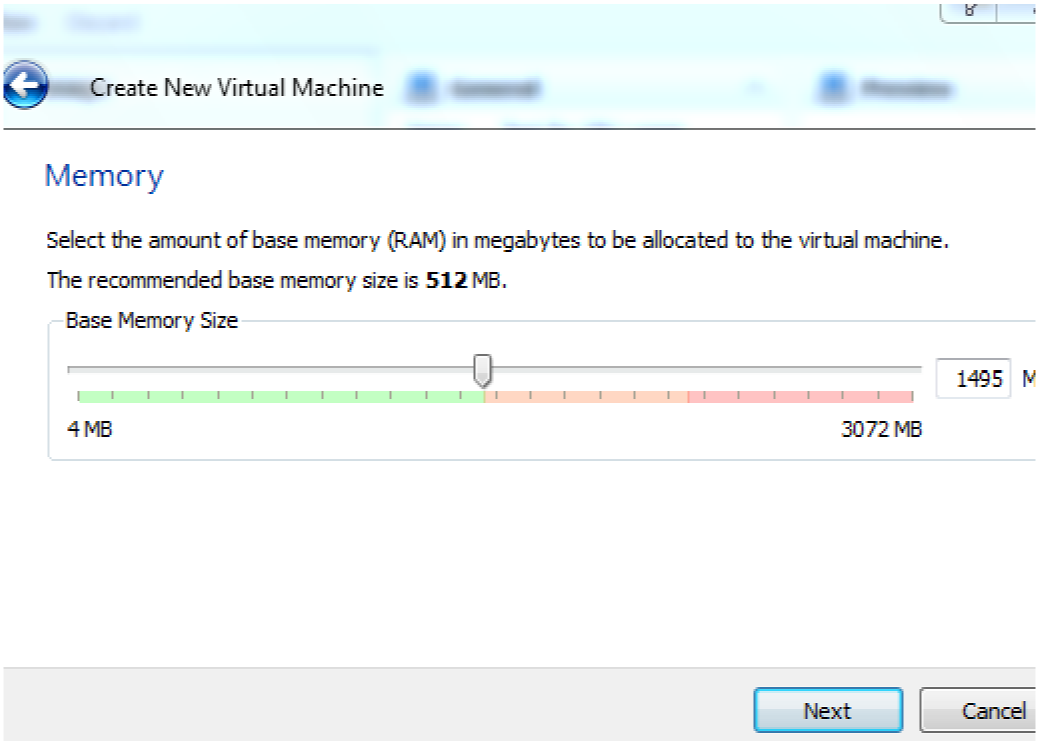
- Limitations

While not necessarily a limitation, but worth noting is the VMware Workstation uses the host systems actual time and adjust itself to match host time. This could be an issue potentially be an issue when attempting to mimic suspect pc's environment.

Oracle VM VirtualBox 4.0

- Installation process and tool dependencies

Similar to VMWARE Workstation installation was quite simplistic not time consuming, with the use of the installation and virtual machine set up wizard. Figure 6 below depicts the RAM allotment selection tool for the Guest OS. Virtual Box also allows the user to select hard disk size, network configuration such as bridged or NAT.



Select the amount of base memory (RAM) in megabytes to be allocated to the virtual machine.

The recommended base memory size is **512** MB.

Base Memory Size: 1495 M

Figure 6: VirtualBox Memory Set Up

- Host performance

A considerable amount of CPU was used and memory was used. When the tool was booted it established process called VBoxSVC.exe, additionally when a new guest OS was added to the tool each established VirtualBox.exe process. When more than one Virtual machine was established with the program each additional process drained quite a bit of the total CPU available on the host. Efficiency of other programs went down considerably, as they were slow to render and execute.

- VM Performance

Virtual Box allows snapshots of Virtual Machines to be taken. This enables the user to save a virtual machine in a particular state and return back to that state at later time. This proves invaluable when it's necessary to pause in the midst of analysis. The tool can establish specific host directories as shared folders, which can be accessed within in the OS being run within Virtual Box. Additionally, USB devices are easily detected by Virtual Box can be used to transfer data between the host and Virtual Machine.

- Vulnerabilities VM presents

VirtualBox's web service allows the host to be controlled remotely.

The connection that is established is not encrypted. This can present risk to the host system. In addition to being able to control the host remotely there is also an added risk present with the teleporting feature. As previously mentioned this feature allows for a VM over a network from one host to another, the vulnerability lies in the fact that anyone who has access to the network can potentially see this transaction execute.

- Limitations

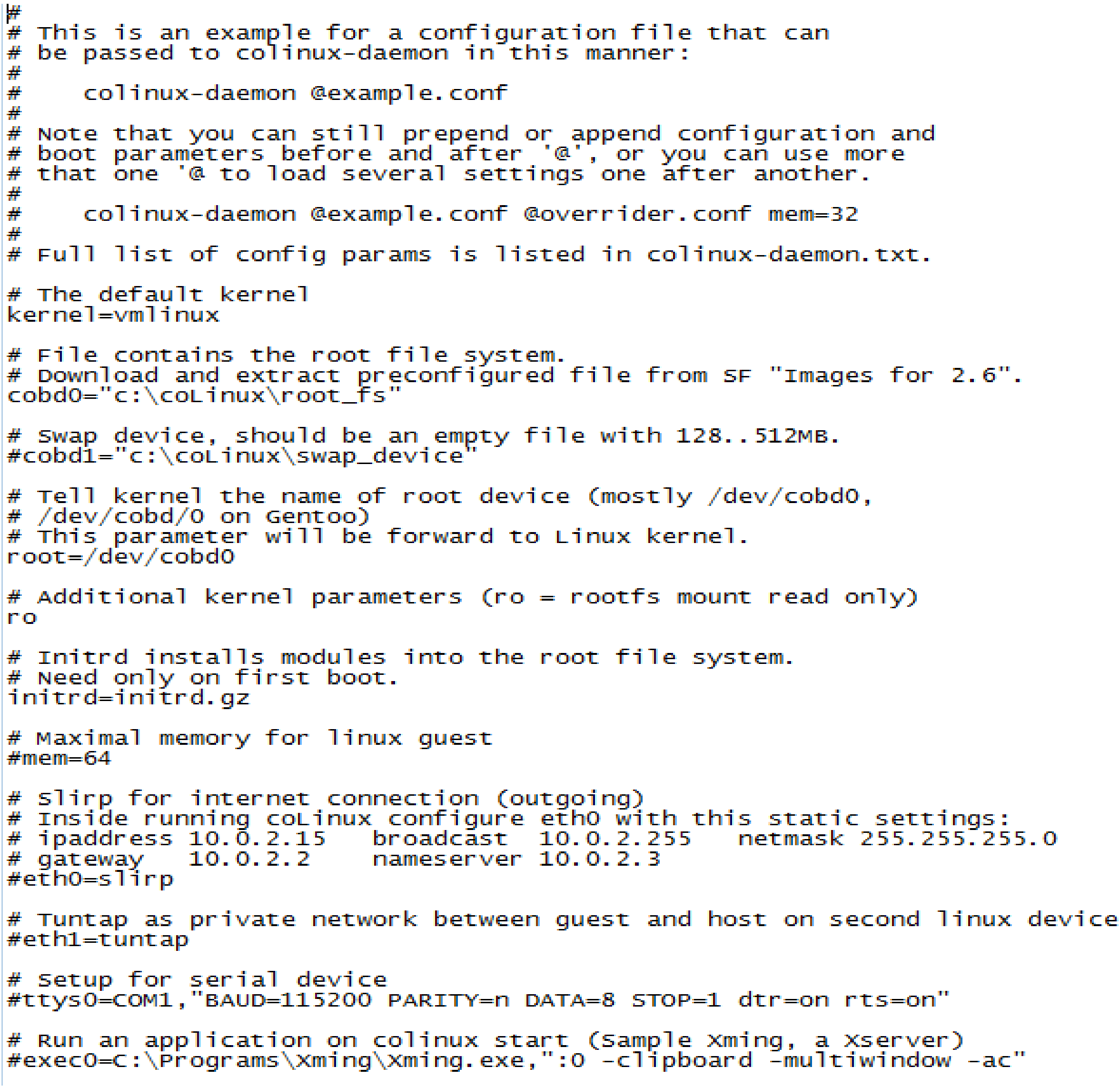
VirtualBox also had the tendency to crash when stressed when more than four virtual machine was running inside the program. Unlike VMWare Workstation, Virtual box doesn't allow booting directly from an Encase virtual (raw) disk. You need to perform additional step. By mounting and concerting the image into a file format can handle. VirtualBox also doesn't support DD raw format natively so some additional conversion would need to be done with those types of images as well. Lastly a key, missing feature is the functionality to cut and paste between virtual machines.

CoLinux v.0.7.3

- Installation process and tool dependencies

CoLinux only works on a 32-bit PC. Images for desired OSes are available for download online. CoLinux needs the WinPCap library for bridged Ethernet support. The majority of the configurations are established in the .conf file such as memory, kernel, networking and Guest OS. See fig.6 for sample config file. CoLinux hard-drive images are standard Windows .fs files of the same size as the virtual hard drive. For example a 1GB Ubuntu guest OS would look something like this "Ubuntu-9.04.ext3.1gb.fs" the config file would reference this image in order to boot an Ubuntu OS. CoLinux emulates hardware Ethernet network via TAP, PCAP, NDIS and SLiRP (see fig.7) utilities. Upon boot putting of the system it starts a col-linux slirp process.

Increasing the size of the file system is simplistic. It was as simple as swapping out one file with another one. During set up the base drive was 2 GB root drive, I was able to increase by simply modifying the file name to 10GB from 2GB. Figure 7 is a snapshot of the configuration file.



```
#
# This is an example for a configuration file that can
# be passed to colinux-daemon in this manner:
#
#    colinux-daemon @example.conf
#
# Note that you can still prepend or append configuration and
# boot parameters before and after '@', or you can use more
# that one '@ to load several settings one after another.
#
#    colinux-daemon @example.conf @overrider.conf mem=32
#
# Full list of config params is listed in colinux-daemon.txt.

# The default kernel
kernel=vmlinux

# File contains the root file system.
# Download and extract preconfigured file from SF "Images for 2.6".
cobd0="c:\coLinux\root_fs"

# Swap device, should be an empty file with 128..512MB.
#cobd1="c:\coLinux\swap_device"

# Tell kernel the name of root device (mostly /dev/cobd0,
# /dev/cobd/0 on Gentoo)
# This parameter will be forward to Linux kernel.
root=/dev/cobd0

# Additional kernel parameters (ro = rootfs mount read only)
ro

# Initrd installs modules into the root file system.
# Need only on first boot.
initrd=initrd.gz

# Maximal memory for linux guest
#mem=64

# Slirp for internet connection (outgoing)
# Inside running coLinux configure eth0 with this static settings:
# ipaddress 10.0.2.15   broadcast  10.0.2.255   netmask 255.255.255.0
# gateway   10.0.2.2    nameserver 10.0.2.3
#eth0=slirp

# Tuntap as private network between guest and host on second linux device
#eth1=tuntap

# Setup for serial device
#ttyS0=COM1,"BAUD=115200 PARITY=n DATA=8 STOP=1 dtr=on rts=on"

# Run an application on colinux start (Sample Xming, a Xserver)
#exec0=C:\Programs\Xming\Xming.exe,":0 -clipboard -multiwindow -ac"
```

Figure 7: CoLinux Configuration File

- Host performance

Although both kernels theoretically have full access to the host PC’s real hardware, technology has not progressed in matter where a system can actually controlled by two different operating systems at the same time. The host kernel is in fact in control of the real hardware and the guest kernel employs special drivers that communicate with the host and provide devices to the guest OS. CoLinux established 3 processes upon start up of the environment; one for networking, console and one for the service between the host.

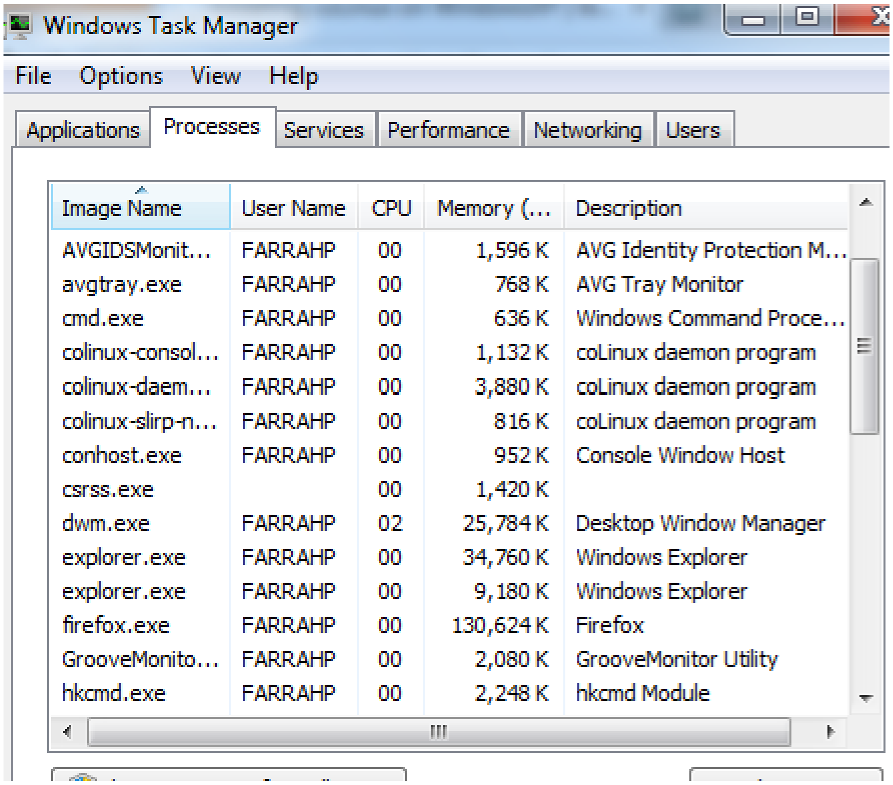


Image Name	User Name	CPU	Memory (...	Description
AVGIDSMonit...	FARRAHP	00	1,596 K	AVG Identity Protection M...
avgtray.exe	FARRAHP	00	768 K	AVG Tray Monitor
cmd.exe	FARRAHP	00	636 K	Windows Command Proce...
colinux-consol...	FARRAHP	00	1,132 K	coLinux daemon program
colinux-daem...	FARRAHP	00	3,880 K	coLinux daemon program
colinux-slirp-n...	FARRAHP	00	816 K	coLinux daemon program
conhost.exe	FARRAHP	00	952 K	Console Window Host
csrss.exe		00	1,420 K	
dwm.exe	FARRAHP	02	25,784 K	Desktop Window Manager
explorer.exe	FARRAHP	00	34,760 K	Windows Explorer
explorer.exe	FARRAHP	00	9,180 K	Windows Explorer
firefox.exe	FARRAHP	00	130,624 K	Firefox
GrooveMonito...	FARRAHP	00	2,080 K	GrooveMonitor Utility
hkcmd.exe	FARRAHP	00	2,248 K	hkcmd Module

Figure 8: Co Linux open processes upon start up

- VM Performance

By chance it was discovered that when graphic intensive applications where functioning the background such as flash player, the linux system experiences issues with memory. There were no other abnormal or outstanding observations discovered in the course of evaluating the image contents or the host machine. Below is a memory usage command executed on the Linux system depicting the amount memory used by the CoLinux.

```
root@CoLinux:~# free –t

             total     used       free      shared buffers cached
Mem:       872100   750000   122100      0    9400    109756
-/+ buffers/cache:   65856     607308
Swap:      557048    477700   79348
 Total:   1293556   808756
```

- Vulnerabilities presented by VM

Running CoLinux involves having the host constantly switch between host OS state and CoLinux kernel state can create some instability in performance on the host. Additionally any kernel issues on the Guest OS can manifest itself onto the host because of sharing nature of the environment. Memory allocation issues can be problem with CoLinux when graphic intensive programs are running on the host PC. So it would appear that the Linux system had run out of memory. Thankfully Linux systems use swap files.   The use of the free command, or top command can give the user and idea of what processes are using the most memory and if the swap file feature is enabled. Also, since the windows files are made available to CoLinux if a Linux system is passes on file that is detrimental to a windows file system and vice versa that could make this type of virtual environment option not optimal.

- Limitations

CoLinux only virtualizes hard disk and network resources but not graphics. Additionally, the host must have an OS kernel that can handle export primitives in order to allow the CoLinux's portable driver to run in CPL0 mode and allocate memory. Lastly, in order to fully use CoLinux you need to have full administrators rights. When installed the program tells Windows its using a level driver, and it needs administrative rights to run it.

CHAPTER FOUR: RESULTS AND DISCUSSION

It was discovered that the propensity of the VM to crash is inversely related to RAM allotted to each VM in correlation to the RAM available on the host machine. As the percentage of RAM allotted to the VM increases vs. what is available the host and VM became unstable, slow, and crashed. Figure 9, below indicates the relationship between the numbers of crashes as it relates to RAM allotment for VirtualBox[1] and VMware Workstation[2]. Both Virtual Box and VMware Workstation allow the user to choose how much RAM to allocate to the guest OS, with 512 being the recommended allotment. VMware and VirtualBox were both installed on Forensic Machine 1, which had 4GB available. More crashes occurred as the memory allotted to the came closer to the amount available on the host machine. Over committing the memory caused swap file issues, which made the host unstable. In terms of computer forensics lab this could prove to be an issue if the machine doesn’t have an enough RAM which would in turn cause possible disrupting activity within the analysis phase.

[1] All references to Virtual Box within this document pertain to Oracle VM Virtual Box version 4.0.8

[2] All references to VMWare Workstation 7 within this document pertain to VMWare Workstation version 7.0.1

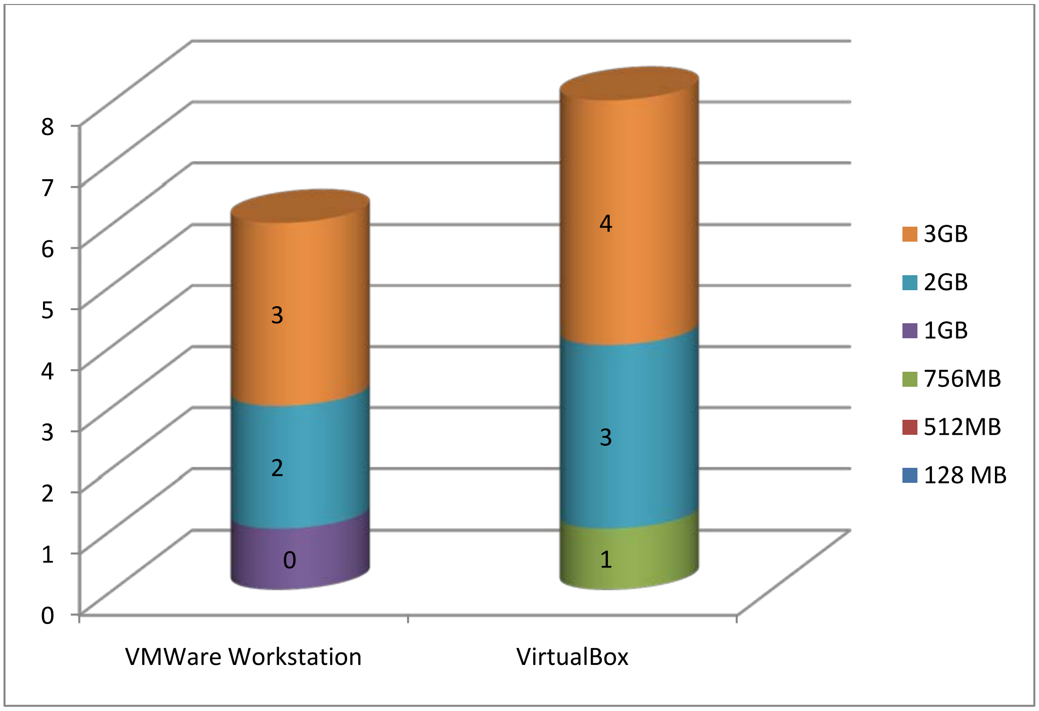


Figure 9: VM crashes in relation to physical memory allotment

Figure 10 below depicts the maximum RAM, CPU consumption, and network resource usage percentage over the 48-hour period the compromised images ran within the VM's. I suspected that since XenDesktop[3] was using the client-server model the network resource usage would be higher in comparison to the other virtualization tools in this study. XenDesktop, VirtualBox, and VM VirtualBox all ran on the 64bit machine. The left axis depicts the percentage of the resource used versus what is available.

[3] All References to Xen Desktop within this document pertain to Xen Desktop version 5.5

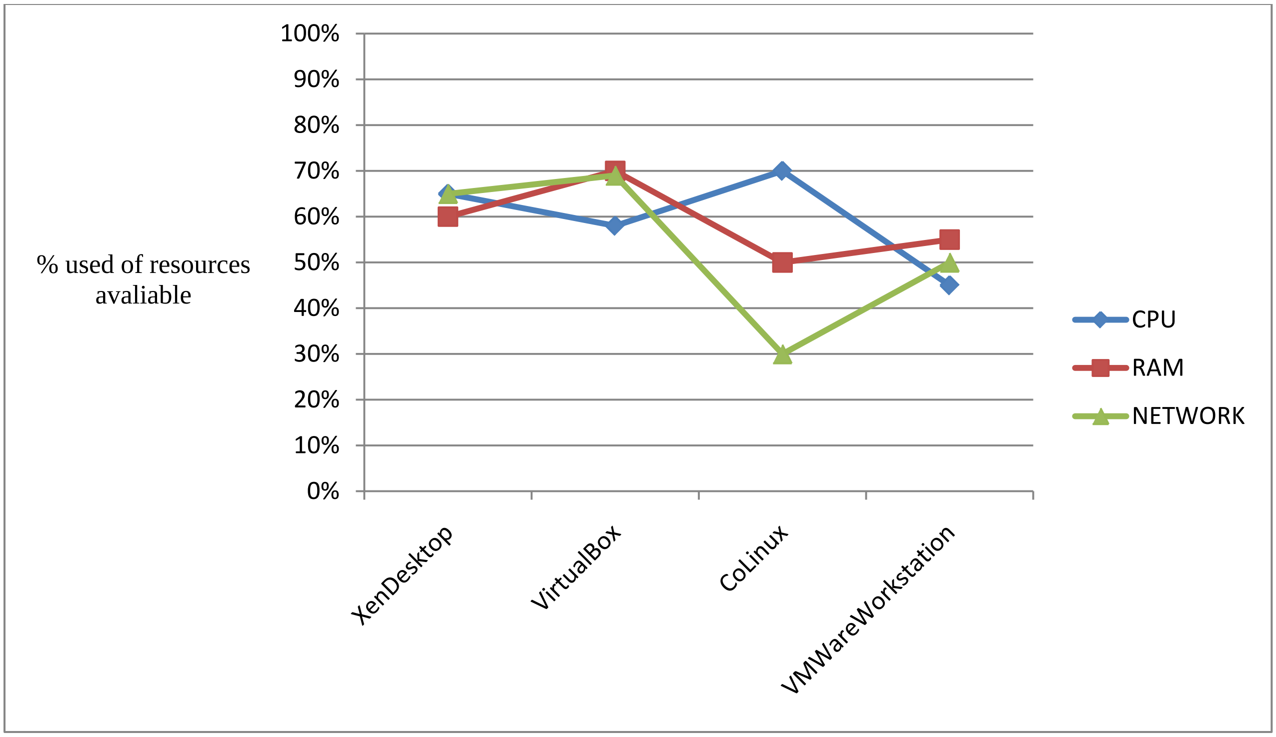


Figure 10: CPU consumption, Memory Usage, Network traffic

It was also discovered that the more time spent setting up the application meant considerable time was spent maintaining it. Figure 11 below depicts in minutes the time it took to install tool vs. the maintenance involved afterwards. XenDesktop took a considerable amount time set up.

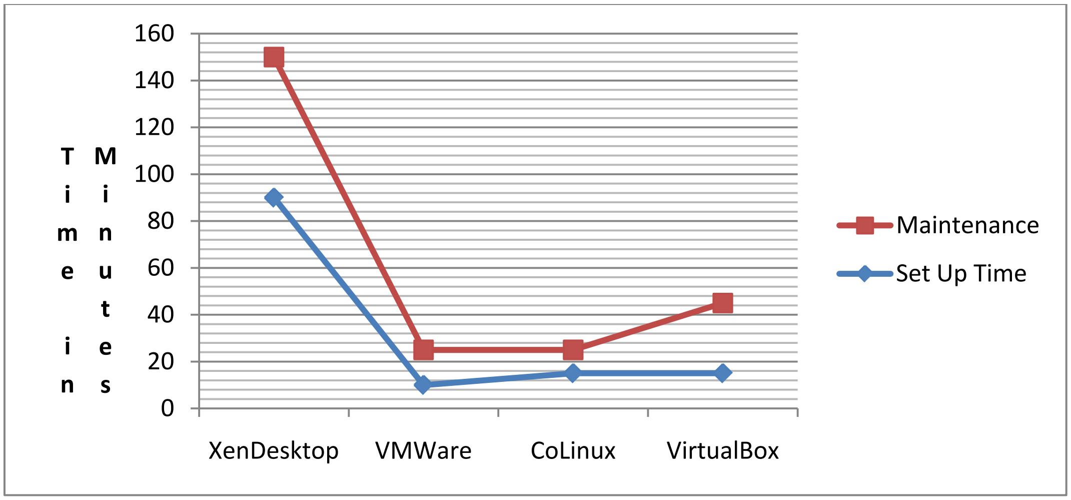


Figure 11: Set up time vs. maintenance time in minutes

Figure 12 below depicts the virtual machines effect on the host in terms of physical memory usage, host network usage, and host swap file usage as a percentage. Xen Desktop consumed the most network resources and Co Linux [4] and VMWare Workstation used the least amount consuming between 20%-25%. Swap file usage is significant measure as it gives insight to how the host is managing memory outside of RAM.

[4] All References to CoLinux within this document pertain to Cooperative Linux version 0.7.3

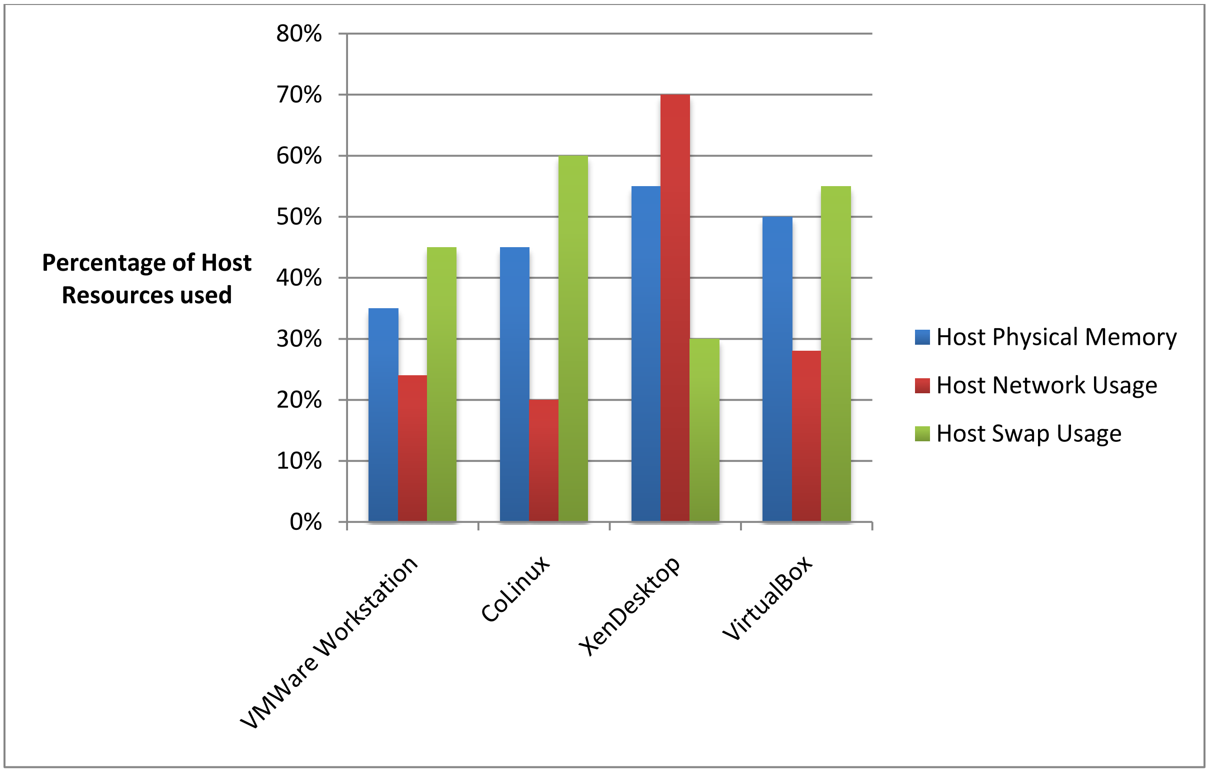


Figure 12: Host Performance with Virtual Machines running

Table 3 below demonstrates some key functional features that each Virtual Machine offered.

Table 3: Functionality Availability Comparison

Functionality	VMWare Workstation	Virtual Box	CoLinux	XenDesktop
Cloning	Yes	Yes	N/A	Yes
Networking options	Bridged and NAT	Bridged and NAT	WinPCap	
Suspend or Pause VM	Yes	Yes	No	Yes
Common Virtual file formats supported	Yes	Yes	N/A	Yes

Each virtual machine had variable limitations noted during experimentation. Table 4 below attempts to condense the limits each Virtual Machine possesses.

Table 4: Limits

Limits	**VMWare Workstation 7**	**VirtualBox**	**CoLinux**	**XenDesktop**
Many File Formats Supported	Yes	No	No	Yes
Co-dependencies outside of host	Low	Low	Low	High
File Transfer from Host to VM	Available	Not Available	Available	Available
Guest OS support	High	High	Limited	High

Here are the definitions of the before mentioned limits:

Many File Formats Supported = High number of virtual file formats supported

Co-dependencies outside of host is a determination as to whether the Virtual Machine has high dependency on other application in order to work properly. i.e. need active directory to manage users such as XenDesktop.

File transfer from Host to VM – direct availability to transfer files from host to Virtual Machine

Guest OS support – means the Virtual Machine supports many different OSes available on the market such Windows XP etc.

CHAPTER FIVE: CONCLUSIONS

5.1 Major Findings

Since digital evidence is so volatile it is imperative that the tools being used to preserve and analyze it are efficient, intuitive, reliable, and accurate. My research uncovered that while Xen is more stable but requires a lot of maintenance to maintain network connections and had many dependencies on various other applications. This could prove cumbersome to an investigator do the expediency in which some evidence needs to be examined. XenDesktop allows a lot of flexibility and options in terms establishing a master image. The tool allowed for the use of an ISO as an image, converting existing Virtual machine formats to ones acceptable by the application. The flexibility to establish a golden image from many different formats is a plus and one less potential task on the part of the investigator.

Virtual Box offers control to the investigator. It offers cloning ability so the investigator can examine the image without the trepidation of damaging the image. This is invaluable feature for any user. Virtual Box installation was very simplistic and the design of the graphical user interface was very intuitive. VMware workstation offer the same control and I found many similarities between the two from interface decision to architecture. Although, when many VM's were paused Virtual Box rendered quicker than employing the same scenario in VMWARE. Rendering on average 20 seconds quicker. VMWare Workstation and Virtual Box shared many similarities.

With CoLinux sharing hardware resources the user must always be cognizant of needs of the host and virtual environment. CoLinux can't really operate as its own machine. A virtual network was implemented between the Linux and the host system. The following quote from

CoLinux.wikia.com sums up the relationship as the following: “If you run a MySQL database on the Linux, there's no problem accessing it via TCP/IP from the Windows. From the host's standpoint, CoLinux is just another computer on the network (Cooperative Linux FAQ).”

Updates where more readily available and come at a more frequent pace for the open source tools. The closed source tools were on a more scheduled release of updates or new versions.

5.2 Implications of the Findings

Investigators should ask themselves the following: Does the tool fit the task, in regards to human resources and lab resources? Does the investigator understand the architecture of the tool to where they may understand produces undesirable behavior? Largely I found that most issues surrounded over committing RAM and causing Swap issues. In terms of design and layout all were intuitive although, XenDesktop had a learning curve to surpass. The many services required to get a complete and fully functional XenDesktop virtual machine took considerable about of trial and failure.

In my opinion XenDesktop is designed to be more of an enterprise solution. Although there can be forensic application it was not designed with a forensic investigators needs in mind.

Each VM exhibited a varying degree of unstable performance in term of the ability to execute a certain task but unstableness did not affect accuracy of the results.   I believe this instability is largely dependent on the stress and workload each VM forced onto the host system. From a budgetary stand point CoLinux and VirtuaBox would agree with most budgetary constraints forensic labs may have as they are free. Absent of any major functional flaws, there is no inherent reason an investigator couldn't count on the open source tools for analysis purposes.

5.3 Limitations of the Study and Suggestions for Further Research

The study is not all inclusive of the VM's available on the market rather it focuses on four particular VM's that span the market. Suggestions for other areas of research include examining case study where VM use has presented an issue for investigators in litigation cases and determine possible solutions to mitigate the damages.

[1] All references to Virtual Box within this document pertain to Oracle VM Virtual Box version 4.0.8

[2] All references to VMWare Workstation 7 within this document pertain to VMWare Workstation version 7.0.1

[3] All References to Xen Desktop within this document pertain to Xen Desktop version 5.5

[4] All References to CoLinux within this document pertain to Cooperative Linux version 0.7.8

LIST OF REFERENCES

Barrett, D., & Kipper, G. (2010). *Virtualization and Forensics: A Digital Forensic Investigator's Guide to Virtual Environments.* Burlington: Elsevier Science and Technology.

Bem, D., & E.H. (2007). Computer Forensic Analysis in a Virtual Environment. *International Journal of Digital Evidence , II* (2), 1-13.

Gibson, S. (2006, July 27). *Virtual Machine History & Technology.* Retrieved August 8, 2010, from GRC| Security Now!!: http://www.grc.com/sn/SN-050.htm

Hay, B., & Nance, K. (2008). Forensic Examination of Volatile System Data Using Virtual Instropection. *ACM SIGOPS Operating Systems Review , XLII* (3), 74-82.

Nance, K., & M.B. (2008). Virtual Machine Introspection: Observation or Interference? *IEEE Security & Privacy , VII* (5), 32-37.

Pollitt, M., Craiger, P., & Nance, K. (2008). Virtualization and Digital Forensics: A Research and Education Agenda. *Journal of Digital Forensics Practice , II* (2), 62-73.

Shavers, B. (2008). *Virtual Forensics: A Discussion of Virtual Machines Related to Forensic Analysis.* Retrieved August 15, 2010, from ForensicFocus: http://forensicfocus.com/downloads/virtual-machines-forensics-analysis.pdf

VMWARE. (2010). *Virtualization Basics*. Retrieved August 8, 2010, from VMWARE: http://www.vmware.com/virtualization/history.html

What is a virtual appliance? (2008). Retrieved March 11, 2011, from Turnkey Linux: http://www.turnkeylinux.org/virtual-appliance

Creeger, M. (2010). *CTO Roundtable: Virtualization Part II* Retrieved August 9, 2011, from ACM QUEUE http://queue.acm.org/detail.cfm?id=1838664

CoLinux. (2010). Cooperative Linux FAQ. Retrieved September 9, 2011, from http://CoLinux.wikia.com/wiki/FAQ

Popek,G., Golberg, R. (1974). Formal Requirements for Virtualizable Third Generation Architecture. Retrieved November 10, 2011. Association for Computing Machinery.