

Georgia Tech “Hands On” Network Security Laboratory

Randal T. Abler, Didier Contis, Julian Grizzard, and Henry L. Owen*
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250 USA

*Corresponding Author:

Henry Owen
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0250 USA
Email: henry.owen@ece.gatech.edu
Phone: 404-894-4126
Fax: 404-894-9959

Abstract

An undergraduate internetwork security-teaching laboratory, which includes both defensive and offensive security laboratory experimentation, is described. This laboratory is oriented toward an introductory internetworking security class for students that have already had an introductory internetworking class. This laboratory is intended to complement more theoretical network security classes and to spark student interest in taking more network security classes. The laboratory is unique in that it uses an isolated laboratory network that attempts to represent the Internet as closely as possible by including a corporate network component, a university component, a “good” Internet service provider, and a “bad” Internet service provider. The use of virtual networking technology allows the physical network topology to be electronically reconfigured into different logical topologies. All of the laboratory assignments are available on the Internet for general community use and modification.

I Introduction

In order to complement the numerous theoretical security classes that exist, we determined the need for a “hands on” oriented laboratory based class that allows students to be exposed to the real world challenges of network security. A common complaint from students who have taken theoretical network security classes was that there did not appear to be any way to legally and ethically obtain practical experience with network security. After a search for textbook or online “hands on” network security laboratory materials, we realized that there did not appear to be an existing set of self contained materials that we could utilize to meet a “hands on” type learning experience. Thus, we set out to create this laboratory and the associated laboratory materials.

The goals of this network security laboratory include exposing students to both defensive mechanisms as well as offensive mechanisms used by the opposition. We do this so that a better understanding of why things are happening in the network security arena may be gained. The laboratory was not intended to just be a “hacker festival” so that students could learn hacking techniques; instead, the intent was to allow both defensive and offensive strategies to be understood and explored. We believe that better protection mechanisms and strategies may be created and employed when there is a full understanding of how attacks are created and how they work. This laboratory is intended to spark student interest in network security so that more traditional theoretical network security classes will be taken beyond the introductory course. The format of this laboratory is one hour of lecture and approximately six hours of laboratory exercises per week.

II The Network

We initially examined the effectiveness and the popularity of a “hands on” internetworking class [1] to determine desirable characteristics of a new laboratory based network security class. Student reviews of this existing laboratory based class indicated that it was a good initial model for our new efforts. Results from this existing class indicated that we needed to make the new network security laboratory network as

realistic, large scale, and interactive as possible. We determined that a structure that contained an internet backbone with distribution routers, a corporate network with excellent security practices implemented, a university network with a more open network but with some access control, an internet service provider with good security practices, and a second internet service provider with no security practices would be the best representation of the Internet for the purposes of this laboratory. This laboratory network architecture represents a very difficult target to exploit (the company network), a network with moderate difficulty to exploit (the good internet service provider), a relatively easy target (a university), and a very easy target (an internet service provider with no security). Thus, students could in theory learn how very easy targets are exploited and work their way up the complexity chain as their understanding of exploits and techniques used by hackers increased. In addition, students may implement good security policies in portions of the network and see the effectiveness of such policies.

The student laboratory network is physically isolated from the Internet so that exploits and information assurance laboratory assignments do not have the potential to escape and proliferate outside the student laboratory. However, it is possible to reconfigure the lab from the Internet through an administrative interface, which is shown in Figure 1. This capability allows multiple instructors and multiple teaching assistants to reconfigure the lab set up remotely. The laboratory network configurations are stored on a Dell Power Edge 650 server. This server also acts as a gateway between Internet users that desire to configure the network and the network itself. A remote user logs onto this gateway server. One network interface of the server connects to the Internet and is how the gateway is accessed. A second network interface is connected to the Ethernet port of a Digi Console CM32 console port manager. The Digi Console 32 allows one to connect to any one of 32 console ports. All router, Intrusion Detection System, firewall, Virtual Private Network Devices, and switch console ports are connected to the Digi Console CM32 and can be configured through it. The Linux gateway server is used to store the network configurations. There is no routable path between the laboratory network itself and the Internet. This means that even the event the Linux gateway server is attacked and

compromised; it is not possible to use the network security laboratory to launch network attacks to the Internet.

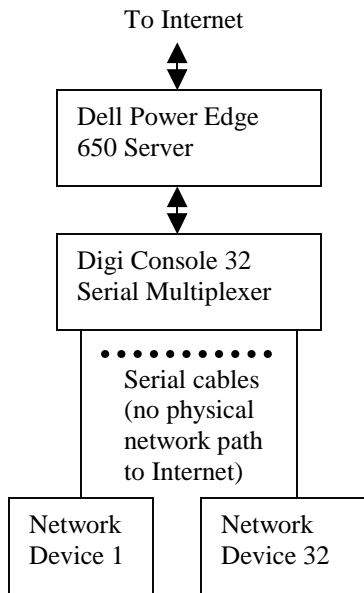


Figure 1 Administrative Interface

The laboratory network topology is shown in Figure 2. The four autonomous systems, an “enterprise”, a “good ISP”, a “university”, and a “bad ISP” are federated by a fifth autonomous system. This fifth autonomous system represents an “Internet backbone” consisting of two “Tier 1” backbone providers. The first Tier 1 provider consists of one Cisco 2621XM router and a virtual router on a Cisco Catalyst 3550. The second Tier 1 backbone provider consists of one Cisco 2621XM router. The backbone routers are used at layer 3 for normal routing.

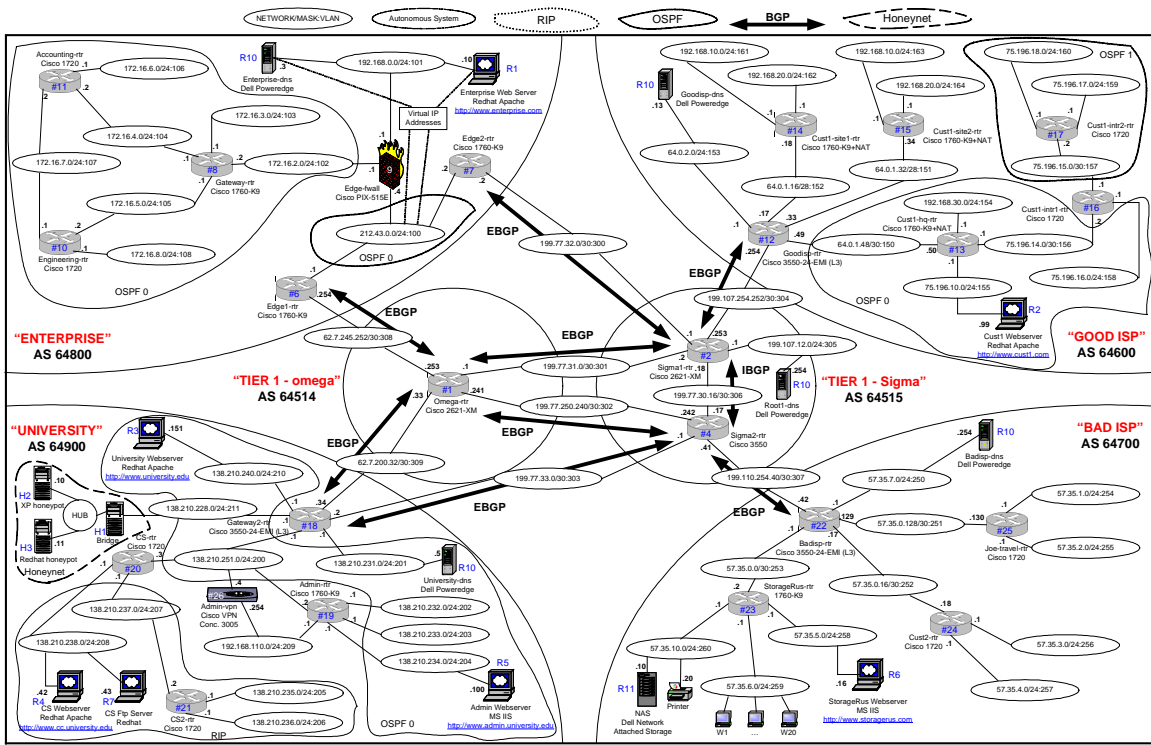


Figure 2 Laboratory Network Topology

The “enterprise” autonomous system consists of two redundant connected Cisco 1760 routers, a Cisco PIX-515E firewall, an access/edge Cisco 1760-VPN/K9, and two Cisco 1720 access routers. A Demilitarized zone that contains a domain name server as well as a web server exists. The “enterprise” allows a student to experiment with a realistic enterprise topology. The “good ISP” autonomous system consists of a Cisco Catalyst 3550, three Cisco 1760-VPN/K9 routers, and two Cisco 1720 access routers. The “good ISP” is set up such that it may contain remote office enterprise connections through both VPNs and clear connections. Network Address Translation is also used inside this “good ISP”. The “university” autonomous system consists of a dual connected Cisco Catalyst 3550, and a Cisco 3005 VPN concentrator. The emulated “university” has no firewall and terminates VPNs from emulated remote users. An access control list is used in the Cisco Catalyst 3550. This autonomous system is intended to look like a university. University use of networks typically entails a large amount of user freedom and thus limited network restrictions. The “bad ISP” autonomous system is a haven for hackers. It consists of a

Cisco Catalyst 3550 virtual gateway router as well as two Cisco 1720s and one Cisco 1760-VPN/K9 for distribution. No access control firewalls, or network filtering is applied internally in the “bad ISP”. This autonomous system is intended to represent the unrestricted access typical of many Internet service providers.

The use of virtual LAN (VLAN) technology and the combined capabilities of the Cisco Catalyst 3550 equipment that have both layer 2 Ethernet switching capabilities and layer 3 routing capabilities allow multiple logical network topologies to be mapped onto the physical topology. Cisco Catalyst 3550s are switches and when they are installed with an “EMI routing image” they have both layer 2 switching and layer 3 routing capability in the same box. Physically all equipment is connected together with enough physical connections that a diverse set of logical topologies may be instantiated through VLAN technology. This allows any port on any piece of equipment to logically appear on any subnetwork. Perl scripts, which we custom created, are used to configure the logical topology and can completely reconfigure the network. Network reconfiguration may be necessary as different laboratory assignments or different classes use the equipment. The need to physically rewire the network is essentially gone. It is possible to move various machines around the network, for example from one autonomous system to another without doing any physical rewiring. It is also possible to change the autonomous system architectures without doing any rewiring. This is possible because the routers VLAN capabilities allow anything that is physically connected to a given Cisco Catalyst 3550 to be logically connected elsewhere in the network.

III Laboratory Assignments

The primary user of this equipment is a class named “Internetwork Security”. This class consists of twelve laboratory assignments and a capstone network security competition, which are all completed in one semester. All of the laboratory exercises are available online [2].

Students are issued their own hard drives and laboratory machines contain hard drive frames (as shown in Figure 3) that allow easy removal of the hard drives. This enables

students to bring to the lab, on their issued hard drive, their own operating systems and tools from previous lab visits and allows them to start at exactly the same system state as when they departed the lab the last time. An alternative approach we have capabilities available for is to maintain fixed hard drives in the end station computers and to download operating system images as needed. We found the large wait times to download gigabyte operating system images inconvenient, and so we have remained with removable hard drives that students can take with them between lab sessions.



Figure 3 Removable Hard Drive Frame

The first lecture material students encounter in the internetwork security class is an ethics presentation and discussion. The first laboratory assignment is Operating System Installation, Network Reconnaissance, Network Mapping, and Vulnerability Assessment lab. The objectives of this first lab include configuring the student issued hard drive to contain the base Red Hat 8.0 operating system, installing VMWare [3] so that students may run multiple operating systems including initially Windows XP and Red Hat 7.2 versions. Vmware is a commercial software package that allows one physical machine to run multiple virtual machines simultaneously. Additionally, the first lab includes installation of initial network security tools. The initial tools that are loaded include a network management tool for mapping and monitoring networks (Cheops-NG) [4], a security auditing and network mapping tool (nmap) [5], a vulnerability-scanning tool (nessus) [6], a Windows XP scanning tool (Super Scan 4) [7], and a Windows reconnaissance tool (Sam Spade) [8]. The main goal of the first lab is to set up the hard drive for future labs and to learn how hackers find targets to attack. Additionally the first

lab highlights the need to carefully consider how much and what information an organization should expose to the Internet in order to minimize exploitation.

The second laboratory assignment involves experimenting with some of the password cracking tools available for Windows and Linux as well as using a network sniffer named ethereal [9] to sniff the network connection between Linux and Windows computers. In addition, address resolution protocol (ARP) and ettercap tools are used to examine how hackers carry out a Man-in-the-Middle attack. To crack passwords on the Windows system, we use a program called L0phtCrack [10]. For the Linux system, we use the software “John the Ripper” [11] to crack the passwords. Ethereal is used to watch a telnet session, capture packets from an SSH session, and to watch the network mapping tool nmap work. In the second part of this lab, Address Resolution Protocol is explored and the theory of address resolution protocol poisoning is experimented with. In addition, the tool hunt [12] is used to hijack a TCP session. The goals of this lab include making students aware of how easy it is to obtain passwords, sniff network traffic, and how attackers can exploit the characteristics of address resolution protocol to take over sessions in the network. In this lab one computer with three virtual machines is used to implement a LAN with three computers attached. The host computer has Red Hat 8.0 (RH 8.0) installed with VMware. The host computer uses VMware to implement two other Red Hat 7.2 computers and a Windows XP computer all on the same network. Defense mechanisms such as password choices and defensive network scanning are used.

The third laboratory assignment introduces the concepts of falsifying identity on a network. Both Ethernet Medium Access Control addresses (MAC) as well as Internet Protocol (IP) address spoofing are examined and experimented with. After successfully spoofing addresses, this lab also allows examination of how attackers may use spoofing for several kinds of denial of service attacks. In particular, a Domain Name System Spoofing tool (DNSspoofer from the dsniff tools) [13] is examined. In addition, a tool named dsniff [13], which includes the capability to kill tcp sessions, is examined. To counter these types of attacks, the tool arpwatch [14], which helps to detect spoofing

attacks, is examined for effectiveness. The tool suite datapool [15], which contains approximately 100 denial of service types of attacks, is also employed in this lab.

The fourth laboratory exercise examines buffer overflows. Since this is the most common technique used to gain control of a target machine, a very detailed study is made of this technique. This lab examines the memory stack used in computer processes and demonstrates how to overflow memory buffers in order to gain root or administrative access. The lab executes several buffer overflow attacks against Linux and Windows XP machines. Techniques for preventing this type of exploit are also examined.

The fifth laboratory involves analysis of rootkits. In this lab three different rootkits are examined, two for Linux and one for Windows. The first rootkit examined is named Irk4 [16], which is a user-level rootkit for Linux. The second rootkit examined is a Linux kernel level rootkit named Knark [16]. Four methods of seeing if a rootkit is installed are examined. The methods used are kern_check [17], chkrootkit [18], strace [19] and rootkit Hunter [20]. For Windows XP the rootkit called Hacker Defender [21] is examined. This rootkit allows one to hide files and processes and creates a backdoor on the machine that has it. The goal of this lab includes understanding how an attacker maintains access to a compromised machine and how one can detect that this has happened.

The sixth laboratory involves backdoors and Trojans. In the first part of the lab, Netcat [22] is used to gain access to a machine. Netcat can also be used to create backdoors on systems. Attackers can use Netcat to create a remote shell on another machine over a TCP or UDP port. Netcat is run on both Linux and Windows machines in the lab. This lab also examines another backdoor that uses Internet Control Message Protocol to obtain a remote access shell (icmp-backdoor) [23] on a Linux machine. In the next part of the lab, the properties of a Windows XP operating system Trojan are examined by using a software package called Virtual Network Computing (VNC) [24]. VNC is an application level Trojan backdoor. Next, a Trojan program called Back Orifice 2000 [25] is examined on Windows platforms. The last part of the lab involves involuntarily downloading an exploit through Microsoft Outlook Express email to demonstrate email

class vulnerabilities. Countermeasures including monitoring machine port activity and network traffic flow are included.

The seventh lab is a honeypot, Network Monitoring, and Forensics lab. In this lab students set up two different honeypots, one on Windows and then one on Linux, to monitor network traffic and look for suspicious activity. Students also use snort to log data and as an Intrusion Detection System. A snort rule is written to detect a buffer overflow attack. The tool Advanced Intrusion Detection Environment (AIDE) [26] is used to determine if any changes have been made to a system. On the forensics side students examine a few files of captured data from real attacks to see if they can find out what was going on. An exercise from the honeynet.org organization called the “Scan of the Month Challenge” [27] is completed. Students use the Forensic and Incident Response Environment [28] to simulate what one would do after a successful attack was performed on a system. This forensics environment includes the Coroner’s Toolkit (TCT) [29], Autopsy [30], Sleuth Kit [31], as well as penetration testing and virus scanning tools. A buffer overflow exploit from a previous lab is again used in this lab to attempt to compromise a system both with and without countermeasures in place.

The eighth lab is a firewall lab. This lab is divided into three major parts. The first explores the Linux firewall implementation in the form of the *iptables* program. The second part is a small introduction to Zone Alarm [32], a popular Windows firewall program. In the third part of the lab, students configure a Cisco PIX 515E firewall for a given network structure. Various testing exercises are conducted after students configure and set up various firewall rules. Students are able to compare the performance of open source versus commercial firewall implementations.

In the ninth lab, two worms are examined. One worm that is examined was designed as a learning tool [33] and the worm AnnaKournikova [34] that was captured on the Internet. Also in this lab, a simple laboratory computer virus, which we authored for use exclusively in our laboratory environment, is examined. In all three cases computers are

actually infected to see how it is done and what happens when we attempt to completely remove the malicious programs.

Lab 10 is a wireless security lab where wireless tools such as Kismet [35] and AirSnort [36] are used to examine vulnerabilities that exist in wireless networks. Packets are sniffed to obtain MAC addresses to get around filters. Techniques to crack a WEP key are used. This lab shows how a hacker can monitor unencrypted traffic, spoof a MAC address, crack WEP and decrypt encrypted packets. Counter measures such as encryption are presented to motivate the next laboratory.

Lab 11 works with Virtual Private Networks. In this lab, the general concept of a VPN (Virtual Private Network) and some different methods of implementing a VPN are examined. First, students implement a very simple SSH based VPN in Linux. They then use IPSec to communicate safely between windows XP computers using IPSec. Finally, a modern IPSec VPN implementation is experimented with using a Cisco 3005 VPN Concentrator. Students are able to compare hardware versus software implementations of VPNs with the laboratory resources provided.

Lab 12 introduces tools that hackers may use to see vulnerabilities on an apache web server and to gain access to private pages. The tool WGET [37] is used to download web sites for examination. The open source tool Nikto [38] that is a web server scanner is used to perform comprehensive tests against web servers.

The last laboratory activity is an attack defend competition scenario that is widely used in Information Assurance classes. The one scenario we have employed was a Capture the flag exercise which involved UCSB, Georgia Tech, Naval Postgraduate School in Monterey, North Carolina State University, United States Military Academy, University of Texas at Austin, and the University of Illinois at Urbana-Champaign. The competing teams were given a VMware image prepared by the organizers [39]. The image contained a number of undisclosed vulnerabilities for different services running. The task of the teams was to find the vulnerabilities, keep them from being exploited on their server, and

exploit the same vulnerabilities to compromise the security of other teams' sites. The teams gained points by keeping their server active and uncompromised and by compromising other teams' servers (that is, "capturing their flag").

IV Course Assessment

At the time of writing, this laboratory environment had been used for the Internetwork Security class for three semester offerings. Student feedback results are shown in table 2. Results are out of a possible 5.0.

	Spring 2003	Fall 2003	Spring 2004
Course seemed well planned and organized	4.4	4.4	4.0
Good job covering course objectives/content	4.6	4.8	4.3
Explained complex material clearly	4.6	4.7	3.8
Number of assignments was reasonable	4.6	4.4	4.3
Exams covered course content/objectives	4.3	4.4	4.9
Exams were of appropriate difficulty	4.0	4.1	4.4
Number of Students Responding to Survey	13	20	20

Table 1 Course Assessment Results

V Conclusions

Having a totally isolated information security laboratory where students are allowed to launch attacks and attempt to defend against them is highly educational and highly motivating. Students having hardened their issued machines and networks and then

seeing them compromised are better prepared to understand how to prevent similar compromises in the future. Having a laboratory environment where students may experiment and be creative in a complex network environment is highly motivating for students.

We find the reconfiguration capability of the lab highly beneficial in that it is possible to change the network topology by just running a configuration script. This means that we may reconfigure the laboratory for example for a firewall laboratory assignment without requiring any physical wiring changes or manual configuration changes to reset the firewall configuration. One of the interesting capabilities is a master reset ability. When the lab is completed, we are able to easily set the network topology back to the original configuration. We have used the laboratory to accommodate up to a total of 60 students a semester. The limiting factor is the number of end station computers (twenty five) that we have at present.

The laboratory we implemented has more network equipment and capability than many small companies. These small companies typically have a full time information technology support person. There is a high workload level associated with maintaining and managing a lab of this type. A highly successful information assurance laboratory may be implemented with far less equipment [40]. What we have discovered by implementing our lab at the other end of the capability and complexity spectrum is that having this level of complexity enables a level of real world realism unmatched by the simpler implementations. Exposure to access control lists, firewall rules, network address translation, etc in a network of the complexity presented here teaches valuable operational issues that theoretical and simpler laboratory implementations never encounter.

Acknowledgements

We would like to acknowledge the CISCO University Critical Infrastructure Assurance Group equipment donation program [41] and training programs as well as Intel

equipment donations, which make this laboratory possible. We would also like to acknowledge that Cisco field representative Jim Berg was instrumental in helping architect, select, and set up the equipment.

References

- [1] Abler, R., Owen, H., and Riley, G., "University Methodology for Internetworking Principles and Design Projects," *IEEE Transactions on Education*, vol. 46, no. 2, pp. 218-225, 2003.
- [2] The Internetwork Security Class home page. [Online] Available: <http://users.ece.gatech.edu/~owen/>, as of August 4, 2004.
- [3] The VMware home page. [Online] Available: <http://www.vmware.com/>, as of August 4, 2004.
- [4] The cheops-ng home page. [Online] Available: <http://cheops-ng.sourceforge.net/>, as of August 4, 2004.
- [5] The insecure.org nmap home page. [Online] Available: <http://www.insecure.org/nmap/>, as of August 4, 2004.
- [6] The nessus home page. [Online] Available: <http://www.nessus.org/>, as of August 4, 2004.
- [7] The snapfiles.com superscan home page. [Online] Available: <http://www.snapfiles.com/get/superscan.html>, as of August 4, 2004.
- [8] The same spade.org windows home page. [Online] Available: <http://www.samspace.org/ssw/>, as of August 4, 2004.

[9] The ethereal.com home page. [Online] Available: <http://www.ethereal.com/>, as of August 4, 2004.

[10] The Security Focus Tools Page. Available: <http://www.securityfocus.com/tools/1005>, as of August 4, 2004.

[11] The John the Ripper Password Cracker Page. Available: <http://www.openwall.com/john/>, as of August 4, 2004.

[12] The Packet Storm Tools page. Available: <http://packetstormsecurity.nl/sniffers/hunt/>, as of August 4, 2004.

[13] The monkey.org dsniff home page. [Online] Available: <http://monkey.org/~dugsong/dsniff/>, as of August 4, 2004.

[14] The LBNL Network Research Group page. Available: <http://www-nrg.ee.lbl.gov/>, as of August 4, 2004.

[15] The Packet Storm DoS Tools page. Available: <http://packetstormsecurity.nl/DoS/indexsize.html>, as of August 4, 2004.

[16] The Packet Storm rootkits Tools page. Available: <http://packetstormsecurity.nl/UNIX/penetration/rootkits/>, as of August 4, 2004.

[17] The samhain lab library home page. [Online] Available: <http://la-samhna.de/library/>, as of August 4, 2004.

[18] The chkrootkit.org home page. [Online] Available: <http://www.chkrootkit.org/>, as of August 4, 2004.

- [19] The strace homepage. [Online] Available: <http://www.liacs.nl/~wichert/strace/>, as of August 4, 2004.
- [20] The rootkit home page. [Online] Available: <http://www.rootkit.nl/>, as of August 4, 2004.
- [21] The Hacker Defender home page. [Online] Available: <http://rootkit.host.sk/>, as of August 4, 2004.
- [22] The GNU Netcat project home page. Available: <http://netcat.sourceforge.net/>, as of August 4, 2004.
- [23] The project-hack.org home page. [Online] Available: <http://www.project-hack.org/back.html>, as of August 4, 2004.
- [24] The real vnc home page. [Online] Available: <http://www.realvnc.com/>, as of August 4, 2004.
- [25] The cultdeadcow.com back orifice home page. [Online] Available: <http://www.cultdeadcow.com/tools/bo.html>, as of August 4, 2004.
- [26] The SourceForge AIDE page. Available: <http://sourceforge.net/projects/aide>, as of August 4, 2004.
- [27] The honeynet.org challenges home page. [Online] Available: <http://www.honeynet.org/misc/chall.html>, as of August 4, 2004.
- [28] The F.I.R.E. home page. [Online] Available: <http://fire.dmzs.com/>, as of August 4, 2004.
- [29] The Computer Forensics home page. [Online] Available: <http://www.porcupine.org/forensics/>, as of August 4, 2004.

- [30] The Autopsy forensic browser home page. [Online] Available: <http://sleuthkit.sourceforge.net/autopsy/desc.php>, as of August 4, 2004.
- [31] The sourceforge.net sleuthkit home page. [Online] Available: <http://sleuthkit.sourceforge.net/sleuthkit/desc.php>, as of August 4, 2004.
- [32] The Zone Labs home page. Available: <http://www.zonelabs.com/store/content/home.jsp>, as of August 4, 2004.
- [33] Church, C., Schmoyer, T., and Owen, H. L., “Design and Implementation of a Simple Class Room Laboratory Internet Worm”, submitted to Journal of Security Education, July 2004.
- [34] The Symantec vbs.ss@mm worm page. [Online] Available: <http://www.symantec.com/avcenter/venc/data/vbs.sst@mm.html>, as of August 4, 2004.
- [35] The kismet home page. [Online] Available: <http://www.kismetwireless.net/>, as of August 4, 2004.
- [36] The airsnot home page. [Online] Available: <http://airsnot.shmoo.com/>, as of August 4, 2004.
- [37] The wget home page. [Online] Available: <http://www.gnu.org/software/wget/wget.html>, as of August 4, 2004.
- [38] The nikto home page. [Online] Available: <http://www.cirt.net/code/nikto.shtml>, as of August 4, 2004.
- [39] The UCSB capture the flag home page. [Online] Available: <http://www.cs.ucsb.edu/~vigna/CTF/>, as of August 4, 2004.

[40] The Information Warfare Analysis and Research Laboratory home page. [Online]
<http://www.itoc.usma.edu/iwar/index.html>, as of August 4, 2004.

[41] The Critical Infrastructure Assurance Group home page. Available:
http://www.cisco.com/security_services/ciag/initiatives/education/equipmentdonation.html, as of
August 4, 2004.