Integration of Virtualization Technology into Network Security Laboratory

Peng Li, Tijjani Mohammed
East Carolina University, lipeng@ecu.edu, mohammedt@ecu.edu

Abstract - Distance education has witnessed a tremendous growth in the past decade. Advances in Internet technologies have made it possible to deliver not only lectures, but also hands-on labs remotely. Virtualization technology enables multiple virtual machines and their applications to run simultaneously on a single physical computer. This eliminates the need to have multiple physical machines host diverse operating systems typically deployed in remote network security labs. In preparation for a course on intrusion detection systems (IDS), the instructor creates pre-configured virtual machines and network trace files for students’ use. These virtual machines are then installed by the students on their personal computers at home and used to conduct lab exercises. The virtual lab approach is different from the centralized remote laboratory because students run the lab on their own computers and do not depend on the remote servers. Additionally, the virtual environments allow rapid changes to be made to the lab exercises or environments thus allowing instruction with up-to-date technologies. Furthermore, the burden of maintaining centralized physical labs has been lifted from the institution’s shoulders.

Our approach to decentralized virtual lab-based distance education has been well received by our students.

Index Terms – Decentralized Remote Laboratory, Distance Education, Network Security, Virtual Machine.

INTRODUCTION

Distance technology education demands cost-effective ways of delivering hands-on labs remotely. A common solution is for schools to house centralized remote laboratories that the students connect to from home to perform lab exercises. Rapid advances in technology along with growing enrollments often stretch centralized lab resources to their limits. This issue is further compounded by shrinking resources that make it difficult for schools to maintain and/or upgrade centralized labs in step with the developments in technology.

This paper presents a plausible solution that integrates virtualization technology into a distance education (DE) course on intrusion detection. Virtualization technology enables multiple virtual machines with guest operating systems and their applications to run concurrently on a single physical machine. This technology significantly reduces, or completely eliminates the necessity to have multiple physical computers needed to host diverse operating systems typically deployed in network security labs.

In preparation for the course, the instructor creates pre-configured virtual machines and network trace files for students’ use. These virtual machines are then distributed to the students to be installed on their personal computers at home. The students can then use their installed virtual machines to conduct hands-on lab exercises in the secure private network formed by the virtual machines.

The virtual lab approach is different from a typical centralized remote laboratory because students run the lab exercises on their own computers and do not depend on the remote servers on campus. Additionally, the virtual environments allow rapid changes to be made to lab exercises or environments thus allowing instruction with up-to-date technologies. Furthermore, the burden of maintaining centralized physical labs has been lifted from the institution’s shoulders, thus allowing meager institutional resources to be redirected to other areas of need.

BACKGROUND

Virtualization technology was first used by IBM in the 1960s to offer concurrent timesharing of their mainframe systems [1]. Virtual machine monitor (VMM) was introduced as a software abstraction layer to manage the mainframe hardware and to support virtual machines [2]. Each virtual machine (VM) runs like a physical machine. Users can install operating systems and execute applications in the virtual machines in the same manner as they would on physical machines.

It was not until the late 1990s that virtualization technology breakthrough occurred on the Intel x86 platforms, which allowed users to run multiple virtual machines (including operating systems and applications) simultaneously on a single personal computer (PC). Figure 1 shows the architecture of hosted virtual machine monitor, which extends the host operating system to support a virtual machine running the guest operating system [3]-[4].

Currently there are many different virtualization solutions available for x86-based personal computers and Macintosh computers. The host operating system can be FreeBSD, Linux, Mac OS X and Windows 2000, XP, Vista or 2003. The commonly used Virtual Machine Monitor (VMM) programs include Microsoft Virtual Server,
Parallels Workstation, QEMU, VMware Server, VMware Workstation, and Xen [5].

Physical Machine

Host Operating System

Virtual Machine Monitor (VMM)

Guest Operating System

Applications

Host Applications

FIGURE 1
THE ARCHITECTURE OF HOSTED VIRTUAL MACHINE MONITOR

VMware was reportedly used in networking lab courses as early as 2003 [6]-[7]. It was, however, mostly deployed in on-campus computer labs. Students either sat in the physical lab to use the virtual machines or connected to the remote servers hosting virtual machines to do lab exercises. The adoption of virtual machine technology in academic teaching has been slow until recent years because:

- Running virtual machines inside a physical computer demanded a lot of memory. The requirements for the CPU and the storage were also quite high. Most personal computers did not meet the requirements.
- The open source virtualization programs QEMU and Xen were not supported under Windows and were difficult to configure for non-experts.
- User-friendly virtualization programs such as VMware were not free, so the implementation of virtualization was quite expensive.
- Virtualization on x86 platforms did not have a long history. VMware Inc. was founded in 1998. Xen was initially released in 2003 and QEMU was started in 2003. The technology is still evolving and has not proven to be stable until recently.

In 2006, VMware Inc. released VMware Server 1.0, a replacement of GSX server, free of charge, for non-commercial personal use. Shortly after that, Microsoft released their Virtual Server for free use. Other virtualization projects such as Xen, also received endorsements from major technology companies. The golden era of virtualization had arrived.

We started testing decentralized approach to remote labs with virtual machines in the fall semester of 2006, after VMware Player and VMware Server became available for free use [8]. Later we joined the VMware Academic Program so our students could use VMware Workstation for coursework free of charge.

VMware was selected over other virtualization products for three main reasons:

1. VMware could be installed under Windows operating systems, which were used by most students at home. The students could continue using their existing operating systems and applications without interruptions. In contrast, Xen, boasting its high performance, did not support Windows as the host operating system and could only be hosted under selected UNIX/Linux systems with modified kernels.

2. VMware supported more guest operating systems than other virtualization programs. VMware virtual machines were capable of running guest operating systems including DOS, Windows, Linux, FreeBSD, NetWare and Solaris. It was possible to emulate a diversified network environment using VMware virtual machines. Microsoft Virtual Server R1 only supported Windows and selected Linux guest operating systems.

3. Our faculty members tested various virtualization programs and found VMware Workstation and Server to be stable and user-friendly.

TRADITIONAL CENTRALIZED REMOTE LABS

In the traditional model of centralized remote labs, physical machines and other physical equipment are needed for students’ use. A typical centralized networking lab is illustrated in Figure 2. The computers in the lab are connected through hubs or switches to a VPN concentrator/server. Remote users (students) connect to the server through Internet using different methods (VPN, Remote Desktop, SSH etc.) to conduct lab exercises. This setup has the following disadvantages:

- The lab equipment, including physical computers, switches, cables, routers, is expensive.
- The cost of maintaining these labs can be quite high. Remote access to these labs, even through broadband Internet connection, can be slow and unstable sometimes.
- Students often have to share the equipment with others. In some implementations, the students are assigned certain lab time slots, or must use a scheduling system to reserve the lab time. Those who are unable to complete their exercises within the designated or assigned time slots may not be able to continue their work at a later time.
Decentralized Virtual Labs

In the decentralized virtual lab approach, pre-configured virtual machines and other lab materials are prepared by the instructor and distributed to the students. The students then install the virtual machines on their personal computers and complete hands-on exercises using the virtual machines. These students no longer need to connect to a central server on the university campus to perform lab exercises.

The decentralized virtual lab has the following advantages:

- It is portable because it allows students to conduct hands-on lab exercises anywhere, at any time, and at their own pace. Students no longer need to share lab equipment with other students in the class.
- The virtual lab is expandable. As long as the host machine has sufficient memory, virtual machines can be duplicated and added easily.
- Because the virtual machines run locally, they are much more responsive than the physical machines accessed remotely via Remote Desktop or Virtual Network Computing (VNC).
- The cost of setting up the virtual lab is low since the labs are conducted on students’ computers.
- The maintenance is a snap since corrupted virtual machines can be restored or replaced within a very short period of time.
- New lab projects can be deployed more quickly through virtual machines.

The decentralized virtual lab, however, has its limitations because not all lab equipment can be virtualized. To that end, the virtual lab is complementary to centralized physical lab but will not replace it in the foreseeable future.

Preparation of Network Traces

Network traffic analysis is perhaps one of the most important skills for an information security specialist. On a real production network, traffic volume is often overwhelming that it is nearly impossible to conduct real-time analysis. Usually interesting packets are captured and saved as trace files for later analyses. However, network traffic recorded on a real public network is most likely unavailable for academic use due to privacy and other legal reasons.

Prior to the availability of virtual machines, network traces were obtained for academic use from two primary sources. The first source was from trace files that were released for public use, such as the 1998/1999 DARPA Intrusion Detection Evaluation Datasets [9]-[10]; and the data recorded during the Capture the Flag Contest at DefCon [11]. In July 2006, Bejtlich started a web site [12] to serve the digital security community as a repository of network traffic captures for open use. More often than not, the data that was made available in the public domain was very limited and sometimes outdated.

The second source was from, the network traffic that was emulated and recorded in isolated private networking labs. The cost of setting up and maintaining these physical labs, however, was often quite high.

Virtualization technology makes it possible to build a private virtual lab for traffic captures. This virtual lab resides on one physical computer and emulates a network consisting of virtual machines running different operating systems, including Windows, Linux, FreeBSD, NetWare and Solaris. The hardware cost of putting this virtual lab
together is minimal compared to a physical lab. For example, instead of an institution having to acquire a dedicated Sun Workstation to host Solaris, a Solaris virtual machine can be added to the virtual network as long as the host machine has enough RAM. To that end, the cost is essentially zero.

The private virtual network can be easily isolated from the public network. Network traffic or attacks generated within the virtual network have no impact on the public network. For instance, malware samples [13] can be executed and studied inside the virtual machines. The malware-related traffic can be contained and captured inside the private network. After the work is done, the infected virtual machine(s) can be deleted and replaced with fresh virtual machine(s) fairly quickly.

To prepare the network trace files for lab use, we created a virtual network composing of virtual machines running Debian 3.0, FreeBSD 5, Red Hat 7.3, Fedora Core 5, Solaris 10 and Windows 2000. Older versions of the operating systems were deployed to provide an opportunity for capturing and studying attacks that exploited the vulnerabilities in the operating systems. For example, we launched an attack exploiting Microsoft RPC DCOM vulnerability from a Linux virtual machine toward a vulnerable virtual machine running Windows 2000 with Service Pack 1. The network traffic of the attack was recorded using Wireshark.

Network traffic was captured in this virtual network and saved as trace files. Various network traces were recorded for HTTP, FTP, SMTP, TELNET, SSH, IRC, RPC, SMB, TCP Connect Scan, TCP SYN Scan, TCP XMAS Scan, Window enumeration, OS fingerprinting, Point to Point file sharing, and network and malware attacks. These trace files were later distributed to our students for analysis and testing on the intrusion detection systems they set up on the virtual machines.

### VIRTUAL LAB IN FALL 2006

We started experimenting with the decentralized remote lab approach in our undergraduate course on Intrusion Detection Technologies during the fall semester of 2006. This was one of the required senior-level courses for Information and Computer Technology (ICT) majors. The three-credit hour course was taught in two lecture and two lab hours each week. At the completion of the course, each student was expected to understand network security threats, identify malicious activities and attacks, and implement intrusion detection systems to monitor and protect given networks.

The students began the course by downloading and deploying a single Fedora Linux virtual machine at the beginning of the semester. The minimal hardware requirements included a 1GHz Pentium III CPU, 512MB RAM, and 10 GB of free hard drive space. The Linux VM contained basic packages such as nano, yum and wget, and development tools such as GCC. Network security and intrusion detection programs were not pre-installed. Instead, the students were required to install, configure and use them in the weekly labs. Many security programs such as OSSEC, Bro and IPAudit were not bundled with Fedora. To that end, the students had to compile the programs from the source code. Compiling is one of the basic skills that the students will need to deploy intrusion detection systems in the real world.

The host machine was used by the students as the client to attack the virtual machine (server). The intrusion detection systems such as Snort were installed on the virtual machine to monitor the attacks. The students performed the following lab projects using the Linux virtual machines on their personal computers: Host-based Intrusion Detection System AIDE; Host-based Intrusion Detection System OSSEC; Loadable-Kernel-Module Rootkit Detection; Packet Capture Tools Tcpdump and Wireshark; Traffic Analysis of Network Scanning; Network-based Intrusion Detection System Snort; Basic Analysis and Security Engine (BASE); Network Traffic Monitoring System IPAudit; Bro Intrusion Detection System; and Tuning Intrusion Detection System.

Considering the fact that some of the students had limited experience with Linux, the first few lab handouts were very detailed, and included step-by-step instructions on how to conduct the experiments. Subsequent labs grew progressively more challenging and had less detailed handouts. The students were expected to apply skills learned in the previous labs in solving subsequent problems. A typical lab handout contained 20 - 50 questions, including observation questions and discussion questions. The observation questions required students to record outputs of lab activities, which evaluated their abilities to complete mandatory tasks. The discussion questions measured the students’ understanding of concepts. After completing the weekly labs, the students submitted electronic copies of their lab reports in form of answer sheets, often with supporting materials, including screenshots, configuration files, logs or packet captures. The lab reports were graded based on the completeness of tasks and the depth of analyses.

The first trial of the virtual lab went fairly well, with mostly positive feedback from the students. The following issues surfaced from our first implementation:

- Lab projects involving multiple hosts were planned but not implemented because some students’ computers did not have enough RAM to support more than one virtual machine at a time.
- The students used the host machine as the client and needed to install many programs (WinPeap, SSH, Nmap, Wireshark etc.) in addition to VMware. Most students’ host operating systems were Windows XP. But there were a few students using Windows 2000, or Linux. The virtual machines still ran on these systems but other programs behaved slightly differently on different platforms.
The students depended on the Internet to download source codes, RPMs, and other data. If the Internet connection was slow or temporarily unavailable, they were not able to perform the labs.

**VIRTUAL LAB IN FALL 2007**

Based on the lessons learnt from the first implementation during the fall term in 2006, some modifications were made to the virtual labs. For example, to improve the portability and the expandability of the virtual lab, two virtual machines (VM1 and VM2) were provided for selected lab projects in the fall semester of 2007. The virtual machines were customized to use as less memory as possible. The students ran the two virtual machines simultaneously on their personal computers. The virtual machine VM1 acted as the client or the attacker, while VM2 worked as the server or the defender. The students used VM1 to attack VM2 and to analyze the results. They also set up network defense systems such as Snort on VM2 to detect and defend against attacks in the labs. Table I shows the specifications of the two virtual machines. Figure 4 is a screenshot of VMware Workstation running a CentOS 5 virtual machine and a Debian virtual machine together. The browser inside the virtual machine displays the BASE console for Snort.

With two virtual machines, it was possible to implement labs using multiple hosts. The Snort/BASE lab was updated. In the old project, the Snort sensor and the BASE console were installed on the same virtual machine. In the updated lab, the students installed the Snort sensor on VM1 and the BASE central console on VM2. Then a secure (Stunnel) connection was established to transmit the alerts from the sensor to the console. This setup was closer to the configuration in the real world.

With VM1 acting as the client, the virtual lab became more portable. Although the students could still use the host machine as the client, it was no longer required. If a student’s laptop was temporarily out of service, she/he could install the virtual lab on her/his desktop computer if there was one. The student could also go to our on-campus computer lab, launch VMware Workstation, open the pre-built virtual machines and perform the labs. There is no need to install any program on the host machine, which is not allowed in most student computer labs.

![Figure 4 VMware Workstation](image)

**TABLE I**

<table>
<thead>
<tr>
<th>Specifications of Pre-Built Virtual Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating System</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
</tbody>
</table>
| Debian Linux 3.1  | 141 MB
| CentOS Linux 5.0  | 368 MB
| Default Memory    | 64 MB 96 MB |
| Compressed Image Size | 141 MB 368 MB |
| Uncompressed VM Size | Up to 1 GB  Up to 4 GB |
| Main Programs Installed | Tcpcap, Wireshark, Nmap, Snort, Nessus, Stunnel, Firefox, Fluxbox | Apache, MySQL, Development Tools and Libraries |
| Purpose            | Client, attacker, network sensor | Server, defender, central console |

**LAB EVALUATION**

To evaluate the effectiveness of the virtual lab, an anonymous, voluntary survey was carried out online before the end of the fall semesters in 2006 and 2007. In 2006, 24 out of 40 students (including 18 on-campus students and 22 DE students) responded to the survey. In 2007, 24 out of 67 students (including 24 on-campus students and 43 DE students) responded to the survey.

In both years, 100% of the surveyed students acknowledged that the virtual lab helped them understand the lecture topics. A few questions were added in the 2007 survey. 86% of students who responded were positive that the hands-on exercises in the virtual lab were as effective as the ones in a physical computer lab. The remaining 14% were neutral. 100% of respondents agreed that lab projects were well organized. Students’ opinions about VMware are summarized in tables II and III.

**TABLE II**

<table>
<thead>
<tr>
<th>Responses to the Question “What do you like about using VMware in the lab?”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fall</td>
</tr>
<tr>
<td>I could study at my own pace.</td>
</tr>
<tr>
<td>I could learn Linux and other new stuff.</td>
</tr>
<tr>
<td>The OS could be easily installed and reinstated.</td>
</tr>
<tr>
<td>I could experiment with the security tools inside a virtual network.</td>
</tr>
<tr>
<td>Other</td>
</tr>
</tbody>
</table>
The biggest complaint received was that VMware used too much CPU/memory/space. Our plan is to further tailor the virtual machines to make them use less memory and space. At the same time, we will encourage the students to upgrade their computers to meet the hardware specifications required by our college. The hardware requirements for VMware were considered very demanding a few years ago, however, the price-performance ratio of personal computers has been steadily falling. We are confident that the virtual lab can run smoothly on a typical 2-year old computer with 512MB RAM.

CONCLUSIONS

Using virtual machines in the Intrusion Detection Lab is part of an ongoing effort to integrate virtualization technology into our network security curriculum. We have successfully deployed virtual machine technology for teaching and laboratory exploration over a period of two years. Our approach to virtualized lab-based distance education has been well received by our students. Class performance and positive course feedback demonstrate the success of this method of instruction.

Following successful utilization of virtual machine technology in the Intrusion Detection Lab, other instructors have either adopted, or are considering using this technology in other laboratory-based courses in our Information and Technology program. Compared to traditional remote access physical labs supported by the institution, the virtual labs offer a cost-effective and viable alternative for hands-on IT courses. Another advantage is the ability to modularize the labs so they could be reused in other courses. For example, the labs on intrusion detection, digital forensics, and ethical hacking may share modules with minimal modifications.

The decentralized virtual lab may not replace the centralized remote lab, but the approach can be very useful for some network security lab courses.

ACKNOWLEDGMENT

The authors would like to thank VMware Inc. for granting licenses of VMware Workstation through the VMware Academic Program. The licenses of Windows operating systems were provided through the MSDN Academic Alliance Program.

TABLE III

RESPONSES TO THE QUESTION "WHAT DON'T YOU LIKE ABOUT USING VMWARE IN THE LAB?"

<table>
<thead>
<tr>
<th></th>
<th>Fall 2006</th>
<th>Fall 2007</th>
</tr>
</thead>
<tbody>
<tr>
<td>It is difficult to use.</td>
<td>0%</td>
<td>5%</td>
</tr>
<tr>
<td>It is time-consuming.</td>
<td>4%</td>
<td>9%</td>
</tr>
<tr>
<td>It is too slow.</td>
<td>25%</td>
<td>5%</td>
</tr>
<tr>
<td>It uses too much CPU/memory/space.</td>
<td>38%</td>
<td>50%</td>
</tr>
<tr>
<td>Other</td>
<td>8%</td>
<td>9%</td>
</tr>
<tr>
<td>None</td>
<td></td>
<td>32%</td>
</tr>
</tbody>
</table>

REFERENCES


AUTHOR INFORMATION

Peng Li, Assistant Professor, East Carolina University, lipeng@ecu.edu.

Tijjani Mohammed, Assistant Professor, East Carolina University, mohammedt@ecu.edu