Using Virtual Machines to provide a secure Teaching Lab environment

Harry Bulbrook

bulbrookh@durhamtech.edu

Durham Technical Community College

1637 East Lawson Street

Durham, NC 27703

Using Virtual Machines to provide a secure Teaching Lab environment

Abstract

Teaching an Information Security curriculum requires special consideration of the problems that students and lab exercises may generate. Many security exercises such as penetration testing, injection attacks, session hijacking, and spoofing will cause numerous problems on production campus networks. This paper will present a method for isolating these problems into a virtual environment, allowing labs to progress in a secure and portable manner. A teaching classroom or lab can provide additional protection for real systems while still allowing a full range of attack and defense exercises by using virtual machines. Specific examples will be presented using VMWare products, but the techniques should be applicable to other virtual environments. The primary focus will be on network-oriented services, although host-based security evaluations will also be supported.

Introduction

Alarms at the university's IT security center light up – pagers go off, phone calls are made, network traffic is captured and analyzed. Penetration scans are being run on a number of critical infrastructure servers, and evidence shows that it is originating from on-campus. Patterns are tracked to a classroom where Professor Packetslinger is running his Computer Security class, and students are working on an assignment to evaluate system security. This event, while providing several interesting examples of the ethics of computer security, illustrates one of the major problems with teaching computer security: methods learned in the classroom can easily overstep boundaries and harm real production systems.

Providing a security curriculum can be challenging, given the problems of understanding and using tools that can be used to compromise systems. Historically, when dangerous tools are involved, the primary method used to ensure that those tools don't cause unintended problems is to use them far removed from potentially vulnerable systems. In the case of network security tools, this often meant creating a separate network, with attacker and target computers unattached to the larger network of the Internet. Besides the additional expense of those additional computers and the time required to setup and install a completely parallel system, such labs were disadvantaged by not being able to validly use the resources of the Internet or the campus network.

For example, while it may be easy to create a scenario where a server is probed by an attacking computer, any supplemental references or information would need to be provided manually. A student having difficulty with setting up the scenario might be well served with a quick Internet search for specific examples of how other attacks may be setup.

In contrast, a virtualized lab provides multiple benefits. The resources of lab computer systems can be utilized more effectively, multiple environments can be configured quickly and easily, and access to external resources can be provided without permitting attacks to those resources.

A caveat: creating a virtualized lab does not ensure complete protection. Just as a physically isolated lab can cause problems if it is inadvertently connected to the campus network, virtualized machines can have their configurations changed and cause issues. The configuration can be locked to a certain extent, but it depends on the configuration and the virtualization software involved. VMWare Player, for instance, prevents the user from changing the network configuration. Part of an effective student policy will dictate that exercises involving dangerous programs should be prevented from affecting the rest of the network.

A Physically Isolated Lab

The simplest approach is to have a physically separate network. The lab is set aside for security curriculum use, and all computers, switches, and other network devices are only interconnected within the room. Programs and files are brought into the classroom via read-only media such as CD, DVD, or write-protected flash drives. No data is brought out of the room. This has the advantage of being extremely easy to create, and very secure, as there is no access possible from the infected or attacking computers. However, the disadvantages of this environment are numerous. Primarily, the lack of Internet access means that every exercise must be thoroughly planned ahead of time, and student research of active attacks is limited. Additionally, since the Internet is not available, there is no capability for showing an active attack from a wild Internet worm such as SoBig or CodeRed without setting it up internally. Another problem is the possibility that a student will bring in a laptop or some kind of

unprotected writable medium to make their own copy of the tools in use. This device may be compromised or infected without the student's knowledge, and give a vector for causing problems in other labs or the existing network. This lack of access to the enormous security resource of the Internet is the main detriment to this type of environment.

A Virtual Network Lab

One of the main reasons to use VM's is their increased manageability. Instead of installing operating system software, application software, and configuring an existing machine, a previously configured virtual machine could be distributed. An exercise that requires several hours of setup (making it difficult to assign to a large class) can be configured once, then distributed. Specific exercises could be tuned for specific virtual machines, allowing for extremely focused topic demonstration (such as a buffer overflow), instead of being concerned about the configuration of one exercise conflicting or interfering with another. A huge benefit is that even if the entire virtual machine is formatted or destroyed, another environment can be deployed as simply as copying a file to the system. Additionally, if system resources are capable enough, several virtual machines can run simultaneously, connected through a virtual network connection. While not all attacks can be modeled using this environment (especially timing-based attacks), the ability to run a demonstration in a normal lab, and indeed even a student's home computer without risk to the computer itself makes this an excellent configuration options for security lab development. Often in the past, this type of easy switching between configurations was done with removable hard drives. While removable disks may still an option, wear and tear on the equipment and the time to reinstall a new operating system and environment are eliminated if VM's are used instead.

The software to provide this environment may be VMWare's family of products, Microsoft's VirtualPC and Virtual Server, and the Open Source QEMU, Plex86, Xen, and various others. VirtualPC may be attractive to those schools with a Microsoft software licensing agreement, as it is designed to work with Windows servers, but it has significant limitations, especially for network use. For instance, while VirtualPC allows up to 4 virtual network adapters installed into a virtual machine, they can only be assigned to either a real network driver on the host, a NAT provided by the host, or a completely private network. Virtual Server does allow assignments of virtual network adapters to shared virtual networks (again with a limit of 4 virtual network adapters per virtual machine), but those assigned virtual networks are not portable. That is, while they will work on that single host configuration, the virtual machines that make up the virtual network could not be moved to another host and run there in the same configuration without extensive reconfiguration. Microsoft states that virtual networks are not portable, and virtual network portability is not realistic.

QEMU and Xen have more usable features than Microsoft's products, but are only supported on Linux host computers, and are more difficult to configure and install. Once installed, the virtual machines actually have more flexibility with network configurations, as an unlimited number of virtual networks can be configured. The main issue with using these products is their lack of support for Microsoft guest operating systems. Windows is unsupported (though has reportedly worked) on QEMU, and will not be supported on Xen until the release of new processor virtualization technology from Intel and AMD. AMD's Pacifica and Intel's Vanderpool technology include on-chip support for virtualization instructions, much as the original 386 chip supported virtual 8086 chips. However, the lack of a Microsoft host platform together with unsupported and/or slow windows virtual machines leave this platform for use in

non-Microsoft environments only. A high level of Linux administration skill will be necessary to support these Virtual machine environments.

All of these options have greater limitations in compatibility and speed than VMWare's products, so VMWare software such as VMWare Player, Workstation, Server, or ESX server is preferred unless the greatest of networking flexibility is required. In a lab environment, the highest performance is not required, so the expensive ESX server is usually not necessary. Player, Workstation, and Server all have similar technical limitations regarding the handling of network interfaces. Only 3 or 4 network adapters per virtual machine are allowed, there is a limit of 10 virtual switches to connect your virtual machines, and 32 virtual machines can be connected to each virtual switch. Effectively, you can have 10 separate networks that are completely isolated from one another, although they can be connected with one another via a virtual machine with two virtual adapters that is bridging or routing between them.

A number of other limitations on the products are notable. VMWare Player and VMWare Server are both available as a free download (though they are not Open Source). While VMWare Server is currently in beta (and according to its license agreement not allowed to be used in a production environment until fully released), it is expected to be released in Q2 2006. VMWare Player is a "run only" environment – creation of Virtual Machines using this product is unsupported, and no changing of the virtual hardware is allowed. VMWare Server does allow creation of virtual machines, and supports most of the features of VMWare Workstation, but will only run on server-class operating systems (including Windows 2003 server, Windows 2000 server, and Linux server host OSes.)

Even with these limitations, the VMWare family provides the easiest framework to manage. Free availability of the software means that the Player can be installed onto computers

in open campus labs and students' home computers without licensing issues, and the portability of the VMWare virtual machines mean that the same virtual environment is seen wherever they are running. Additionally, remote access features built into the Server allow for effective remote access to involved labs. That is, you could configure a powerful server with many different virtual machines, and allow student access via a schedule to run remote labs. In this way, a single server with plenty of RAM and a fast processor can act as several computers, and send each virtual environment's display to a single remote computer. However, the problem of relying on this as a primary means of instructional support is that either multiple high-end servers need to be available to provide for the peak demand of an entire class of students using the VM's, or the resource must be rationed by means of scheduling or other division. Finally, many pre-configured virtual machines are available from the VMWare technology Network (VMTN) for drop-in testing for various environments, greatly reducing the amount of installation time required.

Specific Lab Exercises and Configurations

Firewall configuration and testing.

There are several open-source firewall products available (many Linux-based.) IPCop was forked from SmoothWall, and provides a complete and highly configurable firewall with a web interface and support for IPTables, web proxy, DHCP and DNS, and many other features. This is an excellent platform from which to illustrate many of the issues in properly constructing a firewall, and the VMTN has a pre-built appliance available at around 40MB. This appliance, along with a standard workstation image, will allow configuration and testing of a protected network, and demonstrate and verify the basics of packet filtering, traffic shaping, port forwarding, and general traffic restriction.

Setup

Ensure that the VMWare Server is installed on the host machine properly. Download the IPCop image from the VMTM (linked to vmwzrez.com), and start VMWare Server by double-clicking the .vmx file. Note that inside this virtual machine, there are two interfaces: eth0 will be considered facing the private network to secure, and eth1 will be facing the Internet (or the public, unsecured network.) By default, eth0 is bridged to your physical network adapter, and eth1 is connected to the NAT virtual network provided by VMWare. We will want to connect eth0 to a separate virtual switch, allowing connection to a separate virtual machine that will be protected by the IPCop firewall. With the VM not running, open the settings with Ctrl-D, then select each Ethernet virtual adapter in turn. Change Ethernet to Custom: vmnet2, and Ethernet 1 to NAT (or Custom: vmnet1).

Start the IPCop virtual machine, and check the IP addresses by logging in as root (password: vmwarez) at the command prompt then typing ifconfig. The IP address of eth0 (the protected network) will be 10.1.123.33, and the IP address of eth1 will depend on the configuration of VMWare, but should allow connection with the outside world.

Start a second guest VM, configured with its Ethernet interface connected to vmnet2 as above. The virtual adapter within this virtual machine will need to be configured with an IP address on network 10.1.0.0/16, such as 10.1.123.44/16. The default gateway and DNS server can be set to 10.1.123.33. Verify Internet connectivity in this VM by opening a web browser and connecting to a valid external website.

From the workstation guest VM, access the IPCop configuration page by visiting http://10.1.123.33:81 You will need to accept the presented SSL certificate, as IPCop will by default require an encrypted web browser connection, and the included SSL certificate is not yet

trusted by the guest VM. Once you have accepted the certificate, connect to the IPCop firewall by clicking on the “connect” button on the IPCop system homepage. The username should be admin and the password vmware.

Lab Scenario – Blocking access to specific sites

This first exercise will demonstrate selective filtering of website traffic. Verify that the guest VM can browse to the website http://www.msn.com. Then access the IPCop management interface, and select Advanced Proxy from the Services menu option. Enable the following settings: under Common settings, Enabled on Green, Transparent on Green; under URL filter, Enabled. Clock the save button at the bottom of the section. Next, choose URL Filter from the Services menu option. under Custom Blacklist, Blocked Domains, enter google.com in the available textbox, and check the Enable Custom Blacklist checkbox beneath it. Click the “Save and Restart” button and the bottom of the URL Filter settings section, and wait until the page refreshes. Then, verify that access from the guest VM to http://www.msn.com is no longer allowed by opening a web browser and connecting to the site. The standard msn page should be replaced by a large banner page indicating that access to the page has been denied. This block page can be customized to a certain extent, with images and specific references to information sources such as the campus acceptable use policy.

Lab Scenario – Ping tracing though firewall

This exercise will demonstrate how a firewall filters incoming traffic. For this exercise, the guest VM will need two pieces of software: a packet capture tool and a SSH client. PuTTY is a freely available client that will work well, and Ethereal is similarly available for packet capturing. This exercise captures traffic both on the external interface of the firewall, and the interface of the guest VM (which is connected to the internal interface of the firewall.)

Prepare the workstations to test by pinging the host workstation from the guest VM. The host workstation's IP address may be discovered in Windows by starting command prompt (Start, Run, cmd) and typing ipconfig. The IP address to ping is the one associated with the Ethernet vmnet adapter has been assigned to NAT, which by default is vmnet1. Assuming the IP address of the interface is 192.168.12.1, test connectivity by issuing a ping from the guest VM: ping 192.168.12.1. Demonstrate that the reverse is not true – if the guest VM has an IP address of 10.1.123.44, then ping 10.1.123.44 from the host command prompt will generate no response, and will offer a "Destination Host Unreachable".

On the VM guest, ssh into the IPCop firewall by accessing the IPCop eth0 address (10.1.123.33) and the non-standard ssh port number 222. In PuTTY, start the putty.exe file, type 10.1.123.33 into the Host Name section, and 222 in the port section. Assuming the IPCop firewall is still running and configured from the previous exercise, you will be prompted to accept an unknown ssh key. After accepting the key, you will be prompted for a login name. Use root as the login name, and when prompted enter vmwarez for the password.

Now start the packet capture software. On the guest VM, run Ethereal and from the capture menu, select interfaces. Choose the virtual interface available on that virtual machine, and select capture. Notice that you may not immediately start seeing traffic. While that capture is running, switch to the ssh session connecting to the IPCop firewall. Issue the command tcpdump –i eth1 to begin capturing traffic on the external interface of the IPCop firewall. At this point, you are capturing traffic on two networks, with two tools: tcpdump on the linux hosted IPCop firewall connected to the outside world, and Ethereal on the guest VM, connected only to the inside port of the IPCop firewall.

Once again, try to test connectivity by issuing a ping from the guest VM, and from the host machine to the guest VM. The results should be the same as before, but this time you will see traffic in the ssh connection, and in the results captured by Ethereal. Even though the packets are generated by the same event, they will have very different contents, and the IP address mangling done by the NAT function of the IPCop firewall will be evident.

Port Scanning and advanced probes.

A useful Linux distribution with plenty of security-related tools is Knoppix-STD. This Linux live-cd can be used in standard computer labs by booting to the CD, but any commands run will impact the network directly. Instead, a virtual network can be quickly setup to probe specific virtual machines and identify weaknesses in their configuration.

Setup

Ensure that the VMWare Server is installed on the host machine properly. Download the Linux Live-cd image from the VMTM (linked to vmwzrez.com), and download the .iso file for Knoppix-std. Place the .iso in the live-cd folder, and name it livecd.iso. Change the settings of this virtual machine so that the virtual Ethernet interface is connected to vmnet2. Ensure that your target guest VM also has its adapter set to vmnet2, and that no other VMs are configured for that interface, and start both VMs.

Verify connectivity between the two guest VMs by assigning a valid IP address to the Knoppix STD VM (for instance, 10.1.123.55). Do this by accessing the Start menu (Big K with gears), then KNOPPPIX, then Root Shell. Issue the command “ifconfig eth0 10.1.123.55”. Immediately, you should be able to ping your target guest VM: ping 10.1.123.44.

Lab Scenario – nmap and nessus

Scan the target for active ports by running nmap from a root shell with the defaults: “nmap 10.1.123.44” This should return the open scanned ports for your target guest VM. If you receive no report, or the report indicates that the host is unavailable, check to see if the software firewall is enabled in the target guest VM. Once you have verified that the target is up and has several ports open for probing, run nessus to discover any verified weakness in the configured security of the target.

Start nessus by selecting “Vulnerablility Assessment”, then Nessus from the Knoppix start menu. You will need to authenticate to the Knoppix STD locally running nessus server with a username of knoppix and a password of knoppix. Once authenticated, choose target selection, then enter the target VM of 10.1.123.44, and start scan. The scan will take some time, but when complete you have a vulnerability report for that target OS.

Conculsion

The benefit of running all of these exercises is twofold: none of the damaging or questionable traffic was generated on any production network, and all of the labs could be run not just from the lab, but from a properly configured platform in any location. Virtual Machines allow for the creation of simple files or groups of files that can be distributed with all the configuration necessary to demonstrate topics in a way that does not negatively impact the device or the network the device is running on.

References

* Alec Yasinsac, Jenny Frazier, and Marion Bogdonav, "Developing an Academic Security Laboratory", 6th National Colloquium for Information Systems Security Education 2002, June 3-7, 2002, Microsoft Headquarters, Redmon, Washington. http://www.cisse.info/history/CISSE%20J/2002/yasi.pdf

* Jason Kretzer, Charles E. Frank. Network security laboratories using SmoothWall. Journal of Computing Sciences in Colleges, Volume 21, Issue 1 (October 2005) http://portal.acm.org/ft_gateway.cfm?id=1088800&type=pdf&coll=GUIDE&dl=GUIDE&CFID=73915444&CFTOKEN=99357225

* Nieh, J. Leonard, O. C. EXAMINING VMWARE. Doctor Dobbs Journal 2000, VOL 25; PART 8, pages 70-79 http://www.ncl.cs.columbia.edu/publications/drdobbs2000.pdf

* Ji Hu, Dirk Cordel, Christoph Meinel. A Virtual Laboratory for IT Security Education. Proceedings of the Conference on Information Systems in E-Business and Egovernment (EMISA), Luxembourg, 6-8 Oct 2004, pp. 60-71 http://www.informatik.uni-trier.de/~meinel/papers/Trier-Emisa04-Hu.pdf

* Patricia Y. Logan. Crafting an Undergraduate Information Security Emphasis Within Information Technology. Journal of Information Systems Education, Vol 13(3) http://www.jise.appstate.edu/13/177.pdf

* Patricia Y. Logan, Allen Clarkson. Teaching students to hack: curriculum issues in information security. Technical Symposium on Computer Science Education, Proceedings of the 36th SIGCSE technical symposium on Computer science education (2005)

* T. Andrew Yang, Kwok-Bun Yue, Morris Liaw, George Collins, Jayaraman T. Venkatraman, Swati Achar, Karthik Sadasivam, Ping Chen. Design of a distributed computer security lab (sic: found as comptuer) Journal of Computing Sciences in Colleges archive. Volume 20, Issue 1 (October 2004)

Australian High Tech Crime Centre – Glossary. Terms from wikipedia (http://en.wikipedia.org) http://www.ahtcc.gov.au/glossary.aspx (visited April 17, 2006)

QEMU (11 April 2006). Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/wiki/QEMU (visited April 15, 2006)

QEMU http://fabrice.bellard.free.fr/qemu/ (visited April 15, 2006)

VMWare Player. http://www.vmware.com/products/player/ (visited April 15, 2006)

VMWare Server http://www.vmware.com/products/server/ (visited April 15, 2006)

VMTN Virtual Appliances http://www.vmware.com/vmtn/appliances/ (visited April 15, 2006)

Microsoft Virtual PC 2004. http://www.microsoft.com/windows/virtualpc/default.mspx (visited April 15, 2006)

Microsoft Virtual Server http://technet2.microsoft.com/WindowsServer/en/Library/bcc5e200-88af-4a64-963b-55f1efb251d11033.mspx (visited April 15, 2006)

IPCop http://www.ipcop.org/ (visited April 15, 2006)

IPCop Test Rig for VMWare Player http://www.vmwarez.com/2005/12/ipcop-test-rig.html (visited April 15, 2006)

PuTTY – ssh client for windows http://www.chiark.greenend.org.uk/~sgtatham/putty/ (visited April 15, 2006)

Ethereal – packet capture software http://www.ethereal.com (visited April 15, 2006)

Knoppix STD – Security Tools Distribution http://s-t-d.org/ (visited April 15, 2006)