AC 2007-2262: USING VIRTUAL MACHINE TECHNOLOGY IN AN UNDERGRADUATE INTRUSION DETECTION LAB

Peng Li, East Carolina University

Peng Li is an Assistant Professor in the Department of Technology Systems at East Carolina University. His professional certifications include CISSP, LPIC and SCSECA. He received a Ph.D. in Electrical Engineering from University of Connecticut.

Philip Lunsford, East Carolina University

Phil Lunsford received a B.S. in Electrical Engineering and a M.S. in Electrical Engineering from Georgia Institute of Technology and a Ph.D. in Electrical Engineering from North Carolina State University. He is a registered professional engineer and is currently an Assistant Professor at East Carolina University. His research interests include system simulation, telemedicine applications, and information assurance.

Tijjani Mohammed, East Carolina University

TIJJANI MOHAMMED is an assistant professor in the Information and Computer Technology program, within the Department of Technology Systems at East Carolina University. Currently, Dr. Mohammed teaches both graduate and undergraduate courses addressing a range of issues in the planning, selection, deployment, and securing computer networks.

Lee Toderick, East Carolina University

Lee Toderick received a B.S. in Computer Science from East Carolina University and an MS in Computer Information Systems from Boston University. His professional certifications include CCNP/CCDP and RHCE. He currently serves as teaching instructor in the Department of Technology Systems at East Carolina University. Research interests include remote lab access for distance learning students, firewall implementation, and information security as it applies to computer networks.

Chengcheng Li, East Carolina University

Chengcheng Li is an assistant professor at the Department of Technology Systems of East Carolina University. He obtained his M.S. and Ph.D. in Computer Science from Texas Tech University and MBA degree from the University of Southern Europe. He is holding MCSE and CCNA certifications issued by Microsoft and Cisco. His research interests are in network security, traffic engineering, and image processing.

Using Virtual Machine Technology in an Undergraduate Intrusion Detection Lab

Abstract

Virtual machine (VM) technology was recently adopted in an undergraduate lab on Intrusion Detection Technologies. Each student was provided with a pre-built, but non-configured Fedora Core 5 Linux VM image that was used to complete hands-on labs using the virtual machine on her/his own computer. To prepare the lab environment, a virtual network was built with Windows, Linux, FreeBSD, and Solaris virtual machines to simulate network attacks. Network traces of attacks were generated inside the virtual network using Metasploit Framework and other penetration testing tools. Student exercises included installing and using host-based intrusion detection systems, network-based intrusion detection systems and network monitoring tools. Students used TCPdump, Ethereal, Snort, and Bro to analyze the trace files. Students also performed installation and detection of loadable-kernel-module rootkits inside the virtual machine. A "compromised" virtual machine could be deleted after the lab and a fresh virtual machine could be reopened from the pre-built image in no time. The virtual machine was easy to use and easier to maintain than a real computer.

Using VM technology, it was possible to build a very "real" network environment at a minimal cost. Hands-on exercises of concepts could be set up in the virtual machine. Students were offered various opportunities to test other platforms such as Solaris without acquiring real physical machines. Additionally, the lab was available to students around the clock.

The adoption of VM technology helped students understand basic concepts, increased their interests and improved their troubleshooting skills. In addition, VM technologies expanded the physical boundaries of the lab environment. Students were able to use their own personal computers at home to perform lab exercises that previously would have required multiple machines configured in a dedicated lab room. This flexibility allowed the students to work at their own pace, and extended the lab environment to distance education students.

Using VM technology, we were able to transfer a physical hands-on intrusion detection lab from a Windows-dominated environment to a diversified virtual environment in a very short period. We believe that virtual machine technology can be successfully used in other computer security and networking labs.

1. Introduction

2006 may have been the year of virtual machine (VM) technology. During the year, VMware Inc. released VMware Server as freeware[1]. To compete with free virtualization solutions offered by VMware and Xen, Microsoft announced that Virtual Server 2005 R2 was available as a free download[2]. Virtualization technology enables multiple virtual machines to run concurrently on a single physical computer, with each virtual machine running an isolated operating system[3]. A virtual network is constructed that permits mesh or point-to-point VM connectivity. The pre-

existing operating system, such as Windows XP, does not need to be replaced; instead, the operating system acts as a host to one or more guest operating systems.

Our course on Intrusion Detection Technologies is a three-credit course with two lecture, and two lab hours per week. The course is offered to both face-to-face and distance education (DE) students in College of Technology and Computer Science at East Carolina University. Prior to the Fall semester of 2006, the Intrusion Detection Lab was offered online via virtual private network (VPN) and Remote Desktop connections. This solution had certain limitations:

- Students had to share the limited lab resources and did not have enough time to work individually.
- Remote desktop could be slow and unreliable for students without stable high-speed Internet access. This was especially a problem for DE students.
- The Intrusion Detection Lab involved installation and use of many different applications on different platforms. The cost of equipment and maintenance was high.

The problems enumerated above, along with other issues called for alternative solutions. To that end, we decided to migrate to a lab solution using virtual machines. With the new approach, the lab environment would no longer be centralized; instead, it would be distributed to the students' own personal computers. The adopted virtual lab solution had several advantages including the following:

- Students could run the virtual machines and conduct experiments on their own computers instead of a centralized lab in a remote location.
- Students had control over the lab environment as well as the pace of the lab experiments.
- Virtual machine images could be reinstalled easily and quickly if the need arose.
- The cost of setting up the virtual lab was much lower than that of setting up and maintaining a dedicated physical lab.
- Migration from the old physical lab to VMware was not difficult, as a result, we had more time to add content to the virtual lab.

While virtualization technologies have been around for a while, it has not been feasible for academic institutions to implement them due to significant cost issues. The distributed virtual lab was not feasible even a few years ago because virtualization applications were costly. Even if they were available, the programs were resource-consuming and demanded high-end computers to run. Virtualization technologies have become more mature and stable over the past few years. Several virtualization applications such as VMware Server, VMware Player, Microsoft Virtual Server, and Xen have been released as either open source or free software. Personal computers are more and more powerful and affordable. Currently, our college requires every student to have a computer for coursework and strongly recommends a high-speed Internet access for distance education students.

Among the major free virtualization products reviewed[4], VMware was selected over Microsoft Virtual Server for two reasons:

1. Members of our faculty have been using VMware products for a while and have found them to be relatively easy to use and fairly reliable in our tests.
2. VMware supports more guest operating systems than Microsoft Virtual Server. For example, VMware Workstation 5.5 (license required) supports DOS, Windows, Linux, FreeBSD, Netware and Solaris; whereas Microsoft Virtual Server 2005 R2 only supports Windows and limited Linux distributions. It was important for us to be able to emulate an environment with diversified platforms.

Xen was not considered because it did not support Windows XP as the host operating system. Xen could be only hosted under Linux or NetBSD with a customized kernel. The Mac-based virtual machine solution based on Parallels was not considered because the department policy requires students to purchase PC based machines.

2. Laboratory Setup

2.1 Preparation of the Virtual Machine for Student Use

VMware Workstation 5.5 was used to build the base image of a Linux virtual machine for student use. Fedora Core 5 Linux was chosen as the operating system. The software packages installed on the virtual machine included KDE, GCC, wget, YUM software manager and other applications. Network security utilities such as TCPdump, Wireshark (Ethereal), Snort and Bro were not pre-installed. Installation of these tools from RPM or from source code was to be performed by the students as part of their lab exercises. We chose Linux over Windows as the virtual machine operating system because most network security tools were originally developed on UNIX/Linux; many of them (e.g. Bro) were not available under Windows.

The virtual machine was configured to use Network Address Translation (NAT) networking. In NAT networking configuration, the virtual machine works inside a private virtual network[5] and obtains a private IP address (e.g. 192.168.*.*) from the VMware virtual DHCP server. Virtual machines on the same virtual network communicate with each other internally like physical machines on the same LAN subnet. When external connectivity is needed, the virtual network can be configured to share the IP address of the host operating system. The virtual network is protected behind a NAT firewall.

The pre-built virtual machine was then compressed using 7-Zip[6] before distribution. The size of the compressed virtual machine was 419 MB. The image of the compressed virtual machine was made available on the class ftp server for download. For students without high-speed Internet access, the compressed VM was distributed on CD-ROMs. The uncompressed size of the Linux virtual machine was less than 3 GB. The dedicated memory requirement for the virtual machine was 128 MB.

The primary purpose of the VM lab was to help students understand the concepts and principles of intrusion detection, as well as the deployment and use of intrusion detection systems. The lab was not intended to be a Linux operating system (OS) lab or an ethical hacking lab. The emphasis was on detection of attacks. The students were not required to install the Linux OS or to perform complex network attacks. However, the students were expected to install, configure

and use intrusion detection systems and other network monitoring tools. For example, the students were required to practice the whole process of deploying and using the Snort intrusion detection system (IDS), including downloading the Snort source code, compiling and installing the program with gcc, configuring snort.conf, setting up rules, running Snort as packet logger, running Snort as network-based intrusion system and analyzing network traffic from trace files.

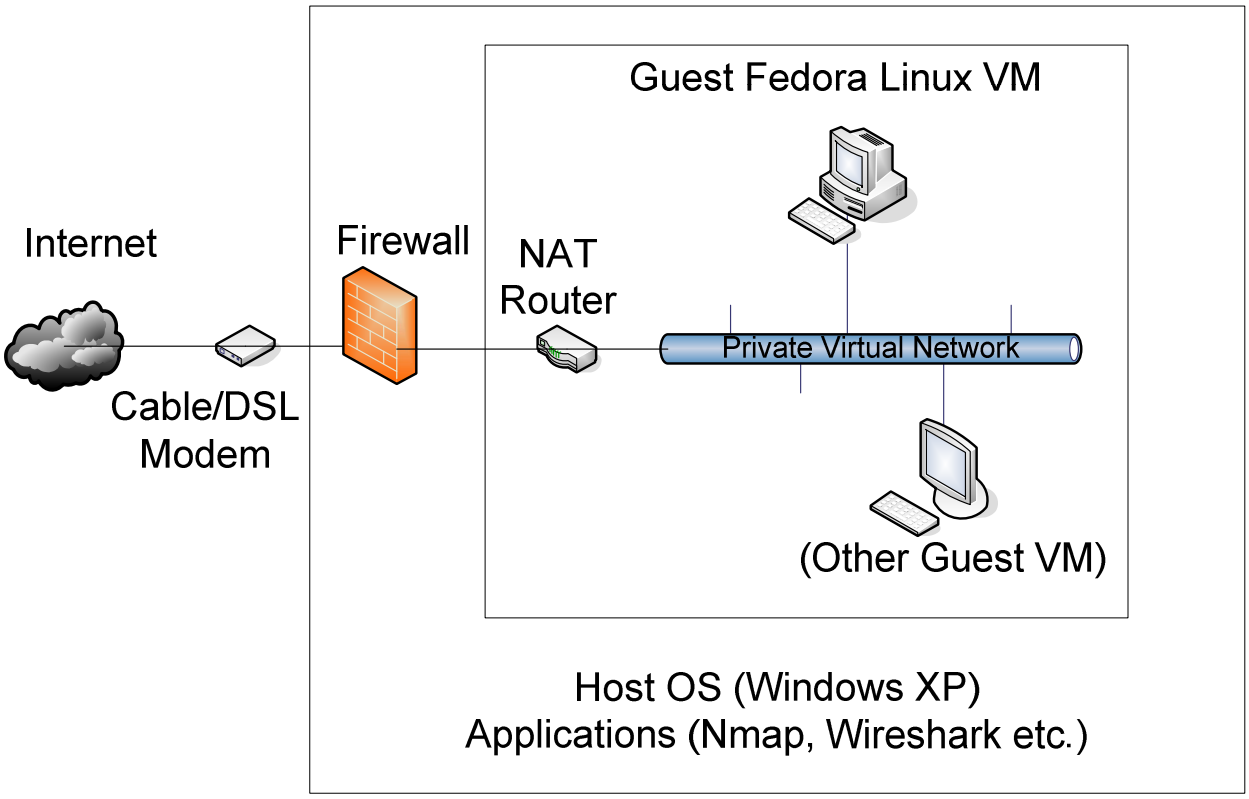

Figure 1. Typical virtual machine(s) inside a Windows XP host operating system.

A typical virtual machine setup[7] for the students' host OS is shown in Figure 1. A single Fedora Linux virtual machine ran on the student's personal computer. Multiple virtual machines were used by the instructor to create traces of network attacks for student analysis.

2.2 Preparation of Network Traces

Analyzing traffic is a fundamental component of network intrusion detection[8]. Network traffic can be recorded as trace files for later analysis. Most trace files provided to students for network traffic analyses were captured inside a virtual network created by VMware Workstation 5.5. The virtual network included a FreeBSD 5 virtual machine, a Solaris 10 virtual machine, a Windows 2000 virtual machine, a Red Hat 7.3 virtual machine, a Debian Linux virtual machine and several Fedora Core 5 virtual machines. Normal traffic using different protocols like FTP and HTTP was recorded for use in the lab on plain-text protocols. Network attacks between virtual machines in the virtual network were generated using published exploit codes and open-source tools like Nmap, Nessus, IDSwakeup and the Metasploit Framework. The traffic of the attacks was saved in trace files. For example, in order to record an attack exploiting the Microsoft RPC DCOM

buffer-overflow vulnerability, we compiled the exploit code and started the attack from a Fedora Linux virtual machine. The targeted victim was a Windows 2000 virtual machine. The traffic of the attack was recorded by TCPdump in pcap format. The recorded trace file was placed on the class ftp server for student to download, analyze and report.

Figure 2 shows Metasploit Framework running inside a FreeBSD virtual machine, which is part of a private network with multiple virtual machines. The instructor used the virtual network to simulate attacks and record traces.

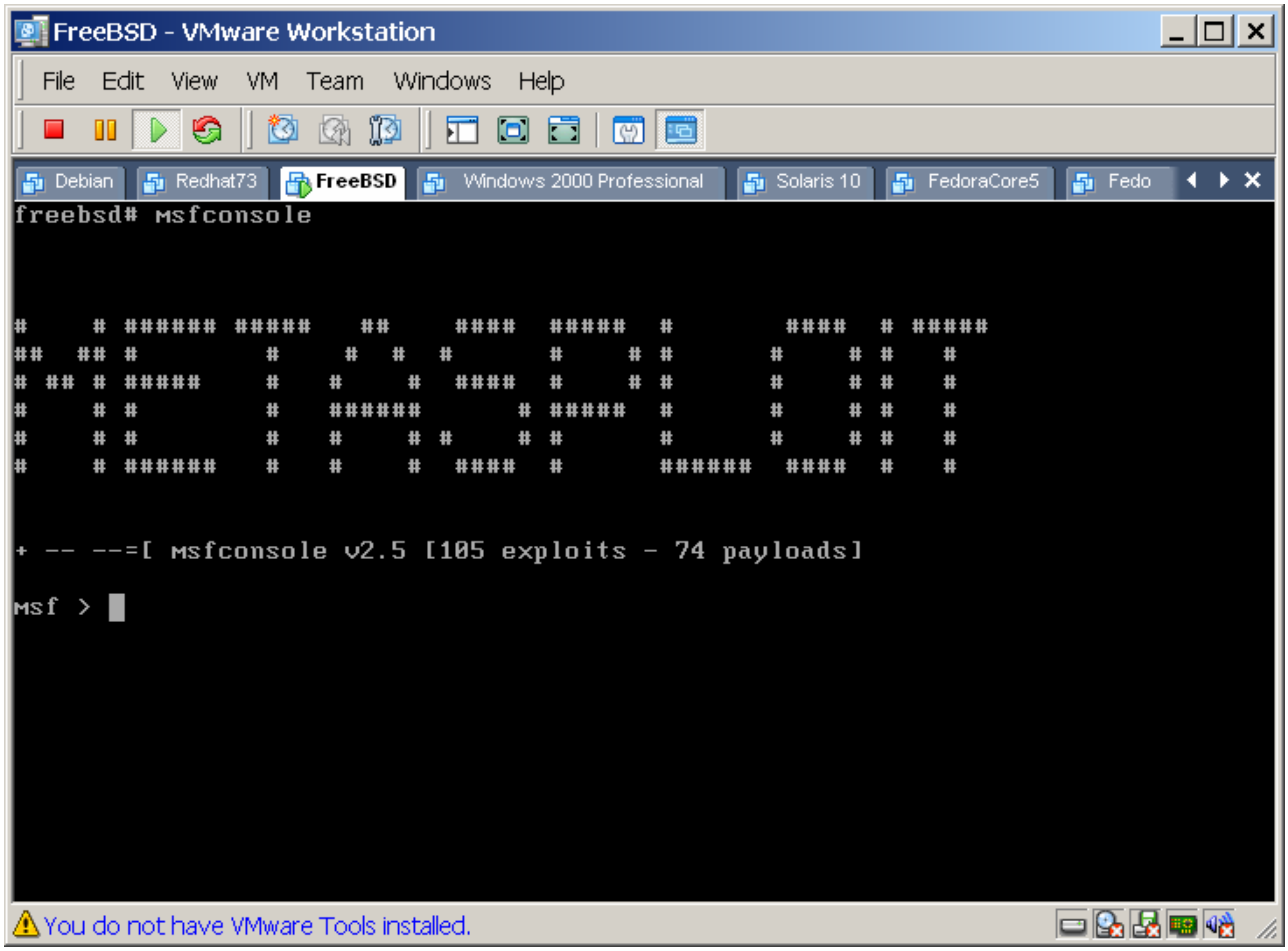

Figure 2. Metasploit Framework is running inside a FreeBSD virtual machine.

2.3 Virtual Machine Installation on the Students' Computers

The minimal hardware requirements for deploying the virtual machine included a 1 GHZ, Pentium III CPU; 512MB RAM; and 10GB free space on the hard drive. The recommended hardware requirements were Pentium IV class CPU, 1GB of RAM and 20GB free space on the hard drive. Windows XP is required for the host operating system. In our department, students without Windows XP are provided with the operating system installation media and license key through the Microsoft Academic Alliance Program.

Once the minimum system requirements have been met, the students were required to install some specific programs under Windows XP on their own computers at the beginning of the semester. A summary of these programs is provided in Table 1.

Table 1. Summary of student-installed programs under the host OS

Program	Purpose	Source
VMware Server or Player 1.0	To run the Linux VM	http://www.vmware.com
SSH Secure Shell Client 3.2.9	To access the VM remotely	http://www.ssh.com
Nmap 4.11	To scan/attack the VM	http://insecure.org/nmap
Wireshark (Ethereal) 0.99.3	To capture network traffic	http://www.wireshark.org
7-Zip	To compress and extract data	http://www.7-zip.org

Windows XP was used as the host operating system, while the guest operating system, Fedora Core 5 Linux, ran in the virtual machine. The students were given both the user and root accounts and had full control over the virtual machine. Our intention was to try and emulate real world situations in which the students worked as the intrusion detection analysts.

Weekly labs typically consisted of two parts. In the first part of the lab, the students installed an Intrusion Detection System (IDS) or a security tool. Some programs were installed using YUM[9]; others were installed from source code. In the second part, they began to use the IDS or the security tool. The major security programs installed and used by the students on the Linux virtual machine included AIDE, OSSEC, TCPdump, Wireshark, Snort, BASE, IPAudit and Bro. In some experiments, the traffic was sniffed by the students in real time. In other experiments, the students analyzed pre-recorded trace files that were created by the instructor. Our observation showed that running one Linux virtual machine was sufficient for the students to meet all of the required objectives in our Intrusion Detection Lab.

An important advantage of the virtualization technology is that a typical VM can be reloaded quickly from a pre-built image relatively quickly. All a user needs to do is to delete the old virtual machine and install the new one from the saved image. The typical process takes just a few minutes to complete. In comparison, if a similar process was to be replicated on a physical computer, it would take much longer and require much more effort to reinstall, even when an auto-install mechanism, like Kickstart, is used. As part of a lab on host-based intrusion detection, for example, students were instructed to install and detect a loadable-kernel-module rootkit. This exercise was a bit challenging for the inexperienced students, however, if they did anything wrong and got stuck, they could delete the "infected" virtual machine and reinstall from a fresh image in no time. This capability greatly enhanced students' confidence in the lab exercises by providing a very fault tolerant work environment. This would have been difficult to accomplish in a physical lab environment due to limited lab hours.

2.4 Topics Covered in the Lab

To familiarize all students with VMware and Linux OS, quick review sessions were provided at the beginning of the semester. Initial lab handouts were furnished with step-by-step instructions to minimize anxiety and the learning curve. Once off the ground, the students completed several

lab exercises and case studies. The following weekly labs were completed by students using the Fedora Linux virtual machine:

a. Installing and running Host-based Intrusion Detection System AIDE, a free replacement of Tripwire
b. Installing and running Host-based Intrusion Detection System OSSEC
c. Loadable-kernel-module rootkit detection
d. Using packet capture and analysis tools Tcpdump and Wireshark (Ethereal)
e. Traffic analysis of network scanning
f. Installing and using Network-based Intrusion Detection System Snort
g. Installing and using Basic Analysis and Security Engine (BASE)
h. Installing and using Network Traffic Monitoring System IPAudit
i. Installing and using Bro Intrusion Detection System
j. Traffic analysis of IRC bots and tuning IDS rules

The following case studies were completed by students without using the virtual machines:

a. Monitoring TCP/UDP Ports and Internet Footprinting
b. Incident Response and Forensics
c. Malware Detection and Handling under Windows

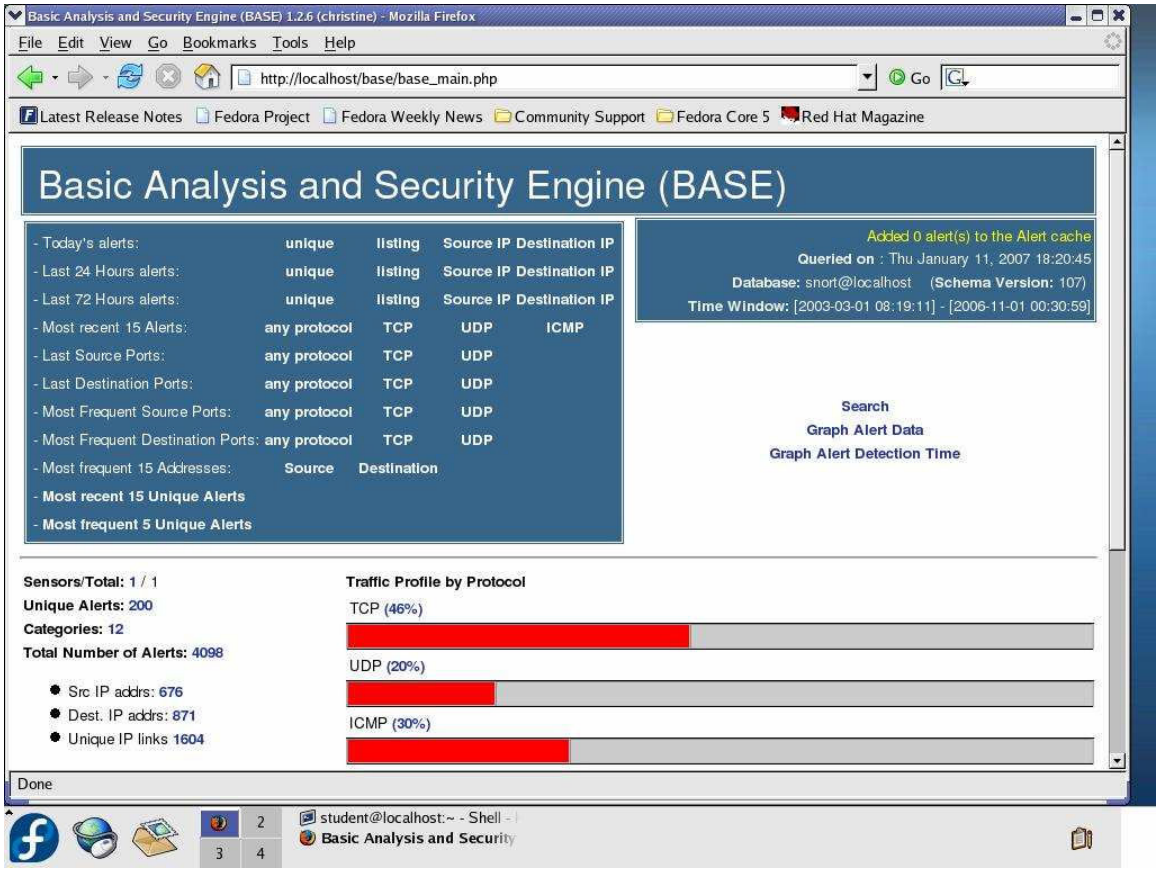

Figure 3. Basic Analysis and Security Engine (BASE) in a Fedora Linux virtual machine.

Figure 3 shows a screen shot of Basic Analysis and Security Engine (BASE) inside a Fedora Linux virtual machine. Snort and BASE were set up by students individually in a two-week project.

3. Evaluation

To assess the effectiveness of instruction and lab experiences, a voluntary, anonymous online survey was administered to the students near the end of the semester. A total of 11 out of 18 on-campus, and 13 out of 22 distance education (DE) students responded to the survey.

100% of the surveyed on-campus students acknowledged that using VMware helped them understand the concepts in Intrusion Detection Technologies, and they may continue using VMware after finishing this course. In comparison, 100% of surveyed DE students acknowledged that using VMware helps them understand the concepts in Intrusion Detection Technologies; while 92% of surveyed DE students said they may continue using VMware after finishing this course.

The students were also asked to comment on what they liked or disliked about the use of VMware in the lab exercises. Their responses are summarized in Table 2 and Table 3.

Table 2. Responses to the question "What do you like about using VMware in the lab?"

	Surveyed on-campus students	Surveyed DE students
I could study at my own pace.	73%	54%
I could learn Linux and other new stuff.	82%	77%
The OS could be easily installed and reinstalled.	82%	77%
I could experiment with the security tools inside a virtual network.	91%	69%
Other	27%	15%

Table 3. Responses to the question "What don't you like about using VMware in the lab?"

	Surveyed on-campus students	Surveyed DE students
It is difficult to use.	0%	0%
It is time-consuming.	9%	0%
It is too slow.	36%	15%
It uses too much CPU/memory/space.	55%	23%
Other	18%	0%

4. Conclusion and Future Work

The fall semester, 2006, was the first time that we have used virtual machines extensively in the undergraduate Intrusion Detection Lab. Compared to physical laboratory environments, VM technology afforded us the opportunity to extend the lab environment to students' homes.

Students' responses regarding the use of VMware have been generally positive. The students can now run the virtual lab on their own computers at any time and at their own pace. They have more time to practice troubleshooting, one of the most essential skills for IT professionals. For both the face-to-face and the DE students, the virtual lab was as useful as the physical lab and supported more diversified platforms like Solaris, Netware and FreeBSD which were not available in our physical lab. Some motivated students were able to conduct further research and experiments using the virtual machines.

Based on our experiences this past semester, there are a few things that we would like to work on in the future. First, we need to strengthen VMware and Linux training in lower-level courses/labs. This will improve students' readiness for more advanced concepts in upper division courses. Second, the students only used one Linux virtual machine, which was sufficient for an undergraduate IDS Lab; however, we would like to explore the possibility of building a virtual security lab using multiple virtual machines with minimal installations. The proposed virtual lab should among other things:

- allow installations that are small enough to run smoothly on a typical home computer with 512MB RAM
- permit teaching other lab-based courses from the undergraduate security curriculum, including Information Assurance Lab, Ethical Hacking Lab and Intrusion Detection Lab using virtualization technologies
- permit distribution via various media, including DVD.

Virtualization technology has worked well for us thus far. We believe that the adoption of virtualization technology in delivering remote lab will not only enhance distance education, but also benefit on-campus education.

Bibliography

1. VMware Server. http://www.vmware.com/products/server
2. Microsoft Virtual Server. http://www.microsoft.com/windowsserversystem/virtualserver/software/default.mspx
3. http://en.wikipedia.org/wiki/Virtual_machine
4. http://en.wikipedia.org/wiki/Comparison_of_virtual_machines
5. Jeremy Sugerman, Ganesh Venkitachalam and Beng-Hong Lim, "Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor", Proceedings of the 2001 USENIX Annual Technical Conference, 2001.
6. 7-Zip. http://www.7-zip.org
7. Gary D. Steffen, "Teaching Local Area Networking in a Secure Virtual Environment", Proceedings of the 2004 ASEE Annual Conference & Exposition, 2004.
8. Richard Bejtlich, "The Tao of Network Security Monitoring", Addison-Wesley, 2005
9. http://linux.duke.edu/projects/yum