2006-2516: OPEN SOURCE SOFTWARE AND LIVE LINUX CDS: ELEMENTS OF SUCCESSFUL LAB MODULES

Cheryl Willis, University of Houston

Ed Crowley, University of Houston

Ed Crowley, a former IS Director, developed the four course security specialization at UH's College of Technology. This is the only NSA/CNSS certified (4011/4014) program in Houston.

Mr. Crowley holds multiple INFOSEC certifications from the National Security Agency (NSA). He has also earned the usual vendor certifications from Cisco, CompTIA, and Microsoft. In addition to having held governmental security clearances, he is a graduate of the Military Police Academy.

Open Source Software and Live Linux CDs: Elements of Successful Lab Modules

Introduction

In addition to the intellectual effort, hands-on lab development can require substantial budget, labor, and laboratory resources. Often, commercial software can require a significant budget commitment. At the same time, system configuration and software installation can also require a significant labor commitment. In many colleges, incorporating laboratory activities into a class necessitates the dedication of a physical room to a single course section. In many environments, obtaining the necessary budget, labor, and room resources for hands-on lab modules is problematic.

This paper describes elements of a methodology for creating hands-on lab activities that minimize budget and support requirements. We have used this methodology to create hands-on activities in Operating System, Networking, and Security contexts. These activities provide students with concrete experience that the student can, in most cases, duplicate and extend outside of the laboratory environment. These structured activities are accompanied by questions and assignments that provoke reflection and stimulate critical thinking.

Our experience has been that Open Source Tools and Live CDs are critical elements in the design of our hands-on learning activities. These two elements produce a synergy that facilitates the creation of learning experiences that would normally require substantially greater budget and support resources.

Live CDs and Open Source Tools: Elements of Success

Four years ago, when we began to develop our security courses, we soon realized that we had neither the time nor the resources required for a conventional development process. We also knew that hands-on activities would be critical. As an analysis of the structure of hands-on activities, Jeanna Matthews[1] has observed:

> Nearly all (laboratory activities) involve three parts:
> 1. Configuring the hardware and software to prepare for a networking experiment (e.g. connecting a series of machines with routers, hubs of switches, configuring the machines with hard coded IP address, etc.)
> 2. Performing an experiment to generate specific network activity and capturing a trace of the activity.
> 3. Analyzing the trace to understand the subtleties of what occurred."

We have found that utilizing Live CDs and Open Source Software has enabled us to minimize the amount of time on Part 1, and concentrate our limited resources on Parts 2 and 3. That is, Live CDs and Open Source Tools offered us the elements that we needed for successful hands-on activities.

In general, the term Live CD refers to self configuring operating systems that may be freely distributed on an optical, or similar, medium. By definition, a Live CD can dynamically configure system hardware as well as boot and run from the CD. Many distributions come standard with an array of useful open source tools. Consequently, they require significantly less support than a conventional dedicated lab.

Since our lab activities are built around Live CDs and open source tools, students are free to use, copy, and distribute the tools they utilize in lab. As many of our students have a hands-on learning style, lab activities, like these, can significantly enhance a student's educational experience. Anecdotal student responses indicate that most students perceive these attributes as enhancing the value of their educational experience.

Because they are freely available, open source tools also mitigate lab development budget requirements. They do not however, limit the range of activities. There are many sophisticated and mature open source tools. Included among these tools are protocol analyzers, network scanners (mappers), vulnerability scanners, and intrusion detection systems as well as conventional utilities.

Live CDs and Open Source Tools: Advantages

Because Live CDs dynamically configure the hardware and can save their settings to a USB memory stick, or other transportable media, the utilization of Live CDs enables the same physical resources to host multiple class sections without increasing the need for dedicated lab support. As Live CDs are self configuring, their use eliminates the need for a support person to pre-configure the lab systems. It also eliminates the time required to set up the physical resources for the lab.

In our undergraduate information systems technology program, students work primarily in the Windows environment. Yet:

> Many advanced TCP/IP assessment utilities are available only for Unix based systems such as Linux, so you will often find that a competent security consultant uses a variety of tools under different operating systems to assess and successfully penetrate a network.[2]

Live CDs provide an expedient way for our students to gain experience with Linux based security tools. Our students have responded very positively to the utilization of open source tools and Live CDs. Anecdotally, many students report a very high perceived value added to the classes in which Live CDs are utilized.

All of the Linux based security tools used in our activities are Open Sourced. These tools are highly developed and widely utilized in the field. For example, a prominent survey reported that 9 of the top 10 security tools are available as Open Source. Many of these tools are available for both Linux and Windows.[3]

Open Source Tools

Many quality open source tools are available for both Linux and Windows platforms. For example, a May '03, survey showed that of the top ten tools, seven are available for both Linux and Windows, two for Linux alone, and one for Windows alone.[3] Nine of the tools are open source. One is commercial with a limited free edition. Specific tools are shown in Table 1-1. Several of these tools are included on the Knoppix Live CD.

Tool	Open	O/S
Nessus	Yes	L
Ethereal	Yes	L/W
Snort	Yes	L/W
Netcat	Yes	L/W
TCPDump / WinDump	Yes	L/W
Hping2	Yes	Linux
DSniff	Yes	L/W
GFI LANguard	Commercial	Windows
Ettercap	Yes	L/W
Whisker/Libwhisker	Yes	L/W

Table 1-1 Open Source Security Tool Survey Ranking

There are a wide variety of Open Source Tools. We have used many of them in class to illustrate important conceptual issues. Table 1-2 provides a brief description of some of the Open Source tools that we have employed.

Tool	Brief Description
Nessus	Nessus automates the discovery of known, local and remote, security vulnerabilities. It identifies vulnerabilities by the CVE standard number. Nessus utilizes a client/server architecture with a Linux/Unix compatible server module. Both Windows and Unix clients are available. Nessus versions, prior to V3, are available for distribution under the GPL.[4]
Ethereal	Ethereal a GUI based network protocol analyzer enables you capture and interactively browse packet data from a live network. It also enables you to analyze a previously saved packet capture. You can view both detailed information about a packet and summary information concerning network traffic. A sophisticated filter language allows you to focus on particular issues.
Snort	Snort, a lightweight network intrusion prevention/detection system, utilizes a rule-driven language to perform real-time traffic analysis and packet logging on IP networks. Snort can detect and log attacks and probes, including buffer overflows, stealth port scans, CGI attacks, SMB probes, OS fingerprinting attempts... Open Source GUIs are available. Snort binaries are available for Linux, Windows, and OSX.
Netcat	Netcat can read and/or write data across network connections, using TCP or UDP. By design, it functions as a "back-end" tool that can be used directly or driven by other programs and scripts. Both Linux and Windows Netcat versions are available under the GNU General Public License (GPL).
TCP/ WinDump	TCPdump is a promiscuous mode sniffer with a command line interface. It allows the user to intercept and display TCP/IP and other packets being

	transmitted or received over a local network. WinDump is a tcpdump port to Windows.
Hping2	In a manner similar to ping, Hping2 sends custom ICMP, UDP, and TCP packets and displays target responses. Hping2 can implement packet fragmentation, spoofed port scanning, and arbitrary packet size. Useful for testing firewall rules and network performance.
DSniff	DSniff is a suite of network auditing and penetration-testing tools. These tools passively monitor a local network for interesting data such as passwords or e-mails or similar. Suite tools arpspoof, dnsspoof, and macof facilitate the interception of network traffic normally unavailable to an attacker (e.g, due to layer-2 switching).
OpenSSL	An SSL/TLS encryption library that enables students to create and utilize symmetric keys and asymmetric key pairs. It facilitates experience with a variety of cryptographic algorithms and methodologies..

Table 1-2 Open Source Tools and Brief Descriptions.

Open Source Security Tools have distinct advantages and risks. Table 1-3 lists several of these that were presented at the 2003 Educause Annual Conference.[2]

Advantage	Risk
Low cost	Limited support
Highly flexible	High technical knowledge requirement
Reliable	Selection can be difficult
Reduced operating expenses	Management and staff reluctance
Reduced deployment risks	Significant FUD

Table 1-3 Open Source Advantages and Risks

Live CDs

By definition, a Live CD can boot, self configure, and run from a CD. Certain Live CDs, such as Puppy Linux,[5] even utilize rewritable compact disks that enable students to save their work to the disk itself. It is important to note that a Live CD does not necessarily require a system that contains a hard drive. Though, certain Live CD activities, such as remastering, may require a hard drive.

Even though Live CDs don't require a hard drive, they still enable students to work with Linux utilities, install applications, save data, and save configurations. While students may save to a hard drive, they may also utilize other storage devices. For example, the Knoppix Live CD will automatically recognize, mount, and utilize a USB memory stick. These capabilities simultaneously empower students and minimize support requirements.

LiveCDs may be freely distributed through a network or on optical mediums such as a CDs or DVDs. For example, the current version of Knoppix is available as either a CD or a DVD. In addition to being free to duplicate and distribute, students may also create customized Live CDs.

Many different Live CDs are currently available on the Internet.[6] Much like conventional Linux distributions, different Live CDs package different software collections for different users. That is, you will find different applications, or groups of applications, on different Live CDs. Each Live CD will have a configuration and an interface designed to meet the needs of their intended audience. Table 1-4 describes four Live CDs that we have utilized with our classes.

Live CD	Description
Knoppix	Based upon Debian, Knoppix is among the oldest and best documented Live CDs. It provides a substantial collection of both developer and office focused tools. It uses KDE for the default interface and is updated frequently.[7]
Auditor	Based upon Knoppix, Auditor focuses on general security with a particular emphasis on penetration testing. It contains a wide collection of sniffing, scanning, cracking, and auditing tools. It uses KDE for the default interface.
Slax	Based upon Slackware, Slax has a goal of providing a wide collection of useful software, while keeping the cd's image small enough to be written to a 185 MB CD. It uses KDE for the default interface and can be expanded through the addition of modules.
Damn Small Linux (DSL)	DSL, originally based on Model_K (a hack down of Knoppix), contains a complete desktop that only requires 50 megabytes of storage space. It uses the FluxBox GUI. By design, it is to be easily customized. It has minimal hardware requirements and supports older hardware well.

Table 1-4 Live CD Distributions

While we work with a variety of Live CDs, Knoppix is our primary Live CD. It is a very versatile, relatively well documented, system. In Klaus Knopper's words, Knoppix was designed to "…be customized as a rescue system, security scanner or platform for presentations and demos, …."[8]

Customizing or remastering Knoppix facilitates the addition of new tools as well as the addition of new support information. Support information may include lab assignments and class modules as well as presentations. Now let's look at what a sample activity might look like.

A Sample Introductory Activity

Here is an overview of a sample activity that can be implemented in most existing computer lab facilities. This activity requires no prelab setup and will not impact an existing hard drive configuration. At the system level, the only requirement is access to a computer with a BIOS that can be set to boot from CD (or DVD). At the network level, the only requirement is Internet Access. (Note should a DHCP server not be available, then an additional step of IP configuration would be required.)

For the sample lab activity, no extra work is required to complete step one. The second part of this example activity requires the student to work through seven steps. During these steps, the students:

1. Check their systems to make sure that they can boot from the CD (or DVD).
2. Place the Live CD in the system and turn it on.
3. Start capturing packets with Ethereal.
4. Start Firefox and browse to Ethereal.com.
5. Navigate to the downloads page and
6. Download the slammer.pcap file.
7. Stop the packet capture

Note that the slammer.pcap file contains a sample packet from the SQLSlammer virus. After a bit of manipulation, the student's console resembles Figure 1.

Note
Some schools may have security policies that do not allow students to capture packets. In those environments, the instructor should first capture the packets and then make them available to the students.

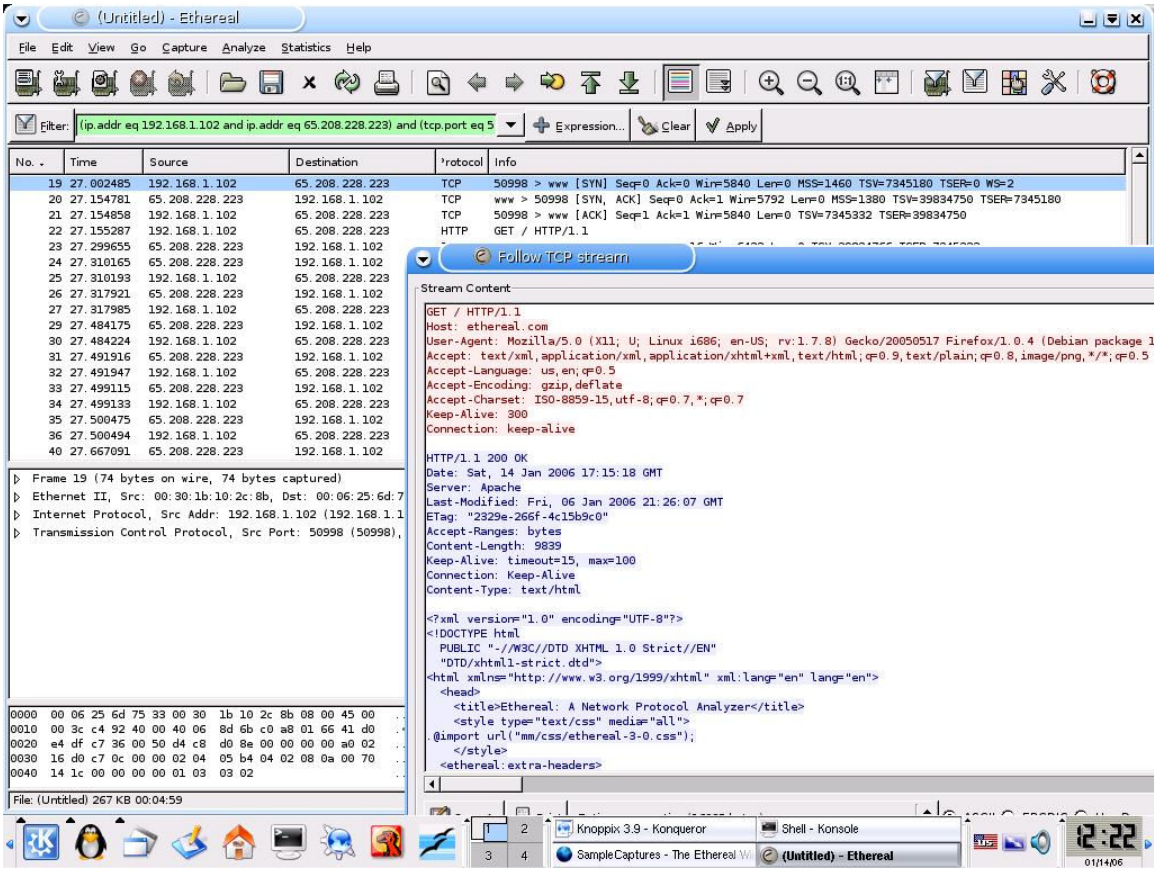

Figure 1 Sample student console snapshot

For part three, where most of the student learning occurs many different analysis activities are possible.[1] Depending on the class subject and the sophistication of the students, several different networking or security analysis activities are possible. Since each frame can be decomposed into Application, Transport, Internetwork, or Network Interface levels many different analysis topics are possible. Table 1-5 presents a few potential analysis topics divided by network layer.

Level	Potential Analysis Topics
Application	The TCP/IP stream can be saved. The resulting web page can be viewed with a browser.
	The server banner can be read. The as the web server software (Apache) can be identified.
Transport	The three way hand shake is visible. Sending and receiving port numbers are available. Packet sequence numbers are available.
Internetwork	IP addresses and headers are available
Network Interface	Mac addresses are available. Finally, the frame header and data portions are available in both hex and ASCII formats.
Security	The SQLSlammer packet could be read and analyzed

Table 1-5 TCP/IP Levels – Potential Analysis Topics

Activities are followed by a set of questions that students answer, in part, by examining the trace using Ethereal. Other questions may require students to research the TCP/IP or other standards. And still other questions may require critical thinking. Space doesn't permit the presentation of the complete set of questions but some of these have been previously published.[9]

Other Activities

Many of the activities that we utilize are Knoppix based. The following table lists selected objectives from activities utilized in a recent data communications class. Typically the students receive a lecture and a demonstration prior to the actual hands-on activity. During the activity they have to gather and analyze information. After the activity, they are required to answer questions for reflection.

Sample Activities Objectives

Activity	Selected Objectives
Into/Setup	Define Live CD List Knoppix Boot Requirements Understand Knoppix Boot Sequence Understand and utilize Knoppix Cheat Codes Boot Knoppix into RAM Define KDE graphical Environment Define and use the Linux Console.
Orientation	Boot Knoppix List locations from where Knoppix may be downloaded Utilize KDE Menu System Use Kinfo to ascertain system resources
Advanced Orientation	List major features of the Knoppix Interface. Identify your Ethernet Configuration(s). Use TCP/IP utilities to identify your network connections. Use basic TCP/IP utilities to footprint an organization
Getting Help	Use the Linux Man pages to obtain help. Explain the GNU General Public License.

	Explain two different levels of Linux User Accounts Compare and contrast the superuser account with a conventional user account.
TCP/IP Utilities	Utilize ping to troubleshoot IP connections. Use TCP/IP utilities to identify the MAC address of a remote computer. Identify the brand of a remote machine on your local network. Articulate and explain the name resolution services used on the Internet. Explain how name services can be used to help footprint an organization. Construct a table that shows how the utilities used here correspond to the layers of the TCP/IP protocol.
Ethereal Introduction	Start and run a Protocol Analyzer. Identify and explain the purpose of the four different Ethereal interface areas. Use Ethereal filtering on a packet capture. In an Ethereal capture, identify the TCP/IP three way handshake.
Footprinting, Enumeration, and Detection	Explain system and network baselining. Analyze protocol analyzer output to identify normal and abnormal network traffic. Utilize Ethereal filters to isolate interesting network traffic. Utilize TCP/IP utilities to remotely identify systems logical and physical addresses. Determine who has registered a particular IP address or address range. Interrogate name servers to gather relevant information. Identify live systems on a network segment. Differentiate between different types of NMAP scans, such as stealth and connect scans. Define a Denial of Service (DoS) attack. Demonstrate how a tool such as NMAP can be utilized to conduct a DoS.

Table 1-6 Sample Activity Objectives

Conclusions

The use of Open Source Tools and Live CDs has been empowering for both faculty and students. Their use has freed the faculty from the time lag imposed in the university purchasing cycle as well as the limits of the software budget. Their use has also freed the faculty from time spent on lab preparation. In addition, their use has also freed the students to copy the lab software and utilize it at home or in other environments.

Students have responded well to these activities. For example, one student presented his work a at a regional security conference. After the presentation, he distributed the customized Knoppix CD containing his presentation to conference attendees.

Open source tools also provide students with an opportunity to demonstrate critical thinking skills. For example, in one class project, students analyze a particular security issue, define a specific problem, survey the available tools, and select an appropriate security tool. The project includes both a conceptual problem/solution analysis and a hands-on class demonstration. In each of their courses, our students develop and utilize these skills in a structured project that utilizes open source software and Live CDs.

Future Work

Future activities may include group projects with the goal of creating custom enterprise security toolkits. These toolkits will likely contain a diverse collection of Open Source Tools. Another type of group project would be for each class to create a custom Live CD that contains each student's semester project. This Project would have a goal of creating the Class Project Live CD in real time while the students are making their project presentations.

References

1 Matthews, Jeanna, Hands-on Approach to Teaching Computer Networking Using Packet Traces, Conference on Information Technology Education, SIGITE, Newark, N.J., 2005.

2 Marten, R., and Myers, C., Selection, Deployment, and Management of Open-Source Security Tools, EDUCAUSE Annual Conference 2003, Retrieved June 05 from: http://www.guardian.maricopa.edu/presentations/educause-2003/siframes.html.

3 Fyodor, "Top 75 Security Tools", Retrieved March, 2005 from: http://www.insecure.org/tools.html.

4 McNab. Network Security Assessment, Sebastopol, CA: O'Reilly Media, Inc., 2004.

5 Puppy Linux, Retrieved June, 2005 from: http://www.guardian.maricopa.edu/presentations/educause-2003/siframes.html.

6 "The LiveCD List", Retrieved March, 2005 from: http://www.frozentech.com/content/livecd.php.

7 Granneman Scott ,Live CD paradise, The Register, 7th May 2005, Retrieved June 05 From: http://www.theregister.co.uk/2005/05/07/live_cd_paradise/.

8 Knopper, Klaus, Building a Self-Contained Auto-Configuring Linux System on an ISO9660 Filesystem, Retrieved Jan 2006 from: http://public.planetmirror.com/pub/knoppix/knoppix-vortrag-als2000/?fl=

9 Crowley, E., Information System Security Curricula Development, Conference on Information Technology Education, SIGITE, West Lafayette, IN, 2003.