Design and Implementation of an Online Secure Linux Laboratory for Networking Courses

Kai Li
East Carolina University
Greenville, NC 27858

Jing Zhao
Virginia Tech
Blacksburg, VA 24060

Tang Xin
East Carolina University
Greenville, NC 27858

Yi Li
Northwestern Polytechnical University
Xi'an, Shaanxi, P.R.China

1. Introduction

The widespread diffusion of Internet has led to the burgeoning growth of distance education, which has provided enormous opportunities for people who wish to further their education. Distance education is a planned teaching/learning experience that uses a wide spectrum of technologies to reach students at a distance. It is designed to encourage students to remotely interact with educators and other students. Critical elements of distance education include access, technical design, virtual environment, interactions within the community, etc.

Distance education through online courses is becoming a widespread standard offered by most universities at both undergraduate and graduate level. Online courses are typically delivered to students via some sort of asynchronous tools, such as bulletin boards, video streaming, email, and web based course assignment submission systems, etc. Some courses may contain a synchronous component with chat facility. Blackboard and WebCT are popular examples. Their main functionalities may include synchronous and asynchronous communication, electronic whiteboards, email, web pages, and calendars. However, these tools show inability to deliver hands-on learning opportunities when used in courses with lab practice.

As an integral part of online networking courses, a remote network lab benefits students in multiple ways. First of all, it strengthens students' understanding of the network concepts in a practical way. Theories of network protocols and architectures are abstract and monotonous for those students without any prior network experience. Active involvement in the laboratory can ignite their interest and give them a "big picture" of the functionality of a network and its

applications. Furthermore, it provides students hands-on opportunities to set up small-scale commercial networks operated as in a real world, which not only permits students to grasp the strategy to efficiently manage a network as administrators but also improve their network problem solving skills. Finally, the remote accessibility of the lab enables students to practice without space and time constraints on a 24/7 basis.

To better serve the needs of online courses, our laboratory was created with a distinct security checking mechanism. The establishment of a secure on-line computer lab for remote access is challenging. There are various ways to perform a secure remote-user-to-network, but the most popular and the most progressive network firewall is to construct a secure Virtual Private Network (VPN) [1][2] over the public Internet rather than traditional private networks. VPN technology has been around for several decades however only during the past decade that it has become more feasible for a wide range of organizations and corporations. Essentially, a VPN is a private data network implemented over some type of shared infrastructure to carry its traffic. Data must be encapsulated within an IP packet before it can be sent through VPN. VPN can be implemented through either hardware or software. For example, Cisco VPN 3000 concentrator provides remote access platforms for data encryption and authentication. However such devices usually cost several thousands dollars and therefore appear to be an expensive solution for those institutions with limited budgets. Software solution is an economic and feasible alternative to hardware one. In our laboratory, we implement a software solution by exploiting Linux built-in security features to develop a script program acting as a firewall on the gateway machine to eliminate the cost for any commercial hardware or software firewall.

This online network lab was originally developed to assist in a senior undergraduate course which mainly teaches network system and applications. It carries three credit hours and is scheduled over a 12-week academic term. Lectures, quizzes, and lab experiments of the course are to be remotely conducted through an internal LAN. All the laboratory work is expected to be done on a single PC being capable of running both Linux and Windows. Throughout the semester, this lab has very well served the purpose of the class. Its proven scalability is also of great values in extending its usage to other online IT courses.

This paper lays out the major challenges, critical issues and problem solving strategies in building a secure on-line lab. We first present the goals and objectives, and also include a survey of related work. Then we describe the components of the online laboratory, -both hardware and software-and the overall system architecture. Next, deployed virtual system technologies utilized in the application are presented, followed by a summary of the main characteristics and mechanisms of remote technologies applied in the lab. A detailed description of a simple and cost-free firewall mechanism to secure remote lab access is also presented. In the end is the discussion of conclusions and future plans.

2. Goals and Related Work

A variety of research attempts to impart practical skills to students via the state of the art computer technologies. In Vernon College, Computer networking courses are taught via virtual machine technology [3]. However, they do not put too much effort on remote technologies. Students are still taught through traditional face-to-face instruction. Other previous work for

network education based on virtual networking lab components can be found in [4][5]. In [6], a remote virtual lab sharing a few similar features with us for networking courses is created using the combination of virtual machine and remote technologies. The security mechanisms for the remote virtual lab, however, are not spelled out in details.

In this paper, our Linux Laboratory is designed with the following objectives:

1. To provide a practical treatment of the fundamental principles of networking protocols, including DHCP, DNS, Network Security, and etc.
2. To consolidate students' understanding of basic networking concepts and improve their hands-on skill.
3. To allow students to remotely perform labs and assignments at anytime from anywhere.
4. To disable external unauthorized access to the lab.

With these objectives in mind, our remote lab is built with several departures from previous work. We deploy virtual system technologies allowing students to perform the configuration of multiple operating systems such as Red Hat Linux, Fedora, and Windows XP on a single PC with a private IP address assigned by the instructor at the beginning of the semester. Secondly, encrypted virtual network computing technology provides a reliable visualization and control of a remote computing desktop environment through any other computer in the Internet with security guarantee. Most importantly, a Linux build-in firewall is applied to the gateway machine. The purpose of the firewall is not only to prevent unauthorized external traffic but also to provide a flexible mechanism to discipline the internal network traffic among lab computers in different practice sections.

3. Descriptions of the Linux Laboratory

The Linux laboratory is a multipurpose laboratory used for teaching practical skills in several courses required for the BS degree in Information and Computer Technology at East Carolina University. The laboratory constitutes the basis for two mandatory computer networking and operating system courses in the third and fourth degree years. Thus, the laboratory is used by approximately 15 students for an average of 30 hours per student every academic semester. This requires the teachers to build a scalable and efficient laboratory to cater to different course flavors. Moreover, the network settings of the laboratory must be flexible enough to facilitate the configuration of the devices and their logical connections.

In the following section, we describe the physical connectivity among different hardware components in the laboratory, the logical functions of each component, the schemes for assignment of private IP address to each machine, and etc.

3.1 Laboratory Setup

The laboratory is structured in two local area networks (LANs) with eight Dell personal computers (PCs) in one and seven in another, as shown in Figure 1. Each PC is featured with 2.8 GHz CPU, 512 MB memory, 40GB hard disk, one internal Ethernet NIC, and multiple USB ports for external devices. Each LAN is constructed by means of a D-Link switch with twenty-

four auto-configured ports. The switches are further linked to different network interface cards installed on a gateway workstation that operates as a firewall router for the laboratory network as

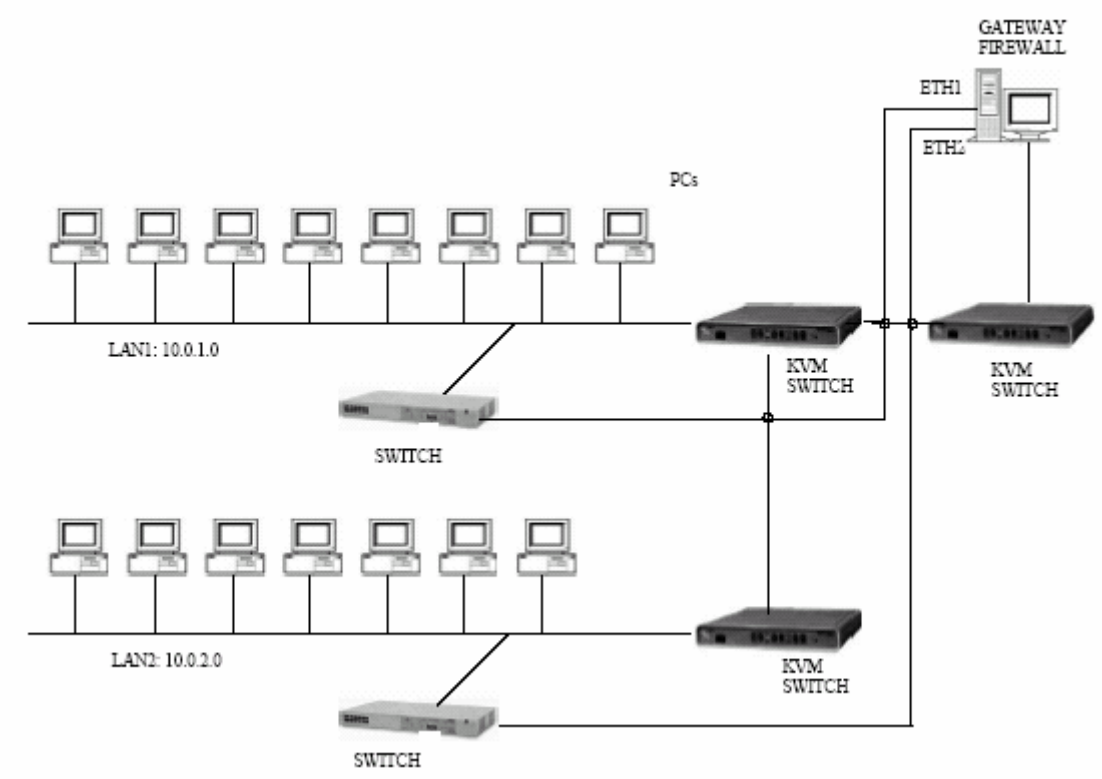

Figure 1. Laboratory network topology

well as connects the internal networks to the public Internet. There exists only one Internet connection for the laboratory.

There are three daisy-chained Belkin KVM (Keyboard, Video, and Mouse) switches used in the laboratory with 8 ports each. The KVM switch is designed to control multiple computers using just one keyboard, video display, and mouse. With KVM Daisy Chain function, up to eight units (banks) can be daisy-chained together to control up to 120 PCs with a single console. Switching among computers is achieved simply by pushing a button on the front panel. The purpose of using KVM switch is to provide more tables and rack space in addition to save the cost of multiple keyboards, mice, and monitors.

The internal network is divided into two LANs with private 10.0.*.0 network address. All internal machines are accessed through a firewall gateway machine mentioned above. This gateway machine has three activated network interfaces eth0, eth1, and eth2. Eth0 is directly connected to the public Internet. Eth1 and Eth2 are the gateway interface for each LAN respectively. There is one internal machine in each LAN configured as a DHCP server which dynamically assigns a fixed IP address to the internal machines according to their MAC address and these DHCP servers cannot be accessed by the students. Each internal machine is assigned a host name according to its connection to different KVM switch ports.

Students are divided into two groups according to the number of LANs. Each student is assigned a private IP address with administrator privileges for that machine at the beginning of the semester. The laboratory permits students to design multiple configurations to test different network protocols under Linux and Windows. The courses include a wide variety of practices based on the following technologies and concepts: Apache web server, FTP, email, DHCP, DNS, routing protocols, network security, network management, active directory, and etc. To implement such practices, one must configure the assigned laboratory PC either individually or collaboratively throughout the semester. Due to the security policy applied to the gateway machine, internal network traffic among students' machines is only allowed for selected activities.

3.2 Installation of OS Images

It is imperative to have multiple operating system platforms running on PCs to allow students to perform various network and system configurations. As one of software packages freely available, Linux lends itself to applications in a network scenario and is also an ideal option for creating a multiple-boot environment. However, a traditional CD installation of Linux on a bunch of machines with pre-installed Windows OS is a tedious process. To automate this task, a network file system (NFS) with a pre-created kickstart file is leveraged for installation.

The principle of NFS is to create a Linux export tree (All Linux files are copied under the tree) shared by the internal networks so that all the other machines can download the files from this tree to perform their own installation instead of switching CDs for each machine. Most often, one of internal PCs with Linux installed in advance plays the roles of both NFS server and DHCP server. The purpose of the DHCP server is to enable the communication between other pre-installed machines and the NFS server. After a NFS server being selected, a kickstart file is created to contain the answers to all the configuration questions that would normally be asked during a typical Linux installation. This file is kept on the server and can be read by the individual computer during the installation. Thus the whole process for installing Linux on multiple machines can be left unattended. Figure 2 shows the kickstart file for our installation.

However this dual boot mechanism only runs one of operating systems at once, and changing to another one means shutting down the currently active one. In another word, the communication established between students and the terminal will be disconnected. To make our lab more flexible, a virtual machine technology is also adopted as a complement.

4. VMWare – A Virtual Machine Technology

A virtual machine technology provides a software environment that encapsulates multiple high performance operating systems working simultaneously inside a main operating system running in a single PC. Each operating system functions as an independent virtual machine with its own hardware profile as if it is an independent system connecting to the same network with other systems. These virtual systems are actually stored as a single file under the main operating system and can be moved from one PC to another without worry of hardware compatibility. The virtual systems share both hardware resource and software application for a given platform, such

as running a copy of Linux on a Windows XP PC. In another word, inside each virtual system, one can access the local mouse, keyboard, floppy drive, etc. The memory and hard disk of the virtual system are shared with those of the main operating system.

```
#Generated by Kickstart Configurator

#System  language
lang en_US.UTF-8
#Language modules to install
langsupport  --default=en_US.UTF-8
#System keyboard
keyboard us
#System mouse
mouse genericps/2 --device psaux --emulthree
#Sytem timezone
timezone America/New_York
#Root password
rootpw --iscrypted $1$m2raeAEv$6Gt8BnmnnkbKY69xOwogG1
#Reboot after installation
reboot
#Install Red Hat Linux instead of upgrade
install
#Use NFS installation Media
nfs --server=10.0.2.1 --dir=/var/ftp/pub/
#System bootloader configuration
bootloader --location=partition
#partition information
clearpart --linux --drives=hda
part /boot --fstype ext3 --size=100 --ondisk=hda
part /share --fstype vfat --size=1098
part swap --size=512 --grow --maxsize=1024 --ondisk=hda
part / --fstype ext3 --size=100 --grow
#System authorization infomation
auth  --useshadow  --enablemd5
#Network information
network --bootproto=dhcp --device=eth0
#Firewall configuration
firewall  --medium --dhcp --port=smtp:tcp --port=http:tcp
--port=ftp:tcp --port=ssh:tcp --port=telnet:tcp
#XWindows configuration information
xconfig --depth=8 --resolution=640x480 --
defaultdesktop=gnome
#Package install information
%packages
@ Everything

%post
```

Figure 2. Example of a kickstart file

VMware is a software package designed to create a complete virtual machine, and it performs very good hardware emulation. All devices are accessed through the underlying host operating system (e.g. windows), and the file system may be a virtual drive that is contained in a file. It may directly access one or more standard File Allocation Table (FAT) 16 or FAT 32 partitions.

However, all access to the Linux file systems is done through the samba open-source file and print server software, which supports Windows clients. Figure 3 shows three virtual systems Red Hat, Fedora Core, and Windows XP running under the Windows XP.

The benefits of virtual machine technology are two folds. The instructors can use it to demonstrate the underlying mechanisms of different operating systems without physically install them on separate machines. On the other hand, it helps students to understand network concepts from different perspectives by applying them on the different systems. Furthermore, it allows students to create their own network by using these virtual systems as if they were configuring physically connecting multiple computers. For example, one system can be configured as a DHCP server, and other systems are specified as DHCP clients. By this way the necessity for collaboration among students is maintained as minimum as possible, which further reduces the risk of the students compromising the computer.

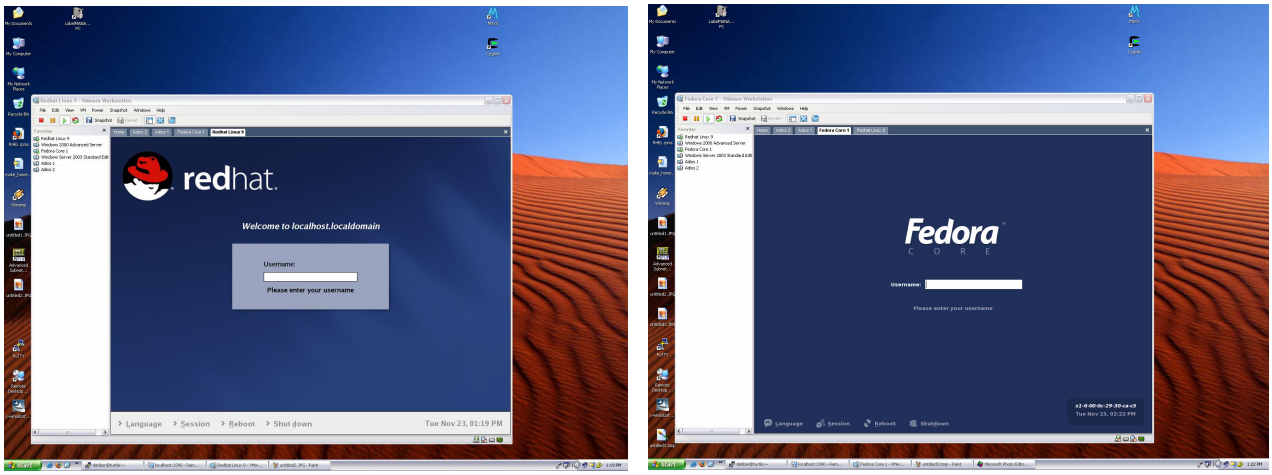

Figure 3. Examples of multiple virtual machines running on Windows XP

Figure 4. A Linux desktop from a Window viewer

5. Tunneling VNC through SSH

To allow students to perform experiments without time and space constraints, it is imperative to provide students with remote accessibility to their virtual systems. Utilizing a VNC or remote desktop technology can achieve this goal.

VNC stands for Virtual Network Computing [7]. It is a remote display system which allows people to view and access an entire desktop environment from any Internet-connected machine and from any computing infrastructure available. The technology underlying the VNC system is based on the client-server paradigm. A VNC server is responsible to encapsulate the system to be displayed and transfer it to the remote VNC client. The endpoint with which a VNC server interacts is called a VNC client. A more descriptive term in day-to-day use rather than client can be referred as viewer. A viewer then displays the received graphical user interface (GUI) on its local screen. The keyboard and mouse of the local machine can be used for this GUI as if they were directly connected to the remote server. The delay of sending keystrokes and mouse movements back to the server is totally depended on the transmission bandwidth of communication link between the server and client.

There are several free commercial VNC software packages including both server and viewer, such as realvnc for Windows XP, X-based server and viewer for Linux. For our courses, students are only required to download a realvnc viewer. We implement VNC server in the laboratory so that students are able to remotely access multiple virtual machines from any computer with a high-speed Internet connection. Figure 4 depicts a Linux desktop being accessed from a window viewer.

Although VNC technology allows students to remotely access the laboratory, the connection information passing through the public Internet will cause critical security issues. VNC uses a random challenge-response mechanism to provide the basic authentication for the user to connect to a VNC server. This is relatively secure. However, once the connection is established, traffic between the viewer and the server is unencrypted, and can be snooped by the hacker to compromise the remote system. To avoid malicious attacks from the Internet, we tunnel the VNC protocol through a secure communication channel established by the Secure Shell (SSH) [8].

SSH is widely used as secure remote terminal software. The SSH can make students login to a remote computer over insecure networks, execute command, and transfer files between a remote computer and a local computer. The SSH software is composed of the SSH server and the SSH client. As one of common SSH clients, Putty can be freely downloaded by students. After students login to the laboratory through SSH, they can use their own pre-assigned private IP address to build a secureVNC connection, as shown in Figure 5.

6. Anatomy of Firewall

A salient feature that makes the laboratory distinct is applying the Linux built-in security mechanisms to the gateway machine to tight the network security instead of purchasing expensive commercial firewall hardware or software. Advantages in using Linux as a firewall

platform include uniform administration, robust kernel-based handling, a wide variety of technical sources, performance and cost efficiency.

Linux firewalls for network security center around packet filtering technologies. A typical packet-filtering firewall checks network traffic (protocol, ip address, port number, etc.) and performs one of three actions: accept, reject, and drop. These actions are performed according to the security policies applied to the firewall. A firewall protects an internal network by accepting only authorized packets. On the contrary unauthorized packets can be either dropped or rejected by a firewall. There is a slight difference between the actions of reject and drop. When a firewall rejects a packet, it sends an ICMP error message to the sender. In contrast, the sender does not get any response once the packet is dropped. The efficiency of Packet filtering lies in the fact that it is a non-intrusive way of protecting the internal network.

Legacy supports for packet filtering in Linux include *ipfw* and *ipchains*. These packet-filtering facilities were used in kernels since the 1.1 kernel series. *Ipchains* is introduced in kernel version 2.2.x to address the *ipfw* limitations such as inability to handle IP configuration. As of kernel 2.4 onward, the functionality provided by *ipchains* has been replaced by netfilter and *iptables*. Netfilter is the name of new packet handling code in Linux kernel 2.4 and more details can be found in [9]. *Iptables* is an evolution of the *ipchains* facility with powerful inspection capabilities and quite flexible logging capabilities.

To implement the packet filtering function of *iptables*, the kernel starts with three build-in chains, INPUT, OUTPUT, and FORWARD. A chain is a set of rules that determine the actions (mentioned above) performed on the packets. When none of the rules match the packet, the kernel consults the chain default policy. The main purpose of INPUT chain is to filter all incoming traffic destined for the local host. All incoming packets destined for this host will pass through this chain, no matter what network interface or in which direction they came from. Meanwhile, the OUTPUT chain is where we filter the packets going out from the local host. Since FORWARD chain only routes the incoming packets to the different networks which pass by OUTPUT and INPUT chain, there is no need to setup FORWARD chain inside our firewall.

Figure 6 depicts the principles of the firewall applied in the laboratory. There are two kinds of network flow: encrypted and unencrypted. The traffic between Internet and our gateway machine are strong encrypted using SSH. The traffic between the firewall and the internal network is unencrypted. So the design purpose of the firewall is to filter unwanted incoming traffic and block the redundant traffic among internal PCs. The INPUT chain consists of INPUT1 and INPUT2. INPUT1 only accepts SSH, SFTP, and VNC packets and INPUT2 accepts SFTP and VNC packets. OUTPUT1 only allows SFTP, VNC packets to pass through while OUTPUT2 only sends out SSH, SFTP, and VNC packets. The only traffic allowed among internal PCs is ICMP for troubleshooting purpose. Other internal traffic like DHCP, DNS will be enabled only for selected course experiments.

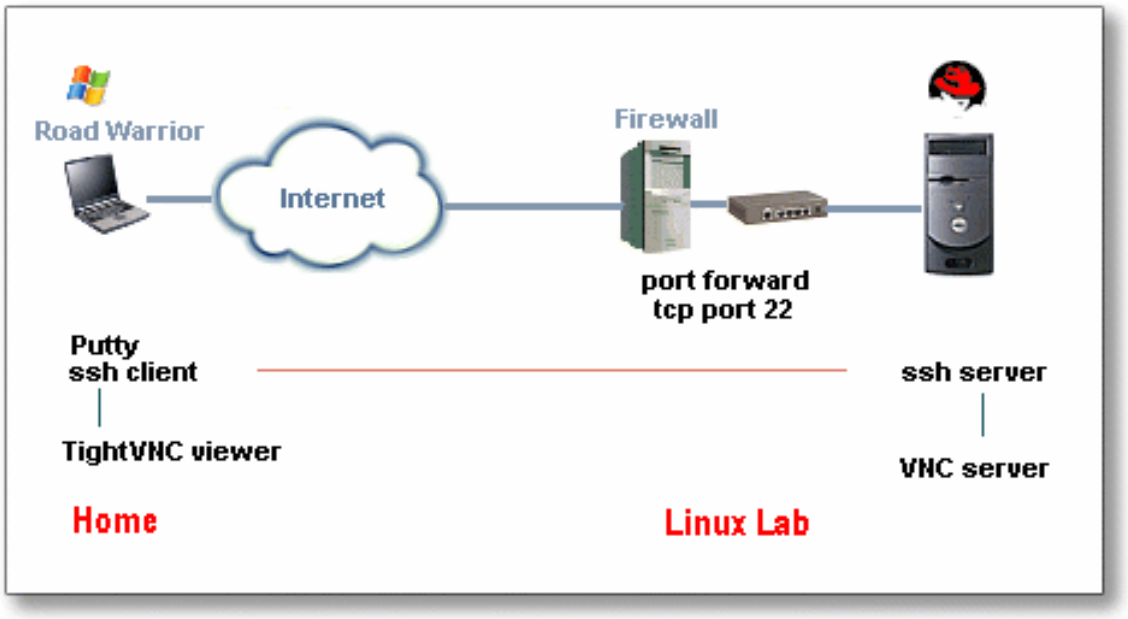

Figure 5. Tunneling VNC through SSH

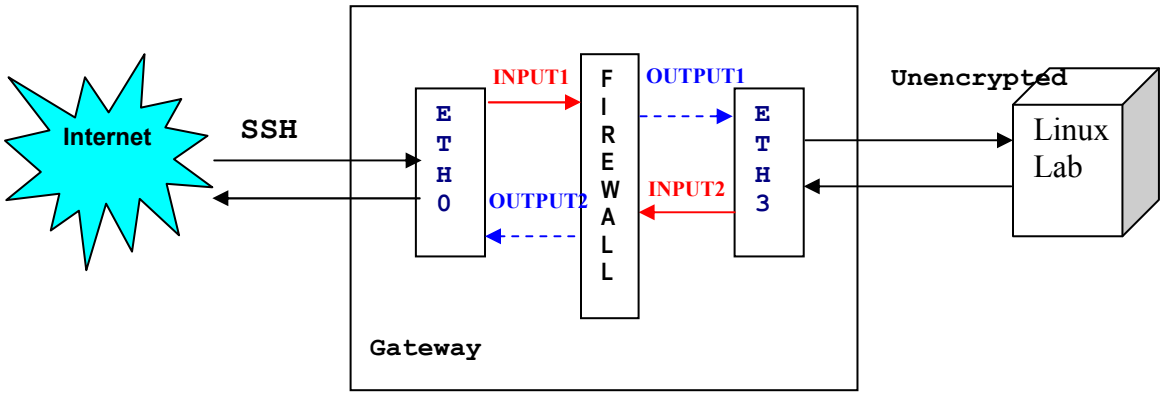

Figure 6. The firewall structure of the laboratory

7. Assessment of the Education Impact of the Lab

As the Linux Operating System rapidly evolves to surpass all others as the competing OS to Windows, the strong demand for professionals with significant Linux exposure and experience is expected to create a major growth sector in the US job market over the next decade. Currently, ECU has no facility to provide students with education on Linux based network management and computing environment. This established online Linux Network Lab has provided the critically needed infrastructure on remote secure network and management to strengthen our DE education and fill the need-demand gap. Furthermore, the Lab has complemented the current Windows OS based network labs available at the Department of Industrial Technology and enhance the education capacity for both DE and on-campus students. Finally, the effects of the Linux networking courses with the online lab companion have brought significant benefits to the department. This can be quantified by number of DE students enrolled. For example, in

Fall 2004, there are only 8 DE students took the course, however in Fall 2005, the number of enrollment is expected to jump to more than 100 DE students[1].

8. Conclusion

A remote Linux laboratory used for practical activities in different computer networking and operating system courses has been designed and implemented. The combination of virtual machine technology with a remote solution provides a real-world environment for students to obtain hands-on skills. To address security issues for accessing the remote laboratory, a simple cost-effective firewall mechanism is designed to alleviate the risk of the malicious attacks from the Internet. As a companion for one of networking course taught in the fall semester of 2004, the laboratory has undergone extensive tests and demonstrated satisfactory robustness and reliability. The results of this work create a prototype for other institutions to adopt to develop their own online secure laboratories.

9. Acknowledgment

This project is funded through UNC OP e-learning initiative grant.

BIBLIOGRAPHY

1. Sixto, O., "Virtual Private Networks: Leveraging the Internet," IEEE Computer Magazine, pp.18-20, Nov. 1997.
2. Cohen, R., "On the Cost of Virtual Private Networks," IEEE/ACM Transactions on Networking, vol. 8(6), pp. 775-784, Dec. 2000.
3. Success Stories: Virtual PC for Student Training. Connectix Website: http://www.connectix.com/success/vpc5w_vernon.html
4. Liu, S., Marti., W., and Zhao, W., "Virtual Networking Lab(VNL): Its Concepts and Implementation," ASEE Annual Conf. Proc., Albuquerque, MN, June, 2001.
5. Fabrega, L., Massaguer, J., Jove, T., and Merida, D.A, "Virtual Network Laboratory for Learning IP Networking," in Proceedings of the 7th Annual Conference on Innovation and Technology in Computer Science Education, June 2002.
6. Stockman, M., "Creating Remotely Accessible Virtual Networks on a Single PC to Teach Computer Networking and Operating System," in Proceedings of ACM SIGITE, Utah, Oct. 2004.
7. T., Richardson et al., "Teleporting in an X Window System Environment," IEEE Personal Comm., No. 3, pp. 6-12, 1994.
8. T., Ylonen, "The SSH (Secure Shell) Remote Login Protocol," Internet Draft, Network Working Group, Nov. 1995.
9. http://netfilter.kemelnotes.org/unreliable-guides/packet-filtering-HOWTO/

BIOGRAPHY

KAI LI is an Assistant Professor in the Department of Industrial Technology at East Carolina University. He

[1] Data provided by the registration office at ECU.

received his M.S. degree in Computer Science and a Ph.D. in Electrical and Computer Engineering, both from the University of North Carolina at Charlotte.

JING ZHAO is a Research Associate at Virginia Tech. She received a MS in Biology from the Georgetown University and MS in Computer Science from Virginia Tech.

TANG XIN is an Assistant Professor in the Department of Industrial Technology at East Carolina University. He received his B. Sc. and M. Sc. from Nanjing University of Aeronautics and Astronautics in 1989 and 1992 respectively, and his Ph.D. in 2002 from New Jersey Institute of Technology, all in electrical engineering.

Yi Li is a System Manager in Software Institute at Northwestern Polytechnical University, China. She received a BS in Electrical Engineering from Xi'an Institute of Technology, China.