Configuring a Lab to Support Community-Based Security Audit Projects

Andrew Holland
Computer Science
University of Wisconsin – Parkside
Kenosha, Wisconsin, USA
syclops@ehollands.net

Susan J. Lincke
Computer Science
University of Wisconsin – Parkside
Kenosha, Wisconsin, USA
susan@lincke.org

***Abstract*— The University of Wisconsin-Parkside cyber-security lab is designed to emphasize training in network security, with a focus on auditing. Audits require extensive security knowledge, including how to configure network equipment, analyze OS configurations, recognize hacks, and compare audit results on a variety of systems. The audits are taught using active learning labs, but implemented via community-based learning. This paper outlines the education plan and describes lab network configuration, including server and workstation implementations, and security software. Diverse operating systems, implemented with VMware, allow for students to simultaneously run, observe, and audit different types of systems. The lab is sufficiently isolated not to allow harm, but sufficiently available so that learning is not restricted, and audit tools can be ported to a customer site.**

Keywords-Security education, laboratory, security audit

I. INTRODUCTION

Computer and network security is an important topic as news of identity theft, computer worms, phishing scams, and hacking become regular features on TV, the newspaper, and in our personal lives. Many security courses focus on 'attack-defend' techniques. However, the instructor emphasizes defense techniques, partly because that is the main goal of security education, and partly because hacks are ephemeral: a problem today, fixed tomorrow. Instead, the University of Wisconsin-Parkside (UWP) lab is used to train students to audit real-world networks as community-based learning projects. Thus, the course emphasizes security techniques and the lab includes portable tools for use anywhere, including a variety of operating systems, software tools, books, and equipment. This paper describes the UWP security lab, which includes attack-defend capabilities, but focuses on security and auditing training.

Hacking is a popular security topic in security courses, because many believe that thinking like a hacker enables one to protect against them [1]. However, Logan and Clarkson argue that teaching hacking techniques is dangerous because it may lead to criminal behavior, takes course time away from important security techniques, can result in damage to the university, and may fail debates during the Security course approval process [2]. However, teaching security techniques can also lead to more stable and up-to-date labs, since hacks are ephemeral. UWP's emphasis on community-based learning encourages instructors to work with real companies and real projects. According to the 2006 CSI/FBI Computer Crime and Security Survey, auditing is used by over 80% of companies to ensure secure networks [3]. Performing security audits with small, local organizations is a good way to train students, but requires that students also be thoroughly trained in the area to be audited.

Because universities often emphasize hacking techniques, the first requirement for a security lab is that it is isolated from campus labs, university departments, and the Internet [4]. A cyber-security lab provides students a safe, secure system in order to perform tasks that would otherwise be deemed "unacceptable" on any commercial or university network. Network sniffing, virus analysis, cryptanalysis, system exploitation, "hacking," and network security are topics of study that network administrators normally ban from any commercial system. To control 'escaped' hacks, an 'isolated' university security lab allows software to be ported in, but disallows anything except paper to leave the lab [4-6]. Brigham Young Univ. implemented an isolated 'sandbox' lab for student-directed security and hacking projects, and a prototype network with De-Militarized Zone to exemplify a private company network [5]. In the sandbox lab, students implement security or hacking projects. Univ. of Idaho has recently implemented a general purpose security lab that is used to teach attack-defend techniques [6]. The features of their lab are that it is multi-use, modular, isolated, and reconfigurable, with rapid transition: i.e., it has dual-boot capability with a number of operating systems. In the last two cases, the labs include one or two firewalls for project use.

The University of Wisconsin-Parkside (UWP) security education and lab differs because the emphasis in on teaching defense strategies to be ported to the real world. This is implemented via community-based learning with audit projects. Therefore, the lab must be set up to provide students the necessary skills to train and practice auditing on a variety of operating systems and networking equipment. Instead of being 'isolated' the lab must be 'portable' to the customer site.

Secondly, UWP emphasizes active-learning labs. All students participate in all labs simultaneously. Therefore, all audit tools are loaded on all nodes. Instead of having two firewalls, the lab has five security-enhanced routers, enabling 5

to 10 teams to work with their own router simultaneously. There are teacher facilities to facilitate group labs.

Thirdly, the lab is integrated with the University of Wisconsin-Milwaukee (UWM) lab. A secure communications link to the UWM cyber-security lab enables inter-campus collaboration, including creating/observing a real Virtual Private Network (VPN) in class, or playing "electronic capture-the-flag" [7] with the UWM's cyber security students.

The UWP security lab was paid for with a grant from National Science Foundation (NSF), as part of its Federal Cyber Service: Scholarship for Service grants, which encourage universities to teach cyber-security. UWP is part of a UW System Coalition that received financial assistance.

This paper discusses all aspects of creating the cyber-security lab. Section 2 outlines the goals of the UWP security education. Section 3 describes physical access restrictions to the lab. Section 4 and 5 describes the lab network configuration and server implementation, respectively. Section 6 describes the workstation implementation with its VMware and audit software. Section 7 discusses security implications. Sections 8 and 9 include Lessons Learned and a conclusion.

II. Requirements for the Lab

The Network Security course has an emphasis on security, and within security an emphasis on auditing. To perform an audit requires skills in network configuration and testing. The skills or concepts that the course covers related to network security include:

- Attack recognition: Recognize common attacks, such as spoofing, man-in-the-middle, (distributed) denial of service, etc.
- Access Control Lists: Configure and test routers and firewalls to filter packets correctly and efficiently, by dropping, passing, or protecting (via VPN) packets based upon their IP and/or port addresses, and state.
- Intrusion Detection/Prevention Systems and/or Proxies: Set and test rules to recognize and report attacks in a timely manner.
- Vulnerability Testing: Test all nodes (routers, servers, clients) to determine active applications, via scanning or other vulnerability test tools.
- Web Vulnerability Testing: Recognize backdoor entry via web page, using SQL injection, inserted comments divulging critical information, java script input, etc.
- Encryption Techniques: Understand techniques to protect the confidentiality, authenticity, integrity, and non-repudiation of data. These must be understood at a protocol and at least partially at a mathematics or algorithmic level, in order to select and implement the algorithm matching the organization's needs.
- Security Network Architecture: Organize border routers, firewalls, Intrusion Detection/Prevention Systems (IDS/IPS), and logging systems to protect networks in a layered manner.

Some audits involve system software. These skills include selecting, setting, and testing the following:

- Password Protection.
- Access Control: Minimize permissions for data and program access.
- Security Settings.
- Logs: Determine which logs to collect and how to handle log file overflows.

Both network and system audits require knowledge of internal operation, in order to recognize normal versus abnormal sequences. This leads to the following skills:

- Baselining: Copy a known good state of the processes, open files, network connections, to compare against future states, in order to recognize attacks.
- Protocol Analysis: Write Access Control Lists (ACLs) or IDS rules correctly and efficiently. Recognize normal from abnormal protocol sequences using sniffers.
- Audit Procedure: Prepare an Audit Plan and Audit Report.
- Continuous Learning: Research in-depth information by oneself. (A limitation on course time and the evolving nature of computer science/networking limits course expectations.)

The rules or ACLs that are implemented must be based on the business requirements as defined by recent legislation: e.g., Sarbanes-Oxley, FISMA, and HIPAA. Hence, additional skills include security evaluation (e.g., risk analysis), security planning (creating of sound policies), and incident response. Of these, incident response (including computer forensics) is an important lab capability.

Students perform audits at local organizations (usually small). These organizations use a variety of versions of operating systems. To prepare for an audit, the lab network must be easily configurable to any operating system. The lab must contain reference texts of various operating systems and network equipment.

The audit capability means that the test tools must be portable to customer premise networks. Therefore, a portable laptop computer is available with audit tools on it. Some tools must be run on the machine being audited. A CD with audit tools is also available for borrowing.

This section has defined the requirements of a security lab to include an audit capability for use in community-based learning. Additional uses of the lab include secure software development (web page and otherwise) and data communications training. The next sections describe the implementation of the lab.

III. Physical Access

The cyber-security lab facility is built in its own room, which is independently locked from the rest of the departmental resources. This is necessary since physical

attacks can defeat any security system much easier than remote attacks.

The lab consists of 6 student workstations, and one teacher workstation. The only difference between the student and teacher workstations is that the instructor's workstation is also connected to a projector so all the students can see that screen for demonstrations and lectures. Otherwise it can serve as another student workstation. Allowing two students per workstation, the room has capacity for 14 students.

A small lockable room attached to the side of the main room is used for storage and to house the network servers, data communications equipment, audit-related, loanable texts, and a portable laptop with audit tools. All the workstations have connections to the network. Each of the four groups of computers (three sets of 2 student workstations, and the teacher computer) have in addition, a Cisco 1841 network service router, and a Linksys 5 port switch. These devices are available for network communications training. Students can use the security-enhanced routers to create subnets and place workstations within our outside the subnets for audit or other purposes. (See Fig. 1)

IV. NETWORK

The lab is not entirely 'isolated', since it allows students to access web pages but not transfer any data outside the lab, except through a proxy. A previous implementation of the security lab network had no interface to the Internet, at the strong request of the system administration staff. The current lab configuration is approved by the university network administrator. Internet access is useful to download updates for security software (e.g. MBSA), perform web page accesses for various labs, and perform traceroutes, digs, and other network commands to demonstrate command responses to equipment not available in the lab or requiring multiple hops.

The network for the cyber-security lab consists of a Cisco 1841 network service router, and a Cisco catalyst 3750 switch. (See Figure 1.) The network cable from the Internet interfaces with this border router. The router acts as many things, a router, a Network Address Translation (NAT) box, a Virtual Private Network (VPN) endpoint, and a firewall. The routing system routes IP packets from the public network segment (connected to the UWP network core) and the internal network, allowing data to get from one segment to another. As it does this the NAT subsystem translates the public IP block (provided by the campus) to an internal 10.0.0.0 network address system. This class A address enables more extensive subnetting than will ever be needed, via additional lab routers. It also abstracts the lab from the campus, so that while we teach students how systems can be attacked/defended, they use separate IP addresses from the campuses servers. The NAT translation also makes it more difficult for unwanted packets to get into or out of the private network. For security reasons, the workstations can not talk outside the segment without going through the border router.

Another function of the router is to act as a firewall in both directions. Most firewalls protect the internal network from the outside environment, creating a small "zone of trust" that is easy to define. Our border router in addition protects the outside networks from our network. The rule set on the router blocks all communications from the workstations except what is needed to run the lab (ping, whois, and a few other minor protocols). The workstations get a connection to the internet through a proxy server running on the main lab server. This allows the proxy to filter what is going out onto the public network or internet and prevent extraneous packets from reaching the internet.

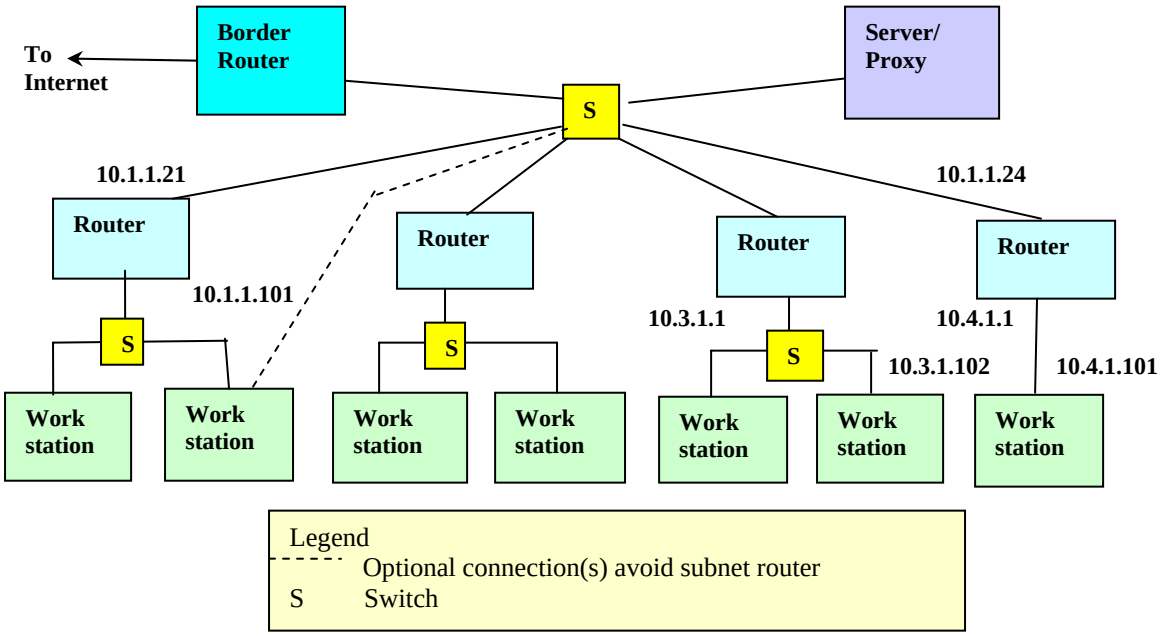

Figure 1. Security Lab Configuration

Finally the router acts as an endpoint for a VPN between us and UW-Milwaukee's Cyber Security lab. The router routes any data destined to 10.128+.0.0 to a corresponding router at the Milwaukee campus acting as the remote end point. This VPN allows our two campuses to collaborate and share information and resources easily and securely.

V. SERVER

The server for the cyber security lab is a custom built system with a RAID 5 data array, dual processors, three gigabytes of RAM, and a one gigabit network connection. This server is used to keep the lab running correctly and provide services to the workstations necessary for the lab maintenance and operation. This server is hardened to withstand most attacks that can be initiated against it, but is not designed to be a target of organized attacks. This server runs many programs, including, DHCP, DNS, Email, SMB, NFS, SSH, SFTP, Squid, TFTP, and CUPS. The interesting processes that require description are SMB, NFS, Squid, and TFTP. Server Message Block (SMB) and Network File System (NFS) are used to allow the Linux and Windows based systems to share files and act as a file store for the workstations. The two systems are both attached to the same physical directory on the server so a file saved via NFS on a Linux system is accessible via SMB on the Windows systems. Because the workstations may become corrupted as a result of daily use, we have created quick restore procedures that reformat the hard drives on the workstation. This means that any data stored on the workstation will be lost. The Server file store allows for students to persistently store files from session to session.

The Squid Process [8] is the proxy server allowing for the workstations to safely connect to the internet. Squid is configured to block most known virus signatures from getting on the internet. If the workstations become infected with a virus such as Code Red or Nimda, they will not be able to propagate the infection to other workstations on the Internet. This proxy is also configured to block a program called The Onion Router (Tor) [9]. Tor is a protocol that can be used to bypass the network security infrastructure of the lab, by allowing the workstations full access to the Internet via a SOCKS5 proxy tunnel. By blocking the Tor protocol we mitigate the risks of that system being used as an attack vector.

Trivial File Transfer Protocol (TFTP) is also active on the system, although it is a security vulnerability. TFTP allows routers to pickup and save their configuration files to a server as backup files.

The server also has a number of custom utilities installed on it that are used to keep the workstations running correctly. They include the UNIX shell scripts “startall” “shutdownall” “onall” and “backonall”. These four scripts are designed for the administrator or instructor to be able to maintain the workstations without having to work on each individual station. Startall uses the wake-on-lan protocol to properly start up all the workstations. (Figure 2 shows how specific Ethernet addresses are specified in the script. The script uses the freely-available etherwake software.) Shutdown uses Secure Shell (SSH) communications to initiate a shutdown of all the workstations. (Figure 3 demonstrates that specific IP addresses are generated during the shutdown process.) These two are used before and after a class to bring all workstations into a useful state. These scripts can also be setup on a schedule to bring all workstations online 15 minutes before a class. The Onall transmits a command to be run on all the workstations sequentially. Backonall performs the same task, except that Backonall performs the commands in the background and all of the commands run in parallel instead of serially across the workstations. This allows for things like software upgrades to be propagated, with out overloading a central server. All of these files are also used in a batch script to update the password files on all the workstations at night.

```
#!/bin/sh
for addr in “77:d6” “e1:2d” “76:eb” “”74:df”; do
        /bin/etherwake 00:11:43:bc:$(addr)
        /bin/etherwake 00:11:43:bc:$(addr)
        /bin/etherwake 00:11:43:bc:$(addr)
done
```

Figure 2. Startall script

```
#!/bin/sh
for machine in `seq 128 2 140` do
        # Next two lines should be one line…
        ssh –i /bin/identity-test/id_rsa
        root@10.1.1.$(machine) “init 0” &
done
```

Figure 3. Shutdown Script

VI. WORKSTATIONS

As with most security labs, the workstations are easily modifiable as well as easily reset to a baseline configuration. This allows for safely performing attacks that will corrupt data on a system with out taking the lab off-line for repairs afterward.

The workstations are Dell single processor workstations with one gigabyte of RAM. These workstations themselves are hardened to prevent attacks. The workstations are designed to be an interface to virtual machines provided by VMware 5. They are setup with their own IPs and own operating systems.

The operating system loaded on the host system is a Debian Sarge Linux kernel, because the UWP CS department uses Debian Linux on all department workstations and lab equipment. The kernel of the system is fairly easy to install except that the workstations we chose had an issue with the video card provided by Dell. In order to get the lab to run, we needed to use the X.org X windows server instead of the Xfree86 X windows server.

Host emulation software is loaded onto each workstation so that students can scan, audit, or attack a variety of popular operating systems. VMware [10-11] was selected as the machine virtualization software because it is the standard in the field. It supports multiple host and guest operating systems. The second reason that VMware was selected is it can do raw packet captures of its own virtual Ethernet address. This allows for utilities like Ethereal, Windump, and Snort to run inside the guest operating system without being overloaded by the host

OS's TCP data. Allowing this feature requires a small modification of the installation for VMware. After VMware is installed, we modified the VMware startup script to change the file permissions of the virtual Ethernet adapter on the host system. This change requires editing the /etc/rc.d/init.d/vmware file and adding a line saying "chmod 777 /dev/vmnet[0-9]". This gives all users full read write permissions on the vmnet device, allowing it to go into raw mode as long as the guest operating system allows.

Other software loaded onto the workstations includes OpenOffice 1.1, Screen, Apache web server, VNCviewer, and a few other miscellaneous tools. OpenOffice 1.1 was installed as a word processor so that students could work on write-ups for the "classwork" part of the lab. Screen is useful as a simple serial port VT100 terminal emulator for configuration of the Cisco routers. Apache web server in installed so that when using the "Black" system images for computer auditing, the student can see something running on the host computers. VNCviewer [11] is the last main utility installed on the host operating system. This tool enables a hacker or system administrator to command multiple workstations from one physical location. Students can try to find the tool during an incident response exercise. Another example use is that it allows a student to sit on one computer, and look at packet captures as three other computers attack his. Because the host OS, and guest OS each has its own Medium Access Control (MAC) address and separation at the lowest layers, Ethereal (or other utilities) will only see its own data. So when attacking the guest OS, running VNC on the host OS will not affect results inside of the guest OS.

A. Virtual machine

Virtual machines allow students to have administrator privileges without causing any damage. They are virtual workstations within VMware with snapshots created in an un-booted configuration. By hitting the "revert" button, the virtual machine reverts to a standard configuration no matter what has been done to it.

Multiple virtual machines have been created. Some of the primary Virtual Machines are Windows XP Black, Debian Sarge Black, Windows 2000 White, Windows 98 White, Novel 6.5 White, Windows NT White, and Windows 2003 server White. The Color associated with each of the images defines what the image is used for. A "White" image is supposed to be a clean image: it has no hacking or administration tools installed inside, so it is safe for any user to operate. White images are designed to be targets for both auditing (seeing what ports are open, how the firewalls work, what services are available, etc) and hacking because they all have "snapshots" so they can be reverted to a useful configuration in a matter of seconds. The "Black" images have administration/hacking tools installed. Black images are used to audit or attack White images. This allows the students to see how different kinds of workstations will respond to the same audit tool. Multiple "Black" images allow us to initiate attacks from multiple OS's, since most attacks are also version oriented to operating system/application version. We can run multiple virtual machines simultaneously, scan/audit/attack all of them, and see which succeed and fail.

1) Guest system "Debian SARGE Black"

One Guest operating system is Debian Sarge. It was installed from the same CDs as the host operating system. A simple, default installation was configured to our tastes. We again installed X.org X windows server onto the guest operating system just to keep things similar. Gnome is the window manager in both the Host and Linux guest operating systems, because it is the standard for the UWP CS department and students are already familiar with the layout and operation.

A variety of network software is loaded onto this virtual machine [11]. Ethereal, and TCPdump are both network sniffers, with Ethereal having a friendly user interface and TCPdump only being usable on the command line. Hping is used to send nearly arbitrary network packets to network hosts, allowing for use by many tools to send data, and to send attack packets. Dsniff which is a higher layer network protocol sniffer, is able to detect and correctly analyze many protocols. Instead of simply looking at the layer 2-4 data like most sniffers, Dsniff looks at the application layer as well, allowing users to read web traffic, watch telnet sessions, and see queries on most SQL server as well as a host of other protocol analysis.

Isic is the IP stack integrity checker [11]. It is used to create arbitrary packets for transmission onto the network. It allows users to craft almost any packet possible, including bad CRCs, setting and clearing flags, sending Acks without Syn's, etc. It is a great tool for teaching networking, or crafting single packet buffer overflows, or to implement most single packet attacks to be caught by a firewall, router, or IDS.

Some hacker tools also are available [11-12]. These tools can be used to demonstrate hacker attacks but also to attempt to recognize hacker tools, by comparison with a known, good OS baseline. Knark is a second generation root kit—a loadable kernel module for Linux systems designed to mask the presence of system activity. Its can be loaded into the kernel and used to hide information for the process table (shown by ps), network tables (netstat), and others. This allows a system to be running applications that the administrator is unaware of, including sniffers, and password grabbers. Back Orifice client is use to connect to the Back Orifice server that is loaded on the Windows XP black image. NetCat and CryptCat are useful utilities because they allow users to forward data from one system to another, either in a non-encrypted or encrypted tunnel, respectively. Both are extremely small programs, which can quickly be placed or hidden on compromised systems. While hackers use them as a back door into those systems, administrators may use them to monitor a hacked system as part of incident response.

Many audit tools are now mainly used by security analyzers, but could be used by hackers [11, 13]. Nessus is a network workstation analyzer. It scans networks to see what workstations are running, and can determine what software is loaded on those workstations. Nessus is also used to create baselines for workstations, as well as keep track of which workstations contain possibility compromised software. It is used by security administrators to find and close holes (or unneeded open applications) in their networks. Mozilla Firefox is installed inside of the window to be use to connect to Nessus, and a few other tools that only have web interfaces. By

installing Firefox inside of the VMware image the image can be used standalone with no connection to the network, or the host operating system.

Pure security tools (not for hackers) include RATS, Snort, and Tripwire [11]. RATS, the Rough Auditing Tool for Security, evaluates source code for applications, and looks for signatures for things like buffer overflows, bad links to other applications, possible race conditions, and other programming vulnerabilities. This very 'rough' tool will not detect all software flaws, but can be used to double check code. Snort is a network IDS that is free for download. Tripwire is a form of host-based IDS software. It keeps a log of all the files, file sizes, and checksums for those files in a protected file, and in memory. Then it checks the workstation against this file at a later time. This allows users to see if files are being modified by a program or by the administration. If properly configured, Tripwire can e-mail an administrator if key files are altered, or patterns of alteration start to form. This is a great tool to be installed on any server that does not have a lot of user interaction, and is included so that students can learn how to use it properly.

2) Guest system "Windows XP Professional Black"

The Guest operating system of Windows XP Black is our windows-based auditing image. The OS is Windows XP Professional with all the updates as of the creation of the image (this includes Service Pack 2). Firewall software is included in the service pack and automatically enabled as the virtual machine is powered up.

The Windows XP Professional Black is tailored with a suite of programs that focus on network auditing goals [11]. Tools installed include L0pht Crack 4, a popular Microsoft Security Accounts Management (SAM) password attack/auditing program. John the Ripper is a Brute-Force and Dictionary attacker that works on Linux passwd and shadow files, Windows SAM, LAN Manager (Lanman), and SMB packet captures.

We also included workstation auditing tools which can determine open applications on servers and hosts [11-12]. Nmap is designed to scan networks and report open ports and it also is a good utility to use for fingerprinting workstations. It has configuration options designed to camouflage its traffic into the mainstream of the network so that IDS, IPS, and firewall software do not detect its scans. Nessus scans a local network for known vulnerabilities and reports possible attack vectors. Tenable NeWT implements the Windows implementation of Nessus. WinFingerprint scans IP ranges looking for workstations and uses multiple techniques (including Microsoft commands) to determine what OS and applications the workstation is running. Sniffers also are used in combination with scanners to audit routers and firewalls to determine open ports: Ethereal and Windump are also included on the XP VMware.

Some tools audit via commands internal to the machine [11-12]. Microsoft Baseline Security Analyzer (MBSA) analyses workstation(s) security options/implementations for correct configuration, including detecting that all outstanding Microsoft System patches are installed. Dumpsec is a manual audit tool for dumping the Windows NT/2000/XP/2003 users, groups, and permissions, etc. PsTools is a popular package for baselining workstations as well as for autopsy/forensics work. ForensicToolkit is fairly self explanatory. Snort is a free network intrusion detection application.

The hacker tool, Back Orifice, is loaded as an example hacker tool. Back Orifice allows remote monitoring and control of a remote host system, and could be used as an attack tool/technique.

This list is not comprehensive, but lists the most important packages. Most of the packages were chosen from a compilation of sources [11-12, 14]. The most comprehensive list of security tools can be found (for free) on www.insecure.org/tools.html. This site includes most of the tools we are using, and a number that we are not. The set of tools we chose to use were selected for overall usefulness, no-cost, and ease of use.

3) White Image Software

All of the "White" images are just the operating system with all known patches applied as of the date of image creation. They are useful for practice audits, when customers have systems installed other than the ones mentioned previously. No other applications have been installed on them. This allows a clean base to be able to add other things to. Any installed software can change the behavior of a machine. Specific audits that require certain software packages may be installed on a per-needed basis, but should not be on the master white image.

VII. Legal Implications

A final important aspect of creating this lab was to work out an addendum to the normal school network Acceptable Use Policy (AUP). Most schools and universities have a standard AUP that all students are required to agree to in order to use the campuses network. The activities that occur in this lab violate some of the terms and conditions of these policies. For the UWP cyber security classes we have the students sign an additional addendum to these policies. This addendum states that the student will not attempt to attack, audit, or penetrate anything that they are not expressly allowed to, that they are only allowed to use the tools of the lab on computers in the lab, except during an authorized audit, and that they will not use this information for malicious purposes elsewhere in or off the campus network. We keep a signed copy of this document on file so that if any student is caught in violation, we have a signed contract with a little bit of teeth for prosecution if we choose to. Legal implications involving hacking are emphasized in lecture.

Students are expected to use security/hacking tools to audit parts of external organizations. Before auditing begins, students prepare an Audit Plan that lists all of the activities they will perform. The audit plan includes a signature page, to ensure that students have written permission from the organization to complete all audit steps. The audit plan also includes a detailed list of tests so that the 'customer' is aware of the specific tests that are performed. All results of audit tests are also forwarded to the customer so that they too can interpret the results. Customers are made aware in an introductory letter

that although the audit can help to close vulnerabilities, the audit cannot ensure that all security attacks will be thwarted.

Appropriate banners are placed on the lab workstations, routers, and server. These indicate that only students of CS classes are permitted to use the system.

VIII. Lessons Learned

Our use of the lab has helped us to refine the process. The most problematic audit test we executed includes scanning a customer firewall to determine open ports. This test must be run outside the university (or any other) firewall, as shown in Figure 4. Obtaining this access required reserving ports outside the university firewall addresses, reconfiguring the Internet Service Provider's DNS to forward to these ports, and obtaining access to the university computer center for the duration of the test. Simpler configurations include scanning via the customer's wireless network, or testing from the student's cable/ADSL connection, with advanced notice to the network provider.

COBIT, which is used to achieve Sarbanes-Oxley compliance, describes a model of stages of Control Reliability, for information technology or information systems departments [15]. To achieve Stage 3, the 'Defined Process', COBIT recommends extensive documentation. Since students are hired to maintain the lab, a written lab requirements description and update procedures have proven critical. Documentation is also critical for the other professors who share the lab.

Further information on the audit process is found at [16].

IX. Conclusion

This paper describes in detail the configuration of a security lab, where students learn auditing and other network security skills in an active learning environment. The lab is sufficiently isolated, but also portable to a customer site. It offers some access to the Internet as necessary for teaching, using a security-enhanced border router, NAT, and proxy. Test audits and preparations for customer-site audits require that the lab be partially portable and support multiple OS versions, with texts describing each. Active-learning requires that all software is available at all nodes but is easily reset to a baseline version – and this is accomplished via VMware. Finally, security software is described.

References

[1] J. Aycock and K. Barker, "Viruses 101", Proc. Of 36th SIGCSE Technical Symposium on Computer Science Education, Association for Computing Machinery, 2005, pp. 152-156.

[2] P. Y. Logan and A. Clarkson, "Teaching students to hack: curriculum issues in information security", Proc. 36th SIGCSE Tech. Symp. on Computer Science Education, Assoc. for Computing Machinery, 2005, pp. 157-161.

[3] L. A. Gordon, M. P. Loeb, W. Lucyshyn, and R. Richardson, 2006 CSI/FBI Computer Crime and Security Survey, Computer Security Institute, 2006.

[4] J. M. D. Hill, C. A. Carver, Jr., J. W. Humphreys, and U. W. Pooch. "Using an isolated network laboratory to teach advanced networks and security", Proc. SIGCSE Techn. Symp. On Computer Science Education, Association for Computing Machinery, 2001, pp. 36-40.

[5] G. W. Rombney, C. Higby, B. R. Stevenson, and N. Blackham, "A teaching prototype for educating IT security engineers in emerging environments", Proc. Fifth International Conf. on Information Technology Based Higher Education and Training, 2004, pp. 662-667.

[6] S. Caltagirone, P. Ortman, S. Melton, D. Manz, K. King, and P. Oman, "Design and implmenetation of a multi-use attack-defend computer security lab", Proc. 39th Hawaii International Conf. on System Sciences, vol. 9, 2006, pp. 220c-227c.

[7] P. J. Wagner and J. M. Wudi, "Designing and implementing a cyberwar laboratory exercise for a computer security course", Proc. 34th SIGCSE Technical Symposium on Computer Science Education, Association for Computing Machinery, 2004, pp. 402-406.

[8] "Squid Web Proxy Cache". www.squid-cache.org, Jan. 2006.

[9] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router", Proc. 13th USENIX Security Sumposium, Aug. 2004.

[10] "VMware Technology Network". www.vmware.com, Jan. 2006.

[11] M. Shema and B. C. Johnson, Anti-Hacker Toolkit, 2nd Ed., McGraw Hill, 2004.

[12] Auditing Networks, Perimeters, and Systems Hands-On Workbook, Audit 507 – Auditing Networks, Perimeters & Systems Course, SANS Institute, 2005.

[13] J. Clemens, "Knark: linux kernel subversion", www.sans.org/resources/idfaq/knark.php, SANS Institute, Jan. 2006.

[14] Insecure.org, "Top 75 security tools", www.insecure.org/tools.html, Jan. 2006.

[15] "IT Control Objectives for Sarbanes-Oxley, 2nd Ed., (Exposure Draft)", IT Governance Institute, http://www.isaca.org, April 30, 2006.

[16] S. J. Lincke, "Network Security Auditing as a Community-Based Learrning Project", Proc. 38th SIGCSE Tech. Symp. On Computer Science Education, 2007, Assoc. for Computing Machinery, pp. 476-481.

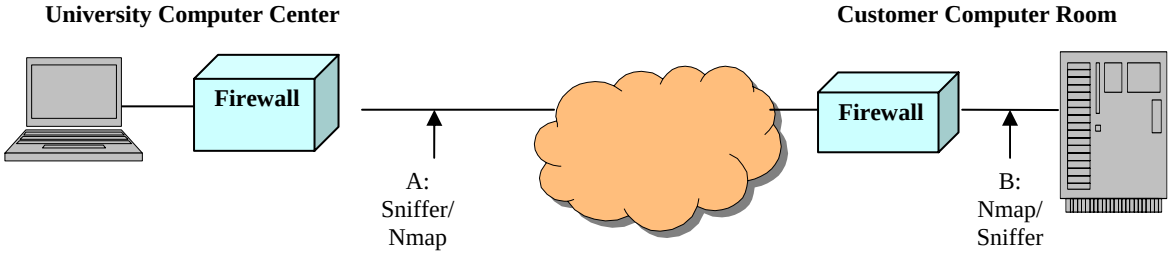

Figure 4. Determining Open Ports on a Firewall