Virtual Machines - An Idea Whose Time Has Returned: Application to Network, Security, and Database Courses

William I. Bullers, Jr.
Anderson Schools of Management
University of New Mexico
1-505-277-4901
bullers@mgt.unm.edu

Stephen Burd
Anderson Schools of Management
University of New Mexico
1-505-277-6418
burd@mgt.unm.edu

Alessandro F. Seazzu
Anderson Schools of Management
University of New Mexico
1-505-277-8451
alex@mgt.unm.edu

ABSTRACT
Virtual machines provide a secure environment within which students may install, configure, and experiment with operating system, network, and database software. This paper describes experiences teaching three advanced courses in system and network administration, information security and assurance, and database administration using VMware workstation in a shared student laboratory. The paper describes benefits and challenges in course and lab configuration, security, and administration.

Categories and Subject Descriptors
K.3 [**Computers & Education**]: Computer & Information Science Education.; K.6.5 [**Management of Computing And Information Systems**]: Security and Protection; D.4.6 [**Operating Systems**]: Security and Protection; C.2.0 [**Computer-Communication Networks**]: Security and Protection; H.2.7; [Database Management]: Database Administration – Security, Integrity & Protection.

General Terms
Design, Experimentation, Security.

Keywords
Virtual machines, VMware, Network, Security, Database.

1. INTRODUCTION
Some computer science and management information system courses require students to install, configure, and experiment with operating system and network computing environments. Students may use one or more operating systems, multiple configurations of a single operating system, and a variety of application software and other tools. Providing needed hardware and software in a laboratory that supports multiple courses and general-purpose uses can be cumbersome or impossible.

SIGCSE'06, March 1-5, 2006, Houston, Texas, USA.

To perform required tasks in many advanced courses, students must have administrative privileges on laboratory computers. Providing such privileged access in a shared student computing environment potentially compromises the integrity and availability of institutional computing resources.

This paper describes support of multiple advanced courses in a shared computing environment using virtual machines. We report on our experiences supporting three concurrent advanced courses in a lab that also supported other courses and general-purpose computing needs. We summarize the benefits and challenges of using virtual machines and describe how we addressed problems and administrative issues within and across courses.

2. VIRTUAL MACHINES
2.1 Definition and History
One of the earliest VM systems was CP-40 [1], developed for an IBM System/360 Model 40. A VM system was "defined as a computing system in which the *instructions* issued by a program may be different from those actually executed by the hardware to perform a given task" [11]. More generally, a VM can be categorized as a "software abstraction with the looks of a computer system's hardware (real machine)" [12].

Advantages of VMs were described by Parmelee [11], including:

- Concurrently executing multiple operating systems supporting different users
- Developing software for one machine on a different machine
- Insulating one software environment executing on a VM from failures in others

VM systems were also purported to enhance computer system security "if the programs of independent and possibly malicious users are to coexist on the same computer system" [7]. It was recognized that "a combined virtual machine monitor/operating system (VMM/OS) approach to information system isolation provides substantially better software security than a conventional multiprogramming operating system approach". In addition, VM systems offered "a very suitable basis for system development" [4] given the independence of VM's and their support for debugging and monitoring of systems activities.

The salient features of VMs appeared to be so fundamental that Parmelee, et. al. [11] closed their review by saying "The utility of the virtual storage and machine concepts is well-established, and in the future, the application and extension of these concepts is

certain to increase." Their prognosis appeared justified throughout the heyday of mainframe computing from the mid-1960's through the 1980's, as embodied in operating systems such as IBM's VM/370. However, VMs waned with the advent of smaller and cheaper computers and ubiquitous networks.

2.2 Rebirth of Virtual Machines

Recently, the use of VM concepts has experienced a rebirth in computing [12]. VM software such as Xen [3] and VMware Server introduces an abstraction layer between operating systems and computer hardware which enables multiple virtual servers to share a common pool of hardware resources. Other VM software such as VMware Workstation [15] creates virtual hardware environments within a host operating system that enables other operating systems and their applications to execute within encapsulated virtual machines. This form of VM software is commonly used to test system and application software under multiple operating systems or operating system versions on a single physical machine. For the remainder of this paper, references to VMware will be for the workstation version only.

Modern VM software such as VMware provides the benefits described earlier including concurrently executing multiple operating systems in separate VMs and isolating each VM from bugs and malicious code in other VMs. Additional benefits include the ability to configure virtual networks. Early VM's consumed about 10-15% of the underlying computing resources [7]. Modern VMs consume just a few percentage points of the physical machine resources [3, 12].

2.3 VM Use in Education

Prior to the return of VM's, the academic computing community was forced to fund dedicated computer labs in order to teach advanced classes covering topics such as system administration, network design, and information security [5, 17, 16]. However, such labs are expensive and often feasible only with external funding such as an NSF grant [9]. The reemergence of VM software enables educational institutions to configure shared, multi-course computing labs in which students in advanced courses can undertake system-level projects [8, 2].

VM features such as isolation, compatibility, and encapsulation allow an instructor to build virtual network topologies [6] encompassing multiple, independent operating systems and networks [13, 8]. These features permit an instructor to create a "virtual kernel development environment in which operating systems can be developed, debugged, and rebooted in a shared computing lab environment without affecting other applications users" [10]. This type of applications development and debugging support is particularly important in the network and operating system "development cycle of plan-implement-reboot-test-debug" [10] and valuable for emphasizing troubleshooting in teaching system administration [14].

3. Virtual Machine Usage in the Classroom

The authors used VMware during 2005 in a shared computing lab to support three advanced courses covering system and network administration, information security, and database administration. The lab also supported an accounting class and occasionally general-purpose use.

Details of the lab environment are described in Table 1. Details of the advanced courses are provided in the following sections.

Components	Details	Cost
Workstations (17)	Windows XP, 3 GHz Pentium 4, 2 GBytes RAM, 40 GByte disk, 100 Mbps Ethernet	$1850
VM software	VMware workstation	$110
Client O/Ss (executing within VMs)	Windows 2000 and XP Professional, Windows Server 2003, and FreeBSD	Free or covered under $400/year institutional license
Network	24 port 100 Mbps Ethernet hub with a single 100 Mbps external connection.	$150
Backup server	Windows 2003 Server, dual 800 MHz Pentium 3, 768 Mbytes RAM, 500 GBytes RAID disk storage	$2500
Approximate total cost		$36,000

Table 1. Laboratory hardware and software

In semesters before VMware was used, lab computer disks were divided into several partitions supporting a standard general-purpose lab configuration and specialized configurations for advanced classes. Student teams used adjacent computers for lab exercises that required multiple machines. Changing a machine configuration required a lengthy reboot and disk space limited the number of partitions that could be supported.

3.1 System & Network Administration

The systems & network administration (S&NA) class covers knowledge and skills necessary to install, configure, and operate small- to medium-size network and server infrastructure based on Windows Server 2003. Most topics are reinforced with hands-on lab or homework exercises. A list of the exercises and homework assignments for the most recent semester follows:

- TCP/IP configuration
- DNS and DHCP installation/configuration
- Active Directory installation/configuration
- Account and user directory creation and access controls
- Shared folders and offline files
- Web-based data exchange for workgroups
- Web and FTP server installation/configuration
- Disk/file system administration, backup, and recovery
- Remote installation services
- Performance monitoring and tuning
- UNIX system administration

As can be seen from the above list, the course covers many topics and students spend considerable time installing and configuring software and performing typical administrative tasks.

VMware enabled each student to operate concurrent client and server VMse, which enabled server configuration changes to be immediately tested. For example, changes to shared directory permissions could be immediately tested by logging into the client VM as a user and verifying which directories and files

were/weren't accessible. Similarly, changes to network configuration and services such as DNS and DHCP could be quickly tested from the client machine. More labs and more complex exercises were scheduled than in previous semesters due to the increased ease of testing.

In a previous semester (before VMware use), all server partitions became infected when a worm (Code Red) penetrated one of the machines during operating system installation and quickly propagated throughout the lab. The penetration wasn't immediately apparent to the students or the instructor and it was not corrected by the application of appropriate security patches immediately after the installation. The instructor returned to school the next morning to find that the campus computing service had disconnected the entire school subnet due to the flood of malicious traffic emanating from the lab.

VMware provided important capabilities to minimize the opportunity for virus infections and worm penetrations. First, client and server machines were connected via a virtual network using network address translation with port address translation, which enabled unrestricted traffic between client and server VMs but effectively filtered unwanted packets from the host machine and the campus network. Second, for sensitive configuration exercises such as operating system installation and application of service packs, the necessary files were copied to a directory on the host computer that was readable by the virtual machines. This reduced the possibility of receiving adulterated files from bogus web sites and other sources.

VMware also made it much simpler for students to back up and recover server images. This is an important capability in a classroom setting since students do make mistakes and those mistakes occasionally cause irreparable damage to the operating system image. Prior to VMware use, students created backups on neighbor machines the Windows backup utility. The primary problems with this approach were that all machines had to be left booted as servers during the typical several hour backup and server images had to be accessible beyond the host machine.

With VMware, disk partitions are stored as ordinary (but large) files on the host computer system. Students performed backups using SFTP from the host operating system to copy the disk image files to a backup server. The upload was started at the end of class after which the student could "walk away" from the computer. When the upload finished, the SFTP server would time out the connection after a few minutes of inactivity. Another student using the same computer the next day would encounter the standard lab configuration, logged in with the general student account, and with a completed and disconnected SFTP session. Although not implemented for this class, the backup procedure could be automated and controlled by the backup server.

A final advantage of VMware was realized during the session covering UNIX system administration near the end of the semester. The session demonstrated differences between UNIX and Windows Server in the areas of permission schemes, shared folders and file systems, and web services. The instructor installed FreeBSD, Samba, and Apache within a VM on a test machine and configured a set of accounts, directories, and services that mirrored those created by students under Windows Server earlier in the semester. The VMware image was then copied directly to all lab workstations enabling the students to compare the Windows and UNIX VMs side-by-side.

One characteristic of VMware that complicated a few lab sessions was the way in which virtual networks configurations are stored on the host machine. By default, VMware installs a few virtual NIC drivers on the host operating system and organizes them in standard configurations that are stored in a file in the VMware directory in the Program Files folder. The standard configurations include adapter configured as a DHCP client with a virtual network hub that implements DHCP, DNS forwarding, and NAT services. There is also a configuration with no network connectivity and another that disables all communication beyond the host machine.

S&NA students use the first configuration for the first couple of weeks. However, once DHCP and DNS are installed on the Windows server VM, both server and client require a virtual network configuration that provides NAT but not DHCP nor DNS forwarding. Unfortunately, such a network must be custom configured and doing so requires administrative privileges on the host operating system to modify the VMware network adapters and network configuration file. The virtual network can be configured and stored by the lab administrator in advance but it must be individually set up on all machines.

Another complexity arises with the way in which VMware chooses an address for the virtual subnet used with the NAT service enabled. VMware generates 1024 host subnets within the standard class C (192.168.0.0) non-routable subnet but does so with random network addresses (e.g., 192.168.4.0 on one machine and 192.168.16.0) on another. As a result, virtual machines on each host machine use different network numbers and default gateways. Thus, using the standard VMware installation in the TCP/IP configuration lab, the instructor can't provide one set of configuration parameters and assume that they'll function correctly on all machines. The author's answer was to manually reconfigure the VMware virtual networks on each host machine with identical network address ranges. But this was tedious and time consuming.

It may be possible to address both of the preceding problems by modifying the installation procedure for VMware to install all needed virtual network configurations with consistent parameters. But the authors did not have the time to investigate whether or how to do so.

3.2 Information Security and Assurance

Three lab exercises were conducted in the information security and assurance (INFOSEC) class through VMware. The first environment that was simulated was the distribution of a Macromedia Shockwave game that had been altered to contain malicious code. The simulation assumes that a public network is used to transfer the game. Generally this is accomplished by sending an e-mail attachment. Other sources can be FTP, HTTP or peer-to-peer networks.

The student begins by crafting a malicious package of the Shockwave game. To complete this task they are issued a lab CD with three software components — a clean copy of the game penguin.swf, a copy of Sub7, and a recent version of WinZip. By design Sub7 contains two objects to its architecture, a server and a client. In this case a package is prepared to deliver and install the server on the victim's system. This version of Sub7 will only attach its malicious payload to executable files, generally .EXE. The full application itself comes with a client, a ready to distribute

server and a utility to customize a server or create a new one. The first step the student will complete is to create a transport module. In order to be successful this has to be attractive to the victim, thus the selection of a game. The penguin.swf file is first zipped and the resulting file converted into a self-excecutable archive. These changes provide a carrier for the malicious code to be transported in. We should note that this step would not be necessary if we had an executable game or if the malicious tools we were using could directly attach themselves to a Macromedia file.

With the bait in place the student now modifies the package by inserting the server malicious code using the customizing utility. In most versions of Sub7 the one component to consider while crafting the package is to ensure that the server becomes active when the code executes on the victim's system. There are several options to activate the server and make many other modifications but most of them, though useful for pedagogical reasons, are ignored at this time. The last step is to inject the game with the code and save the new package as a .EXE file.

At this point in the exercise we are faced with the challenge of transporting our payload. The students have already configured an attacker system on VMware with Windows XP, SP1 running in host only mode. A second VMware host is configured with host-only IP settings that correspond to the attacker subnet and powered on with either Windows XP or 2000.

The most common attack transport is via e-mail attachments as we see almost daily with worms and other self replicating malicious code. In our exercise e-mail is unavailable as are most common internet transports mentioned earlier. Other options include USB storage devices, floppies or CD-ROMs. Given the size of the game file created, less than 400KB and the availability of free floppies to the student, this becomes our method. From the attacker, the file is copied to the floppy disk. The victim's VM can access the floppy disk once control has been relinquished by the attacker's system. A few more clicks and the victim will be compromised. We assume that the victim can double click the attachment, install the Shockwave plugin from the lab CD if necessary and start playing the game. What is transparent to most users is that at the same time a new service (or server) started running on the victim's system. A close look at the task manager will reveal the new application though there are several techniques available to mask its presence, one simplistic one being to call the service svchost.exe or services.exe.

At this stage of the exercise with the correct IP number of the victim, if no password was configured and the default port was left unchanged, the attacker has access to all the resources available on the victim's system. There are also other playful features an attacker could use, such remotely opening CD trays, flipping the desktop upside down and others. It is questionable though if any of these would be used in a real case since they would immediately provide symptoms of the compromise.

By default Sub7 uses a high port to communicate in its client and server architecture. The port is somewhere in the 27000 range where it is safe to assume that current firewalls will intervene and block that traffic. What seemed to peak the class interest was to see if we could determine a noticeable difference in traffic patterns if the Sub7 server was configured to communicate through more common ports.

In order to do so a second exercise was designed in which the steps above were followed with one specific change. During server configuration, before it was attached to the game, the server port was modified for three common ports, HTTP on port 80, FTP on port 21 and DNS on port 63. At the end of the packaging stage we would thus have three deliverables.

The victim's system was compromised on three separate occasions with minimal configuration work by using the "revert" feature in VMware. The majority of the work done by the students was based or repackaging the malicious server, transporting it and capturing the traffic generated while the attacker connected. The traffic being generated by the server and client modules of Sub7 was captured using Ethereal. The results for each different package were then compared to what would be expected in a normal HTTP, FTP or DNS communication. This exercise pointed to the need for some type of intelligence in the protection of one's assets that would go beyond the historical capabilities of a firewall. Packet analysis or inspection would be required to counter this type of threat. These tools would then recognize that traffic other than HTTP, FTP or DNS was occurring on those ports.

The last exercise and probably the most time consuming to set up was a man in the middle attack. Three VMs were used with the adapters configured as host only. Two were Windows 2003 servers and one Windows XP. The first Windows 2003 server was configured with two Ethernet adapters. Each adapter was configured on a different IP subnet (A and B) and would route traffic from one subnet to the other. The Windows XP system was placed on subnet A, while the second Windows 2003 server was placed on subnet B. Users having physical access to the Windows XP system were given different accounts with varying privileges on the second Windows 2003 server. When they connected to access a resource on the Windows 2003 server they would be asked to authenticate themselves. Using a network sniffing application such as oxid.it's "Cain and Able" on the routing server, which could see all the traffic passing back and forth, the students were able to test how quickly they could crack usernames and passwords from Windows traffic.

3.3 Database Administration

The database administration course covered topics in database administration (DBA) and data warehousing. Students installed the Oracle 10g Database Server (version 10.2.0.1) under Windows XP Professional. VMware provided an isolated installation platform so that students could be granted administrator privileges for the install. Each student assigned passwords to Oracle DBA users, SYSTEM and SYS, in order to administer the database through the Oracle Enterprise Manager (OEM). Students used OEM to carry out DBA tasks including configuration of the database listener and Oracle client, startup and shutdown of the Oracle Instance, and management of database storage structures. Students administered users and database security by creating roles, profiles, and DBA and user accounts. Students occasionally encountered installation errors, usually the result of invalid parameter entries requested by the Oracle Installer. In these cases the VMware "revert" capability allowed them to quickly recover from a botched installation by restoring the pre-installation operating system image.

The data warehouse module was based upon an Oracle Warehouse Builder 10g Reviewer's Guide distributed at an Oracle Development Tools User Group (ODTUG) conference. Students attempted to install the Apache HTTP Server 9.0.4.0.0, Oracle Workflow Server 2.6.3, and Oracle Warehouse Builder 10.1 (OWB), in conjunction with the previously installed Oracle Database Server 10.2.0.1. However Oracle Workflow Server 9.0.4.0.0 turned out to be incompatible with the Oracle Database Server 10.2.0.1. Thus, the Oracle 10.2 Database Server needed to be replaced by an Oracle 10.1 version. Normally this would require an extensive procedure to de-install the HTTP Server and the 10.2 Database Server. Instead, students were able to quickly revert to a pre-installation VM image, then install compatible versions of the Oracle software. Even with compatible software versions, a couple of installs aborted while executing a script that configures the OWB repository. These students were still able to undertake a subsequent data warehouse design exercise by copying a pre-installed OWB version within minutes, instead of struggling endlessly with the software installation.

A final problem occurred during data warehouse design exercises that utilized an Oracle sample data warehouse. The pre-configured warehouse contained errors. These needed to be corrected before students could complete the tutorial exercises. Although OWB allows specifications to be exported from the instructor's repository and imported into the student's, copying a VMware snapshot simplified the distribution of working OWB definitions.

4. SUMMARY AND CONCLUSIONS

Several benefits of VMware usage accrued within each course and for the lab as a whole. First, students were able to exercise administrative privileges within VMs while logged into the host operating system as ordinary users. Lab machines were never compromised and always available for the next class or general-purpose use. Second, the ability to restore a previous configuration via the VMware revert command or by copying a previously saved image enabled students to quickly recover from failures. Third, the ability to concurrently execute multiple VMs enabled many complex exercises, demonstrations, and side-by-side comparisons. Finally, and possibly most importantly, VMs enabled instructors to increase course content and use a greater number of more complex exercises, thus providing a richer student learning experience.

VMware usage was not without its challenges. First, consistent network configuration and configuration of custom networks was problematic because it required administrative privileges on the host operating system. Instructors need to ascertain their need for consistent or custom network configurations in advance and implement appropriate manual or automated procedures to create them. Second, adequate backup of student VM images was problematic due to their number and size. Instructors and lab administrators need to ensure sufficient network bandwidth, storage capacity, and free time to complete needed backups.

REFERENCES

[1] Adair, R.J., Bayles, R.U., Comeau, L.W. & Creasy, R.J. A Virtual Machine for the 360/40. IBM Corp., Cambridge Scientific Center, Report No. 320-2007, (May 1966).

[2] Adams, J.C. & Laverell, W.D. Configuring a Multi-Course Lab for System-Level Projects. *SIGCSE Bulletin*, 37, 1 (2005), 525-529.

[3] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., & Warfield, A. Xen and the Art of Virtualization. *Proc. of the 19th ACM Symp. On Operating System Principles*, Bolton Landing, NY, (Oct. 2003), 164-177.

[4] Berthaud, M., Jacolin, M., Potin, Ph. & Savary, H. Coupling Virtual Machines and System Construction. *Proc. of the Workshop on Virtual Computing Systems*, Cambridge, MA (1973), 1-14.

[5] Hill, J.M.D., Carver, C.A., Humphries, J.W. & Pooch, U.W. Using an Isolated Network Laboratory to Teach Advanced Networks & Security, *SIGCSE Bulletin*, 33, 1 (2001), 36-40.

[6] Kneale, B., DeHorta, A.Y. & Box, I. VELNET (Virtual Environment for Learning Networking), *6th Australasian Computing Education Conf. (ACE2004)*, 161-168.

[7] Madnick, S.E. & Donovan, J.J. Application and Analysis of the Virtual Machine Approach to Information System Security and Isolation. *Proc. of the Workshop on Virtual Computing Systems*, Cambridge, MA (1973), 210-224.

[8] Mattord, H.J. & Whitman, M.E. Planning, Building and Operating The Information Security and Assurance Laboratory. *InfoSecCD Conference '04*, (Oct. 2004), 8-14.

[9] Micco, M. & Rossman, H. Building a Cyberwar Lab: Lessons Learned Teaching cybersecurity principles to undergraduates. *SIGCSE Bulletin*, 34, 1 (2002), 23-27.

[10] Nieh, J. & Vaill, C. Experiences Teaching Operating Systems Using Virtual Platforms and Linux. *SIGCSE Bulletin*, 37, 1 (2005), 520-524.

[11] Parmelee, R.P., Peterson, T.I., Tillman, C.C. & Hatfield, D.J. Virtual storage and virtual machine concepts, *IBM Systems J.*, 11, 2 (1972), 99-130.

[12] Rosenblum, M. The Reincarnation of Virtual Machines, *ACM Queue*, 2 (2004).

[13] Stockman, M. Creating Remotely Accessible "Virtual Networks" on a Single PC to Teach Computer Networking and Operating Systems. *CITC4'03*, (Oct. 2003), 67-71.

[14] Vollrath, A., & Jenkins, S. Using Virtual Machines for Teaching System Administration. *Journal of Computing Sciences in Colleges*, 20, 2 (2004), 287-292.

[15] *Workstation 5 User's Manual*. (2005). vmware, Inc.. Retrieved August 2, 2005, from http://www.vmware .com/pdf/ws5_manual.pdf.

[16] Wagner, P.J. & Wudi, J.M., Designing and Implementing a Cyberwar Laboratory Exercise for a Computer Security Course. *SIGCSE Bulletin*, 36, 1 (2004), 402-406.

[17] Yasinsac, A., Frazier, J., & Bogdanov M. Developing an Academic Security Laboratory. *6th National Colloquium for Information Systems Security Education*. Redmond, WA, June 4-6, (2002).