Calhoun: The NPS Institutional Archive

Theses and Dissertations | Thesis Collection

2004-09

Develop, build, and test a virtual lab to support vulnerability training system

Akgul, Turgut

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/1468

http://www.nps.edu/library

Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

NAVAL POSTGRADUATE SCHOOL

THESIS

DEVELOP, BUILD, AND TEST A VIRTUAL LAB TO SUPPORT A VULNERABILITY TRAINING SYSTEM

by

Coskun Kargin
and
Turgut Akgul

September 2004

Thesis Advisor: Richard M. Harkins
Second Reader: Wen Su

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2004	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Develop, Build, and Test a Virtual Lab to Support a Vulnerability Training System			**5. FUNDING NUMBERS**
6. AUTHOR(S) Coskun Kargin and Turgut Akgul			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT ***(maximum 200 words)***

A computer security virtual lab architecture was developed and tested for functionality and performance. Four Dell PowerEdge 1650, dual processor, blade servers were configured as host machines with VMware and VNC running on a Linux RedHat 9 Kernel. An Apache-Tomcat web server was configured as the external interface to lab users. Web content was created, the site was secured with SSL, and Java Servlet functionality was enabled. Host machine performance was tested under various load conditions. Analysis indicated that, for our architecture, that the average host machine CPU load was ~12 % while the average memory load was ~33 %. We conclude that for the cost and space requirements of 5 1U blade servers we have configured an equivalent 20 computer lab. Performance tests show that the virtual lab could scale easily from 4 – 30 computers.

14. SUBJECT TERMS Virtual Lab, Virtual Network, Virtual Machine, Web Server, Apache, Tomcat, VMware, VNC, SSL			**15. NUMBER OF PAGES** 93
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

DEVELOP, BUILD, AND TEST A VIRTUAL LAB TO SUPPORT VULNERABILITY TRAINING SYSTEM

Coskun Kargin
1st Lieutenant, Turkish Army
B.S., Turkish War College, 1998

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

and

Turgut Akgul
1st Lieutenant, Turkish Army
B.S., Turkish War College, 1999

MASTER OF SCIENCE IN COMPUTER SCIENCE

Submitted in partial fulfillment of the
requirements for the degree of

from the

NAVAL POSTGRADUATE SCHOOL
September 2004

Authors: Coskun Kargin

Turgut Akgul

Approved by: Richard M. Harkins
Thesis Advisor

Wen Su
Second Reader

Peter J. Denning
Chairman, Department of Computer Sciences

Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

A computer security virtual lab architecture was developed and tested for functionality and performance. Four Dell PowerEdge 1650, dual processor, blade servers were configured as host machines with VMware and VNC running on a Linux RedHat 9 Kernel. An Apache-Tomcat web server was configured as the external interface to lab users. Web content was created, the site was secured with SSL, and Java Servlet functionality was enabled. Host machine performance was tested under various load conditions. Analysis indicated that, for our architecture, that the average host machine CPU load was ~12 % while the average memory load was ~33 %. We conclude that for the cost and space requirements of 5 1U blade servers we have configured an equivalent 20 computer lab. Performance tests show that the virtual lab could scale easily from 4 – 30 computers.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

The authors would like to thank Professor Richard M. Harkins for his assistance in acquiring the lab and necessary equipment, as well as his helpful guidance to conduct this thesis research. We also appreciate the Turkish community at NPS for the amiable social atmosphere of friendship and understanding, without which we would have a much harder time engaging in our studies.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Students of Computer Science and Information Systems Technology have experienced that learning computer related topics—whether it is software development, network building and configuration, or network security—often requires hands-on practice. It is "widely recognized this type of learning enhances the education experience.[1]

Yet, creating a physical network environment to practice the necessary skills may generate some problems:

Time: Since the labs are often dedicated to a number of classes, the students are generally faced with time restrictions. Most often, the time allotted for the lab hours is two hours a week. The students may find extra time when the lab is free, but since that is not a guaranteed time, the instructors tend to give assignments that can be finished in two hours time. Also, since not all the students possess the same skill level, instructors have to adjust the level of their assignments so that the lowest skill level student can complete them in two hours.

Cost: Creating such an environment takes money. Depending of the number of users intended to use the same environment, the costs of establishing and maintaining the necessary hardware and software in a quantity sufficient for each user to have access to the network laboratory for learning and practicing purposes, both during and after the scheduled times, requires substantial funding.[2] The expenses include the cost of the

[1] Bruce Kneale, Ain Y. De Horta and Ilona Box, (2004) "Velnet: Virtual Environment for Learning Networking," (This paper appeared at the sixth Australian Computing education Conference (ACE2004), Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, vol. 30. Editors, Raymond Lister and Alison Young), <http://portal.acm.org/citation.cfm?id=979990&dl=ACM&coll=portal> (16 July 2004) The authors cite a number of references to support this idea, but were unable to check those sources to confirm.

[2] Kneale, Horta, Box, (2004), "Velnet: Virtual Environment for Learning Networking," p. 161; Russell Elliott, "Creating a Home Test Lab, Cases Study in Information Security," (SANS Institute, February 19, 2003), <http://www.giac.org/practical/GSEC/Russell_Elliott_GSEC.pdf > (11 July 2004), p. 5.

computers and their components, the costs of the required software (operating systems, etc.), and other secondary costs (routers, switches, cables, etc. – not to include the cost of the room).[3]

Space: A physical network laboratory for an average number of students in a class entails a large space. Large air-conditioned rooms, full of computers, connected to routers and switches are often required.[4] Considering the ever-present space restrictions in the buildings, this is a practical issue that affects the design of the lab.[5]

Structure: The money and space restrictions often force the instructors to group the students. Group working, despite its positive aspects, may have some negative consequences as well. Sometimes one member of the group ends up doing the bulk of the work, leaving the other(s) without the benefits of hands-on experience.[6] Even if the group members share the work properly, it still means less practical experience than having the chance to do everything themselves.

Maintenance: The maintenance of the computers is always an issue. It takes time and effort to keep large numbers of computers up and functioning in the lab.

Restrictions: Legal restrictions, as well as the "lack of a secure network environment, "in which actions do not damage the other services on the network, makes it necessary to isolate the network.[7] First, students must be physically present in the lab. Second, the lab must be air gapped.

Thus the problem of addressing same of these deficiencies arises as well as being particularly interested in designing a virtual network environment for practicing Network Security skills. This thesis study concentrates on a particular solution to this problem: the virtual lab (VL).

---

[3] Elliott, "Creating a Home Test Lab," p. 6.

[4] Kneale, Horta, Box, "Velnet," p. 161; Elliott, "Creating a Home Test Lab," p. 5.

[5] Elliott, "Creating a Home Test Lab," p. 5.

[6] Kneale, Box, "A Virtual Learning Environment for Real-World Networking," p. 672.

[7] Kneale, Horta, Box, "Velnet: Virtual Environment for Learning Networking," p. 161.

The concept of virtual labs is a widely used phenomenon currently. Many organizations feature virtual labs for a number of purposes. With the help of the advances in the computer hardware, and the software simulation techniques, it is now possible to visit virtual physics, biology, chemistry, or mathematics labs online.[8] They generally run Java applets, or some software programs, to simulate the necessary environment for the virtual labs, and they feature experiments, which may or may not need physical laboratories. Thus, depending on their application, they often provide a virtual hands-on experience to the users, and facilitate the educational process.

Although the idea of virtual labs is not new and there are a number of implementations of virtual labs for a variety of purposes, to the author's surprise, it has been noticed that virtual computer network lab implementations for the purpose of network security classes are not very common.

According to Kneale, a virtual lab must have the following attributes in order to be an effective substitute for a physical network.

It should be available to every student for a long enough time to complete the assignments and preferably more.[9]

It should provide the students with the ability to stop and resume an exercise over time.[10]

The environment should be configured to be accessed securely 24 hours a day, seven days a week.[11]

---

[8] Some of these websites featuring virtual lab environments are: <http://www.enc.org/resources/records/full/0,1240,016555,00.shtm> (16 July 2004), which "provides over 500 web links to applets, simulations, and virtual labs that illustrate visually difficult physics concepts"(an Eisenhower National Clearinghouse website);

<http://www.math.uah.edu/stat/> (16 July 2004), which provides "Virtual Laboratories in Probability and Statistics" (University of Alabama, Huntsville Website);

<http://www.biointeractive.org/> (16 July 2004), which concentrates on biology;

<http://www.chem.ox.ac.uk/vrchemistry/> (16 July 2004), which focuses on chemistry;

<http://www.jhu.edu/~virtlab/virtlab.html> (16 July 2004), which features a "Virtual Engineering/science Laboratory course" (A Johns Hopkins University Website);

<http://www.physics.nwu.edu/ugrad/vpl/> (16 July 2004), which is a " Virtual Physics Laboratory" (A Northwestern University Website)

[9] Kneale, Box, (June 2003), "A Virtual Learning Environment for Real-World Networking," p. 672.

[10] Ibid.

It should deal with the money and space restrictions. It should preferably be financially cheaper than a physical network lab, both in the building phase and the maintenance phase, and it should not take too much space, at least not more than a physical network.[12]

It should provide every feature present in a physical network, including different operating systems (OS).

A. SURVEY OF RELATED WORK

Two reviewed examples illustrate the concepts:

1. Velnet

Velnet was developed by the School of Computing and Information Technology, and presented by Bruce Kneale at the University of Western Sydney, Australia.[13] Figure 1 demonstrates the underlying architecture for a virtual network (VN) education system.

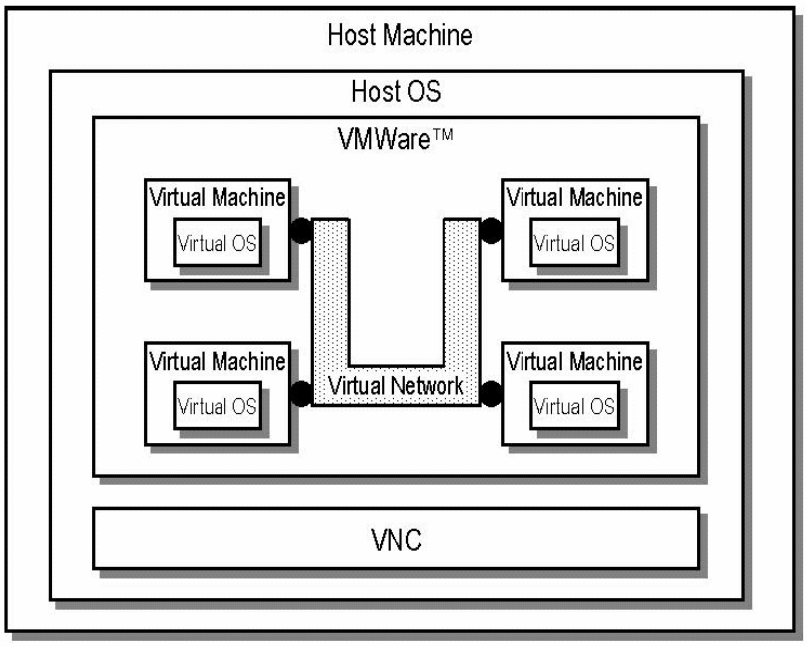


Figure 1. The Components and the Configuration of Velnet[14]

[11] Ibid.

[12] Russell Elliott, "Creating a Home Test Lab," p. 6.

[13] Ain Y. De Horta, Bruce Kneale and Hona Box, "Development of a Virtual Overlay for Velnet (Virtual Environment for Learning Networking)," School of Computing and Information Technology, University of West Sydney, Australia, December 7-10, 2003, <http://proceedings.informingscience.org/IS2003Proceedings/docs/090Kneal.pdf> (16 July 2004).

The goal of Velnet is to provide computer science students with a virtual environment in which they can have hands-on experience with computer networking without having to deal with all the possible problems of a physical laboratory. The solution they offer against all these problems is to create a virtual lab, Velnet. They refer to the six assumptions out of 14 propositions developed by Winn and Jackson (1999)[15] and briefed in the paper titled as "The Effects of Virtual Environments on Recall in Participants of Differing Levels of Field Dependence."[16] The assumption is that Virtual Environments (VE) are cheaper and safer. VE allows students to experience metaphorical concepts and undetectable phenomena. Students are more likely to do well in VE. VE allows students to take what is familiar to them and add to their knowledge. VE simulates learning in real context.

The rest of the paper concerns the actual implementation of the project. To build their virtual lab, they take a single machine with a host OS installed on it. The hardware configuration of the machine and the choice of host OS depend on the amount of the work to be written to that machine. Inside this outer layer, they install a software called VMware in order to simulate multiple virtual machines (VM) with various operating systems within the host OS. They also use another software called Virtual Network Computing (VNC) to have access and control the other virtual machines remotely from either one of the virtual machines or the host machine itself. Velnet is the tool by which they can create different scenarios by establishing various networking configuration on this system. The paper ends with some of the results obtained from this study and their future research plans.

2. Home Test Lab

In practical and theoretical terms, the second study, "Creating a Home Test Lab" by Russell Elliott, published at the SANS Institute[17], which is similar to this project, is much like the first. Yet, there are some minor differences. The first difference is the

---

[14] Horta, Kneale, Box, "Development of a Virtual Overlay for Velnet," p. 162.

[15] Ibid., p. 161.

[16] Todd Ogle, "The Effects of Virtual Environments on Recall in Participants of Differing Levels of Field Dependence," <http://scholar.lib.vt.edu/theses/available/etd-04252002-112047/unrestricted/etd.pdf> (April 11, 2002), pp. 16-19.

[17] Elliott, "Creating a Home Test Lab."

purpose of the study. While the goal of the first study was to provide students with hands-on experience in the computer networking they need by means of a virtual lab, the study in the second paper tries to provide hands-on experience for security professionals. Another difference is that VNC software is not used in this study.

The paper first discusses the advantages and the disadvantages of setting up a home network for network security professionals. Although this is not the author's intent, Elliot's study, nevertheless, provided some helpful hints. The paper discusses why having a single system with several virtual machines in it would be beneficial over having another system for every operating system to be studied. According to his calculations (which exclude the cost of space), the costs of building a five-computer physical network, and building a virtual network doing the same job are not much different, but when space is a limiting factor, the virtual lab alternative is clearly recommended.[18]

The paper continues with the discussion on why having a single system with several virtual machines in it would be beneficial over having another system for every operating system to be studied. Again, the costs related to each option depend on which is preferred. In this case, the first is preferred. The paper continues with a detailed comparison of the hardware, two types of virtual machine software on the market, host, and the virtual operating systems installed on that single system. The networking among the host and the virtual machines was also explained in the paper. It ends with the results of some studies made with the virtual lab.

Both samples have common characteristics with the author's study. In terms of goals, the second study is more relevant, and therefore, a virtual lab was built for the benefit of students taking computer security courses. Yet, it can be used for networking studies with some configuration changes. The physical design and implementation is more like the first sample, however. The authors also used VNC software as in the first study.

Although many similarities exist between this study and those mentioned previously, there are three issues making this study different and more complicated than both samples. First, five hosts machines interconnected among themselves and a separate

[18] Elliott, "Creating a Home Test Lab," p. 7.

server machine were used. Second, virtual machines were made available via the Internet to the users who do not require them to be physically present in front of the host machine. Lastly, this study involved one last phase. The system was tested to be able to determine the amount of workload that can be put on the host machines given the present configurations. An attempt was also made to ascertain how many virtual machines and how many connections for each one of them should be available in order for the system to run efficiently.

THIS PAGE INTENTIONALLY LEFT BLANK

II. VIRTUAL LAB MODEL

The required components for the virtual lab include:

- The Host Machine and Operating System
- Virtual Machine Software and Network
- Guest Operating Systems
- Remote Desktop Display
- Remote Access Server

This list of components is similar to the model introduced in Figure 1. The only addition to that model is the server machine, which enables the users to have external access to the virtual lab.

A. THE HOST MACHINE AND HOST OPERATING SYSTEM

1. Hardware

The host machine forms the base for all of the components in the virtual lab. Therefore, it must meet certain requirements to support the scope of the project. Ideally, a computer with a large hard drive, a fast Central Processing Unit (CPU) and a lot of memory would be needed to host multiple Virtual Machines effectively. The intent was to choose a hardware software combination that would perform well under heavy loads. A heavy load is interpreted to mean multiple VMs installed and running on the host with multiple connections to outside users. Considering performance objectives and cost constraints, the following hardware was selected:

- Dell Rack System, with five identical two-processor PowerEdge 1650 computers
- A single monitor connected via a Keyboard - Video - Mouse (KVM) switch.
- CPUs with a 1400 Megahertz (MHz) bus speed
- 40 Gigabyte (GB) Small Computer System Interface (SCSI) hard-drive
- 1GB Random-Access Memory (RAM).

Detailed information on hardware configuration is given in Chapter III.

2. Software

It is important that the host OS support Internet protocols such as Transmission Control Protocol/Internet Protocol (TCP/IP). It also had to support the virtual machine and the remote desktop display software that was intended to be installed. Running as a standalone workstation was another desired trait since the virtual machines were going to be created on workstations rather than personal computers (PC). Among all the different versions of Windows and Linux based operating systems, Red Hat Linux 9.0 was selected. Red Hat 9.0 provided a means to develop and modify the Kernel. It had multiple configurable security features and had a reputation for stability under heavy load and was fully supported by the Virtual machine software chosen to use.

B. VIRTUAL MACHINE SOFTWARE AND VIRTUAL NETWORK

Virtual machine software is a tool that makes it possible to run multiple operating systems and their applications at the same time on the same physical computer. These operating systems and applications are isolated in secure virtual machines that co-exist on a single piece of hardware.[19]

For this experiment, the decision was to use VMware Workstation 4.5. VMware is a mature product that gives the virtual machine an interface with the host machine's hardware and peripherals. This includes:

- The network card,
- The Compact Disc/Digital Versatile Disc (CD/DVD) drive,
- The Universal Serial Bus (USB) and serial communications ports
- The printer port.

Some of the components of a physical network such as switches and network adapters are emulated in the virtual network. Routing is supported by operating systems, which have this function and are installed on the virtual machines.[20]

[19] "Workstation 4 User Manual," <http://vmware-svca.www.conxion.com/software/ws45_manual.pdf> (26 August 2004).

[20] Ibid.

C. GUEST OPERATING SYSTEMS

The VMware Workstation emulates the Intel x86 hardware architecture, and therefore, supports any operating system that can run in that environment. The following is a brief list of supported operating systems as stated in the VMware manual:[21]

1. Microsoft Windows

- Windows, code-named Longhorn, beta (experimental)
- Windows Server 2003 Web Edition, Windows Server 2003 Standard Edition,
    - Windows Server 2003 Enterprise Edition
- Windows XP Professional and Windows XP Home Edition with Service Pack 1 or Service Pack 2RC (listed versions also supported with no service pack)
- Windows 2000 Professional Service Pack 1, 2, 3 or 4; Windows 2000 Server
    - Service Pack 1, 2, 3 or 4; Windows 2000 Advanced Server Service Pack 3 or 4
    - (listed versions also supported with no service pack)
- Windows NT® Workstation 4.0 Service Pack 6a, Windows NT Server 4.0 Service Pack 6a, Windows NT 4.0 Terminal Server Edition Service Pack 6
- Windows Me
- Windows 98 (including all Customer Service Packs) and Windows 98 SE
- Windows 95 (including Service Pack 1 and all OSR releases)
- Windows for Workgroups 3.11
- Windows 3.1

2. Microsoft MS-DOS

- MS-DOS 6.x

3. Linux

- Mandrake Linux 8.2, 9.0
- Red Hat Linux 7.0, 7.1, 7.2, 7.3, 8.0, 9.0
- Red Hat Enterprise Linux 2.1, 3.0
- Red Hat Linux Advanced Server 2.1
- SuSE Linux 7.3, 8.0, 8.1, 8.2, 9.0, 9.1

[21] Ibid, pp. 23-24, (26 August 2004).

- SLES 7, 7 patch 2, 8
- Turbolinux Server 7.0, Enterprise Server 8, Workstation 8

4. Novell NetWare

- NetWare 5.1, 6, 6.5

5. FreeBSD

- FreeBSD 4.0–4.6.2, 4.8, 5.0

6. Solaris

- Solaris x86 Platform Edition 9 (experimental), 10 beta (experimental)

This study used Windows 2000 Professional, Windows XP Professional, and Red Hat Linux 9.0 for the guest operating systems.

D. REMOTE DESKTOP DISPLAY (RDD)

Since one of the goals of the project was to make the virtual machines externally available, a remote desktop display tool was necessary. There are several commercial products on the market that serves this purpose. The one selected was VNC (Virtual Network Computing).

VNC is an open-source, free, cross-platform remote desktop display package developed by ATT Labs. The software allows connectivity between different types of operating systems. By using VNC, one has the full control of a remote machine from any other computer or mobile device anywhere on the network. VNC consists of a server that runs on the machine to be remotely controlled and a client installed on the machine that would connect to the server. It also has a built-in Java viewer, which makes it reachable within a browser without having to install the client software.[22]

E. REMOTE ACCESS SERVER

External connectivity to the VMs was realized by a Remote Access Server. The server was identical in makeup to the other host machines with regard to hardware and software. However, Tomcat and an Apache web server were installed to provide support for external use.

The Apache Server was installed with Secure Sockets Layer (SSL) functionality for secure Internet connectivity. Then, another web server, Tomcat, was installed, which

[22] "Java VNC Viewer," <http://www.realvnc.com/javavncviewer.html> (27 August 2004).

would run concurrently with Apache. Tomcat supports Java Servlets and Java Server Pages (JSP) specifications. "Servlets are modules of Java code that run in a server application to answer client requests."[23] They are used for extending and enhancing web servers. Servlets are useful because they can be built component-based, and platform-independent.[24]

In this project, Tomcat's Java compatibility was beneficial in two ways. First, it made it possible to display the VNC server through a web browser. Second, it was possible to use Servlets in the web pages if any were necessary.

[23] "An Invitation to Servlets," <http://www.novocode.com/doc/servlet-essentials/chapter1.html#ch_1_1> (27 August 2004).

[24] "Java Servlet Technology," <http://java.sun.com/products/servlet/> (27 August 2004).

THIS PAGE INTENTIONALLY LEFT BLANK

III. VIRTUAL LAB SETUP

A. EQUIPMENT

Dell PowerEdge 1650 Blade Servers with dual processors on each rack was used to create the network for the lab. The system consisted of:

- 5 Dell PowerEdge 1650 mountable racks (Figure 2)
- 1 Dell PowerConnect 3024 Switch (Figure 3)
- 1 Dell 8 Port KVM (Keyboard, Video, and Mouse) Switch (Figure 3)
- 1 APC Smart UPS (Uninterruptible Power Supply) 2200 Power Supply (Figure 4)
- 1 Dell™ PowerEdge™ Rack Console 15FP Flat-Panel Monitor (Figure 6)
- 1 PS/2-style keyboard with integrated mouse (Figure 6)

The technical specifications of the system are listed below (Table 1):

Microprocessor	
Microprocessor type	Two (2) Intel Pentium III, 1.4 GHz Processors
Front side bus (external) speed	133 MHz
Internal cache	512 KB Level 2 cache
Math coprocessor	Internal to microprocessor

Expansion Bus	
Bus type	PCI
Expansion slots	two dedicated PCI (one full-length and one half-length 64-bit, 66-MHz slot, or optionally, one half-length 64-bit, 66-MHz slot with one full-length 32-bit, 33-MHz slot, 5-V compatible on separate buses)

Memory	
Architecture	72-bit ECC PC-133 SDRAM DIMMs, with 2-way interleaving
Memory module sockets	four 72-bit wide 168-pin DIMM sockets
Memory module capacities	128-, 256-, 512-MB, or 1-GB registered SDRAM DIMMs, rated for 133-MHz operation

RAM	1 GB RAM at 4*256MB. Configurable to 4 GB.
Drives	
Diskette drive	3.5-inch, 1.44-MB diskette drive
SCSI hard drives	up to three 1-inch, internal Ultra3 SCSI
IDE hard drives (optional) *(our system did not include this)*	up to two internal (not hot-pluggable), ATA-compatible
CD or DVD drive	*CD drive*

Ports and Connectors	
Externally accessible:	
Rear:	
SCSI	68-pin Ultra3 SCSI connector
Serial	9-pin connector
USB	4-pin connectors
NIC	2(two) RJ45 connectors for integrated 10/100/1000 NICs; one RJ45 connector for optional remote service card (10 Mbit Ethernet controller) used for remote system administration)
Video	15-pin connector
PS/2-style keyboard	6-pin mini-DIN connector
PS/2-compatible mouse	6-pin mini-DIN connector
Front:	
Video	15-pin connector
USB	4-pin connector
PS/2-style keyboard/mouse	6-pin mini-DIN, keyboard default (mouse optional with combination Y cable)

Video	
Video type	ATI Rage XL PCI video controller; VGA connector
Video memory	8 MB

Power	
Power supply:	
Wattage	275 W (AC)
Voltage	100–240 VAC, 50/60 Hz, 3.9–2.0 A

Heat dissipation	1033 BTU/hr maximum per power supply
Maximum inrush current	Under typical line conditions and over the entire system ambient operating range, the inrush current may reach 25 A per power supply for 10 ms or less.
System battery	CR2032 3.0-V lithium coin cell

Physical	
Height	1.67 inches
Width	19 inches

Table 1. Dell PowerEdge 1650 Specifications[25]

The "Dell™ PowerEdge™ 1650 Systems Installation and Troubleshooting Guide" on the company website was used to assemble and set up the system hardware.[26] All the components were mounted onto the system as described in the guide manual (Figure 6). The KVM switch afforded single monitor and keyboard access to all the servers via the Print Screen Menu (Figure 5).

Figure 2. Dell PowerEdge 1650 Rack

[25] "Technical Specifications," <http://docs.us.dell.com/support/edocs/systems/pe1650/en/ug/8g540aa0.htm#1039239> (20 August 2004).

[26] "Dell™ PowerEdge™ 1650 Systems Installation and Troubleshooting Guide," <http://docs.us.dell.com/support/edocs/systems/pe1650/en/it/index.htm> (20 August 2004); "Dell™ PowerEdge™ 1650 Systems," <http://support.dell.com/support/edocs/systems/pe1650/en/> (20 August 2004).

Figure 3. Dell PowerConnect 3024 Switch (top), and Dell 8 Port KVM Switch

Figure 4. APC Smart UPS 2200 Power Supply

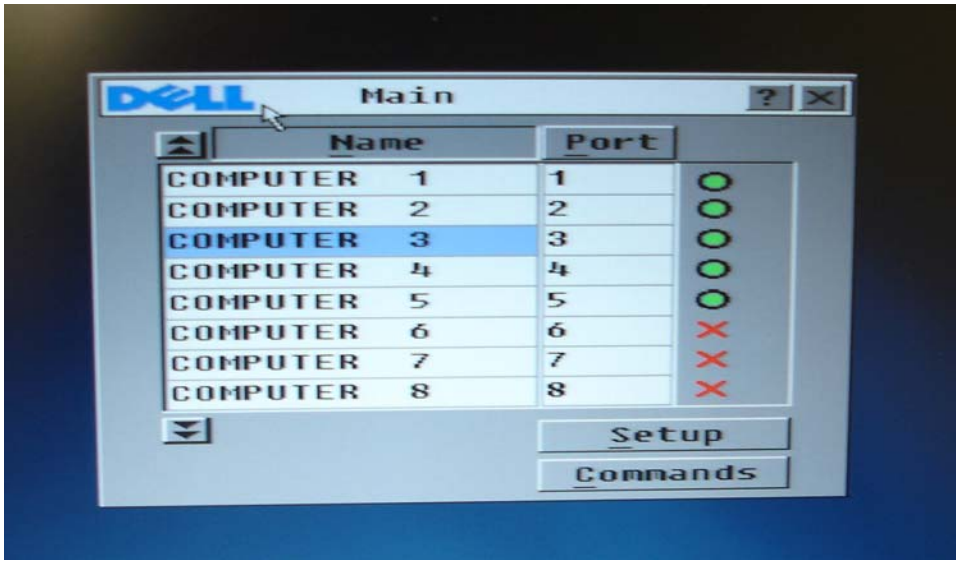


Figure 5. Print Screen Menu

Figure 6. Dell PowerEdge 1650 Rack System

B. NETWORK CONFIGURATION

Each host computer was equipped with two network cards. One was disabled and is reserved for future work. Both network cards were used on the web server for external and internal connectivity. The web server and host computers were connected by a switch, as shown in Figure 7.

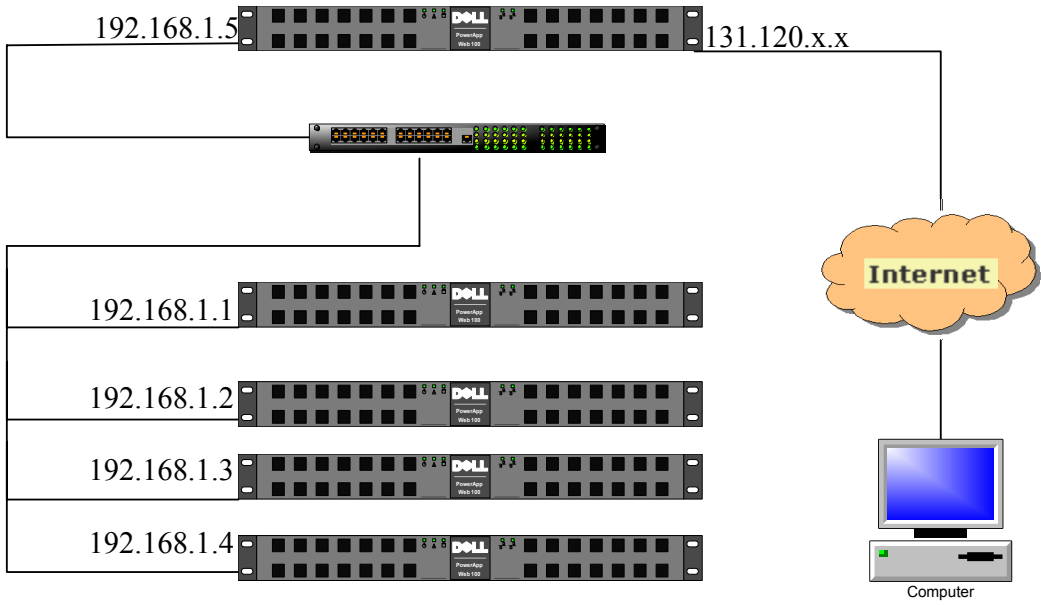


Figure 7. Network Configuration of NIC Cards

VMware was installed in the bridged mode on the hosts (Figure 8). The server was configured with Apache and Tomcat connected via the mod_jk connector. SSL functionality was invoked for security. For installation details, see Appendix A for the hosts and Appendix B for the server.

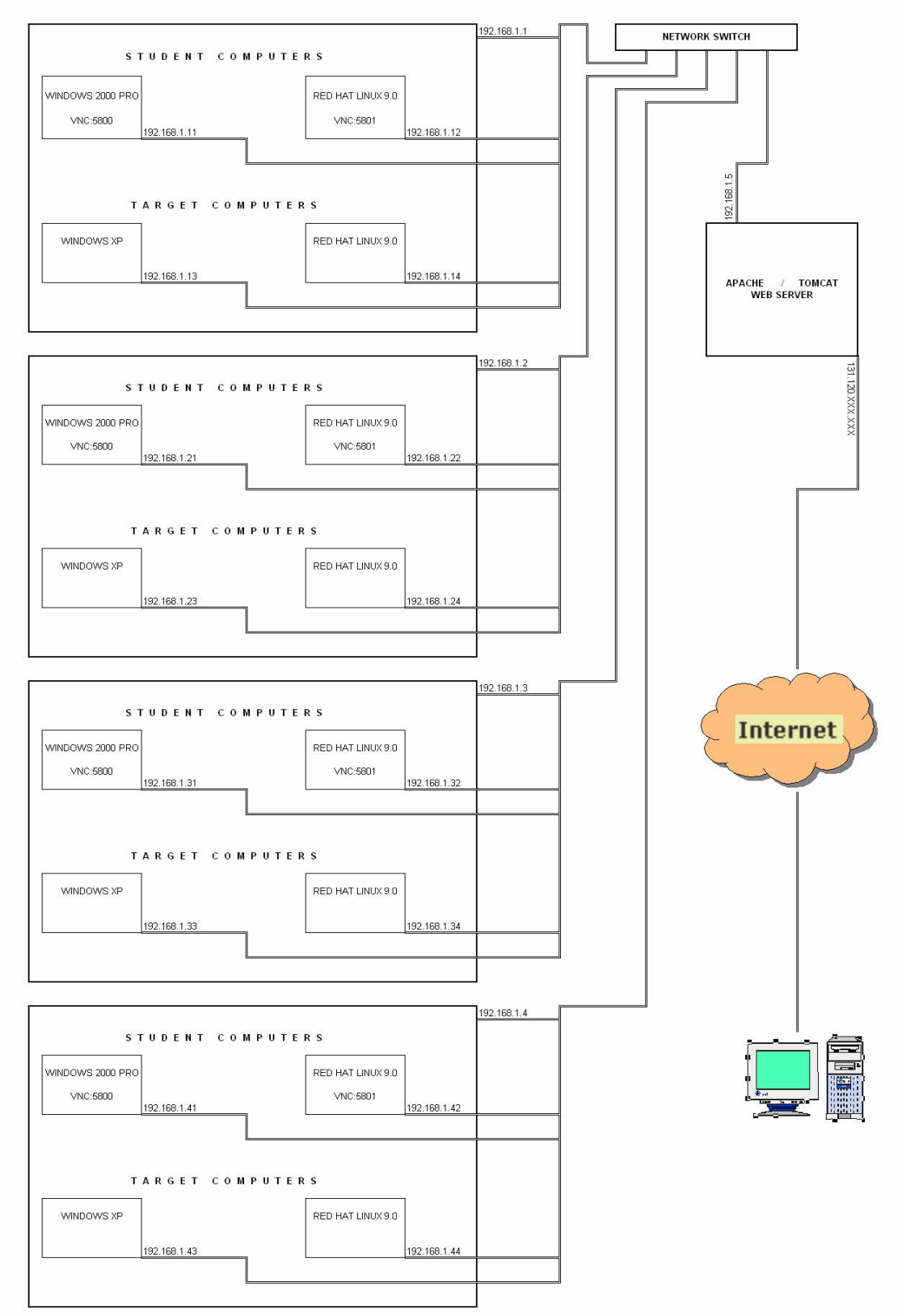


Figure 8. The Network After Installing the Virtual Machines

C. THE WEB SERVER

The web server was configured to provide the initial interface to the external user. The following services and tools were enabled:

- Secure Socket Layer (SSL) for the web server
- Tools for Java Servlets
- Web Content with links to the target virtual machines (Figure 8)

Red Hat Linux 9.0 was selected as the web server operating system because it is considered a stable and reliable kernel for web applications. The Apache-Tomcat interface architecture initially proved to be a bit difficult to get correct. However, Carillo's guide to installing web services[27], provided the necessary guidance for this thesis. The procedure followed was mostly based on the information taken from this paper, and explained in detail in Appendix A. Below is a brief summary of the steps taken:

- Configure the Environment Variables: **etc/profile** was modified to make the Environment Variables fit to the changes made such as installation directories of Java, the Tomcat server and the Apache Web Server.
- Install Java: Java was needed to support Java applications and Servlets.
- Install the Tomcat (web) Server: Tomcat is a free, open-source server solution based on the Java Platform that supports the Servlet and JSP specifications.[28] It serves the same purpose in this project as the main Web Server. Tomcat uses a different default port number (8080) for Hyper Text Transfer Protocol (HTTP) and SSL connections. The port number for SSL connections can be configured, depending on the user's preference. Port 8009 was selected for this project.
- Install OpenSSL: OpenSSL is a cryptography toolkit. The network protocols that OpenSSL uses are: The Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1).It also uses relevant cryptography standards that these protocols require.[29] As the official website for OpenSSL states, it can be used for the following purposes:[30]

---

27 "HOWTO: Installing Web Services with Linux/Tomcat/Apache/Struts/Postgresql/OpenSSL/JDBC/JNDI," <http://www.linuxjava.net/howto/webapp/> (20 June 2004).

28 "Tomcat FAQ Home Page," <http://www.jguru.com/faq/Tomcat> (26 August 2004).

29 "OpenSSL," <http://www.openssl.org/docs/apps/openssl.html> (26 August 2004).

30 Ibid.

- Creation of RSA, DH and DSA key parameters
- Creation of X.509 certificates, Certificate Signing Requests (CSR) and Certificate Revocation Lists (CRL)
- Calculation of Message Digests
- Encryption and Decryption with Ciphers
- SSL/TLS Client and Server Tests
- Handling of Secure/Multipurpose Internet Mail Extensions (S/MIME) signed or encrypted mail

In this project, it is used to create keys and certificates that would be used for authentication purposes.

- Install the Apache Http (web) server: The Apache HTTP Server is an open-source, HTTP/1.1 compliant web server. It is very powerful and flexible, and implements the latest protocols. . The Apache web server can run on Windows NT/9x/XP, Netware 5.x and above, OS/2, and most versions of UNIX and Linux as well as several other operating systems.[31] Some of its features listed on the official Apache website are:[32]
  - DBM databases for authentication
  - Customized responses to errors and problems
  - Multiple DirectoryIndex directives
  - Unlimited flexible URL rewriting and aliasing
  - Content negotiation
  - Configurable Reliable Piped Logs
  - Virtual Hosts (This allows the server to distinguish between requests made to different IP addresses or names mapped to the same machine).

In addition to Tomcat, the Apache Web Server was installed because it uses the standard default ports for HTTP (port 80) and SSL (port 443). The next step was to use the mod_jk connector so that Tomcat and Apache could operate together as a single website listening on port 80 with the ability to handle Java Servlet requests.

- Install the mod_jk Connector: The mod_jk connector is used to connect Apache to Tomcat so that they can operate together as single application. The connector was configured as follows: Once Apache and Tomcat are installed separately, they must be connected so that Apache can process

[31] <http://httpd.apache.org/docs/misc/what> (26 August 2004).

[32] Ibid.

JSP requests by handing them off to Tomcat, and Tomcat can handle http requests destined to port 80 or SSL requests destined to port 443. Although several methods exist for this purpose, **mod_jk** was used.[33] "Mod_jk contains a **Connector** component that communicates with a web connector via the JK protocol (also known as the AJP protocol)"[34]. This is used when Tomcat 4 is integrated into an existing Apache server, which enables Apache to handle the static content of the web application, and/or utilize Apache's SSL processing.[35] "In short, mod_jk is a connector that allows a web server, such as Apache HTTPD (Hyper Text Transfer Protocol Daemon) or IIS (Internet Information Server), to act as a front end to the Tomcat web application server."[36]

- Configure Apache web server for mod_jk connector: "**httpd.conf"**, the configuration file for the Apache web server, was modified in order to make the Apache server recognize the mod_jk connector.
- Configure Tomcat server for the mod_jk connector: "**server.xml"**, the configuration file for the Tomcat server, was modified and **workers.properties** was created in order to make the Tomcat server recognize the mod_jk connector. "*workers.properties"* is the name of the file where the Tomcat workers are defined. Also, a Tomcat worker is a Tomcat instance waiting to run Servlets on behalf of some web server. In this case, the Apache web server forwards Servlet requests to a Tomcat worker running behind it.[37]
- Configure the Apache web server for non-SSL connections: "**httpd.conf"** was configured to make the Apache web server allow non-SSL connections.
- Configure the Apache web server for SSL connections: **ssl.conf**, the configuration file SSL connections, was configured to make the Apache web server allow SSL connections.
- Enable SSL on Apache and Tomcat: Certificates and keys for them were issued using OpenSSL and integrated into the system.
- Build the website: The website was built using one of the templates offered by Microsoft Office Publisher 2003. A site with three pages, a welcome page, a page with links to the virtual machines, and one last page for posting announcements and assignments, was created.

33 "JSP Quick-Start Guide for Linux," <http://www.sitepoint.com/article/jsp-quick-start-guide-linux/4> (26 August 2004).

34 "Server Configuration Reference," <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk.html> (26 August 2004).

35 Ibid.

36 "Apache Tomcat mod_jk Connector 1.2.5 Released," <http://www.serverwatch.com/news/article.php/3091461> (26 August 2004).

37 Gal Shachor, "Tomcat workers.properties," <http://jakarta.apache.org/tomcat/tomcat-3.3-doc/Tomcat-Workers-HowTo.html> (26 August 2004).

The web source code was placed in the "/usr/local/tomcat/webapps/ROOT/", which is the default root directory of the Tomcat server for the documents to be published. The document root of the Apache web server is set to "/usr/local/apache/htdocs/nonsecure" in **httpd.conf.** Since the requests destined for the Apache web server are forwarded to the Tomcat server by the mod_jk connector, and the document root directory of Apache is not used for publishing documents. Also, the **"Redirect / https://localhost.localdomain"** line in **httpd.conf** file forwards the requests to the SSL port (port 443).

This last step ensures that every component of the server machine is installed and configured to work together, which means that the Tomcat server is compatible with the Java Servlets and JSPs that can handle the regular http and SSL requests coming to the Apache web server

D. HOST AND VIRTUAL MACHINES

Four of the computers were configured as host machines. This section will cover how the host machines and the virtual machines are set up and what other software and configuration changes are made.

The OS of the host machines was again Red Hat Linux 9.0. Refer to Appendix B for host machine setup details. The configuration summary is listed below.

- Install VMware: VMware Workstation 4.5 was installed on the host machines.
- Install Virtual Machines: The number of virtual machines that could be supported by the host computer was a function of the hosts' hardware configuration. Although hard disk space and processor speed was a concern, it resulted that the amount of host memory was of primary importance. It was determined that with 1 GB of RAM, the host could support 4 VMs comfortably. Taking such considerations into account, four operating systems were chosen to install as guest machines. Two were Red Hat Linux 9.0 (Desktop Version), and the other two were Windows 2000 Professional, and Windows XP Professional.
- Install VMware Tools: After installing the guest machines, VMware Tools must be installed. VMware Tools is a pack of tools integrated in the VMware for each supported operating system to increase the graphics performance, to support shared folders, and drag-and-drop operations. "Other tools in the package support synchronization of time in the guest operating system with time on the host, automatic grabbing and releasing

of the mouse cursor, copying and pasting between guest and host, and improved mouse performance in some guest operating systems."[38]

- Configure the virtual machines to run automatically on startup by running a script: A short script was written to start the virtual machines (See Appendix B).
- Configure the virtual machines for automatic logon: The virtual machines had to be configured for automatic log on so that they could be started with the script written without the need for logging in to each machine every time.
- Configure the virtual machines for persistent mode: The goal was for the system to discard all the changes made by a user and revert back to its initial stage once it was powered off or restarted.
- Install VNC (Virtual Network Computing) on the guest machines: As mentioned in Chapter II, VNC consists of two components: A server and a client. For this project, only the VNC server was installed on "Student" virtual machines, one Linux VM and one Windows VM, on each host machine (Figure 3.7).

Access to the VNC server can be accomplished via the standard VNC client or through a Java enabled web browser. The VNC Server binds to two default ports. The default ports may change according to the version used. Two different versions of the VNC server were used. The VNC version 4.0 was used for Windows machines, which listens to ports 5800 and 5900 by default. For Linux machines, the VNC package, included in the Red Hat Linux 9.0, was used for the installation of CDs. The default VNC ports for Linux are 5801 and 5901.[39] It is possible to change the ports, but the default ports were used.[40] Port 5901 is used for the connections from the VNC client software. Thus, this port is used if desiring to connect to the VNC server via the VNC client. Port 5801, on the other hand, is used for serving the Java viewer via http. The connections from the browsers need to connect to port 5801 for accessing the VNC Server. After the initial connection and authentication, the VNC Server transfers back to port 5901. Thus, after the initial connection, the server reverts back to the same port it uses for connections

---

38 "Workstation 4 User's Manual," <http://VMware-svca.www.conxion.com/software/ws45_manual.pdf> pp. 81-94 (26 August 2004).

39 To avoid further confusion, only ports 5801 and 5901 will be mentioned. However, in reality, these ports should be understood as 5800 and 5900 for Windows machines.

40 For the configuration of the VNC server, please refer to: "VNC Server 4.0 for Windows," <http://www.realvnc.com/v4/winvnc.html#4> (20 August 204).

from the VNC client software. This situation prevented the configuration of guest virtual machines in the Network Address Translation (NAT) mode. Therefore, the virtual machines were configured in bridged mode as described in Appendix B.

IV. RESULTS

A. ARCHITECTURE

- Apache and Tomcat servers were installed, connected, and run together successfully.
- The VNC server, the choice for the remote desktop display tool, proved to be efficient for the scope of this study and performed nicely.
- All the configuration options for VMware program worked as advertised. The configuration of virtual machines in bridged mode was successful.
- Scripts were developed to automate VM start-up and stop (see Appendix B).
- Individual targets were configured, successfully in the snapshot, non-persistent mode. Therefore, students could literally wipe out the target Operating System and the VM would fully recover upon re-boot.
- Although the NAT configuration of the virtual machines was successful, access to the virtual machines with the browser via NAT through the host machines could not be implemented because of the specific way the VNC Server functions. It was alternating to another port once the connection was established, and the attempts to configure the system to follow this process in the NAT produced no result.
- A student web interface was developed and a user guide was produced.
- Host-machine performance was tested for CPU and memory load under different conditions, see Section B of this chapter.
- The Virtual Lab provides the user with a look at up to 20 targets (four hosts and 16 VMs.) at the cost of only five actual computers. Power, space and time are conserved with this lab architecture.

B. PERFORMANCE

The Successful Virtual Lab installation, as discussed in Chapter III, afforded the opportunity to test the setup under different user conditions. Since the lab is designed to provide a target network to students conducting computer security research and education, it was important to discover its usefulness and viability under different user loads. In other words, would a user tolerate the performance hits inherent to the Virtual Lab Architecture? Two functional areas were initially evaluated: network bandwidth through put and host-machine performance.

It was soon realized that network bandwidth throughput evaluations were of no value for this experiment because the Virtual Lab was to exist in a closed environment with more throughput capacity than the lab would ever require. In other words, the network bandwidth throughput would never be a performance bottleneck under the current method of employment.

The interest, however, is in a subjective, qualitative view on the difference in the performance between a web browser connection and the standard VNC viewer application over this internal network. The concept was to stay with the web browser configuration because it was a more general solution for more users. However, it was virtually impossible to not notice that when the web browser was removed from the architecture and the VNC client was used to connect to the target, the experience felt more like an actual computer. This was partly due to the border sizing issues inherent in any browser but also because, in general, the response time was perceptively faster and smoother. This was attributed to the optimization of the VNC client for the VNC application and the more time required for the translation to html content provided to the client browser.

The host machine performance, on the other hand, was certainly a candidate for performance evaluation, since it would be the only real source for potential bottlenecks. Although careful thought was given to the selection of the blade servers used for host machines (dual processor with a lot of RAM), it was necessary to collect some data to support the author's claim.

The application **"top",** a system usage statistic tool, was used to evaluate host-machine performance. It is bundled with most Linux/Unix operating system and is, by default, a formatted text dump to the screen (Figure 9).

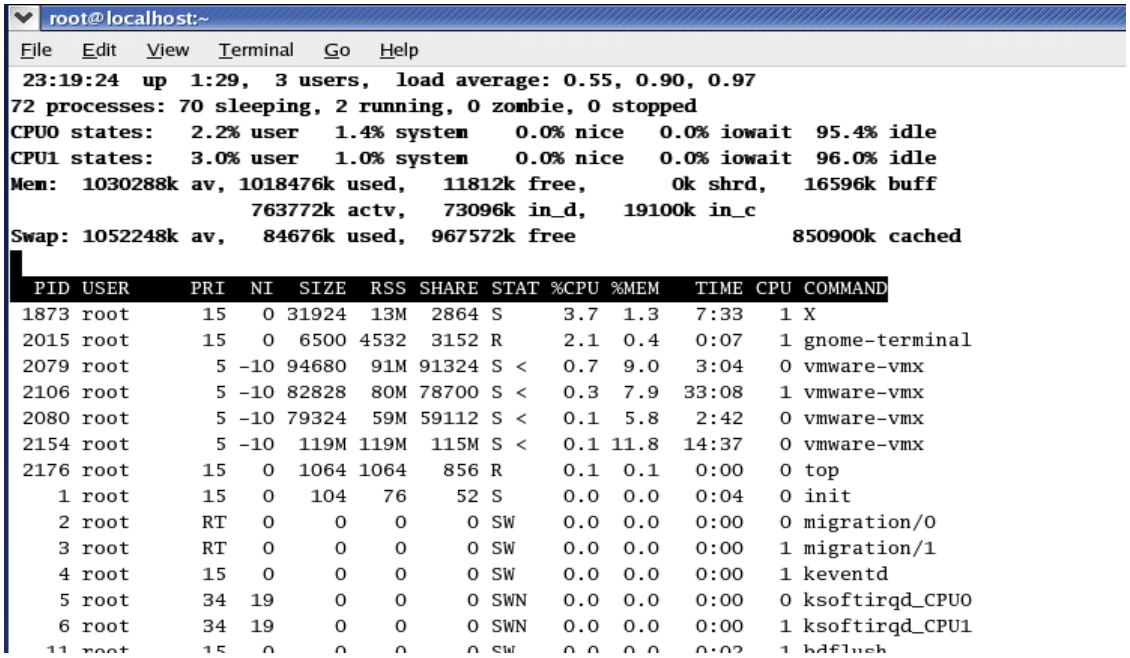


Figure 9. The Screen Output of "top" Command

There are 13 column outputs starting with PID and ending with the name of the application. Notice that both CPUs are accounted for in column 12 and that vital systems data is easily parsed. From the shell,

[root@localhost root]# top | grep vmware-vmx

which produces output similar to Figure (10).

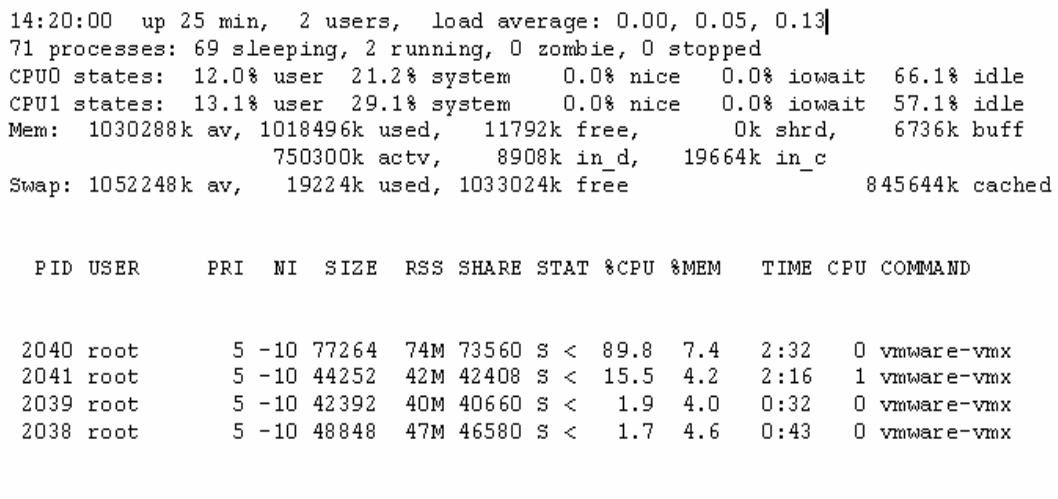
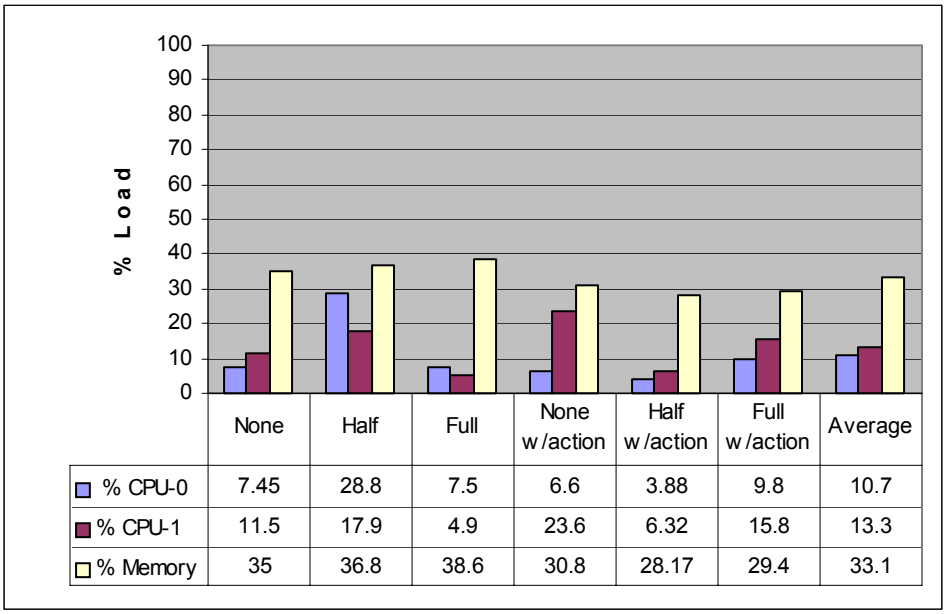


Figure 10. The Screen Output of "top" Command Together with "grep" Command

The analysis included the evaluation % CPU, and the % memory of the host machine under different student VM load conditions. The results are tabulated below.



	None	Half	Full	None w/action	Half w/action	Full w/action	Average
% CPU-0	7.45	28.8	7.5	6.6	3.88	9.8	10.7
% CPU-1	11.5	17.9	4.9	23.6	6.32	15.8	13.3
% Memory	35	36.8	38.6	30.8	28.17	29.4	33.1

Table 2. The Percentage of CPU and Memory Use by Virtual Machines

None = no users connected

Half = 1 users connected

Full = 2 users connected

Action = User application Interaction

The averages for % CPU and % Memory clearly indicate that the host machine configuration was more than sufficient to support the number of virtual machines required for the virtual lab. Notice that even under full load with action, the host machine is not close to being threatened by performance problems. This confirmed the author's architecture design and indicated that there was excess CPU and memory capacity. This reveals that the virtual lab would easily scale with the addition of more VMs per host machine.

C. VIRTUAL LAB USERS GUIDE

Finally, a user guide was developed to assist the student during the lab. This guide appears below:

1. Host Machine and Web Server Start-Up Procedure

- Start up the host machines. If you created the virtual machines in ***/root*** directory, make sure you either log on as **root** or gain **root** privileges by using ***su*.** Otherwise, it is not possible to be able to have access to the virtual machines.
- Check also to see if the script mentioned in Appendix B starts the VMware Workstation 4.5 and all the virtual machines automatically without any problems.
- Ensure the VNC servers on two of the virtual machines designated as student machines on each host, Red Hat Linux 9.0 and Microsoft Windows 2000 Professional, are up and running. On Windows-based systems, an icon representing the VNC server is displayed in a system tray when it is running. On Linux-based systems, the following command can be entered in a shell to verify if the VNC server is running:

  [root@localhost root]# ps –A | grep vnc[41]

- Start up the web server machine with **root** privileges.
- The Tomcat server should be started before the Apache web server. Therefore, enter the following command to start it:

  [root@localhost root]# /usr/local/tomcat/bin/startup.sh (You should wait at least 30 seconds for Tomcat to complete the startup process. Then, check that you have a file called **usr/local/tomcat/conf/auto/mod_jk.conf** and that the timestamp on that file is recent.[42])

- Start the Apache web server using the following command:

  [root@localhost root]# /usr/local/apache/bin/apachectl sslstart (The pass phrase chosen for SSL is asked here.)

- With the web server running, connect externally with a browser to the website of the Virtual Lab, which would be provided by the Virtual Lab administrator.
- The servers should be stopped in the reverse order, thus, Apache is the first one to stop and Tomcat the second. The following command stops Apache:

---

[41] As a result of this command, **Xvnc,** the daemon for VNC server, should be displayed.

[42] John Turner, "Apache 2.0.47/Tomcat 4.1.27/mod_jk for Red Hat 9.0," <http://johnturner.com/howto/apache2-tomcat4127-jk-rh9-howto.html> (03 September 2004).

[root@localhost root]# /usr/local/apache/bin/apachectl stop

- Stop the Tomcat server using the following command:

  [root@localhost root]# /usr/local/tomcat/bin/shutdown.sh

2. Student User Guide

- Enter the Uniform Resource Locator (URL) of the virtual lab in the address space of the Internet browser.[43]
- The certificate issued by the website can be accepted permanently or just for the current session (Figure 11). The details of the certificate can be displayed by hitting the "Examine Certificate" button (Figure 12).

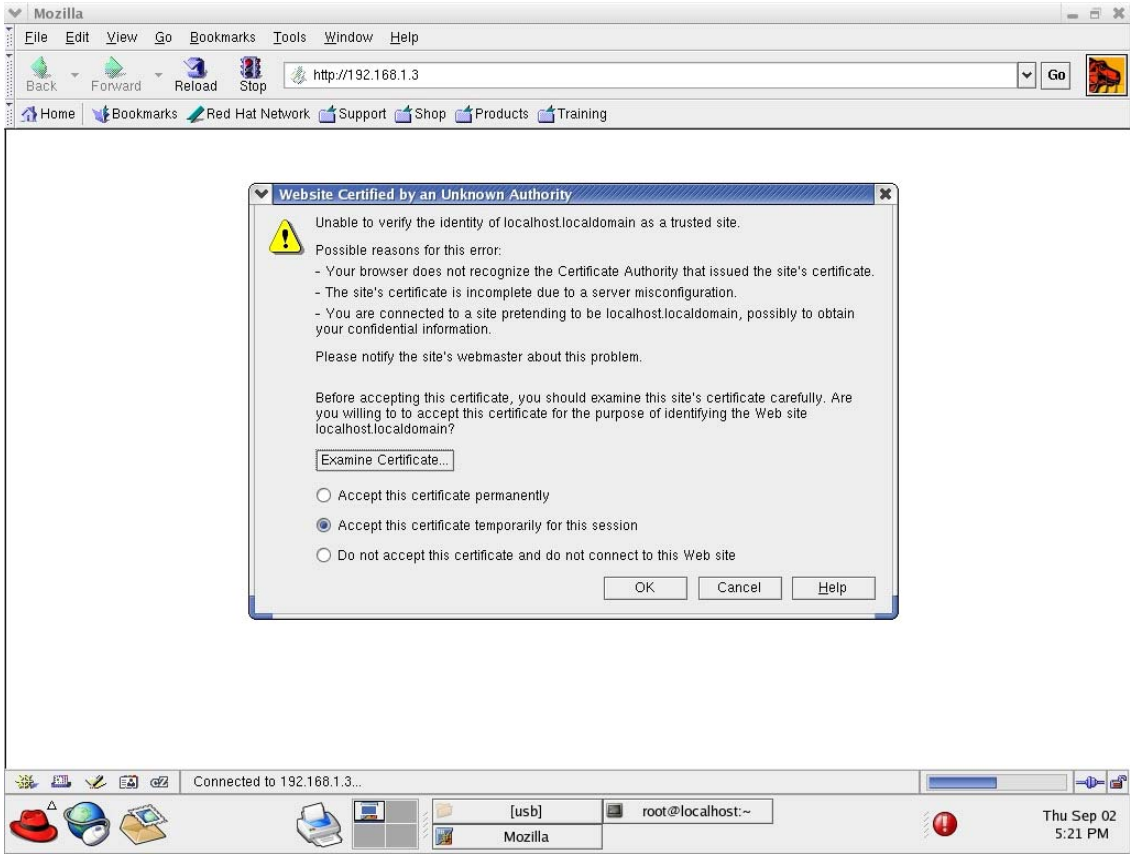


Figure 11. Accepting the SSL Certificate

[43] In this project, the virtual lab was not actually connected to the Internet. Thus, <http://localhost.localdomain/ or http://192.168.1.5/> was used as the URL of the Virtual Lab website in order to test the system. These addresses are defined in the Virtual Host configuration section of the **httpd.conf** file as shown in Appendix A.

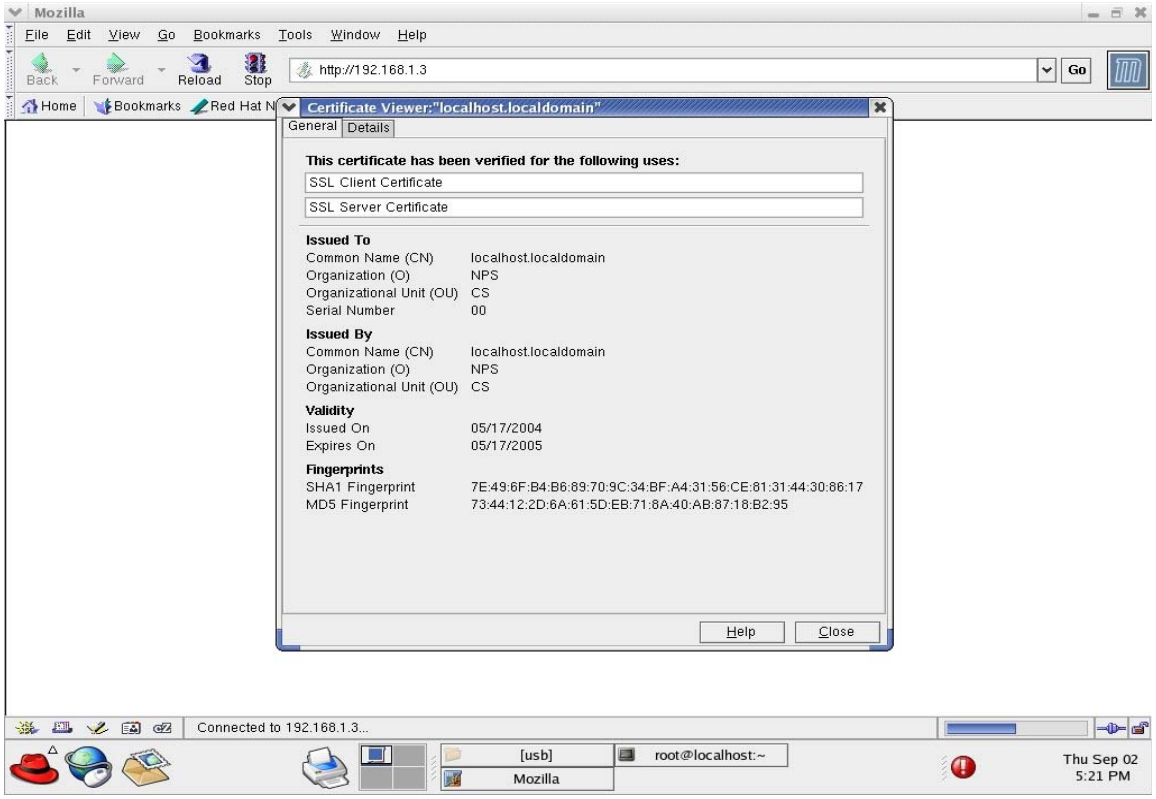


Figure 12. Details of the Certificate

- When the certificate is accepted, the home page of the website is displayed (Figure 13).

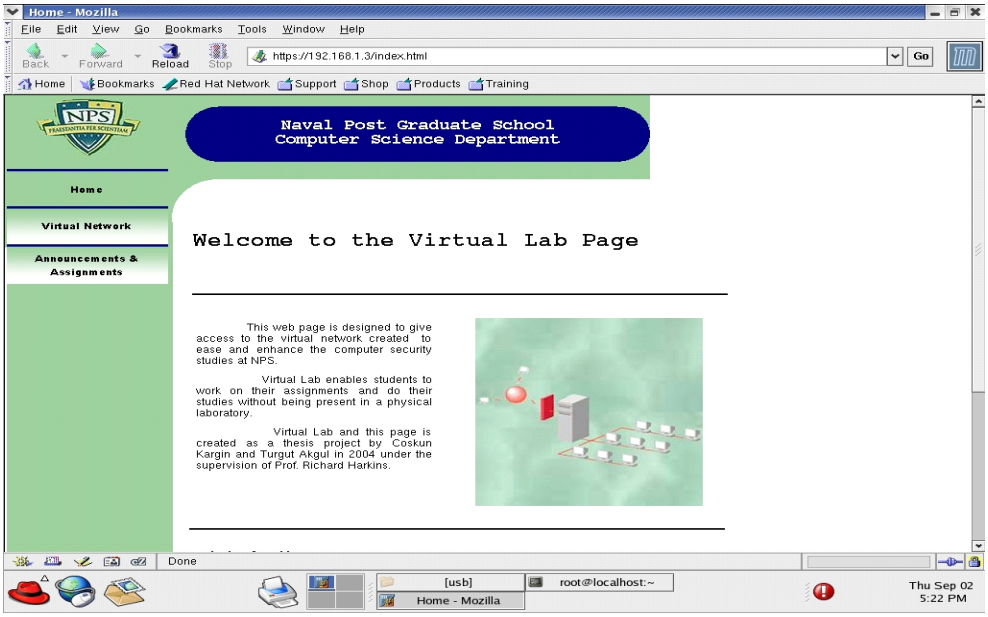


Figure 13. Home Page of the Website

- To connect to the virtual machines, go to the Virtual Network page, and click on the link of the virtual machine to which to connect (Figure 14).

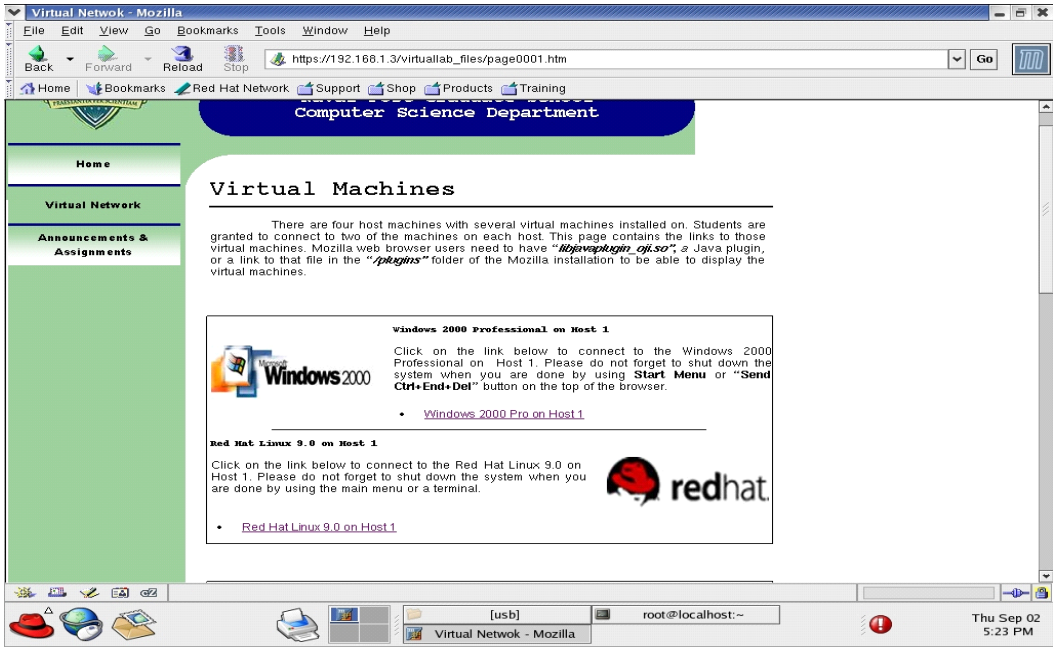


Figure 14. Virtual Network Page of the Website

- You will connect to the VNC server running on the virtual machine you choose. The built-in Java viewer comes up with a password screen for authentication (Figure 15).

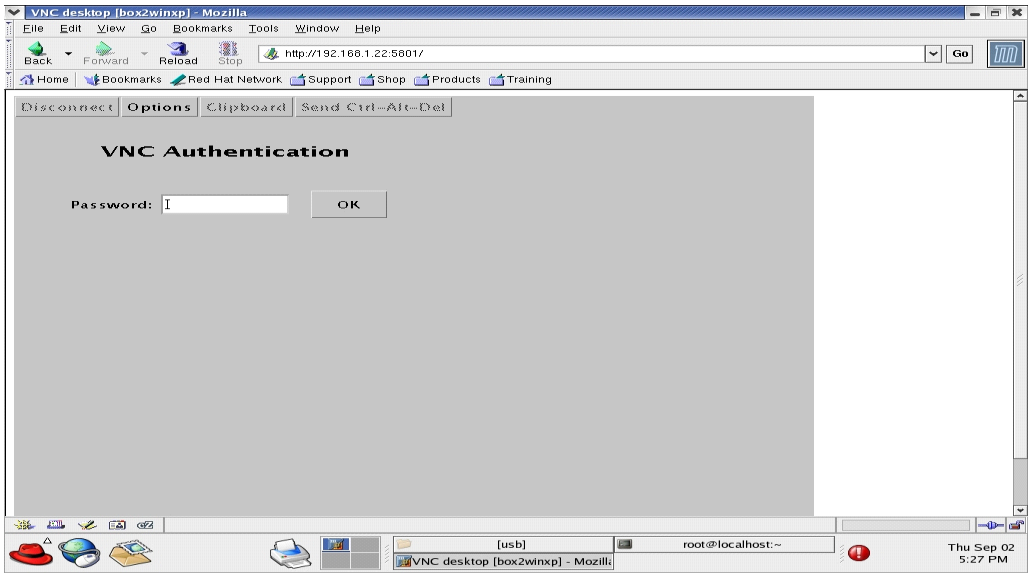


Figure 15. Password Authentication of VNC Server

- Once the password, which would be provided by the Virtual Lab administrator, is entered, you will be able to see the desktop of the virtual machine inside the web browser (Figure 16).

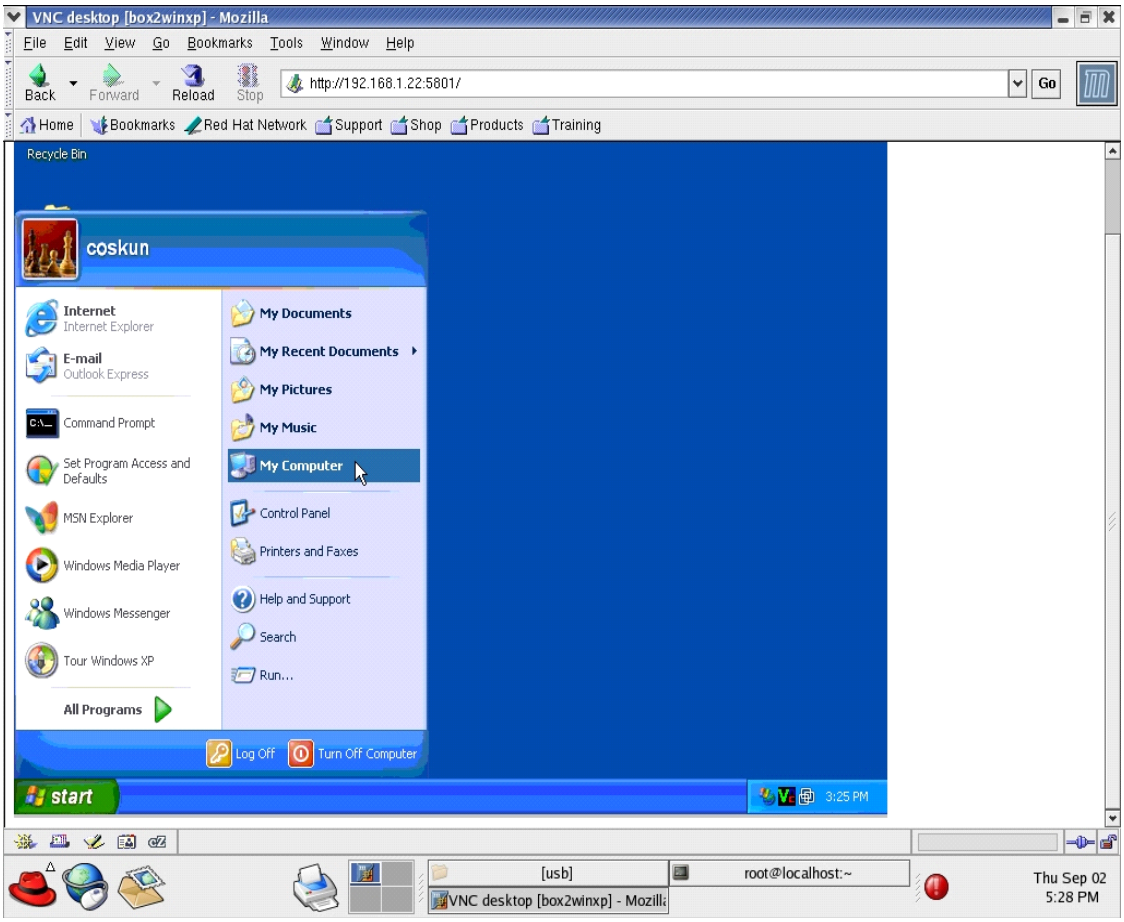


Figure 16. Displaying Virtual Machines Inside Web Browser

- The virtual Machine must be **shut down** instead of **restart** once the user has completed his/her studies because the **persistent mode** of the VMware Workstation does not work when the machine is restarted as explained in Appendix B.
- Go to the Assignments and Announcements page to see the updated information (Figure 17).

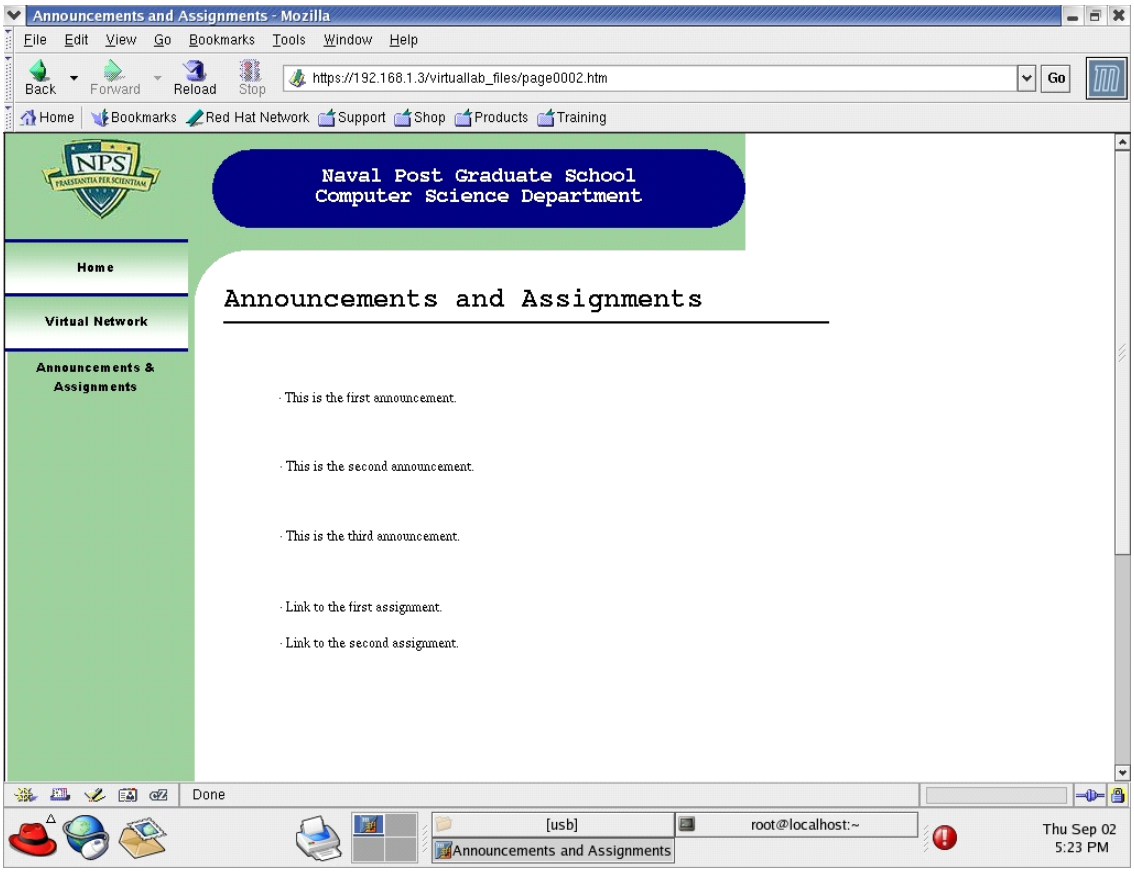


Figure 17. Announcements and Assignments Page of the Website

V. CONCLUSIONS AND FUTURE WORK

A. CONCLUSIONS:

- The architecture for a Virtual Lab was developed, implemented, and tested to support future Naval Postgraduate School (NPS) computer security research and education.
- The results indicate that a simple browser connection to the Virtual Lab, from the client, is a reasonable and cost effective alternative to an equivalent, fully equipped, lab. Performance analysis indicates that the VL is scalable and that porting to an external Internet client base is feasible.
- The web browser VNC interface, although usable, is not quite as fast and user friendly as the stand-alone VNC client. This distinction, however, would only affect users that did not have Broadband access to the lab. It is feasible to say that in a final VL implementation, a dial-up external user would find the experience frustrating.
- The web browser VNC interface is the solution of choice because most users will have a Java enabled browser and the need for the actual VNC client would not be required.
- Users could choose to use the standard VNC client if better performance was desired, which is only a matter of downloading and installing the client on their own.

B. FUTURE WORK

- Scale the VL by adding more, at least three, target machines and porting the lab for external Internet use. Recalculate and analyze host performance parameters.
- Populate, test and employ a full compliment of computer security tools and exploits in the Virtual lab environment.
- Reconfigure the Tomcat server so that it provides user interaction via Servlets. This will require the editing and use of the VNC viewer Java Servlet freely available from ATT labs.
- Increase lab security by isolating the target machines from the user with the use of NAT on the host computer.
- Invoke Public Key Infrastructure (PKI) credentials during user login. The user would be required to send and register their public key with the VL prior to the first login.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. WEB SERVER INSTALLATION

A. WEB SERVER CONFIGURATION

A customized *Server* version of Red Hat Linux 9.0 was used. First, the partition table with *Disk Druid* was modified as shown below (Table 3):

Hard Drives	Mount Point	Type	Format	Size	Start	End
/dev/sda						
/dev/sda1	/	ext3	√	33683	1	4294
/dev/sda2		swap	√	1028	4295	4425

Table 3. Partition Table of Server Machine

For the Network Configuration, both Network Interface Cards (NIC), *eth0* and *eth1*, were selected to be active on boot. In the Firewall Configuration section, *eth0* and *eth1* were set as trusted devices, and World Wide Web (WWW), Secure Shell (SSH), and Dynamic Host Configuration Protocol (DHCP) connections were allowed.

The chart below shows the RPM (Red Hat Package Manager) packages installed on the server machine (Table 4):

Desktops	X Window System	All of the packages
	GNOME Desktop Environment	All of the packages
	KDE Desktop Environment	None of the packages
Applications	Editors	Default packages
	Engineering and Scientific	Default packages
	Graphical Internet	Default packages
	Text-Based Internet	Default packages, *lynx*
	Office/Productivity	Default packages
	Sound and Video	Default packages
	Authoring and Publishing	Default packages
	Graphics	None of the packages
	Games and Entertainment	None of the packages
Servers	Server Configuration Tools	Default packages except *httpd*
	Web Server	None of the packages
	Mail Server	None of the packages

	Windows File Server	None of the packages
	DNS Name Server	None of the packages
	FTP Server	None of the packages
	SQL Database Server	None of the packages
	News Server	None of the packages
	Network Servers	Default packages, *dhcp, krb5-server*
Development	Development Tools	All of the packages
	Kernel Development	All of the packages
	X Software Development	All of the packages
	Gnome Software Development	All of the packages
	KDE Software Development	None of the packages
System	Administration Tools	All of the packages
	System Tools	Default packages
	Printing Support	Default packages

Table 4. Installation Packages of Server Machine

B. THE SOFTWARE

The following list of files were downloaded from the Internet and copied to **/usr/local/src/**.

- httpd-2.0.49.tar.gz (source)[44]
- openssl-0.9.7d.tar.gz (source)[45]
- j2sdk-1_4_2_04-linux-i586.bin (binary)[46]
- jakarta-tomcat-4.1.30.tar.gz (binary)[47]
- jakarta-tomcat-connectors-jk-1.2-src-current.tar.gz (source)[48]

Before these applications were installed, a check for older versions was conducted to prevent version conflicts and to resolve dependency issues.

44 <http://httpd.apache.org/download.cgi> (21 June 2004)

45 <http://www.openssl.org/source/> (21 June 2004).

46 <http://java.sun.com/j2se/1.4.2/download.html> (21 June 2004)

47 <http://jakarta.apache.org/site/binindex.cgi> (21 June 2004).

48 <http://jakarta.apache.org/site/sourceindex.cgi> (21 June 2004).

C. CONFIGURING ENVIRONMENT VARIABLES

The *vi* editor was used to add the following lines to **"/etc/profile"**. The directory structure can be modified depending on the user preference and since */etc/profile* maps proper path relationships for new application installations, it is important to spend some time to ensure correctness.

JAVA_HOME=/usr/local/java/java (sets the default directory of Java to "/usr/local/java/java")

CATALINA_HOME=/usr/local/tomcat (sets the default directory of Tomcat web server to "/usr/local/tomcat")

PATH=$JAVA_HOME/bin:$PATH:$HOME/bin:/sbin:/usr/sbin (adds "/usr/local/java/java/bin which holds the binary files of Java to PATH)

CLASSPATH=$CATALINA_HOME/bin/bootstrap.jar:$JAVA_HOME/lib/tools.jar:$CATALINA_HOME/common/lib/servlet.jar:.

The classpath is a string consisting of directories that tells the JVM where to look for classes it needs to load. In Linux, it is set as an environmental variable. Here, it sets where the jar files are located.

The export line in "/etc/profile" should also be modified as below:

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC JAVA_HOME CATALINA_HOME CLASSPATH

After making these changes and saving the file, it was necessary to log out and then log back in for the changes to take effect. Logging out and logging in after every installation step is a good practice. To see the latest environment variables, the following command was used:

[root@localhost root]# env

D. INSTALLING JAVA:

The following commands were invoked from the shell in order to install Java:

[root@localhost root]# mkdir /usr/local/java (creates a folder titled "java" under "/usr/local")

[root@localhost root]# cd /usr/local/src (changes directory to "/usr/local/src" where all the source or binary files of the tools are located)

[root@localhost src]# sh j2sdk-1_4_2_04-linux-i586.bin (extracts out "j2sdk1.4.2_04" directory)

[root@localhost src]# mv j2sdk1.4.2_04 /usr/local/java (moves "j2sdk1.4.2_04" directory under "/usr/local/java")

[root@localhost src]# cd /usr/local/java (changes directory to "/usr/local/java", which is something that has to be done in order to make the following command work properly)

[root@localhost java]# ln -s j2sdk1.4.2_04 java (creates a symbolic link called "*java*" to the directory "j2sdk1.4.2_04", which means *java* and *j2sdk1.4.2_04* point to the same directory under "/usr/local/java")

E. INSTALLING JAKARTA TOMCAT SERVER

The following commands installed Jakarta TOMCAT:

[root@localhost root]# cd /usr/local/src (changes directory to "/usr/local/src" where all the source or binary files of the tools are located)

[root@localhost src]# tar xvfz jakarta-tomcat-4.1.30.tar.gz (extracts the contents of file jakarta-tomcat-4.1.30.tar.gz)

[root@localhost src]# mv jakarta-tomcat-4.1.30 /usr/local/ (moves the extracted "jakarta-tomcat-4.1.30" directory under "/usr/local")

[root@localhost src]# cd /usr/local (changes directory to "/usr/local" which is something that has to be done in order to make the following command work properly)

[root@localhost local]# ln -s jakarta-tomcat-4.1.30 tomcat (creates a symbolic link called "tomcat" to the directory "jakarta-tomcat-4.1.30", which means tomcat and jakarta-tomcat-4.1.30 point to the same directory under "/usr/local")

F. INSTALLING OPENSSL

The following commands installed OpenSSL:

[root@localhost root]# cd /usr/local/src (changes directory to "/usr/local/src" where all the source or binary files of the tools are located)

[root@localhost src]# tar xvfz openssl-0.9.7d.tar.gz (extracts the contents of file openssl-0.9.7d.tar.gz)

[root@localhost src]# cd openssl-0.9.7d (changes directory to "/usr/local/src/openssl-0.9.7d", which is extracted after running the previous command)

[root@localhost openssl-0.9.7d]# ./config

[root@localhost openssl-0.9.7d]# make

[root@localhost openssl-0.9.7d]# make test

[root@localhost openssl-0.9.7d]# make install (This command together with the previous three commands conclude and configure SSL installation, and at the end OpenSSL is installed in "/usr/local/ssl")

G. APACHE HTTP (WEB) SERVER INSTALLATION

The following commands installed the Apache web server:

[root@localhost root]# export CFLAGS="-I/usr/kerberos/include/ -L/usr/kerberos/lib" (Sets the include and library path for Kerberos, which is needed since the web server will be configured as SSL enabled.)

[root@localhost root]# cd /usr/local/src (changes directory to "/usr/local/src" where all the source or binary files of the tools are located)

[root@localhost src]# tar xvfz httpd-2.0.49.tar.gz (extracts the contents of file httpd-2.0.49.tar.gz)

[root@localhost src]# cd httpd-2.0.49 (changes directory to "/usr/local/src/httpd-2.0.49", which is extracted after running the previous command)

[root@localhost httpd-2.0.49]# ./configure --prefix=/usr/local/apache --enable-so --enable-rewrite --enable-ssl --with-ssl=/usr/local/ssl --enable-proxy (configures the web server to work with SSL and proxy, ands sets the root directory as "/usr/local/apache")

[root@localhost httpd-2.0.49]# make

[root@localhost httpd-2.0.49]# make install (installs the Apache web server together with the previous command.)

In case of an unexpected error, **"make uninstall"** command can be used to remove the installation from the system.

H. BUILDING/INSTALLING MOD_JK CONNECTOR

From the shell:

[root@localhost root]# cd /usr/local/src (changes directory to "/usr/local/src" where all the source or binary files of the tools are located)

[root@localhost src]# tar xvfz jakarta-tomcat-connectors-jk-1.2-src-current.tar.gz (extracts the contents of file jakarta-tomcat-connectors-jk-1.2-src-current.tar.gz)

[root@localhost src]# cd /usr/local/src/jakarta-tomcat-connectors-jk-1.2.5-src/jk/native (changes directory to "/usr/local/src/jakarta-tomcat-connectors-jk-1.2.5-src/jk/native", which is extracted after running the previous command)

[root@localhost native]# ./buildconf.sh (complies and builds mod_jk connecter)

[root@localhost native]# ./configure --with-apxs=/usr/local/apache/bin/apxs (configures mod_jk by using apxs which is located in "/usr/local/apache/bin". apxs is a tool for building and installing extension modules for the Apache server.)

[root@localhost native]# make (creates mod_jk.so file which is configured to work with the current Apache web server on the system.)

[root@localhost native]# cp apache-2.0/mod_jk.so /usr/local/apache/modules (copies mod_jk.so file to "/usr/local/apache/modules")

I. CONFIGURING APACHE WEB SERVER FOR MOD_JK CONNECTOR

The following lines must be added to the **"/usr/local/apache/conf/httpd.conf"** file just before the line "*NameVirtualHost"*.

<IfModule !mod_jk.c>

LoadModule jk_module modules/mod_jk.so

</IfModule>

J. CONFIGURING TOMCAT SERVER FOR MOD_JK CONNECTOR

In order to configure the Tomcat server properly for the mod_jk connector, two things must be done. First, **"server.xml"** in **"/usr/local/tomcat/conf/"** is modified as follows:

- After this line:

<Server port="8005" shutdown="SHUTDOWN" debug="1"\>

- Add these lines:

<Listener className="org.apache.ajp.tomcat4.config.ApacheConfig" modJk="/usr/local/apache/modules/mod_jk.so (Defines the location of *mod_jk.so* file)

workersConfig="/usr/local/tomcat/conf/jk/workers.properties"/> (Defines the location of *workers.properties* file, which is explained in detail in Chapter III)

- After this line:

<Host name="localhost" debug="0" appBase="webapps">

- Add these lines:

<Listener className="org.apache.ajp.tomcat4.config.ApacheConfig" append="true" forwardAll="false" modJk="/usr/local/apache/modules/mod_jk.so" />

- Comment out JK2 connector, and uncomment AJP1.3 connector (jk).
- Change instances of **localhost** to whatever the domain name is, e.g. in this case, it was **localhost.localdomain.**

Second, create a file called **workers.properties** with the following contents and place it in "**/usr/local/tomcat/conf/jk/".**

Setting Tomcat & Java Home

workers.tomcat_home=/usr/local/tomcat

workers.java_home=/usr/local/java/java

ps=/

worker.list=ajp13

worker.ajp13.port=8009

worker.ajp13.host=localhost

worker.ajp13.type=ajp13

K. CONFIGURING THE APACHE WEB SERVER FOR NON-SSL CONNECTIONS

Before modifying *httpd.conf*, the following directories should be created:

[root@localhost root]# mkdir /usr/local/apache/htdocs/nonsecure (creates a directory called "nonsecure" in "/usr/local/apache/htdocs" .)

After creating the folders, the "/usr/local/apache/conf/httpd.conf" file was modified according to this system. Below is the modified version of httpd.conf:[49]

Section 1: Global Environment

[49] The lines in italic show the parts of httpd.conf file modified according to the system in this project.

ServerRoot "/usr/local/apache"

<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>
#LockFile logs/accept.lock
</IfModule>
</IfModule>

<IfModule !mpm_netware.c>
<IfModule !perchild.c>
#ScoreBoardFile logs/apache_runtime_status
</IfModule>
</IfModule>

<IfModule !mpm_netware.c>
PidFile logs/httpd.pid
</IfModule>

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule prefork.c>
StartServers 5
MinSpareServers 5
MaxSpareServers 10
MaxClients 150
MaxRequestsPerChild 0
</IfModule>

<IfModule worker.c>
StartServers 2
MaxClients 150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
MaxRequestsPerChild 0
</IfModule>

<IfModule perchild.c>
NumServers 5
StartThreads 5
MinSpareThreads 5
MaxSpareThreads 10
MaxThreadsPerChild 20

```
MaxRequestsPerChild  0
</IfModule>

<IfModule mpm_winnt.c>
ThreadsPerChild 250
MaxRequestsPerChild  0
</IfModule>


<IfModule beos.c>
StartThreads              10
MaxClients                50
MaxRequestsPerThread      10000
</IfModule>

<IfModule mpm_netware.c>
ThreadStackSize     65536
StartThreads          250
MinSpareThreads        25
MaxSpareThreads       250
MaxThreads           1000
MaxRequestsPerChild     0
MaxMemFree            100
</IfModule>

<IfModule mpmt_os2.c>
StartServers          2
MinSpareThreads       5
MaxSpareThreads       10
MaxRequestsPerChild   0
</IfModule>

Listen localhost.localdomain:80
Listen 192.168.1.5:80

### Section 2: 'Main' server configuration

<IfModule !mpm_winnt.c>
<IfModule !mpm_netware.c>

User nobody
Group #-1
</IfModule>
</IfModule>

ServerAdmin you@example.com
```

ServerName localhost.localdomain:80
ServerName 192.168.1.5:80

UseCanonicalName Off
DocumentRoot “/usr/local/apache/htdocs”

<Directory />
  Options FollowSymLinks
  AllowOverride None
</Directory>

<Directory “/usr/local/apache/htdocs”>
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>

UserDir public_html

DirectoryIndex index.html index.html.var
AccessFileName .htaccess
<Files ~ “^\.ht”>
  Order allow,deny
  Deny from all
</Files>

TypesConfig conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>
  MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn
LogFormat “%h %l %u %t \”%r\” %>s %b \”%{Referer}i\” \”%{User-Agent}i\”“
combined
LogFormat “%h %l %u %t \”%r\” %>s %b” common
LogFormat “%{Referer}i -> %U” referer
LogFormat “%{User-agent}i” agent
combinedio

CustomLog logs/access_log common
ServerTokens Full

```
ServerSignature On

Alias /icons/ “/usr/local/apache/icons/”

<Directory “/usr/local/apache/icons”>
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

AliasMatch ^/manual(?:/(?:de|en|fr|ja|ko|ru))?(/.*)?$ “/usr/local/apache/manual$1”

<Directory “/usr/local/apache/manual”>
   Options Indexes
   AllowOverride None
   Order allow,deny
   Allow from all

   <Files *.html>
      SetHandler type-map
   </Files>

   SetEnvIf Request_URI ^/manual/de/ prefer-language=de
   SetEnvIf Request_URI ^/manual/en/ prefer-language=en
   SetEnvIf Request_URI ^/manual/fr/ prefer-language=fr
   SetEnvIf Request_URI ^/manual/ja/ prefer-language=ja
   SetEnvIf Request_URI ^/manual/ko/ prefer-language=ko
   SetEnvIf Request_URI ^/manual/ru/ prefer-language=ru
   RedirectMatch 301 ^/manual(?:/(de|en|fr|ja|ko|ru)){2,}(/.*)?$ /manual/$1$2
</Directory>

ScriptAlias /cgi-bin/ “/usr/local/apache/cgi-bin/”
<IfModule mod_cgid.c>
</IfModule>
<Directory “/usr/local/apache/cgi-bin”>
   AllowOverride None
   Options None
   Order allow,deny
   Allow from all
</Directory>

IndexOptions FancyIndexing VersionSort
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
```

```
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*
AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core
AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^
DefaultIcon /icons/unknown.gif
ReadmeName README.html
HeaderName HEADER.html
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
AddLanguage ca .ca
AddLanguage cs .cz .cs
AddLanguage da .dk
AddLanguage de .de
AddLanguage el .el
AddLanguage en .en
AddLanguage eo .eo
AddLanguage es .es
AddLanguage et .et
AddLanguage fr .fr
AddLanguage he .he
AddLanguage hr .hr
AddLanguage it .it
AddLanguage ja .ja
AddLanguage ko .ko
AddLanguage ltz .ltz
AddLanguage nl .nl
AddLanguage nn .nn
AddLanguage no .no
AddLanguage pl .po
```

```
AddLanguage pt .pt
AddLanguage pt-BR .pt-br
AddLanguage ru .ru
AddLanguage sv .sv
AddLanguage zh-CN .zh-cn
AddLanguage zh-TW .zh-tw
LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl pt pt-BR ru sv
zh-CN zh-TW
ForceLanguagePriority Prefer Fallback
AddDefaultCharset ISO-8859-1
AddCharset ISO-8859-1  .iso8859-1  .latin1
AddCharset ISO-8859-2  .iso8859-2  .latin2 .cen
AddCharset ISO-8859-3  .iso8859-3  .latin3
AddCharset ISO-8859-4  .iso8859-4  .latin4
AddCharset ISO-8859-5  .iso8859-5  .latin5 .cyr .iso-ru
AddCharset ISO-8859-6  .iso8859-6  .latin6 .arb
AddCharset ISO-8859-7  .iso8859-7  .latin7 .grk
AddCharset ISO-8859-8  .iso8859-8  .latin8 .heb
AddCharset ISO-8859-9  .iso8859-9  .latin9 .trk
AddCharset ISO-2022-JP .iso2022-jp .jis
AddCharset ISO-2022-KR .iso2022-kr .kis
AddCharset ISO-2022-CN .iso2022-cn .cis
AddCharset Big5        .Big5       .big5
AddCharset WINDOWS-1251 .cp-1251   .win-1251
AddCharset CP866       .cp866
AddCharset KOI8-r      .koi8-r .koi8-ru
AddCharset KOI8-ru     .koi8-uk .ua
AddCharset ISO-10646-UCS-2 .ucs2
AddCharset ISO-10646-UCS-4 .ucs4
AddCharset UTF-8       .utf8
AddCharset GB2312      .gb2312 .gb
AddCharset utf-7       .utf7
AddCharset utf-8       .utf8
AddCharset big5        .big5 .b5
AddCharset EUC-TW      .euc-tw
AddCharset EUC-JP      .euc-jp
AddCharset EUC-KR      .euc-kr
AddCharset shift_jis   .sjis
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddHandler type-map var
BrowserMatch “Mozilla/2” nokeepalive
BrowserMatch “MSIE 4\.0b2;” nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch “RealPlayer 4\.0” force-response-1.0
BrowserMatch “Java/1\.0” force-response-1.0
BrowserMatch “JDK/1\.0” force-response-1.0
```

BrowserMatch “Microsoft Data Access Internet Publishing Provider” redirect-carefully
BrowserMatch “^WebDrive” redirect-carefully
BrowserMatch “^WebDAVFS/1.[012]” redirect-carefully
BrowserMatch “^gnome-vfs” redirect-carefully

<IfModule mod_ssl.c>
Include conf/ssl.conf
</IfModule>

<IfModule !mod_jk.c>
LoadModule jk_module modules/mod_jk.so
</IfModule>

Section 3: Virtual Hosts

NameVirtualHost localhost.localdomain:80
NameVirtualHost 192.168.1.5:80

<VirtualHost 192.168.1.5:80>
ServerAdmin rharkins@nps.navy.mil
DocumentRoot /usr/local/apache/htdocs/nonsecure
ServerName 192.168.1.5:80
ErrorLog logs/19216815-error_log
CustomLog logs/19216815_public-access_log common
Redirect / https://192.168.1.5/

Alias /mywebapp “/usr/local/tomcat/webapps”

<Directory “/usr/local/tomcat/webapps”>
Options Indexes FollowSymLinks
DirectoryIndex index.jsp
</Directory>

<Location “/mywebapp/WEB-INF/”>*
AllowOverride None
deny from all
</Location>

<Location “/mywebapp/META-INF/”>*
AllowOverride None
deny from all
</Location>

JkMount /.do ajp13*

JkMount /.jsp ajp13*
JkMount / ajp13
JkMount / ajp13*
</VirtualHost>

<VirtualHost localhost.localdomain:80>
ServerAdmin rharkins@nps.navy.mil
DocumentRoot /usr/local/apache/htdocs/nonsecure
ServerName localhost.localomain:80
ErrorLog logs/localhost.localdomain_public-error_log
CustomLog logs/localhost.localdomain_public-access_log common
Redirect / https://localhost.localdomain/

Alias /mywebapp "/usr/local/tomcat/webapps"

<Directory "/usr/local/tomcat/webapps">
Options Indexes FollowSymLinks
DirectoryIndex index.jsp
</Directory>

<Location "/mywebapp/WEB-INF/">*
AllowOverride None
deny from all
</Location>

<Location "/mywebapp/META-INF/">*
AllowOverride None
deny from all
</Location>

JkMount /.do ajp13*
JkMount /.jsp ajp13*
JkMount / ajp13
JkMount / ajp13*
</VirtualHost>

JkWorkersFile "/usr/local/tomcat/conf/jk/workers.properties"
JkLogFile "/usr/local/tomcat/logs/mod_jk.log"

SSLCertificateFile /usr/local/apache/conf/localhost.localdomain.cert
SSLCertificateKeyFile /usr/local/apache/conf/localhost.localdomain.key

L. CONFIGURING APACHE WEB SERVER FOR SSL CONNECTIONS

Before modifying ssl.conf, the following directories must be created:

[root@localhost root]# mkdir /usr/local/apache/htdocs/secure (creates a directory called "secure" in "/usr/local/apache/htdocs" .)

[root@localhost root]# mkdir /usr/local/apache/htdocs/secure/securedomain (creates a directory called "securedomain" in "/usr/local/apache/htdocs/secure .)

After creating the folders, the "/usr/local/apache/conf/ssl.conf" file was modified according to this system. Below is the modified version of ssl.conf:[50]

<IfDefine SSL>
Listen 443
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
SSLPassPhraseDialog builtin
SSLSessionCache dbm:/usr/local/apache/logs/ssl_scache
SSLSessionCacheTimeout 300

SSL Virtual Host Context

<VirtualHost _default_:443>

DocumentRoot "/usr/local/apache/htdocs/secure/securedomain"
ServerName localhost.localdomain:443
ServerAdmin ckargin@nps.navy.mil
ErrorLog /usr/local/apache/logs/error_log
TransferLog /usr/local/apache/logs/access_log

Alias /mywebapp "/usr/local/tomcat/webapps"
<Directory "/usr/local/tomcat/webapps">
Options Indexes FollowSymLinks
DirectoryIndex index.jsp
</Directory>

<Location "/WEB-INF/">*
AllowOverride None
deny from all
</Location>

<Location "/META-INF/">*
AllowOverride None
deny from all
</Location>

[50] The lines in italic show the parts of ssl.conf file modified according to the system in this project.

JkMount /.do ajp13*
JkMount /.jsp ajp13*
JkMount / ajp13
JkMount / ajp13*

SSLEngine on
SSLCipherSuite
ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNUL
L
SSLCertificateFile /usr/local/apache/conf/localhost.localdomain.cert
SSLCertificateKeyFile /usr/local/apache/conf/localhost.localdomain.key

<Files ~ "\.(cgi|shtml|phtml|php3?)$">
SSLOptions +StdEnvVars
</Files>
<Directory "/usr/local/apache/cgi-bin">
SSLOptions +StdEnvVars
</Directory>

*SetEnvIf User-Agent ".*MSIE.*" *
*nokeepalive ssl-unclean-shutdown *
downgrade-1.0 force-response-1.0
*CustomLog /usr/local/apache/logs/ssl_request_log *
"%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
</IfDefine>

M. ENABLING SSL ON APACHE AND TOMCAT

The following commands were used to create and publish the author's SSL certificates on the website:

[root@localhost root]# cd /usr/local/ssl/bin (changes directory to "/usr/local/ssl/bin" where the binary files of OpenSSL are located)

[root@localhost bin]# ./openssl genrsa –des3 –rand /etc/host -out localhost.localdomain.key 1024 (creates a 1024-bit key by using the *des3* algorithm and writes it to the file *localhost.localdomain.key*)

[root@localhost bin]# ./openssl req -new -key localhost.localdomain.key -out localhost.localdomain.csr (creates a request form to issue a certificate by using the key created earlier.)

[root@localhost bin]# ./openssl x509 -days 365 -req -in localhost.localdomain.csr -signkey localhost.localdomain.key -out localhost.localdomain.cert (creates a certificate in x509 format that would be

valid for 365 days by using the request form and the key, and names it *localhost.localdomain.cert*)

[root@localhost bin]# cp localhost.localdomain.* /usr/local/apache/conf (copies localhost.localdomain.key, localhost.localdomain.csr, and localhost.localdomain.cert to "/usr/local/apache/conf")

At this point, the following lines must be added to the "/usr/local/apache/conf/httpd.conf" file as shown in the modified httpd.conf file above.

SSLCertificateFile /usr/local/apache/conf/localhost.mydomain.cert

SSLCertificateKeyFile /usr/local/apache/conf/localhost.mydomain.key

APPENDIX B. HOST AND VIRTUAL MACHINE CONFIGURATION

A. RED HAT LINUX 9.0 INSTALLATION ON THE HOST MACHINES

The *Workstation* version of Red Hat 9.0, which is different from the version used on the server machine, was installed on each host machine. The partition table for the host was configured exactly the same as the server's, as shown in Appendix A.

Eth0 was selected to be active on boot and trusted by the *Firewall.* WWW, SSH, and DHCP connections were also activated.

The table below shows the RPM packages selected and installed (Table 5):

Desktops	X Window System	All of the packages
	GNOME Desktop Environment	All of the packages
	KDE Desktop Environment	None of the packages
Applications	Editors	Default packages
	Engineering and Scientific	None of the packages
	Graphical Internet	Default packages
	Text-Based Internet	Default packages
	Office/Productivity	None of the packages
	Sound and Video	None of the packages
	Authoring and Publishing	None of the packages
	Graphics	Default packages
	Games and Entertainment	None of the packages
Servers	Server Configuration Tools	Default packages except *httpd*
	Web Server	None of the packages
	Mail Server	None of the packages
	Windows File Server	None of the packages
	DNS Name Server	None of the packages
	FTP Server	None of the packages
	SQL Database Server	None of the packages
	News Server	None of the packages
	Network Servers	Default packages, *dhcp, krb5-server*
Development	Development Tools	Default packages
	Kernel Development	None of the packages

	X Software Development	Default packages
	Gnome Software Development	Default packages
	KDE Software Development	None of the packages
System	Administration Tools	All of the packages
	System Tools	Default packages
	Printing Support	None of the packages

Table 5. Installation Packages of Host Machines

B. VMWARE INSTALLATION

It was first necessary to verify that the computers met the minimum hardware requirements by referring to supporting documentation. *VMware-workstation-4.5.1-7568.i386.rpm* was copied to the */root* directory and after extracting the RPM, it was configured as follows:

```
[root@localhost root]# /usr/bin/vmware-config.pl
```

The configure dialog is listed below as a screen shot and reveals many of the default installation values and paths:

```
Do you want networking for your virtual machines? (yes/no/help) [yes]
Configuring a bridged network for vmnet0.
Your computer has multiple ethernet network interfaces available: eth0,
eth1.
Which one do you want to bridge to vmnet0? [eth0]
The following bridged networks have been defined:
. vmnet0 is bridged to eth0
Do you wish to configure another bridged network? (yes/no) [no]
Do you want to be able to use NAT networking in your virtual machines?
[yes] no
Do you want to be able to use host-only networking in your virtual
machines? [no]
Do you want this program to automatically configure your system to
allow your virtual machines to access the host's filesystem?
(yes/no/help) no
 Starting VMware services:
   Virtual machine monitor                                 [  OK  ]
   Virtual ethernet                                        [  OK  ]
   Bridged networking on /dev/vmnet0                       [  OK  ]
The configuration of VMware Workstation 4.5.1 build-7568 for Linux for
this running kernel completed successfully.
You can now run VMware Workstation by invoking the following command:
"/usr/bin/vmware".
```

1. Network Modes

VMware has three types of networking setups from which to choose. They are:

- Bridged Networking
- Network Address Translation (NAT)
- Host Only Networking

a. Bridged Networking

Bridged networking is the default networking option in VMware. This default setting can be changed during or after the installation. In Bridged mode, the VM will be assigned a network IP as if it were standalone computer. The host machine acts as a bridge on behalf of the VM. Bridged networking makes the virtual machine a full participant in the network. It can access other machines on the network and can be accessed by other machines on the network as if it were a physical computer on the network.

b. Network Address Translation (NAT)

If there is no need for a separate IP address for the virtual machine but access to the Internet and the other virtual machines on the same host machine by using the host computer's dial-up or broadband connection is desired, Network Address Translation (NAT) should be used. NAT sets up a private TCP/IP network on the host machine by using a Token Ring adapter. The virtual machine gets an IP address on that network from the VMware virtual DHCP server. The VMware NAT device also identifies incoming data packets intended for each virtual machine and sends them to the correct destination.

c. Host-Only Networking

If a private or isolated TCP/IP network is desired between the virtual machines, then host-only networking should be selected. In this case, the VMware DHCP server provides a non-routable IP the addresses to each VM. In this configuration, the Virtual Machines cannot network off of the host machine. Networking selection can be changed in virtual machine settings editor (**VM >Settings**).[51]

---

[51] "Workstation 4 User's Manual," <http://VMware-svca.www.conxion.com/software/ws45_manual.pdf> pp. 212-215 (26 August 2004).

The Bridged Networking configuration was chosen for this experiment.

2. Virtual Machines Installation

The necessary installation instructions for this section were taken from the "VMware User's Manual."[52] To start the VMware Workstation, the following command was entered in a terminal:

```
[root@localhost root]# vmware &
```

Selecting **File >New Virtual Machine,** when VMware is started, brings up a wizard for creating a new virtual machine. The next window in the wizard asks if the preference is a Typical or Custom configuration, and the custom configuration was selected. A prompt appears to identify the OS to install (Figure 18).

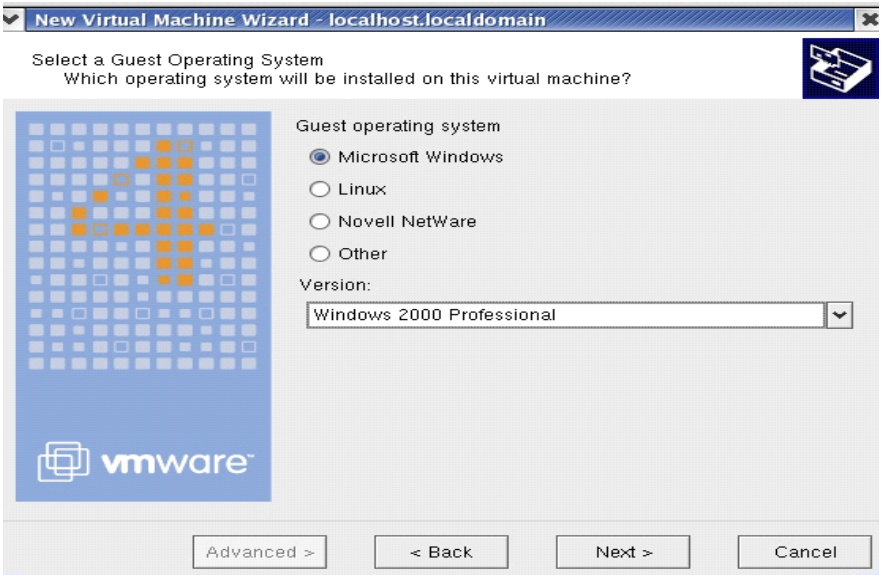


Figure 18. Selecting the OS

In the next window it is possible to define the preferred name of the virtual machine and where to create it on the hard drive. VMware creates a new folder with the name chosen, and puts every file related to that specific virtual machine in that directory. Deleting this directory simply removes that virtual machine from the hard drive.

[52] Ibid., pp. 65-80 (26 August 2004).

The snapshot below shows the step in which the memory for the virtual machine is specified. The amount of memory can be selected depending on the number of virtual machines, their operating systems, and the actual RAM of the host machine. The amount of RAM spared was 256 MB of RAM for Windows-based virtual machines and 128 MB of RAM for Linux-based virtual machines (Figure 19).

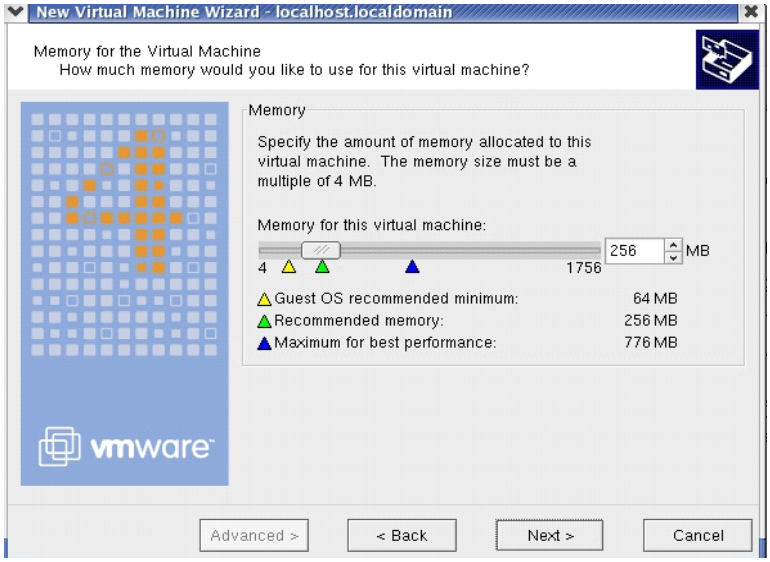


Figure 19. Determining Virtual RAM Space

The networking setup options appear in the next window. The wizard shows the networking types VMware supports together with brief descriptions of these networking types, and asks which one is desired. Bridged Networking was selected for the aforementioned reasons (Figure 20).

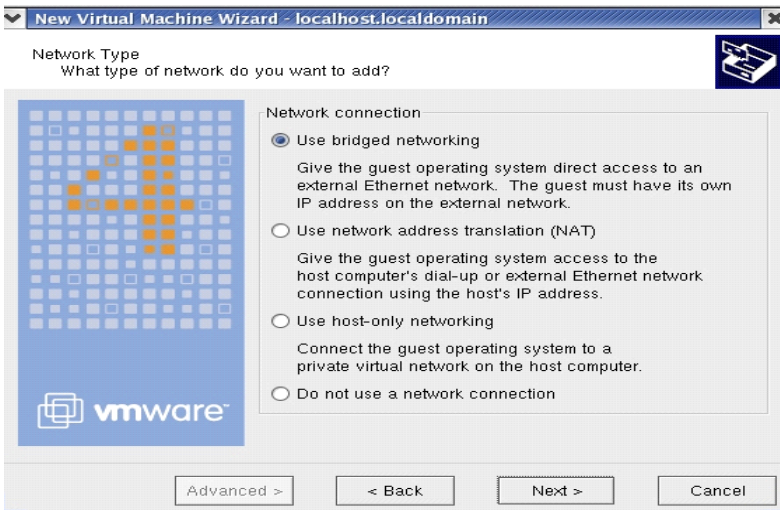


Figure 20. Networking Options

In the next window, the Input/Output (I/O) adapter types for both the Integrated Drive Electronics (IDE) and SCSI devices are chosen. ATAPI (AT Attachment Packet Interface) is the only option for the IDE adapter, and BusLogic is the default option as the SCSI adapter for most of the operating systems. The other option for the SCSI adapter is LSILogic (Large-Scale Integration Logic). The default settings are chosen in this window.

The last three steps of the wizard allow the creation of the virtual hard disk. The selection was to create a new virtual SCSI disk of 4GB for each virtual machine. Also, the option "Allocate disk space now" was checked, which creates a 4GB **wmdk** file with the name specified. VMware spares that size of actual hard drive for the virtual machine under the directory of that virtual machine (Figure 21).

Figure 21. Creating a Virtual Disk of a Specified Capacity

Creating the virtual hard disk finishes the wizard, which means new virtual machines are now ready to be started, and the guest operating system chosen is ready to be installed. Simply put the first installation CD of the guest operating system in the CD-ROM of the host machine and click on “Start this virtual machine”. Within the window of VMware Workstation, note that the new virtual machine boots up, recognizes the installation CD, and starts a regular operating system installation as if it were on an actual machine (Figure 22).

Ctrl + **Alt** + **Enter** puts the VM in full screen mode, while **Ctrl** + **Alt** exits the virtual machine and returns to the host machine.

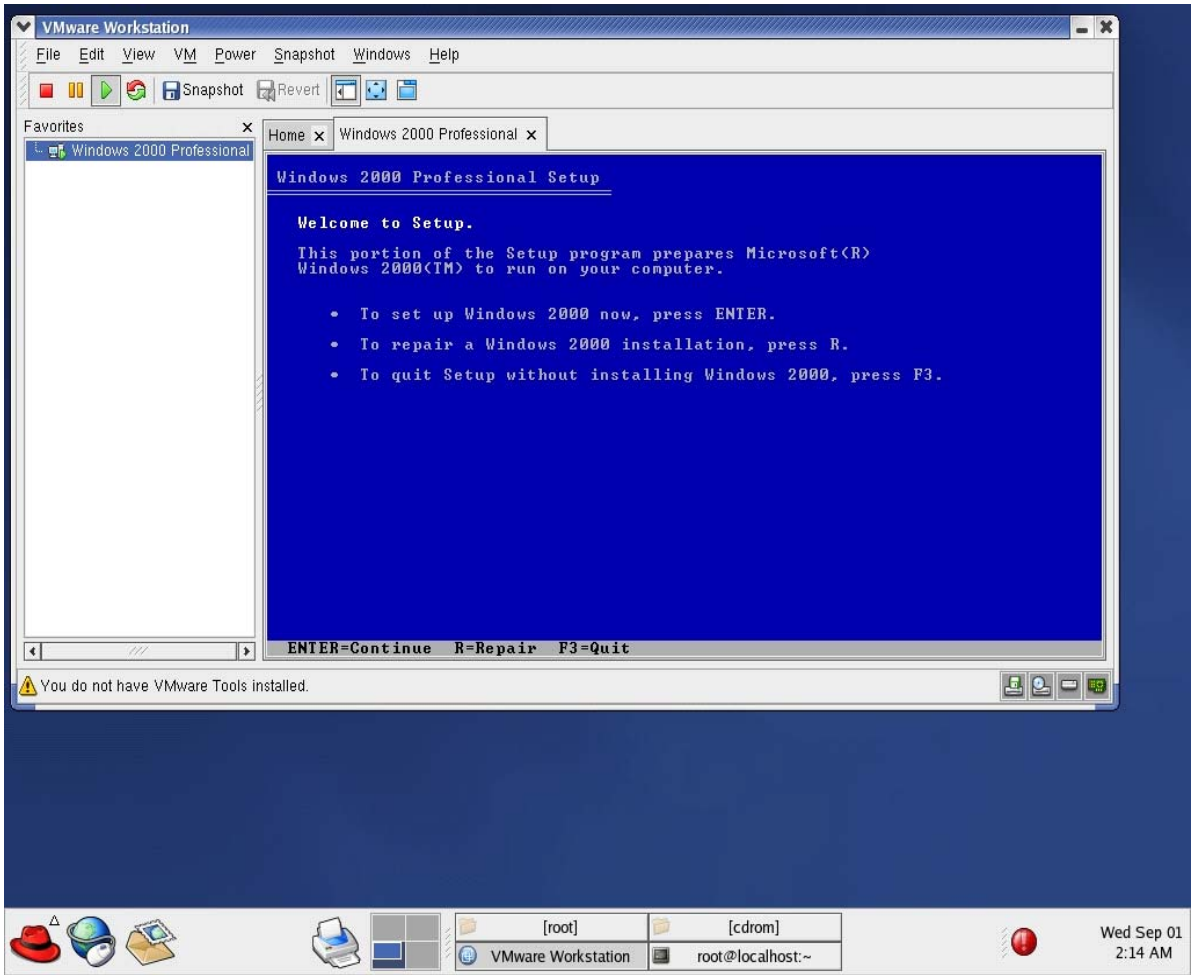


Figure 22. Starting a Virtual Machine for the First Time

3. Installing VMware Tools

The necessary installation instructions for this section were taken from the "VMware User's Manual." [53] VMware Tools must be installed when the guest operating system is up and running. VMware Tools can be installed by going to **VM>Install VMware Tools** on Windows-based guest operating systems. VMware starts the installation on the guest operating system (Figure 23).

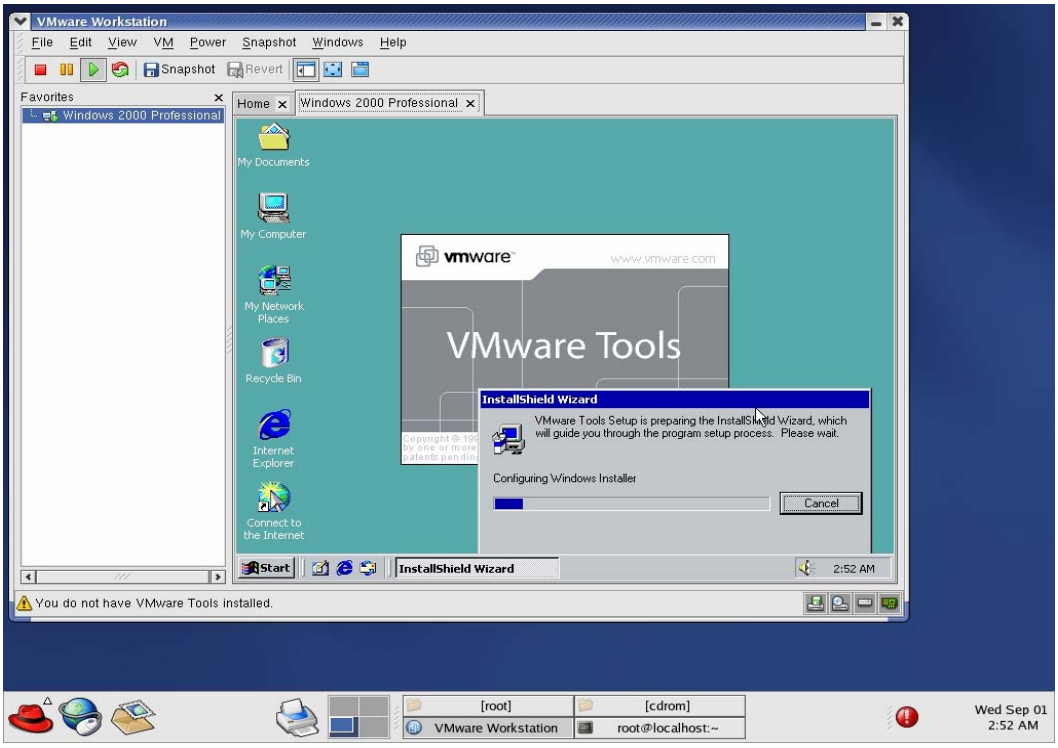


Figure 23. Installing VMware Tools on Windows-Based Systems

On Linux-based guest operating systems, it is not necessary to switch to *text mode* after using the **VM>Install VMware Tools** since VMware Tools cannot be installed in an X window session. Hitting **Ctrl + Alt + F1** or another function key at the same time starts the text mode session. Once in the text mode, the following commands must be entered as *root*:

[53] Ibid., pp. 81-94 (26 August 2004).

[root@localhost root]# mount /dev/cdrom /mnt (mounts the International Organization for Standardization (ISO) image of VMware Tools to */mnt* directory)

[root@localhost root]# cd /tmp (changes directory to */tmp*)

[root@localhost tmp]# tar zxf /mnt/vmware-linux-tools.tar.gz (extracts the contents of the compressed file)

[root@localhost tmp]# umount /mnt (unmounts the ISO image)

[root@localhost tmp]# cd vmware-tools-distrib (changes directory to the newly extracted */tmp/VMware-tools-distrib*)

[root@localhost vmware-tools-distrib]# ./vmware-install.pl (runs the configuration file)

Running the configuration file completes the installation. Afterwards, the X window (graphical environment) can be restarted. The following command runs the VMware Toolbox in the background.

[root@localhost root]# vmware-toolbox & (runs the VMware Tools background application)

4. Configuring the Virtual Machines to Run Automatically on Startup by Running a Script

To start the virtual machine, it is necessary to execute the "**vmware"** command. To use the script, first find the ".vmx" file in each virtual machine's directory on the host and use the "-x" suffix to run it in the following script called "startvm":

```
vmware -x /root/Windows\ XP\ Professional/Windows\ XP\
Professional.vmx &

sleep 10

vmware -x /root/Red\ Hat\ Linux/Red\ Hat\ Linux.vmx &

sleep 10

vmware -x /root/Red\ Hat\ Linux\ Second/Red\ Hat\ Linux\
Second.vmx &

sleep 10

vmware -x /root/Windows\ 2000\ Professional/Windows\ 2000\
Professional.vmx &
```

When the above script runs, the three virtual machines specified in the script are started one by one. The "sleep 10" command keeps a 10 second delay in between. The "&" suffix used after each *vmware* command in the script is very important. The authors did not use the script the first time it was written. The script would start the first virtual machine and stop there. Only after powering down the first virtual machine would the second line in the script execute. This was because the code would wait for the first line to be executed, and as long as the virtual machine was running, it would consider it as still executing the first line in the code. It basically works the same way as the commands are executed in the shells. For example, typing the "gedit" command in the shell will not allow a new command to be typed until the gedit window is closed. The "&" suffix makes the code work in the background, thus, enabling input for new commands.

For the next step, it was necessary to automate the execution of the script after start-up. This was done by using the GUI (Graphical User Interface) available on the GNOME desktop with the following path: Desktop>Start Here>Preferences>More Preferences>Sessions. Once the "Sessions" window was displayed, the "Startup Programs" tab was configured to reflect the appropriate path of the "startvm" script as seen in Figure 24.

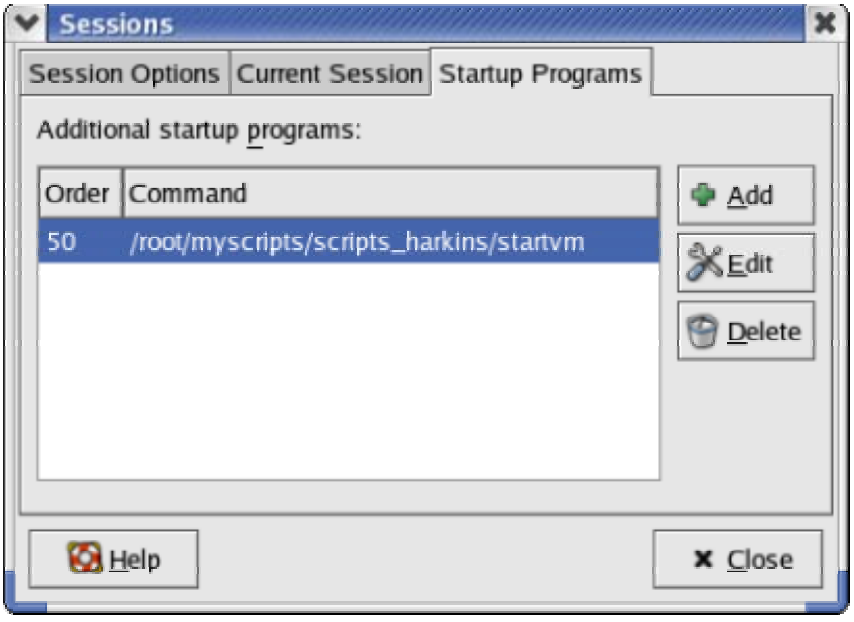


Figure 24. Adding the Script to Startup Programs

Once the starting of the scripts was automated, it was then necessary to automate the powering down of all the virtual machines with a script that could be run from shell, instead of having to shut down all the virtual machines manually. Three separate scripts were used for this purpose.

The "get_pid" script would get the code number of the processes running for each virtual machine, and extract them into a file called: "pidfile." The script was:

```
#!/usr/bin/perl

# Change the path as required

#cd /root/myscripts/scripts_harkins/;

$number = 4;
$path[0] = "/root/Red\ Hat\ Linux/vmware.log";
$path[1] = "/root/Red\ Hat\ Linux\ Second/vmware.log";
$path[2] = "/root/Windows\ 2000\ Professional/vmware.log";
$path[3] = "/root/Windows\ XP\ Professional/vmware.log";

open(OUT, ">pidfile") || die("could not open pidfile \n");

for ($i=0; $i<$number; $i++){

     open(IN, $path[$i]) || die("could not open \n");
     $first_line[0] = <IN>;
     $eq=-1;
     $v=-1;
     $small_v="v";
     $equal="=";
     $eq=index($first_line[0], $equal);
     $v=index($first_line[0], $small_v, $eq);
     $len=$v-$eq-2;
     $PID=substr($first_line[0], $eq+1, $len);

print OUT  $PID, "\n";
close(IN);
}
close(OUT);
```

The "kill" script would extract those process ids from the "pidfile," and kill those processes. The script was:

```
#!/usr/bin/perl
#Usage perl -w kill.pl
$no_of_vms=4;
$pid_file= "/root/myscripts/scripts_harkins/pidfile";

open(IN, "$pid_file") || die("could not open $pid_file: $!\n");

print "Taking Down VMs\n";

for ($i=0; $i<$no_of_vms; $i++) {
```

```
        $first_line[$i] = <IN>;
        $PID=$first_line[$i];
        print "kill 9 $PID\n";
        kill 9, $PID;
}


close(IN);
```

The last script, which was called "STOP," was to decrease the number of executed commands by combining the two commands executing the previous two scripts into one single script. Thus the only thing needed to shutdown the virtual machines on a host would be to execute the "./STOP" command from the shell. The script was:

```
#!/bin/sh

./get_pid.pl
./kill.pl
```

5. Configuring the Virtual Machines for Automatic Log on without Prompting a Username/Password

This is done with standard operating system configurations with no extra changes to the VMware settings. For Windows 2000 Professional, Start>Settings>Control Panel>Users and Passwords was used. It was necessary to:

- *Unclick the box:* "Users must enter a username and password to use this computer"
- *Click OK*

In the pop up window, it will ask for the username and password for the user that will be automatically logged on each time the machine reboots. That is all that is required!

For Windows XP Professional, it was not necessary to specify any password for any user during the installation. Thus, the system automatically logs on as "administrator." Finally, for Red Hat 9 Linux machines, System Settings>Login Screen was used (Figure 24). Next, click on the box "Login User Automatically on First boot up," and chose the user to be logged on every time the machine powers on.

Once the necessary information is entered, the system will automatically log on that user every time it starts. Therefore, it is possible to use the scripts to start the machines without having to enter the username and passwords each time one is powered on. Simply remember to take the snapshot after making the configuration.

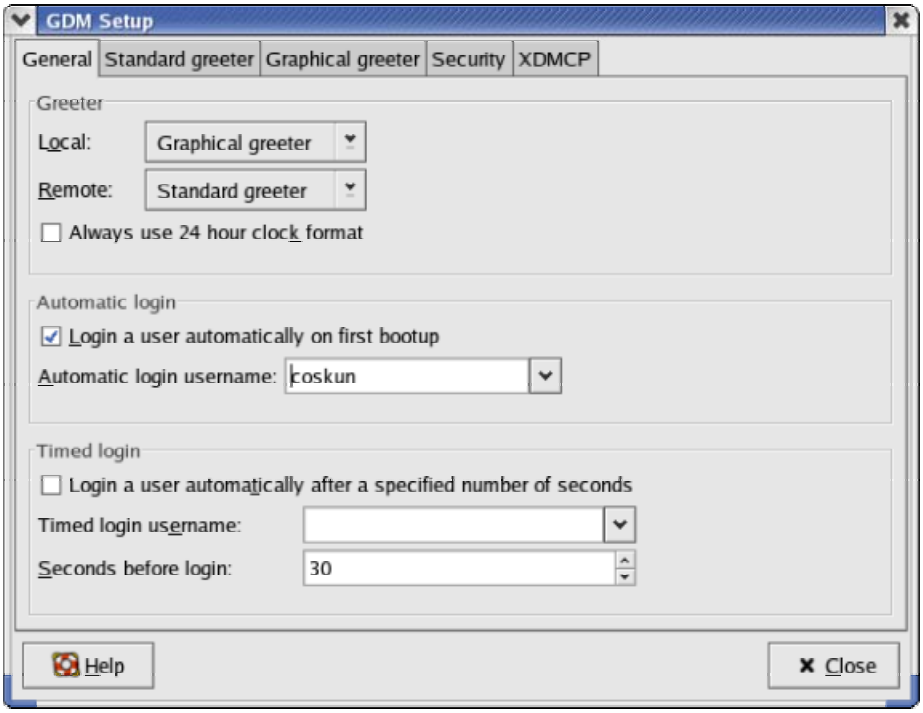


Figure 25. Login Screen Configuration on Red Hat Linux 9.0

6. Configuring the Virtual Machines for Persistent Mode

To do the following configurations, VMware documentation online was used.[54] The VMware program can be configured for persistent mode. For this purpose, it is necessary to take the snapshot of the computer in the stage to which it is desired to revert every time it is powered on. Before taking the snapshot, the system was configured in order to make the reboot process faster. The hardware components not needed were uninstalled from the VM>Settings window (Figure 25) by choosing the hardware component and clicking on the remove button.

54 "Preserving the State of a Virtual Machine," available online at: <http://www.VMware.com/support/ws4/doc/ws40_preserve.html#1018532> (31 August 2004).

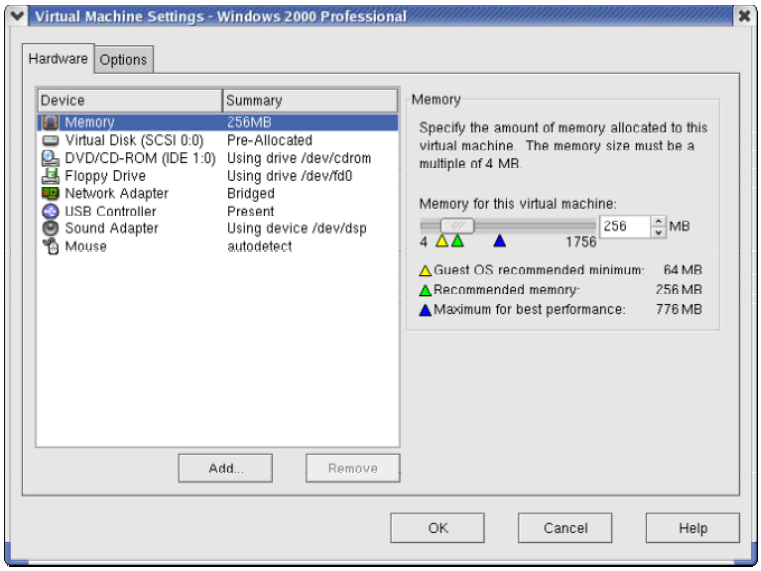


Figure 26. Virtual Machine Settings Screen

The “Sound Adapter,” and “Floppy Drive” were removed. Later the snapshot was taken by clicking on the “Snapshot” button (Figure 26).

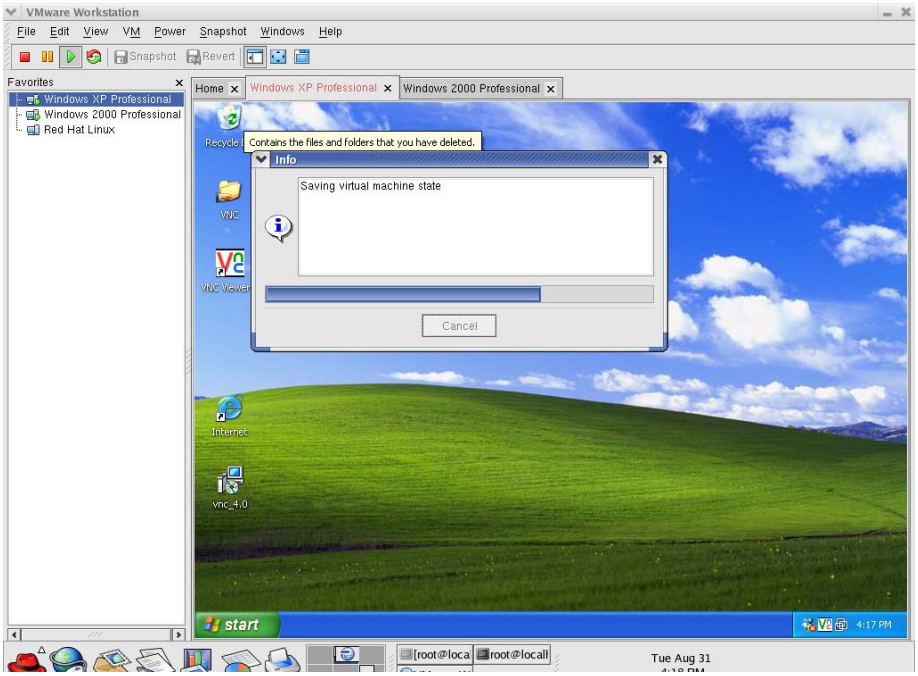


Figure 27. Taking Snapshots

Now the system will revert to this same stage regardless of a user’s changes. It is possible to configure the specific feature in such a way that it will not revert to the

snapshot instance in case users are allowed to make some changes. This is done in the VM>Settings>Options menu (Figure B-10). Here the "Snapshot" has four different options from which to choose, and choosing "Update the Snapshot," will update any changes made to this feature. However, since this was to be avoided, no changes were made to the default snapshot option, which is "Revert to the snapshot" for all settings.

Once the snapshot was taken as desired, no changes were to be made. The system can keep only one snapshot in memory, and it can be updated very easily by clicking on the snapshot button. It can be done while the machine is up or down in every stage. To prevent a user from accidentally changing the snapshot, it can be "lock" by choosing the "Lock the snapshot" box in the VM>Settings>Options menu (Figure 27). This was done for every hardware component in the VM>Settings menu.

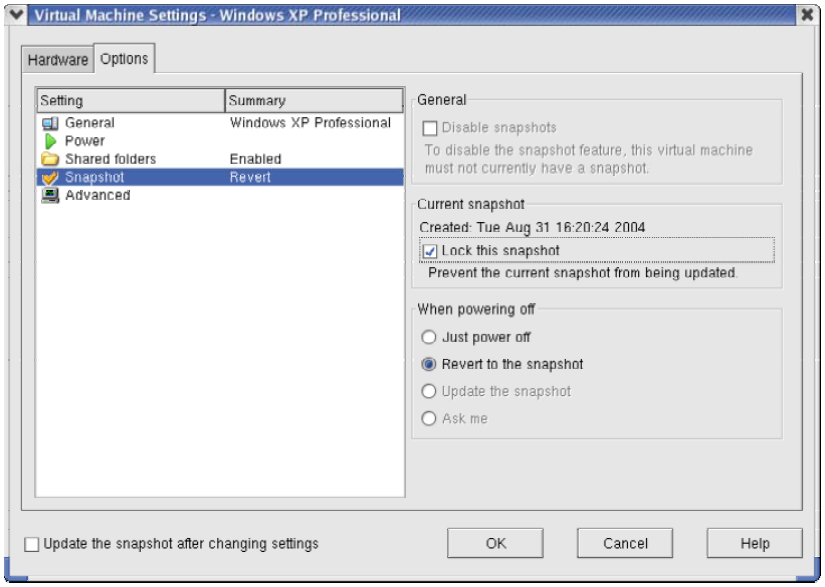


Figure 28. Snapshot Settings on VMware Workstation

It is important to note that in the persistent mode configuration, the system reverts to the snapshot in two ways. One is by clicking the "Revert" button in the VMware window. This will not be available to users who will be logged in from their browsers through VNC. The second option is to "Turn Off"(or for Linux machines, to "Shut Down") the machine. Once turned off, instead of powered down, the virtual machine automatically reverts to the snapshot and discards all the changes made by the user. However, this does not work for "Restart." If you restart the machine, it does not revert

back to the snapshot, but restarts to the last saved settings. Therefore, in this system, the users will be asked to turn off instead of restarting the virtual machines, so that they will not be affected by whatever changes a previous user may have made to the system.

7. Installing VNC (Virtual Network Computing) on Guest Machines

To install the VNC server on Windows-based guest operating systems, *vnc-4.0-x86_win32.exe*, was downloaded, which is an executable file for the latest version (VNC 4.0), from the website <http://www.realvnc.com/download.html>, copied to the guest operating systems, and then run. On Windows based machines, the VNC server starts up by default every time the computer is booted.

For Linux-based systems, there are two options to install the VNC server and client. First, the compressed file *vnc-4.0-x86_linux.tar.gz* can be downloaded from the same website noted above, and installed. Second, the previous versions of the VNC server and client are already among the RPM packages on installation CDs for Linux-based systems. They can either be selected during the initial installation of the operating system or can be installed afterwards by going to **System Settings>Add/Remove Applications**. The server part**, vnc-server,** is located in ***Servers/Network Servers,*** and the client part, **vnc,** is located in ***System/System Tools.*** The decision was to add it to the installation packages. To run the VNC server on Linux machines, the following commands must be entered:

[root@localhost root]# vncpasswd (sets a password of user's choice for VNC server)

[root@localhost root]# vncserver & (starts the server running on the background)

To avoid having to run **"vncserver"** command manually everytime the host machine was booted; the execution of this command was automated. To do this, the file "rc.local" in the "/etc" directory was edited. At the end of this file, the following line was added: "/usr/bin/vncserver." After this, every time the virtual machine was restarted, the vncserver would be run automatically by the system. One important point to remember is to take the snapshot of the system after this configuration, so that when the system reverts to the default mode it shall not lose this property.

LIST OF REFERENCES

"An Invitation to Servlets," <http://www.novocode.com/doc/servlet-essentials/chapter1.html#ch_1_1> (27 August 2004).

"Apache Tomcat mod_jk Connector 1.2.5 Released," <http://www.serverwatch.com/news/article.php/3091461> (26 August 2004).

De Horta, Ain Y., Kneale, Bruce and Box, Hona, "Development of a Virtual Overlay for Velnet (Virtual Environment for Learning Networking)," School of Computing and Information Technology, University of West Sydney, Australia, December 7-10, 2003, <http://proceedings.informingscience.org/IS2003Proceedings/docs/090Kneal.pdf> (16 July 2004).

"Dell™ PowerEdge™ 1650 Systems," <http://support.dell.com/support/edocs/systems/pe1650/en/> (20 August 2004).

"Dell™ PowerEdge™ 1650 Systems Installation and Troubleshooting Guide," <http://docs.us.dell.com/support/edocs/systems/pe1650/en/it/index.htm> (20 August 2004)

"HOWTO: Installing Web Services with Linux/Tomcat/Apache/Struts/Postgresql/OpenSSL/JDBC/JNDI," <http://www.linuxjava.net/howto/webapp/> (20 June 2004).

<http://java.sun.com/j2se/1.4.2/download.html> (21 June 2004)

<http://httpd.apache.org/download.cgi> (21 June 2004)

<http://jakarta.apache.org/site/binindex.cgi> (21 June 2004).

<http://jakarta.apache.org/site/sourceindex.cgi> (21 June 2004).

<http://www.openssl.org/source/> (21 June 2004).

<http://httpd.apache.org/docs/misc/what> (26 August 2004).

"Java Servlet Technology," <http://java.sun.com/products/servlet/> (27 August 2004).

"Java VNC Viewer," <http://www.realvnc.com/javavncviewer.html> (27 August 2004).

"JSP Quick-Start Guide for Linux," <http://www.sitepoint.com/article/jsp-quick-start-guide-linux/4> (26 August 2004).

Kneale, Bruce, De Horta, Ain Y. and Box, Ilona, (2004) "Velnet: Virtual Environment for Learning Networking," (This paper appeared at the sixth Australian Computing education Conference (ACE2004), Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, vol. 30. Editors, Raymond Lister and Alison Young), <http://portal.acm.org/citation.cfm?id=979990&dl=ACM&coll=portal> (16 July 2004).

Ogle, Todd, "The Effects of Virtual Environments on Recall in Participants of Differing Levels of Field Dependence," <http://scholar.lib.vt.edu/theses/available/etd-04252002-112047/unrestricted/etd.pdf> (April 11, 2002), pp. 16-19.

"OpenSSL," <http://www.openssl.org/docs/apps/openssl.html> (26 August 2004).

"Preserving the State of a Virtual Machine," available online at: <http://www.VMware.com/support/ws4/doc/ws40_preserve.html#1018532> (31 August 2004).

Russell Elliott, "Creating a Home Test Lab, Cases Study in Information Security," (SANS Institute, February 19, 2003), <http://www.giac.org/practical/GSEC/Russell_Elliott_GSEC.pdf > (11 July 2004)

"Server Configuration Reference," <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/config/jk.html> (26 August 2004).

Shachor, Gal, "Tomcat Workers.Properties," <http://jakarta.apache.org/tomcat/tomcat-3.3-doc/Tomcat-Workers-HowTo.html> (26 August 2004).

"Technical Specifications," <http://docs.us.dell.com/support/edocs/systems/pe1650/en/ug/8g540aa0.htm#1039239> (20 August 2004).

"Tomcat FAQ Home Page," <http://www.jguru.com/faq/Tomcat> (26 August 2004).

Turner, John, "Apache 2.0.47/Tomcat 4.1.27/mod_jk for Red Hat 9.0," <http://johnturner.com/howto/apache2-tomcat4127-jk-rh9-howto.html> (03 September 2004).

"VNC Server 4.0 for Windows," <http://www.realvnc.com/v4/winvnc.html#4> (20 August 204).

"Workstation 4 User's Manual," <http://VMware-svca.www.conxion.com/software/ws45_manual.pdf>(26 August 2004).

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Richard M. Harkins
   Physics Department
   Naval Postgraduate School
   Monterey, California

4. Wen Su
   Computer Science Department
   Naval Postgraduate School
   Monterey, California

5. Kara Harp Okulu Kutuphanesi
   Bakanliklar
   Ankara, Turkey

6. Kara Kuvvetleri Komutanligi Kutuphanesi
   Bakanliklar
   Ankara, Turkey

7. Coskun KARGIN
   Baruthane Cad. No: 136/9 Ferikoy
   Istanbul, Turkey

8. Turgut AKGUL
   Erciyes Evler Mah. Billur Cad. Kardelen Apt. No: 188/20 Kocasinan
   Kayseri, Turkey