Segun Adodo

CNA LABORATORY ENRICHMENT BY VIRTUALIZATION

Bachelor's thesis
CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES
Degree Programme in Information Technology
May 2011

CENTRAL OSTROBOTHNIA
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Department	**Date**	**Author**
Information Technology	20 May 2011	Segun Jame Adodo

Degree programme
Degree Programme in Information Technology
Name of thesis
CNA Laboratory Enrichment by Virtualization

Instructor	**Pages**
Männistö Sakari	45 + 7(Appendix)

Supervisor
Männistö Sakari

This study is related to networking and the aim is to enrich the laboratory work aspect of the networking study module as implemented by the school's information technology department. The existing practice lab structures in the institution's information technology degree programme rely more on Microsoft windows than other operating systems. As a result, experience in non-window operating systems is low in students.

To increase the awareness of the Linux operating system among networking students, this thesis work conceived a cost-efficient system that comprises a network-aware virtual environment deployed with the Linux platform. The platform is equipped with system administration and monitoring tool and also server applications common to Unix systems. This system provides a sandbox that students can use to experiment at no risk to experience a non-windows platform and acquaint with system administrator tasks usually carried out in enterprises business setup. A prototype laboratory exercise that uses the Linux platform was developed to engage students with this operating system. Furthermore, this thesis work proposed a cable labelling system aimed at facilitating improved cable management system in the networking laboratory to enable students working in the laboratory focus on the targeted task and learn accordingly. Also, the school IT support can reallocate their time resource to more productive use.

Keywords
network, virtual, system, environment, protocol, Linux, sandbox

FOREWORD

It is a milestone achievement to begin and culminate any endeavor on an inspiring note. This variant of achievement is usually hard to come by without support from single or multiple sources. In realization of this fact, I will not but express my hearty gratitude to the author of creation for making me attain this feat.

My sincere thanks to my thesis supervisor, Sakari Männistö for his guidance and support throughout the entire thesis work process. Many thanks, also to all lecturers of the department of information technology for their several tutoring efforts aimed at making me an asset to my environment.

Finally, I wish to acknowledge the studies office personnel for their ever incredible patience and understanding, my colleagues and friends generally for their moral support during my study period in our prestigious citadel of learning. Thank you all.

Table of Contents

ABBREVIATIONS....6

1 INTRODUCTION....1

1.1 Background....1

1.1.1 Current CCNA Teaching Practice....1

1.1.2 Areas of Enrichment....3

1.2 Goals of the Thesis....4

2 THE CONCEPT OF VIRTUALIZATION....6

2.1 What is Virtualization?....6

2.1.1 The term Hypervisor....8

2.2 Why is Virtualization important?....10

2.3 Virtualization techniques....12

2.3.1 Guest Operating System Virtualization....12

2.3.2 Shared Kernel Virtualization....13

2.3.3 Kernel Level Virtualization....15

2.3.4 Hypervisor Virtualization....16

3 VIRTUALIZATION AS SOLUTION....18

3.1 Problem Domain Analysis....18

3.1.1 Understanding Existing Structure....18

3.1.2 Establishing the Need for Improvement....19

3.1.3 Requirement Analysis....20

3.2 System Design of the Sandboxed System....22

3.2.1 Top-level Details....22

3.2.2 Low-level Details....23
3.2.3 Selected Use Cases....25
3.3 Sandboxed System Implementation....26
3.4 Recommendation for Cable Management....27
4 CNA LABORATORY ENRICHMENT....29
4.1 Introduction to UNIX OS....29
4.2 Platform and Network related Applications....30
4.2.1 Domain Name System....30
4.2.2 Trivial File Transfer Protocol....31
4.2.3 Secure Shell....32
4.2.4 Wireshark....33
4.2.5 Webmin....33
4.2.5 LAMP and PHPMyAdmin....33
4.3 Prototype Laboratory Task....34
5 TESTING OF THE SANDBOXED SYSTEM....43
6 CONCLUSION....44
REFERENCES
APPENDICES

ABBREVIATIONS

CCNA	Certified Cisco Networking Associate
CNA	Cisco Networking Academy
COUAS	Central Ostrobothnia University of Applied Sciences
DNS	Domain Name System
GUI	Graphical User Interface
IP	Internet Protocol
MAC	Media Access Control
NAT	Network Address Translation
OS	Operating System
PC	Personal Computer
TFTP	Trivial File Transfer Protocol
VM	Virtual Machine

1 INTRODUCTION

1.1 Background

At present, the Central Ostrobothnia University of Applied Sciences offers a degree programme in information technology. Essentially, the programme focuses majorly on software engineering study track even as the other specialization options of Media and Telecommunication – hereafter referred to as MTel, as well as industrial management exist. The Media and Telecom option broadly relates to data networks. Its objective is to develop competency to design and implement safe-aware efficient data networks and also services to use the networks. Ways of improving this study track will form the objective of this thesis exercise. It will attempt to carefully understudy and highlight areas of possible improvement in the CNA course module of this specialization option and make recommendations as appropriate. (Guide for Thesis Writers 2008, 170.)

1.1.1 Current CCNA Teaching Practice

The current CCNA module teaching practice as offered in the information technology program is a fully developed knowledge bundle. The focus is mainly on the academic version of CCNA module. This version provides "an integrated and comprehensive coverage of networking topics, from fundamentals to advanced applications and services, while providing opportunities for hands-on practical experience and soft-

skills development" (Cisco 2011). Emphasis is on network applications, network protocols and services the lower layers in the stack provide to these applications.

The networking content of the Media and Telecommunication option is fully implemented in line with the original CCNA program designer's intent. As benefits, students who participate in the CCNA exploration modules are provided with the technical competency to build, implement, troubleshoot and protect computer networks. The combination of theory and practical hands-on knowledge have played huge role in this regard. (Cisco 2010.)

However, the computers in the networking laboratories use solely windows operating system. Students are only provided with non-administrator level privileges to PCs engaged for the laboratory tasks. Also, cable management often present a level of challenge to students as each laboratory task requires migrating from school enterprise network to the smaller CCNA laboratory network; the effect of which is settings and physical cabling alterations. This thesis undertaking proposes the use of virtualization – which will be further discussed in later sections, to address some of these situations. It is important to state that the proposed solution has already been partly implemented prior to the start of this thesis work, although it was primarily conceived to be one of the deliverables of this exercise; this is due to its urgent need and could not be delayed until the completion of this thesis work. Hence, effort will therefore also be concentrated on reporting the completed part, however, analysis and design will be discussed as pre-conceived at the conception of thesis topic and scope of work.

1.1.2 Areas of Enrichment

In the course of this research undertaking, the thesis writer discovered opportunity areas to enrich the current COU CCNA lab environment and the Media and Telecom study track generally. It is important to state that this work does not in any way consider the Media and Telecom specialization as presently being offered as less valuable. The effort only seeks to recommend enrichment points.

From preceding subsection, some of the aspects that require improvement have been briefed. The following aspects of the present COU CCNA program implementation have been identified as opportunity areas and possible solution will be discussed in the subsequent sections of this piece:

- Introduction of Linux as an alternative operating system to carryout laboratory task.
- Introduction of system administration, server applications and monitoring tools such as Webmin, DNS or TFTP setup; their usage and related lab exercises.
- Sandboxing system to grant students full privilege to operating environment and to eliminate settings alterations and migration of cabling from enterprise network to CNA local network required to begin each laboratory exercise session.

1.2 Goals of the Thesis

The goal of this effort is simply to enrich the CNA laboratory environment along the identified opportunity areas the effect of which will also be felt at the larger information technology Media and Telecom specialization option level. On a higher level of abstraction, the frequent use of the Linux operating system for desktop and server purpose broadens student knowledge on contemporary OSs. It increases familiarization and builds confidence in students to increasingly consider the use of UNIX OS like Linux for personal use. The fact that moderate low level operational knowledge is required to fully exploit the possibilities offered by this variant of OS not only increase technical knowledge but also widens student thinking horizon. It will represent additional knowledge for participants of the CCNA studies.

Also being proposed is the use of system administration, server applications and network monitoring tools. These include: Wireshark, DNS, TFTP, SSH, Webmin, PHPMyAdmin, Apache and the open source database, MySQL. These tools are (and many others) constantly put to use in enterprise networks across many business organizations. Theoretical and practical knowledge of the underlying concepts behind the workings of these applications are extremely valuable. A prototype lab to unravel the possibilities offered by the selected tools in this category in UNIX environment will be developed, experimented and presented as part of this project's deliverables. The value is embedded in the UNIX experience.

To streamline the laboratory task routines and achieve good abstraction so that students can focus on learning only, this work proposes an improved method of cable

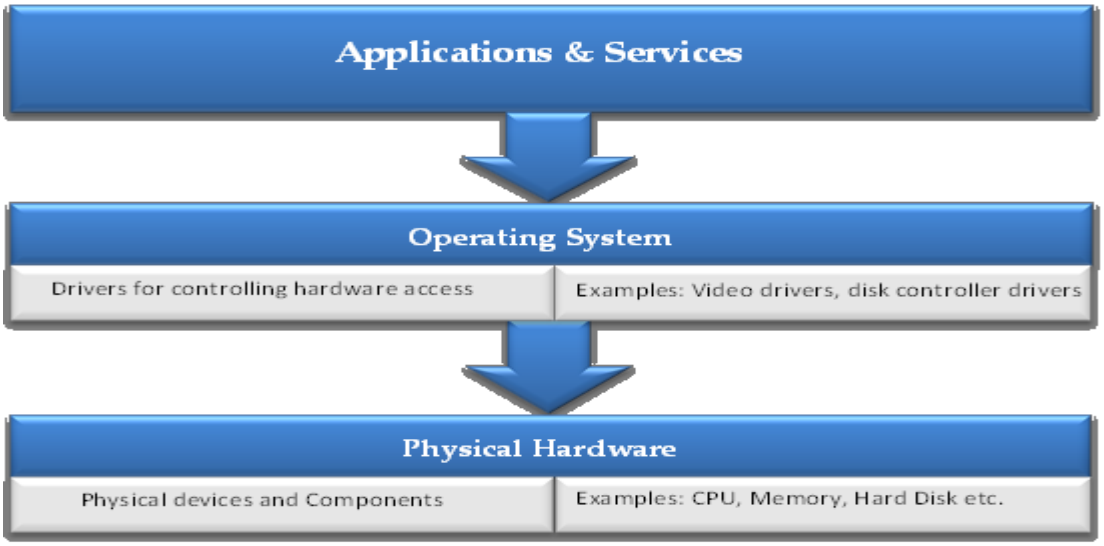
management, provisioning of system that eliminates privileges restrictions and settings alterations required to begin each laboratory exercise session (since it adds no value). By implementing this proposal, course participants are better suitably positioned to benefit from hands-on laboratory activities. All the aforementioned form the objectives of this thesis undertaking.

2 THE CONCEPT OF VIRTUALIZATION

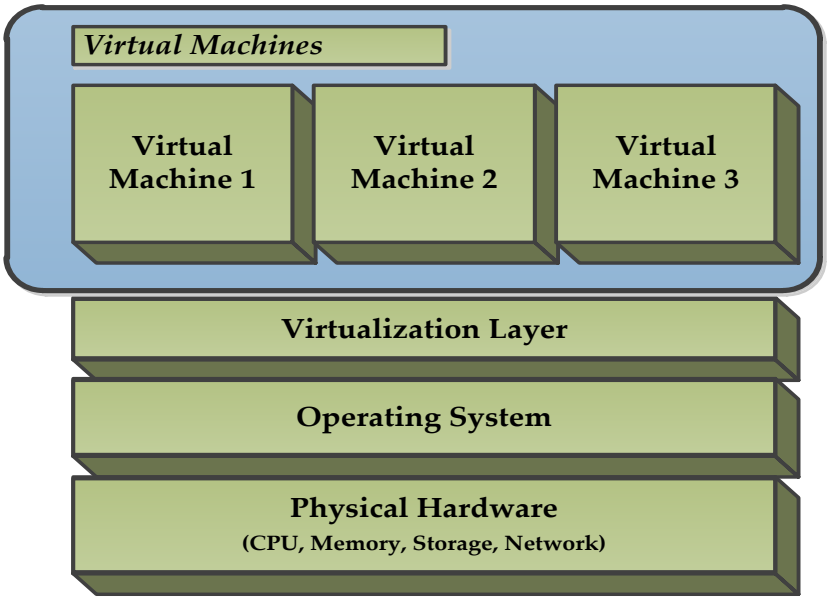
2.1 What is Virtualization?

In a typical computing system paradigm, technological resources such as servers and desktop computers run on single operating systems. Each hardware has its own individual OS installation. A concept referred to as virtualization extends this traditional model. It facilitates distributive use of available technological resources to match business needs. More technically, virtualization involves running multiple operating systems on the same physical machine hardware simultaneously and each of the OS shares the resources of the host hardware system. The shared resources include CPU, memory, storage and network connectivity. (Desai 2007, 3-4; Virtuatopia 2010.)

Virtualization infrastructure simply simulates multiple computer systems. For instance, running windows Vista and Ubuntu in a virtualized environment on a windows 7 host machine all on a single computer system eliminates the need for three physical computers to run each of the OSs. Graph 1 illustrates the relationship between OS and hardware. (Virtuatopia 2010.)



GRAPH 1. Illustration of relationship between OS and hardware (adapted from Desai, 2007)



GRAPH 2. Structure of Virtualization (adapted from Desai, 2007)

Graph 1 depicts operating system as a layer just above the hardware. Application and service are layered on OS. From Graph 2, it can be observed that an abstraction layer is created serving as an interface between the virtual machines and the OS controlling access to system resources.

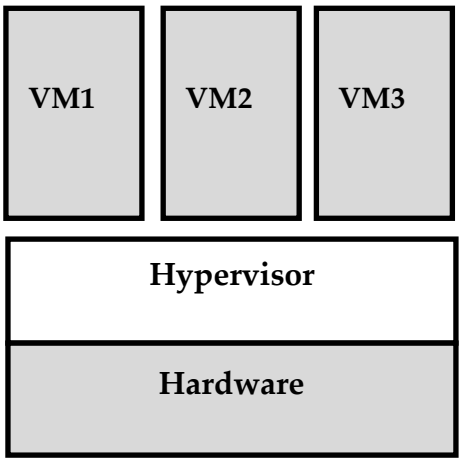
2.1.1 The term Hypervisor

The terminology hypervisor is synonymous with IBM. It dates back to 1972, when the control program for the System/370 mainframe was re-adapted to be virtualization implementation capable. Hypervisor as a technology represented a mileage achievement in computing as it eliminates "architectural limitation" and exorbitant cost of engaging mainframe systems. (Mitch 2009, 23-26.)

It is a virtualization software that emulates, in its entirety, hardware environment thereby creating what is called virtual machine monitor. Multiple operating systems embedded in virtual machines can be easily loaded to the VMM. Hypervisor is used to serve this purpose and to handle access requests between guest OS running in the VMM and native hardware resources present in the physical computer. There are different variants of Hypervisor obtainable and are classifiable by type (i.e. does it runs directly on bare hardware or within an OS) and design – i.e. monolithic or microkernel. Only hypervisor by type -classification is described in the succeeding sections. (Mitch 2009, 23-26.)

Type 1 Hypervisor

This type of hypervisor runs directly on bare metal, in other words, on hardware of the host computer and works as a "control program". Individual guest operating systems can then run on the multiple virtual machines placed above the hypervisor as illustrated in the Graph 3. (Mitch 2010, 24-27; Benard 2009, 14)

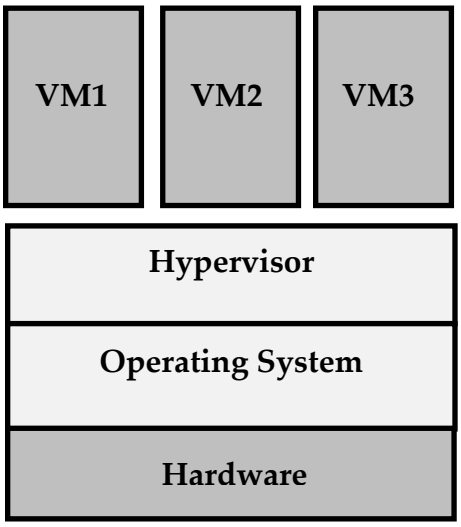


GRAPH 3. Type 1 Hypervisor (adapted from Mitch 2009, 23)

Generally, Type 1 hypervisors are known to provide best performance due to direct interaction with bare metal.

Type 2 Hypervisor

Also called *hosted virtualization*, type 2 hypervisor is designed to run inside an operating system running on a host system (i.e. computer). In this case, the hypervisor is not layered to interact directly with the native hardware. Guest OSs will run inside virtual machine layered just above the hypervisor as shown in Graph 4. (Mitch 2009, 23-24.)



GRAPH 4. Type 2 Hypervisor (adapted from Mitch 2009, 24)

2.2 Why is Virtualization important?

The enterprise landscape has witnessed tremendous change over time. World economy is consistently producing challenges of threatening scale, and this is particularly evident in the dire quest by firms to sustain business continuity. Business enterprises are coerced to seek ways of trimming cost in all their operations. Since information system forms an integral component of any organizational setup, deploring resources to efficient and cost-savings solutions is expedient for survival. By implementing virtualization certain yearnings of businesses can be offset. The following expatiates upon the benefits accrued to this technology. (Golden 2010, 3.)

At present, many desktop and data centers implement their operations at a greatly reduced capacity. Hardware resources of computing systems are grossly under-

utilized. Statistics unveils that data centers only utilize 10 or 15 percent of their total processing capacity. In effect, more than 80 percent of hardware resources are lying fallow. Virtualization solves this redundancy problem by allowing multiple systems to run on a single hardware. As an effect, resource utilization level is elevated to 70 or 80 percent. The extended effect is reduced energy consumption since, for instance, multiple servers are replaced with single physical server, and less number of servers is used. (Golden 2010, 3-9.)

For firms operating huge data centers, lot of spaces are lost to multiple hardware systems and they are now fast running out of space. This development is not unconnected to the latest trend of non-paper based operations adopted by organizations. The capability of virtualization technology to host multiple guest OSs on single hardware (i.e. server in the case of data centers) paves way for space recovery and eliminates the need to invest in building expensive data centers or server rooms. Same volume of data can be stored on same hardware nevertheless with less space consumption. (Golden 2010, 3-9.)

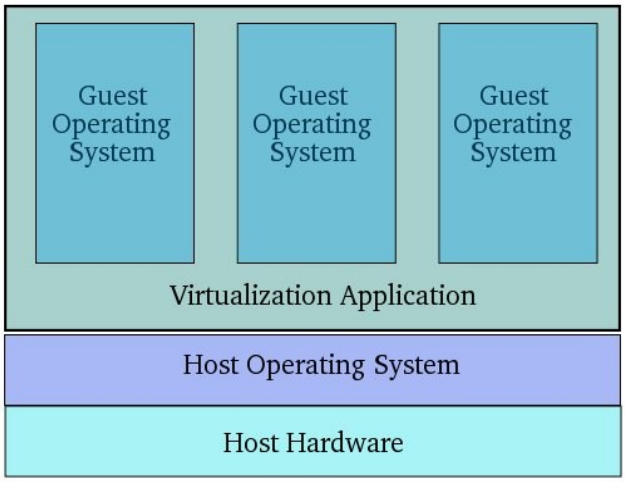
Additionally, it is a common practice especially in computing resources-intensive setup to build IT disaster response system to quickly restore IT operations in the event of disaster. This system is implemented in form of second backup which translates to extra expenses for the business. Maintaining data consistency between backup data centers can also be daunting as data easily become out-of-date. Virtualization mechanism can be engage to provide a rather cost effective solution. “Virtual machines can be easily transferred within seconds or minutes to a backup data center; in tough circumstances, many virtual machines can be run on a smaller

number of physical servers, reducing the cost of physical resources required for disaster recovery". (Golden 2010, 8-9.)

2.3 Virtualization techniques

2.3.1 Guest Operating System Virtualization

This is the simplest technique easily understandable. In this technique, a host computer runs traditional operating system such as windows or UNIX. Similar to other application software - graphics editing, word processor or media player, virtualization software is installed on the OS. It is inside this virtualization application environment that one or more virtual machines are created and run. The role of the virtualization application involves stopping, starting and administrating each virtual machine. It also includes managing interaction of guest OS with host system hardware. Virtualization applications are also responsible for binary rewriting which essentially involves instruction stream scanning and replacement of privileged instructions with safe and secure emulations; these instruction streams emanate from the guest operating systems. Binary rewriting process creates false impression to the guest OS that it interacts directly with system hardware. VMware workstation and virtualbox are some of the example guest operating system virtualization technologies. Graph 5 below illustrates this technique. (Virtuatopia 2010.)



GRAPH 5. Guest Operating System Virtualization Technique (adapted from Virtuatopia 2010)

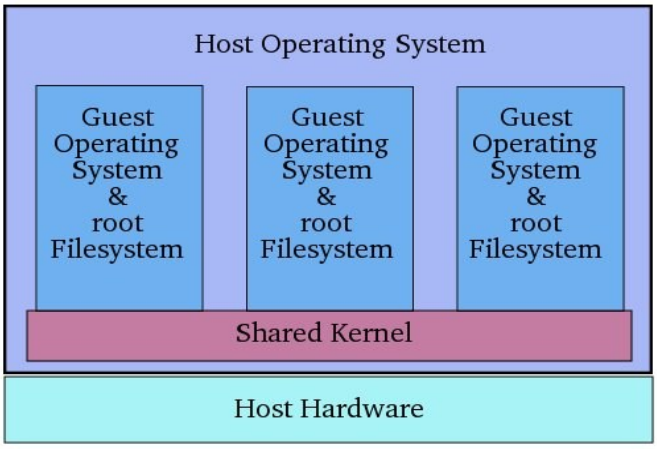
Graph 5 describes guest operating system running in virtual machines which however operates within the virtualization application installed on OS in like manner as any other application.

As shown above, the multiple abstraction layers between guest OS and system hardware inherent in this technique is less supportive of high virtual machines throughput for critical uses; however, it suffices and will be engaged for the purpose of this thesis project undertaking. (Virtuatopia 2010.)

2.3.2 Shared Kernel Virtualization

This technique is also referred to as system level or operating system virtualization. It exploits the design architecture of Linux and UNIX- oriented operating systems. The

architecture of this operating system supports a concept called *chroot.* As generally known, UNIX based OSs comprise two main components, kernel and root file system. The root file system includes libraries, files and utilities. Kernel essentially manages interaction between operating system and the hardware. Chroot involves the capability of the kernel to dynamically swap running root file with another different root file system without recourse to system reboot. Shared kernel virtualization implementation is such that each guest operating system has its own root file system but share the kernel of host operating system. The following figure illustrates the Shared Kernel virtualization technique. (Virtuatopia 2010.)

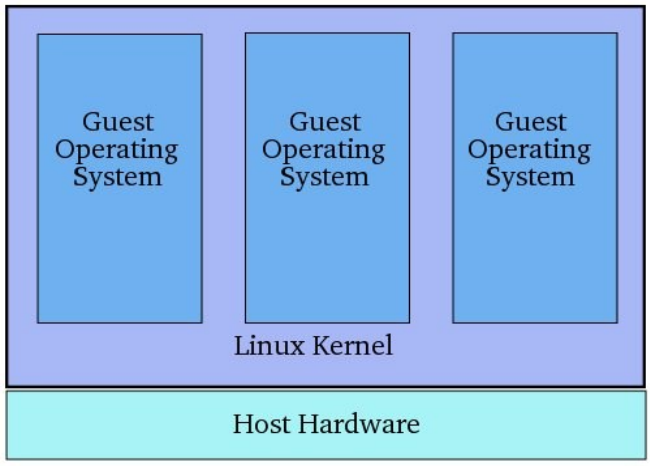


GRAPH 6. Shared Kernel Virtualization Technique (adapted from Virtuatopia 2010)

Graph 6 diagrammatically describes the Shared Kernel virtualization technique. It is relevant to note that this technique requires guest OS be compatible with the kernel (i.e. kernel version must be compatible guest OS version). (Virtuatopia 2010.)

2.3.3 Kernel Level Virtualization

This technique involves custom-designed operating system specifically optimized with kernel which "contains extensions designed to manage and control multiple virtual machines". The individual virtual machines contains guest OS. In simpler terms, each guest OS has its own kernel unlike Shared Kernel technique where kernel is shared. However, the guest OS must be compatible with the kernel in which its runs. (Virtuatopia 2010.)



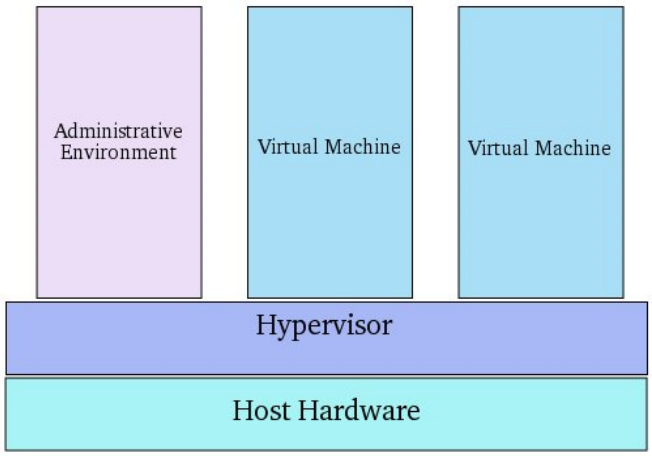
GRAPH 7. Kernel Level Virtualization Technique (adapted from Virtuatopia 2010)

The figure above – Graph 7, is an illustration of the Kernel Level virtualization technique. User Mode Linux and Kernel-based virtual machine are some of the virtualization technologies using this technique. (Virtuatopia 2010.)

2.3.4 Hypervisor Virtualization

There are variants of "protection levels" or *rings* in x86 CPU family (Virtuaopia, 2010).The highest low-level privilege is obtainable at ring 0 and this is the mode in which operating system kernel operates. Other application codes run in lesser ring, ring 3. (Virtuatopia 2010.)

Hypervisor Virtualization implementation uses hypervisor. The hypervisor interacts directly with the host system hardware by running in *ring 0.* It is responsible for the management and allocation of system resource and memory for the virtual machines. Additionally, it provides interfaces for tools useful to manage the virtual machines. An overview of the hypervisor virtualization technique is illustrated in Graph 8 below. (Virtuatopia 2010.)



GRAPH 8. Hypervisor Virtualization Technique (adapted from Virtuatopia 2010)

However, on account of the fact that operating system kernels are designed to run in ring 0 which hypervisor will not allow in this case, as only it is privileged to run in

ring 0 mode, several other hypervisor virtualization based solutions were developed to eliminate this challenge. (Virtuatopia 2010.)

3 VIRTUALIZATION AS SOLUTION

3.1 Problem Domain Analysis

3.1.1 Understanding Existing Structure

The current laboratory environment relies on windows operating system. All laboratory tasks and exercises are carried out in this familiar environment. This has benefited students by increasing their windows OS awareness. Resultantly, focus on other contemporary platforms as will be expected is therefore low. Also, the existing teaching structure has no concrete system to explore the fundamentals of system or network administration on UNIX. Lastly, before now, the CNA hands-on laboratory exercises always encourage system exposure. As it is the case, the Instructor is compelled to grant students privileged access to production PCs the adverse effect of which is only limited to imagination. Overall, the present structure (note: this is no longer the case for solved issues as stated in the beginning of this piece) highlights the adumbrated issues.

As an unrelated concern, cable handling and management especially during laboratory works and when IT personnel carry out maintenance on workstations constitute a matter for concern. The present PC cabling arrangement on the CNA local network is such that color coding scheme combined with LAN port labeling are engaged to facilitate cable continuity troubleshooting during lab hands-on. It principally serves to aid network cabling. Owing to the fact that IT personnel are unaware of the color coding scheme (and perhaps, since the scheme is loose it may be

ignored), in the course of maintenance or other operations the coding scheme may be disoriented and correcting this becomes an avoidable routine tasks. It is also the case that students are at times confused about the coding scheme convention and may equally disorient the color coding arrangement. The problem is more apparent when productions PCs are migrated between the CNA local and production network and IP settings re-configured accordingly. This routine operation is required for individual laboratory hands-on sessions.

3.1.2 Establishing the Need for Improvement

The reality of present-day information technology age and the challenges of satisfying the ever insatiable consumers of information system products and services leave educational institutions with no option than to positively react to the industry stimuli. It will be totally out-of-place for institutes of learning focused on the produce of resource personnel in information technology related disciplines to ignore these stimuli. The crust of the challenge is to advance into market competitive and highly versatile minds that have the knowledge, skills and technical know-how to solve problems and consequently create value. Any lag in this category results to less marketable products lacking in knowledge versatility. With this in mind and considering the case of the largely mono-platform experience and the lack of basic system administration skills highlighted in preceding subsection, a system must be provisioned to tackle this issue. This system will also solve problems related to undue access to administrative privileges.

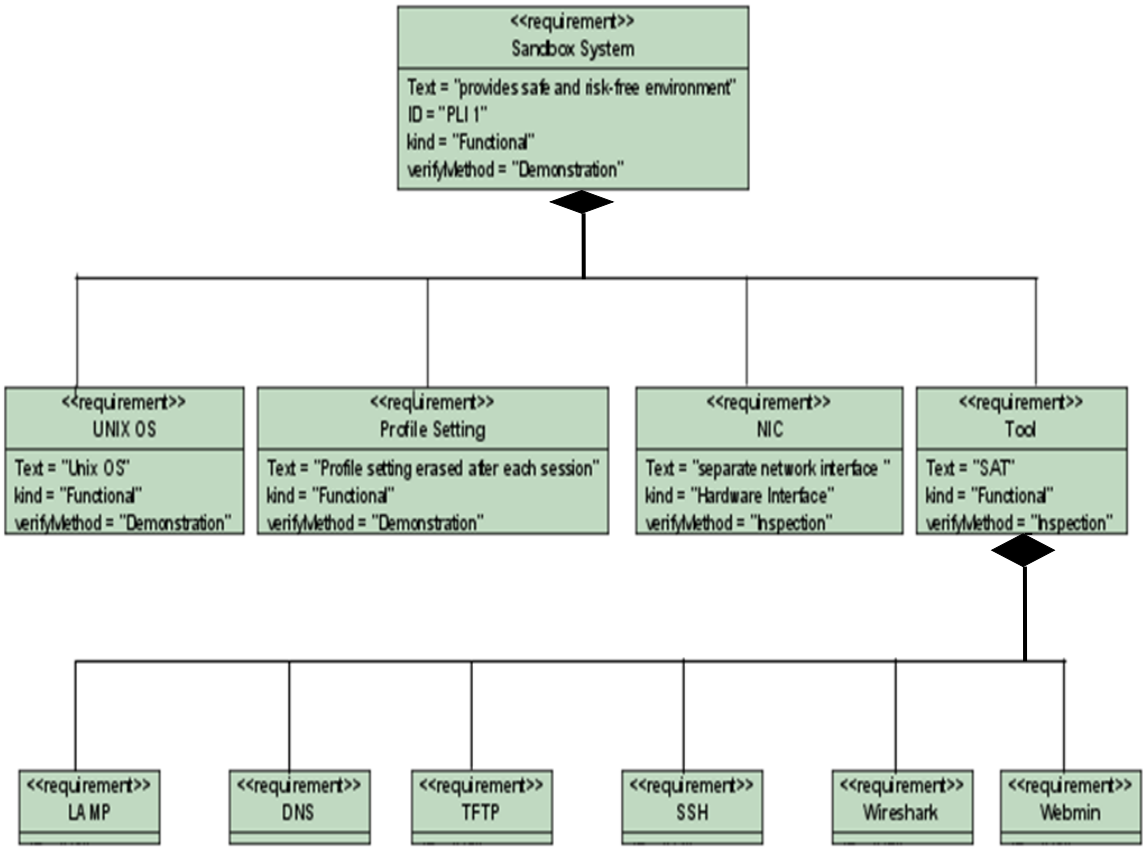
Also, it is ineffective use of time resources to dedicate attention to avoidable routine especially when it adds little or no value to the learning process. Since sustaining the cable color coding scheme and general cable handling is integral to the CNA lab exercise, this thesis undertaking highlights the need for a cable labeling system that will aid cable handling.

3.1.3 Requirement Analysis

To exploit the opportunities presented by the improvement points outlined in previous section this thesis proposes a composite solution, a sandboxed system environment and cable labeling system.

The sandboxed system environment is to, in the minimum, fulfill the following:

- The system runs popular, simple and lightweight open source UNIX OS that is easily manageable
- The user shall be able to create multiple instances of OSs
- The OS instance created by user shall not alter configuration settings on windows OS
- The system discards profile settings at end of each usage session
- It shall provide separate network interface for connection to private CNA lab network
- The system shall provide common network related applications (i.e. DNS, TFTP, SSH, Wireshark, Webmin, PHPMyAdmin) and the LAMP platform. The following is the requirement diagram.



GRAPH 8. Sandboxed System Requirement Diagram

Also, the cable labeling system is to fulfill the below:

- label inscription explicitly points cable to its housing port
- inscription facilitates and guides reconnecting displaced or disconnected cable
- eliminates concern for proper cable connection

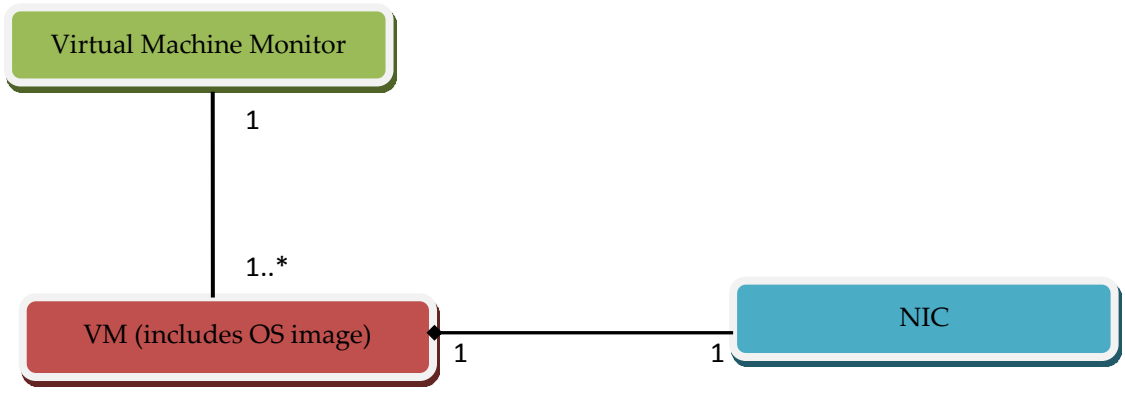
GRAPH 8. Sandbox System Requirement Diagram

3.2 System Design of the Sandboxed System

The sandboxed system is essentially meant to provide platform to facilitate the introduction of lessons related to system administration tools using Linux platform. It further serves to eliminate production PC exposure when students are issued with administrator privilege to carry out laboratory exercises. Also, configuration alterations on production PCs common due to the need to convert these same PCs for labs uses will be eliminated. The following section uses software modeling technique to describe details of the sandbox system.

3.2.1 Top-level Details

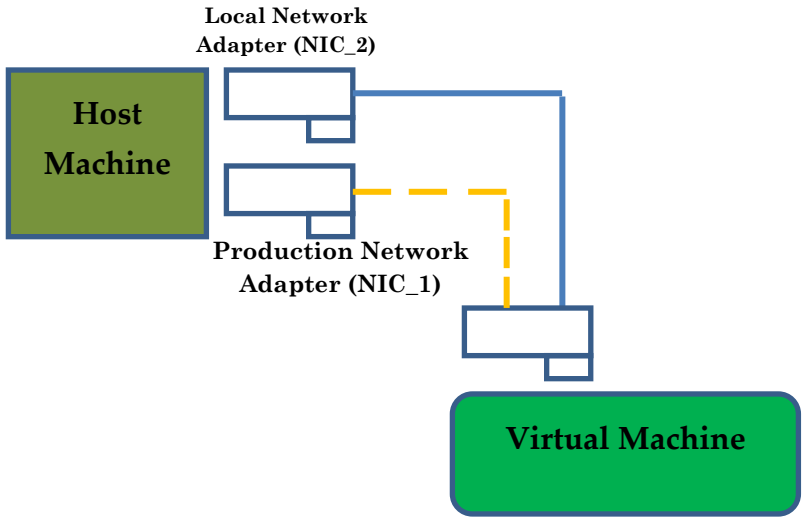
The sandboxed system is composed of hypervisor, virtual machine OS image and network interface card. The hypervisor simulates hardware by providing the required instruction set architecture to the virtual machine (Desai, 2007). However, the virtual machine image - which is the operating system, is used to create a virtual machine while the NIC functions to connect the system to network. The figure below describes these three main objects. It also shows the interaction between them.



GRAPH 9. Top-level Diagram of the Sandboxed

3.2.2 Low-level Details

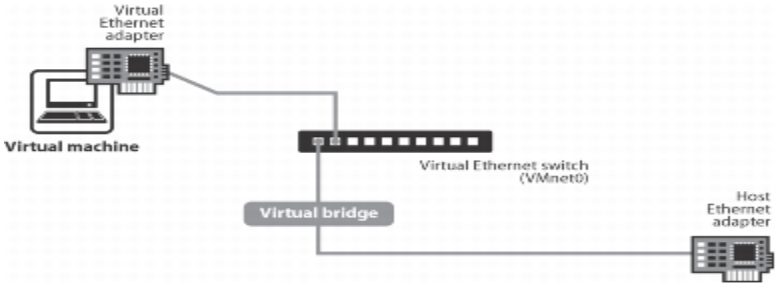
The following is the low-level details of the NIC object in the system above.



GRAPH 10. Low-level Diagram of the NIC Object

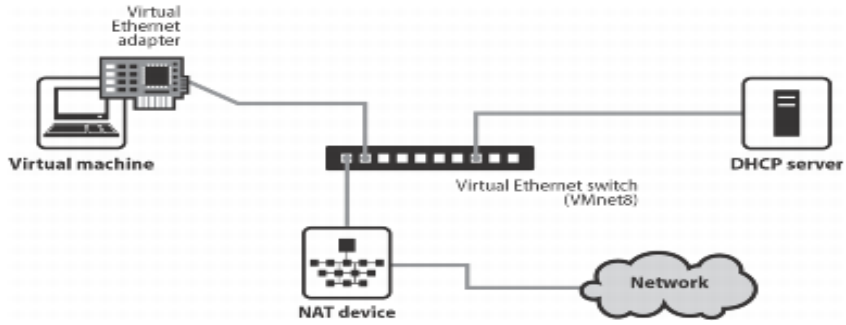
In the figure above, there are two NIC cards on the host machine. NIC_1 connects host machine to production network and NIC_2 to local CNA network. The virtual machine can be configured to engage either of the adapters depending on the desired network. From the figure, the broken arrow indicates another networking possibility as the two connections can only function one at a time.

Also, the Graph presents two networking scenarios, NAT and bridged. The NAT network connection to the virtual machine is implemented with NIC_1 adapter while "bridged" derives from NIC_2 network adapter. The bridged connection is facilitated by the virtual bridging service facilitated by the hypervisor and provides the virtual machine with a discrete identity and "full participant" status on the local network. Graph 12 below describes the bridged-networking further. (VMware 2011.)



GRAPH 11. Bridged-Network Networking (adapted from VMware 2011)

Network address translation provides the virtual machine connection to TCP/IP network (e.g. internet) using the host PC internet connection – such as broadband. The virtual machine can use NAT to communicate with extraneous network via standard TCP/IP protocols such as Telnet, HTTP and FTP. The benefit of NAT to this design is the possibility to refrain from allocating IP addresses to individual virtual machines; in lieu of this, the host system creates private network and an in-built feature VMware virtual dhcp server which allocates addresses from this network to VMs and manages packet dispatch and delivery(both intra and inter) for the virtual machines. Consider the figure below. (VMware 2011.)



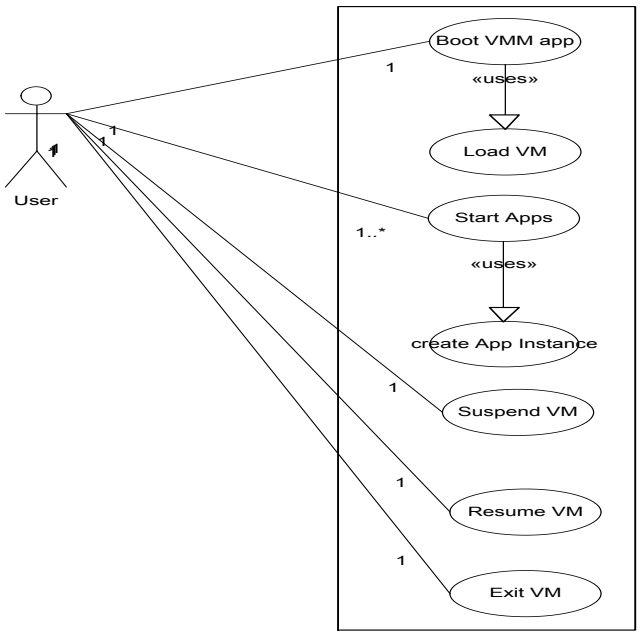
GRAPH 12. NAT Networking (adapted from VMware 2011)

NAT in this application enables the virtual machines to be isolated from external networks; this forms another benefit of NAT to this design. The bridged network however serves to simulate real PC by offering full identity to VMs on the local CNA network. Also, it eliminates concerns on granting administrative privileges to

students on production PCs. The alteration of IP configuration settings on production PCs during hands-on lab exercises and its consequent effect of offsetting the base configuration of production PCs are completely designed out. (VMware 2011.)

3.2.3 Selected Use Cases

The sandboxed system will be put to different use case scenarios. Graph 13 below illustrates selected use cases of the system.



GRAPH 13. Sandboxed System Use Case

A use case description of a typical usage session of the system is presented below. It integrates all use cases highlighted in the graph above.

Use Case Descriptions:

Use case Name: Setup virtual machine in CNA network.

Version: 1.0.

Summary: Use case describes the process of booting virtual PC to desktop and setup machine in CNA local network.

Frequency: As may be needed.

Usability Requirement:

Actors: Student, instructor.

Precondition: Virtual machine has been pre-created and host PC is running.

Description: User starts up the virtual machine monitor, edits networking configurations on hypervisor to select either bridged or NAT and loads virtual machine to virtual machine monitor. System boots to desktop. User configures the machine with IP address and subnet mask to be able to participate on targeted network.

Exception: None.

Illustration: NA.

Post-condition: On success, virtual PC is setup in the CNA network.

3.3 Sandboxed System Implementation

The sandboxed environment was setup as prescribed in the design section. All existing PCs in the CNA laboratory were provisioned with each of extra network interface card, VMware workstation virtual appliance and Live Ubuntu virtual machine.

As stated in preceding sections, the network card component of the design is dedicated to secure virtual machines with connection to the local CNA network. On the settings sections of the virtual machine, two interfaces were created (i.e. Bridged and NAT network). For bridged network option, the virtual machine shares the newly installed network card with host system to access the CNA local network; however the guest virtual machine has its full blown identity – MAC address, on the local network. Owing to the NAT service pre-installed in the virtual appliance, the virtual machine can also access internet by NAT network as explained in section 3.2.2. For reasons of consistency, the Live Ubuntu OS (i.e. of the virtual machine) has been pre-configured not to store profile settings. At the end of each lab session, all user settings configured by students are discarded once virtual machine is shutdown or rebooted. The benefit of this is to provide a tidy computing environment to enable students focus on targeted task.

A test Virtual machine was launched on the CNA LOCAL network by removing NAT card from virtual appliance so that only bridged network is available before booting virtual machine. To connect to internet however, the NAT card is added to the virtual machine at shutdown time and rebooted accordingly. The entire system satisfied the pre-defined design and requirement specifications in section 3.2.

3.4 Recommendation for Cable Management

The cable labelling system requires little design work and as such this thesis exercise, in this category, focuses on recommending solutions that suffice to satisfy the initially stated requirement criteria in the analysis section.

In fulfillment of the requirement specification of the labelling system, this thesis proposes a labelling scheme as below:

"IP": "Network" e.g. 10.177.3.166: CNA_NET, 10.177.3.165: PROD_NET

"IP" is substituted with the PC IP address and "Network" is replaced with the running network which could either be CNA network (i.e. CNA_NET) or production network (i.e. PROD_NET); this idea works since each PC in the CNA lab are individually tagged with IP. The "IP" simply identifies the individual PCs and "Network" differentiates between production and non-production network. The label is strapped around the cable neck at both ends and it will be such as in the figure below:

GRAPH 14. Tie-on Wrap Cable Tagging

4 CNA LABORATORY ENRICHMENT

4.1 Introduction to UNIX OS

As part of this thesis undertaking, UNIX-like operating system is being postulated for consideration as an integral element of the IT degree program of COU UAS. The status quo deploys meager attention to this OS variant. The succeeding sections will examine this operating system and some basic system administration tools commonly used with the UNIX platform.

Unix OS is an operating system originally developed by AT&T in 1969. It is built with multi-user and multitasking capability. The popularity of UNIX that started in the 1970s stemmed from its penetration into the academic community owing to its modular design that is said to aid teaching. The University of California BSD variant specifically contributed immensely to the huge wide-scale adoption of UNIX. System V and V7 were the early release of this OS. Others are Digital Unix, HP-UX and MAC OS. Essentially, operating systems that have same characteristics as these releases are generally accepted as UNIX OS while others that only "resembles" are considered UNIX-like systems. Examples of UNIX-like kernels include: FreeBSD, OpenBSD and NetBSD and Linux(less technically speaking) which is of specific interest to this thesis undertaking. (Daniel & Marco 2005; UNIX 2003.)

"Linux is a member of the large family of UNIX-like operating systems" (Daniel & Marco 2005). The Linux kernel was developed by Linux Torvalds in 1991. It is a

typical example of free and open source software which translates to the potential legal possibility for anyone to modify, redistribute the kernel freely for commercial and non-commercial intent under the GNU General Public License. In the attempt to transform Linux kernel into full-fledged operating platform the GNU free software foundation provided the required supporting utilities and libraries which all integrated together with the kernel to form the platform officially termed GNU/Linux. Generally, Linux is packaged in distributions. Debian, OpenSUSE and Fedora are some of the mainstream Linux distros. There are however, derivatives of Debian one of which is Ubuntu. Ubuntu is the Linux distro deployed for the purpose of this thesis work. (Daniel & Marco 2005; Linux 2011.)

4.2 Platform and Network related Applications

This thesis selected some of the common applications in this category. More precisely, Webmin, DNS, TFTP, SSH, Wireshark PHPMyAdmin and LAMP tool pack. These tools and platform are discussed in the following sections.

4.2.1 Domain Name System

Domain name system, also DNS is a hierarchical namespace network naming service that uses distributed database to translate hostnames to IP address and vice versa owing to the fact that network devices only understand numerical representation of addresses. This service is used by resources such as computers, mobile equipment

connected to internet ecosystem to communicate. This service is used by TCP/IP applications to facilitate routing. The protocol that enables communication between servers and clients is provided by DNS. The concept is such that each internet service providers have DNS server which can be queried by other external systems as well and this explains the distributed nature of the DNS database. (Daniel & Marco 2005; Richard 1993.)

This understanding can be demonstrated in a small-size laboratory local network where hosts engage the service of DNS to implement IP resolution. An activity of this nature will provide students with expanded understanding of how DNS works. (Daniel & Marco 2005.)

4.2.2 Trivial File Transfer Protocol

TFTP (trivial file transfer protocol) is an application layer protocol. Like FTP, it simply serves to implement automated file transfer operations. It is useful to transfer boot and configuration files between hosts - particular diskless nodes, in local networks; as it concerns this thesis, TFTP is useful to implement backup and restoration of configurations and images in CNA network. TFTP is generally considered as a scale-down version of FTP. FTP is relatively heavy-weight as it is full-featured to perform wider array of file transfer and manipulation operations and provides complex functionalities. However, due to FTP's complex implementation nature and for cases where only basic functionalities are required, TFTP was

developed. Students can acquaint and experiment with this application and use it as file transfer solution in more complex laboratory tasks. (tcpipguide 2005.)

4.2.3 Secure Shell

Secure shell or SSH is an application layer network protocol that facilitates data exchange between two nodes on a network using "secure channel". It is a "public-key based authentication protocol" (Wembao 2003). Essentially, SSH protocol allows a user to log onto a remote machine or host over vulnerable network link such as internet but in a secure manner. With remote connection to geographically displaced hosts, it is possible to safely execute commands on the remote machines. The protocol is specifically built for and works mainly on UNIX machines (widely used on UNIX servers) owing to UNIX's support for interactive command sessions for remote users. (Wembao 2003.)

Secure shell is designed to replace Telnet and other remote shells due to their insecurity levels. Telnet for instance dispatches packets in plain text which can be intercepted and sniffed by packet sniffers and analyzed with analyzers. It is the security exposure accruable to existing remote shell solutions that prompted the development of SSH. This is expository to students. Students can configure routers and switches from same terminal without changing physical port connection between these devices. The knowledge and usage of this tool often in labs creates the culture of security and encourages students to be professionally-minded in their approach to solving problems. (Wembao 2003.)

4.2.4 Wireshark

Wireshark is a tool for network troubleshooting, evaluating network security issues, to debug and validate protocol implementations during software development. For the purpose of this thesis, it is useful for packet analysis paving way for students to examine the network protocol details in a UNIX environment. It works by capturing network packets and generating details about the captured packets. (Wireshark 2011.)

4.2.5 Webmin

Webmin provides web-based interface to administer UNIX systems. With this tool it is possible to implement basic operating system configuration operations such as creating and managing users, disk management. It also makes it possible to manage applications such as Apache server or PHP. It simplifies the process of managing UNIX system with the user-friendly GUI concept. Networking students can use this tool to manage their Linux system in a faster way instead of resorting to console commands as the case may demand. (Webmin 2009.)

4.2.5 LAMP and PHPMyAdmin

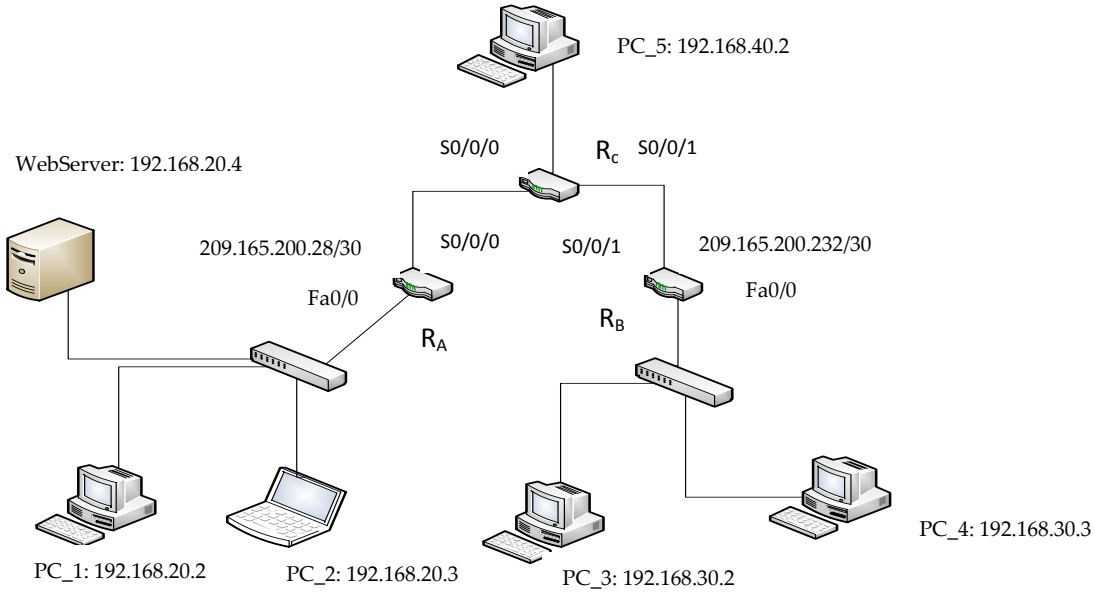
LAMP is an acronym for a bundle of open source software. It comprises Apache HTTP Server, MySQL and PHP. Apache is a webserver for hosting web pages;

MySQL is a database application used to build databases (e.g. for web applications) while PHP is a programming language for creating dynamic webpages that can be served on webservers. PHPMyAdmin provides graphical interface to the MySQL database application providing an alternative to administering the database from the console. (Paul 2006; Apache 2011; Phpmyadmin 2011; Php 2011.)

These applications can work in an integrated manner to manage websites. The operations implemented by these tools are common in enterprise business environments and will therefore represent a valuable benefit for students to understand and know how the tools are used.

4.3 Prototype Laboratory Task

TOPOLOGY:



Addressing Table

TABLE 1. Addressing Table

Device (Hostname)	**IP Address**	**Subnet Mask**	**Interface**
Webserver	192.168.20.4	255.255.255.0	NIC
PC_1	192.168.20.2	255.255.255.0	NIC
PC_2	192.168.20.3	255.255.255.0	NIC
PC_3	192.168.30.2	255.255.255.0	NIC
PC_4	192.168.30.3	255.255.255.0	NIC
PC_5	192.168.40.2	255.255.255.0	NIC
R_A	209.165.200.230 192.168.20.1	255.255.255.252 255.255.255.0	S0/0/0 Fa0/0
R_B	209.165.200.234 192.168.30.1	255.255.255. 252 255.255.255.0	S0/0/1 Fa0/0
R_C	209.165.200.229 209.165.200.233 192.168.40.1	255.255.255. 252 255.255.255.252 255.255.255.0	S0/0/0 S0/0/1 Fa0/0

Learning Objectives

Setup a network following given topology

Configure PCs and Webserver

Test Webserver

Create and Restrict access to a specific folder on webserver

Restrict access to specific folder base on host IP address,

Change PC settings with Webmin

Use SSH connection to edit settings on server

Scenario

In this lab exercise, students will learn to setup and configure apache webserver with different options. Student will establish secure remote connection to webserver and remotely modify settings. Also, this laboratory introduces students to a system configuration integrated GUI-based UNIX tool. All operations will be completely implemented in Linux environment. Note that you would be required to enter root user level password to proceed. Enter the passwords appropriately.

Task 0: Configure all network participating devices

Set hostname.
Set DNS lookup to "disabled".
Set EXEC mode password
Set console connection password
Set vty connection password
Save running configuration to NVRAM

Task 1: Setup network according to topology

Boot the provided Ubuntu virtual machines on three different PCs and cable the network as shown above.

Task 2: Configure PCs and startup webserver

Step 1: Configure the PCs and web server with the given IP addresses and subnet Masks in the table.

Step 2: Startup web server by typing to terminal

sudo /etc/init.d/apache2 start

Note, the terminal will prompt for superuser password. Enter password accordingly.

Task 3: Test Webserver

Step 1: Confirm that server is running by typing to terminal

service apache2 status

Step 2: Launch browser in web server and type http://localhost or http://127.0.0.1; if webserver is up and running the page should display among others the text "it works".

Step 3: Further test webserver on host PC_1 and PC_2 by typing webserver IP address on browser address bar. The browser will display same content as in step 1.

Task 4: Create and restrict access to specific folder on web server

Step 1: Create a directory with this command. To achieve this, do the following in order.

Create directory: *sudo mkdir /var/www/testdir*

Create file in the directory with: *sudo touch /var/www/testdir/test.html*

Step 2: Create and store file in directory created in step 1

Open test.html for editing with: *sudo gedit /var/www/testdir/test.html*

Paste the below html codes in the file, save and then close the file:

<html>
<head>
<title>Your Page Title</title>
</head>
<body>
<h2>I am about to render my parent directory invisible. <h2>
</body>
</html>

Step 3: Check to confirm the folder/directory is visible before proceeding to restrict access. Launch any browser and write to address bar http://localhost/testdir/test.html.

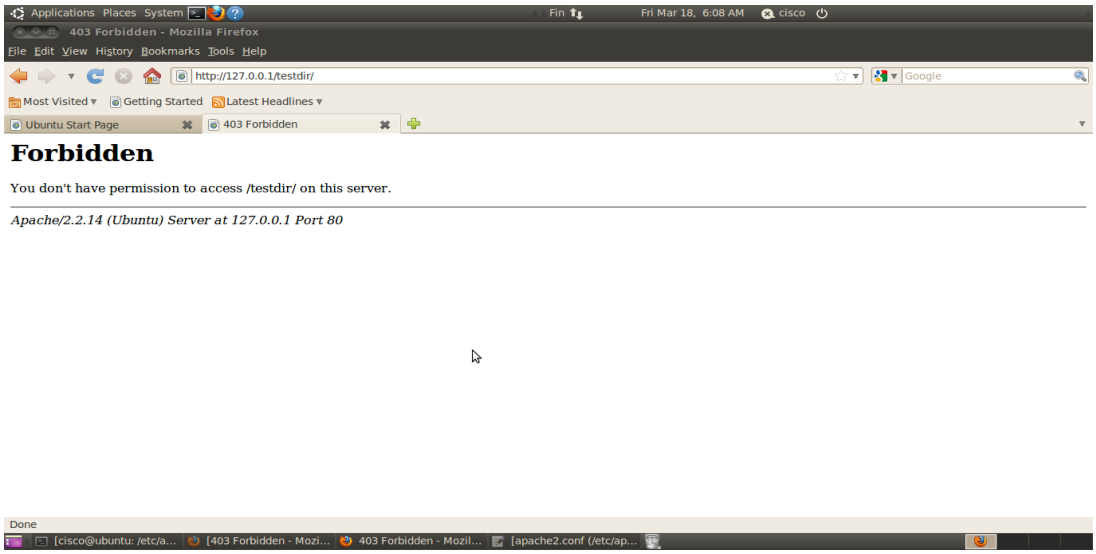
Step 4: Restrict access to directory /var/www/testdir/**.** This will render subfolders and files in this directory invisible**.** To achieve this, do the following in order.

Open apache2.conf file: sudo gedit /etc/apache2/apache2.conf

Paste the below script directive in the file, save changes and close

```
#
# The following lines disable access to folder /var/www/testdir
# THIS IS SEGUN ADODO'S ACTION
#
<Directory "/var/www/testdir">
  Order deny,allow
  Deny from all
</Directory>
#
```

Now, launch any browser and write to address bar http://localhost/testdir/test.html. You see a page as below



GRAPH 16. Snapshot showing output page after access restriction

Task 5: Restrict PC_2 and PC_4 from accessing the test.html page.

Step 1: First remove access restriction implemented in task 4. To achieve this, do the following.

Open apache2.conf file: sudo /etc/apache2/apache2.conf

Then find and delete or comment out (i.e. put the # symbol before each line of the script) the below script directive in the file, save changes and close.

```
#
# The following lines disables access to folder /var/www/testdir
# THIS IS SEGUN ADODO'S ACTION
#
#<Directory "/var/www/testdir">
#  Order deny,allow
#  Deny from all
#</Directory>
#
```

Step 2: Test "testdir" is now available. Type http://localhost/testdir/test.html to any browser. The text "I am about to render my parent directory invisible" will be displayed on the browser.

Step 3: To restrict PC_1 and PC_4 from accessing testdir folder on the webserver while other hosts will still have access to same page, do the following.

Open apache configuration file for editing: sudo gedit /etc/apache2/apache2.conf

Copy and paste the below script directive in the file, save and then close.

```
# The following lines disables access to folder /var/www/testdir by hosts PC_2 and
# PC_4
# THIS IS SEGUN ADODO'S ACTION
```

```
#
<Directory "/var/www/testdir">
  deny from 192.168.40.2 192.168.40.2
  deny from 192.168.40.2 192.168.20.3
 </Directory>
#
```

Then start and stop the webserver by using the following command on the server machine

to stop type: sudo service apache2 stop

to start type: sudo service apache2 start

Step 4: Test to confirm PC_2 and PC_4 cannot access the webserver. Type *192.168.20.3* to browser in PC_2 and PC_4. Text displayed on browser will suggest

Task 6: Change system settings with Webmin. Webmin is a web-based tool. It is used to implement system administration UNIX system.

Step 1: Launch browser and use the follow credentials to access Webmin

Link: https://ubuntu:10000/
Username: admin
Password: cisco

Step 2: Explore the tabs and change system configuration as you may wish.
On PC_3 type to browser https://192.168.20.4:10000/

Task 7: Establish remote SSH connection to server from PC_1. SSH connection is secured unlike counterpart Telnet. Packets are sent in encrypted format to deter eavesdropping or snooping.

Step 1: Establish connection by typing to terminal

ssh “username@webserver ip-addresss" (e.g. ssh cisco@192.168.20.3)

then type YES to the question that follows. Now, you are remotely connected to the webserver. Note that “username” is the username of the remote machine while “webserver ip-addresss” is the ip address of the remote server.

Step 2: Carryout task highlighted in TASK 3(Step 1 and 2).

Step 3: Exit SSH connection by typing “exit” to terminal

5 TESTING OF THE SANDBOXED SYSTEM

The entire system is setup as discussed in section 3.3. The virtual machine is launched in the CNA local network using bridged network only.

The virtual machine network setting is configured with only bridged-network as provided by the virtual machine manager in use; VM is booted to desktop environment. To put the VM on CNA local network, the network settings is edited by:

- first deleting any existing network connections
- clicking to add new connection
- selecting IPV4 Setting
- changing connection option to “MANUAL”
- inputting IP address numerics
- and applying this configuration

These steps setup the VM on the CNA local network. To verify this, another VM was created on another host PC and successful pinging operation was carried out. Overall, to test the entire system the prototype lab presented in section 4.3 was engaged to implement an extensive testing exercise. The results of the testing are provided in the appendix section of this literature.

6 CONCLUSION

This thesis undertaking has worked out as a worthwhile experience with immense value. Exploring the UNIX and the Linux platform in particular presents a somewhat different insight into the computing experience. The idea of interacting more with the character-based terminal than the traditional GUI evolves a new challenge that must be tackled to fully unlock the embedded opportunities in Linux. In the course of this thesis undertaking, it became evident that I had to understudy this relatively unfamiliar environment, the effect of which translated to a knowledge sufficient enough to complete this work. Additionally, I have built more self-confidence in using and expanding my knowledge of the Linux OS environment.

Cardinal to this thesis undertaking is the introduction of a rapidly evolving open source computing platform. By way of achieving this at little or no cost implication to the school, this work conceived a virtual environment to deploy the Linux platform. And to engage the OS, the provisioning of operating system with system admin and monitoring tools such as Webmin was proposed and implemented. However, to put this new environment and tools to practical use by students, a prototype laboratory exercise was developed. The prototype lab exercise is conceived as sample tasks that can be designed to make students acquaint with Linux and the system administration and network related applications. The stated solution is designed to be understandable and to help students navigate more confidently into the Linux world. A combination of this OS and the solution presented by this thesis work will create additional value. However, further development of this solution can create extra impact.

The writer of this thesis posits that the study curriculum be expanded and optimized to accommodate a broad and well-thought out Linux oriented study plan which on implementation will transform the windows-oriented minds to one balanced along the lines of multiple possibilities.

REFERENCES

Aaron, B., Fred, S., Paul, K., Andreas, T., Wes, N., Carsten, W., Craig, S., Davif, W. 2009.Virtualization for Security Including Sandboxing, Disaster Recovery, High Availability, Forensic Analysis and Honeypotting. Massachusetts, MA: Syngress Publishing Inc.

Apache 2011. Available: http://httpd.apache.org/docs/2.0/. Accessed 18.3.2011.

Bovet, P. & Cesati, M. 2005. Understanding the Linux Kernel(3rd Edition). California, CA: O'Reilly Media Inc.

Cameron, J. 2004. Managing Linux Systems with Webmin. New Jersey, NJ: Pearson Education Inc.

Chiangr, Roger, Keng, Bill. 2009.System Analysis and Design: Techniques, Methodologies, Approaches, and Architectures. New York, NY: M.E Sharp Inc.

Cisco Networking Study Module 2011. Available: http://www.cisco.com. Accessed 31.01.2011.

Desai, A. 2007. The definitive Guide to Virtualization to Virtual Platform Management. California, CA: Realtimepublishers.

Dubois, P. 2006. MySQL CookBook(2nd Edition). California, CA: O'Reilly Media Inc.

Keski-Pohjanmaan Ammattikorkeakoulu. 2008. Guide for Thesis Writers. Finland, Kokkola: Keski-Pohjanmaan Ammattikorkeakoulu.

Lo, J. 2005. VMware and CPU Virtualization technology. California, CA: VMware Inc.

Microsoft Virtualization Technology 2011. Available: http://www.microsoft.com. Accessed 8.3.2011.

Muller, A. & Seburn, W. 2005. Virtualization with VMware ESX Server. Massachusetts, MA: Syngress Publishing Inc.

Phpmyadmin 2011. Available: http://phpmyadmin.net. Accessed 22.3.2011.

Php 2011. Available: http://www.php.net. Accessed 22.3.2011.

Pressman, R. 2001.Software Engineering A practitioner's approach (5th Edition). New York, NY: McGraw-Hills.

Protocol Analyzer 2011. Available: http://wireshark.org. Accessed 22.3.2011.

Shelly, Rosenblatt. 2009. System Analsysis and Design(8th Edition). Massachusetts, MA: Course Technology.

Stevens, W. 1993.TCP/IP Illustrated. Massachusetts, MA: Addison-Wesley Professional.Wolf, C. & Hatler, M. 2005. Virtualization From the Desktop to the Enterprise. New York, NY: Apress.

Tcpip Guide 2011. Available: http://tcpipguide.com. Accessed 22.3.2011.

Virtualization 2011. Available: http://www.virtuatopia.com. Accessed 23.1.2011.

VMware Workstation 2011. Available: http://www.vmware.com. Accessed 9.3.2011.

Wembao, M. 2003. Modern Cryptography: Theory and Practice. New Jersey, NJ: Prentice Hall.

Xen Virtualization Technology 2011. Available: http://www.xen.org. Accessed 8.3.2011.

Router Configuration Scripts

```
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname R1
!
enable password cisco
!
ip subnet-zero
!
process-max-time 200
!
interface Ethernet0
 ip address 192.168.20.1 255.255.255.0
 no shutdown
!
interface Serial0
 ip address 209.165.200.230 255.255.255.252
 clockrate 64000
 no shutdown
!
interface Serial1
```

```
shutdown
!
router rip
 version 2
network 192.168.20.0
 network 209.165.200.0
!
ip classless
no ip http server
!
line con 0
transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
end
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname R2
!
```

```
enable password cisco
!
ip subnet-zero
!
process-max-time 200

!
interface Ethernet0
 shutdown
!
interface Ethernet1
 ip address 192.168.40.1 255.255.255.0
 no shutdown
!
interface Serial0
 ip address 209.165.200.229 255.255.255.252
 no shutdown
!
interface Serial1
 ip address 209.165.200.233 255.255.255.252
clockrate 64000
 no shutdown
!
router rip
 version 2
```

```
network 192.168.40.0
 network 209.165.200.0
!
ip classless
no ip http server
!
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password cisco
login
!
end
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname R3
!
enable password cisco
!
ip subnet-zero
```

```
!
process-max-time 200
interface Ethernet0
 ip address 192.168.30.1 255.255.255.0
 no shutdown
!
interface Serial0
 shutdown
!
interface Serial1
 ip address 209.165.200.234 255.255.255.252
 no shutdown
!
router rip
 version 2
 network 192.168.30.0
 network 209.165.200.0
!
no ip classless
no ip http server
!
!
line con 0
 password cisco
 transport input none
line aux 0
```

```
line vty 0 4
 password cisco
 login
!
end
!
version 11.1
service udp-small-servers
service tcp-small-servers




!
hostname R3
!
enable password cisco
!
ip subnet-zero
!
process-max-time 200
!
interface Ethernet0
 ip address 192.168.30.1 255.255.255.0
 no shutdown
!
```

```
interface Serial0
shutdown
!
interface Serial1
 ip address 209.165.200.234 255.255.255.252
no shutdown
!
router rip
 version 2
 network 192.168.30.0
 network 209.165.200.0
!
no ip classless
no ip http server
!
line con 0
 password cisco
 transport input none
line aux 0
line vty 0 4
 password cisco
 login
!
```

end

!

Ping Result

Webserver ping P_3

cisco@ubuntu:~$ ping 192.168.30.2 -c 2

PING 192.168.30.2 (192.168.30.2) 56(84) bytes of data.

64 bytes from 192.168.30.2: icmp_seq=1 ttl=61 time=48.4 ms

64 bytes from 192.168.30.2: icmp_seq=2 ttl=61 time=48.3 ms

Webserver ping P_4

cisco@ubuntu:~$ ping 192.168.40.2 -c 2

PING 192.168.40.2 (192.168.40.2) 56(84) bytes of data.

64 bytes from 192.168.40.2: icmp_seq=1 ttl=62 time=24.6 ms

64 bytes from 192.168.40.2: icmp_seq=2 ttl=62 time=24.8 ms

PC_4 Ping Webserver

cna@ubuntu:~$ ping 192.168.20.4

PING 192.168.20.4 (192.168.20.4) 56(84) bytes of data.

64 bytes from 192.168.20.4: icmp_seq=1 ttl=62 time=35.7 ms

64 bytes from 192.168.20.4: icmp_seq=2 ttl=62 time=24.6 ms

PC_3 Ping Webserver

cna@ubuntu:~$ ping 192.168.20.4 -c 2

PING 192.168.20.4 (192.168.20.4) 56(84) bytes of data.

64 bytes from 192.168.20.4: icmp_seq=1 ttl=61 time=48.3 ms

64 bytes from 192.168.20.4: icmp_seq=2 ttl=61 time=48.1 ms