AC 2009-2278: DEVELOPMENT OF UNDERGRADUATE NETWORK SECURITY LABS WITH OPEN-SOURCE TOOLS

Arif Uluagac, Georgia Institute of Technology

Arif Selcuk Uluagac is a Ph.D. student in the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta, GA as a member of the Communications Systems Center Laboratory. He received his B.Sc. in Computer Engineering from Turkish Naval Academy and M.Sc. degrees in Electrical and Computer Engineering from Carnegie Mellon University in PA, in 1997 and 2002, respectively. He is a member of IEEE, ACM, and ASEE. He is currently teaching the undergraduate level network security class as an adjunct instructor at Southern Polytechnic State University.

Thomas Fallon, Southern Polytechnic State University

Thomas J. Fallon received his BS and MS degrees in Electrical Engineering from the Georgia Institute of Technology and his Ph.D. degree in Astronomy from Georgia State University. He is an Associate Professor of Electrical and Computer Engineering Technology at Southern Polytechnic State University and is coordinator of the BSTCET program. He has 20 years of telecommunications- related experience, conducts networking workshops, and is author of the book The Internet Today. His astrophysics Ph.D. research at Georgia State University involved remote operation of a telescope array via the Internet.

Walter Thain, Southern Polytechnic State University

Walter E. Thain received his BS, MS, and Ph.D. degrees in Electrical Engineering from the Georgia Institute of Technology. He is an Associate Professor in Electrical and Computer Engineering Technology at Southern Polytechnic State University and teaches courses in voice and data networking, communications systems, and analog and RF electronics. Research interests include voice and data network design and management, network security, RF communication systems, and digital signal processing. He spent 12 years in industry, where he designed mixed analog-digital systems, including, short-pulse radars and antennas, low-noise analog circuits, RF circuits, pulse generators, frequency synthesizers, switching power supplies, and high-speed digital circuits. He is co-inventor on a patent for the design of electronic instrumentation used to steer oil wells while drilling.

John Copeland, Georgia Institute of Technology

John A. Copeland holds the John H. Weitnauer, Jr., Chair in the School of Electrical and Computer Engineering at the Georgia Institute of Technology, and is a Georgia Research Alliance Eminent Scholar. He is the Director of the Communications Systems Center (CSC). He served as Director of the Georgia Center for Advanced Telecommunications Technology (GCATT) from June 1993 to Nov. 1996. Prior to joining Georgia Tech in 1993, Dr. Copeland was Vice President, Technology at Hayes Microcomputer Products (1985-1993), and Vice President, Engineering Technology at Sangamo Weston, Inc. (1982-1985) and served at Bell Labs (1965-1982). He founded Lancope, Inc. (2000), and invented the StealthWatch network security monitoring system that is being added to government, corporate, and university networks around the world. He began his career at Bell Labs conducting research on semi-conductor microwave and millimeter-wave devices. Later, he supervised a group that developed magnetic bubble computer memories. In 1974, he led a team that designed CMOS integrated circuits, including Bell Labs' first microprocessor, the BELLMAC-8. His last contributions at Bell Labs were in the area of lightwave communications and optical logic. At Sangamo Weston he was responsible for R\&D groups at ten divisions. At Hayes was responsible for the development of modems with data compression and error control, and for Hayes' representation on CCITT and ANSI standards committees. Dr. Copeland received B.S., M.S. and Ph.D. degrees in physics from the Georgia Institute of Technology . He has been awarded 41 patents and has published over 60 technical

articles. In 1970 he was awarded IEEE 's Morris N. Liebmann Award for his work on gallium arsenide microwave devices. He is a Fellow of the IEEE and has served that organization as the Editor of the IEEE Transactions on Electron Devices. He served on the Board of Trustees for the Georgia Tech Research Corporation (1983-1993).

Development of Undergraduate Network Security Labs with Open Source Tools

Abstract

Undergraduate level network security classes are usually taught during the junior or senior year of an undergraduate education, because it is assumed that students have acquired the necessary background material in previous classes, such as algorithms, programming, and networks. Although students should have had enough exposure to the background material, they still potentially face difficulties in grasping theories and concepts related to the network security field. One way to address this concern is to give homework, or require laboratory exercises with programming assignments. Programming assignments provide the students with an excellent opportunity to digest the concepts. However, they are usually focused too much on one aspect of the problem rather than overall picture of the particular topic of interest. Thus, it is vital to design network security labs that combine theory and available applications representative of the lecture component of the class.

In this paper we discuss the development of the laboratory component of an undergraduate network security course for the Telecommunications Engineering Technology (TCET) program at Southern Polytechnic State University. As with other engineering technology programs, the TCET program maintains an application-oriented approach in all of its courses. Creating the laboratory component is often the most challenging part of the overall course development task. Fortunately, the availability of numerous open source security tools provided resources for all of the lab exercises as well as many of the lectures. The tools were selected in order to better enable student to comprehend the complexities and intricacies of security-related topics. Furthermore, the lab exercises can be used with stand-alone labs, or aid in the completion of programming assignments or other forms of homework. Descriptions of the security course lab exercises and features of the open source tools that were utilized are included.

1. Introduction

In our era, information is distributed across many uncontrolled domains (e.g., Internet) and we have become more dependent on technology and the Internet.. For instance, we have many flavors of distributed networks today: wired, wireless, GPS, hand-held devices, sensor networks[1], etc., with almost the same set of rich networking functionalities (e.g., multimedia.) However, as the number of cyber crime incidents increase[2], the security of these diverse set of networks and their services has become an integral part of most businesses.. For instance, the Internet Crime Complaint Center (IC3)[a] received over 200,000 cyber crime-related incidents in 2007 alone. Therefore, the situation necessitates the teaching and better education of today and

a The Internet Crime Complaint Center (IC3) is a partnership between the Federal Bureau of Investigation (FBI), the National White Collar Crime Center (NW3C), and the Bureau of Justice Assistance (BJA).

tomorrow's workforce in terms of network security practices. From this perspective, one can claim that network security classes comprise an important element of the undergraduate curriculum.

Undergraduate-level network security classes are usually taught during the junior or senior years of an undergraduate education. As the lecture content of the network security classes are generally based on complex and rigorous mathematical foundations, it is assumed that students have already acquired the necessary background in previous classes, such as algorithms, programming, and networks.

Nonetheless, students still face difficulties in grasping and understanding the complexities and intricacies of security-related topics during lecture. One way to address this issue is to give homework, or require laboratory exercises with programming assignments. Programming assignments provide the students with an excellent opportunity to digest the concepts. However, they are usually oriented toward just one aspect of the problem rather than the overall picture of the particular topic of interest. Thus, it is vital to design network security labs that combine theory and available applications representative of the lecture component of the class.

On the other hand, there has been a proliferation of various open source Linux operating system distributions[3]. While it is true that they may not be as dominant as their counterparts (e.g., MS-Windows) in the market, they provide a rich set of flexibilities to the end-user. Inter alia, one nice feature of these Linux derivatives is that they contain a lot of security-related tools and most importantly they are available free of charge.

Available open source security tools can be utilized to facilitate the teaching and the learning of relatively difficult lecture topics in network security classes. Lab exercises that couple theoretical topics directly with open-source Linux security tools benefit both students and instructors. There are further benefits of this “tool-oriented” lab approach. First of all, there is a minimal cost associated with the tools as they are primarily free software. Second, students have a chance to see how theoretical lecture topics (e.g., asymmetric encryption, digital signatures, etc) in the security field are realized in the real world. Third, students increase their understanding of the complex material via hands-on labs. Finally, the labs can be extended easily for the implementation of other more advanced concepts in the security field.

The benefits and availability of open source security tools permitted the efficient development of the laboratory component of an undergraduate network security course for the Telecommunications Engineering Technology (TCET) program at Southern Polytechnic State University. TCET courses emphasize application of key course concepts to enhance student learning. This paper discusses ten lab exercises and some of the associated lecture content developed from Linux-based security tools. The exercises can either be stand-alone labs[4] or can aid in the completion of programming assignments or other homework assignments.

Several contributions can be articulated for this paper. First, we show and discuss which Linux-based free security tools can be used to complement the teaching of network security lectures via the created labs. Secondly, we list the details of the labs. Then, we show an example lab exercise.

Lastly, we discuss the type of other labs and lecture topics that could be created with Linux-based open-source security tools.

The paper proceeds with a description of the lab exercises in Section 2. An example lab is articulated in Section 3. Section 4 addresses other open-source Linux-compatible security tools that may be used in lab exercises. A discussion of the cost and benefits of tool-oriented labs is given in Section 5, followed by concluding remarks in Section 6.

2. Descriptions of Labs Created with Open Source Tools

A number of simple or complicated security labs could be written using open source Linux security tools, depending on the need. This section describes the exercises written using such tools. The lab includes 20 Sun Ultra 24 Workstation with Suse 10.1 as the Linux distribution. An illustration of the laboratory environment is given in Figure 1. Table 1 summarizes the Linux-based open source security tools used in the 10 exercises developed and the lecture topic addressed by the tool.

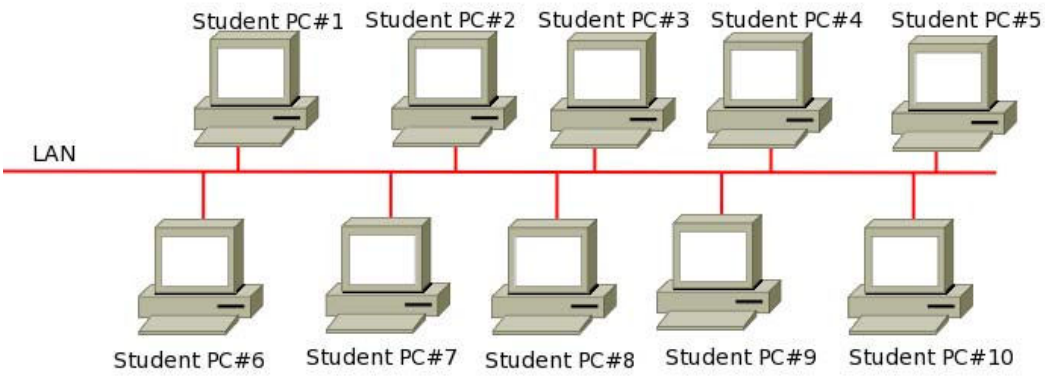

Figure 1: An illustration of the laboratory environment

a. Introduction to Encryption Mechanisms: In this lab, students are exposed to some fundamental knowledge and concepts of popular encryption mechanisms. Specifically, the student is introduced to the basics of symmetric and asymmetric encryption[5] (aka public key-based) mechanisms using the free Gnu Privacy Guard (GPG[6]) tool. The lab's objective is to complement the students' understanding of the utilization of primitive encryption and cryptography operations, which, in general, are based on rigorous mathematical theories in the textbooks. More specifically, students learn how to create their own key, encrypt a text message, and upload their keys on available public key servers. After obtaining and learning how to use their keys, they use them to communicate with the class instructor during the semester in a

secure fashion[b].

b. Digital Signature Systems: One other common use of public-key cryptography (except for encryption) is digital signature systems. In this lab, students learn the basics of digital signature systems using the GPG tool. Specifically, they learn how to sign and verify a document using the keys created with the GPG tool in the previous lab.

c. Network Sniffing Introduction: Often times, attackers/hackers perpetrate their malicious activities on available networking resources. For instance, even if one has the most secure system, one must check if the networked computing resources are under attack or determine the presence of any suspicious activity in the network. Network sniffers, or network analyzers, allow one to accomplish such tasks easily by capturing packets on a specified physical network segment. Thus, in this lab, students learn the basics of the network sniffing using one of the most widely used network analyzers, Wireshark[7],and learn some of its capabilities like filtering certain types of network packets and constructing an observed http session from the captured packets.

Tool Name	Lecture Topic Covered
Gnu Privacy Guard (GPG[6])	Symmetric and asymmetric encryption principles Digital signature mechanisms
Wireshark[7] Openssh[8]	Network sniffing Advances reconnaissance
Openssl[9]	Digital certificates
Nmap[10]	Port scanning
Nessus[11]	Vulnerability scanning
JohnTheRipper[12]	Password policies
Snort[13]	Intrusion detection
Dig[14] Whois[14] Netstat[14] Top[14]	Simple reconnaissance

Table 1. Class content and the security tools.

d. Network Sniffing Advanced: From laboratory 1 through 3, students learn the basics of secure and insecure communication using tools like openssh[8] and GPG to encrypt messages and traffic, and also learn to use Wireshark to capture the network communications. Thus, this lab combines students' previous skills and allows them to see the difference between encrypted and unencrypted traffic streams. They can gather valuable information like passwords and usernames when the traffic is not protected.

b The idea of students' communicating to their instructors with their keys was inspired from the Network Security class (ECE 6612) offered in the School of ECE at Georgia Institute of Technology.

e. Digital Certificates: Many real world applications utilize public key-based systems (aka asymmetric key-based) as their choice of encryption mechanisms. However, encryption techniques using public and private keys need a public-key infrastructure (PKI) to support the distribution and identification of public keys. Digital certificates bundle public keys, and the pertinent information about them such as the algorithms used, owner or subject data, the digital signature of a Certificate Authority (CA) that has verified the subject data, and an expiration date. Thus, the digital certificates provide users a mechanism to ensure that a public key contained in a certificate belongs to the entity to which the certificate was issued. In this lab, using the openssl[9] tool, students learn how to run their own root certificate authority, create normal, root and user certificates, and how to insert certificates into applications, specifically into web browsers.

f. Port Scanning: Port scanners allow enumeration of network resources to discover what machines are connected and which networking services are running on those machines. It is a useful task for system administrators; they can identify unauthorized or illicit services, malicious programs like spyware, Trojan horses, or network worms. Thus, this lab teaches students the basics of port scanners. Specifically, they are introduced a tool called nmap[10] to scan a network and identify ports and services available on other hosts.

g. Vulnerability Scanning: After enumeration network resources to discover what machines are connected and what services/ports are running on those machines using nmap, it is extremely important to know if those services are vulnerable to any exploit or not. In other words, identifying security level of services in a network is an important task. This can be accomplished with a security scanner or vulnerability scanner. Students in this lab learn the basics of vulnerability scanners. Specifically, they use a tool called nessus[11] to scan a network and identify vulnerabilities about their and other machines associated with particular services and open ports.

h. Strong Password Policies: Usually securing any networking entity involves setting up passwords. In this lab, students learn how weak passwords may be vulnerable to attacks and how they can be broken easily by the password-cracking tools. They experiment with John The Ripper[12], one of the most famous password cracking tools. Hence, learning the basics of password crackers from an attacker's perspective, the students comprehend the importance of having robust and strong password policies.

i. Intrusion Detection Systems (IDS): Intrusion detection is an integral part of a security architecture in any corporation and organization in the Internet. It allows detection of unwanted attempts at the instant when a malicious activity like accessing, manipulating, and/or disabling of computer systems is under way. Students in this lab learn the basics and capabilities of the Snort[13] IDS tool. They create certain user rules, and configure automated actions like alerts if the analysis of incoming packets matches any of the rules.

j. Other simple reconnaissance tools: In this lab, students learn to use several Linux-based small tools that are useful in network reconnaissance. These tools are especially handy for learning more information about the source of malicious activities in a network or host. With some of these tools, students can collect valuable information about other hosts, identify whether an

unknown host is legitimate or not, and they can obtain information about how they can contact them if needed. Moreover, with some, they can track the activities of the processes in a host to check the presence of any adversary. The simple reconnaissance tools used in this laboratory exercise include dig[14], whois[14], netstat[14], and top[14].

3. A Sample Laboratory Exercise for Digital Certificates

In this section, we elaborate on one of the labs created as an example. As discussed in the previous section, digital certificates helps provide users a mechanism to ensure that a public key contained in a certificate belongs to the entity to which the certificate was issued.

For instance, students can use the following commands to create their own root certificate (Step 1), user certificate signing requests (Step 2), sign the request using the root certificate created (Step 3), and finally insert both the root certificate and user certificates (Step 4) into their preferred browser.

Step 1: openssl req -new -x509 -extensions v3_ca -keyout private/cakey.pem -out cacert.pem -days 3650 -config ./openssl-3904.cnf
Step 2: openssl req -new -nodes -keyout myprivkey.pem -out signrequest.csr -config ./openssl-3904.cnf
Step 3: openssl ca -out mysignedcert.pem -config ./openssl-3904.cnf -infiles signrequest.csr
Step 4: openssl pkcs12 -export -out mypkcs12cert.pfx -in mysignedcert.pem -inkey myprivkey.pem -name "My ECET 3904 Certificate"

Moreover, as shown in Figure 2, the students can insert their certificate in to their preferred browsers. With the root certificate, they can create and sign more user certificates for various other purposes.

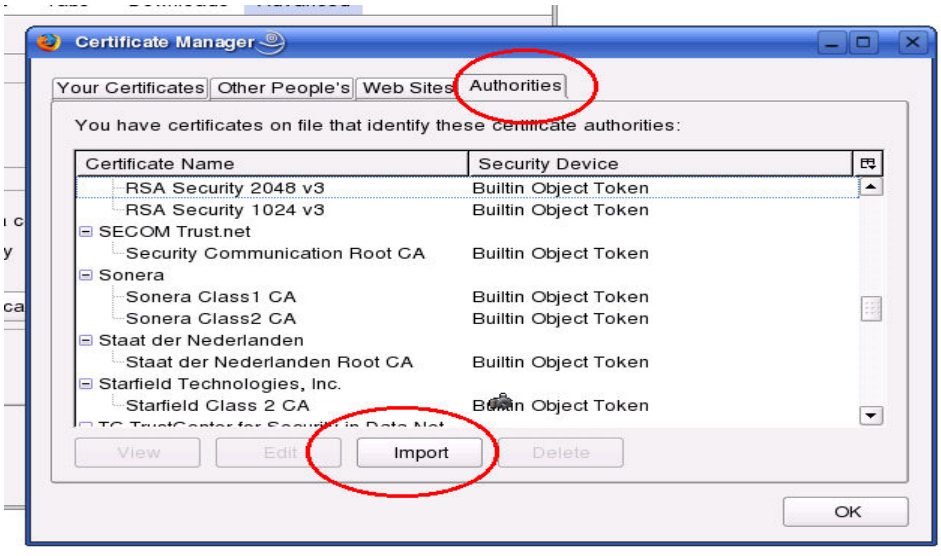

Figure 2: Inserting a root digital certificate into a web browser

4. Other Open Source Linux Security Tools Available

In this section, we articulate other available open source security tools in the Linux-based systems. It is clear that the community contributing to the open source software is increasing day-by-day. There may be yet other tools that have not been covered in this study, and there will certainly be other tools available as time passes. It is also important to note that some of the tools may include multiple features that can span different lecture topics. Table 2 lists some of these tools and appropriate lecture topics. Comments on the tools and how they might be utilized follow.

a. Authentication and Access Control Lists (ACL[15]): In real world security implementations, there may be situations where information about users, passwords, home directories belonging to a particular domain (e.g., a company) need to be authenticated and managed from a centralized location. OpenLDAP[16], open lightweight directory access protocol, can be utilized for such scenarios. Additionally, ACLs is a very common way of setting up permissions over an object that needs to be protected, and is fundamental in many security lectures. For the demonstration of ACLs topic, features of OpenLDAP can also be utilized.

b. IPsec: IPsec is a framework for securing communications over IP networks. There are several tools that implement IPsec as part of their function. These include Openvpn[17], Openswan[18], and IPsec[19]. One nice direct application of Ipsec is a Virtual Private Network (VPN[5]), and the Ipsec lecture topic can be addressed via demonstrations of VPN tunnels.

Tool Name	Textual Topic That Can be Addressed
OpenLDAP[16]	Authentication Access Control Lists (ACLs)
Ipsec[19] Openvpn[17] Openswan[18]	IPsec VPN
Iptables[20]	Firewalls NAT
Tripwire[21] Clamav[22]	Malicious software Anti-virus
Kismet[23]	Wireless security

Table 2. Further class content and the security tools that can be utilized.

c. Firewalls: The Iptables[20] command tool allows setting up host-based firewalls on Linux-based machines. Firewall or network address translation (NAT) concepts can both be addressed using this tool.

d. Malicious Software – Anti-virus: Lectures about malicious software and anti-virus techniques are also an integral part of any network and computer security classes. In such lectures, students learn fundamental concepts regarding combating against the malicious software by checking

integrity of the files for any possible changes or checking for the presence of any virus in the systems. Tools like tripwire[21] and clamav[22] can supplement the lectures on malicious software and anti-virus, respectively.

e. Wireless Security: The field of wireless security, albeit popular, is not yet part of most of the popular textbooks[5, 15, 24, 25] used for network or computer security classes. Nevertheless, wireless security topics can be addressed using Kismet[23] tool, which is a passive wireless packet sniffer for IEEE 802.11 wireless local area networks.

5. A Discussion of the Cost and Benefits of Tool-Oriented Labs

In this section, we discuss several benefits and related cost of tool-oriented labs.

- In network security courses, the lecture topics are usually based on material requiring understanding of complex and rigorous mathematical concepts. So, tying theory with tool-oriented labs help the students' learning process. Initial observations of the students who took the network security class using the described 10 laboratory exercises, support the same result.
- Linux-based security tools are freely available in the Internet and they can be easily installed in many of the Linux distributions. So, there is a minimal cost in terms of software related items.
- With the tools-oriented labs, students have a chance to learn how theoretical lecture topics (e.g., asymmetric encryption, digital signatures, etc) are reflected and utilized in real world applications.
- Students have a chance to configure certain files in order to accomplish some of the tasks required in the labs. This experiential learning approach increases the students' understanding of how theoretical aspects of the security field are implemented and maintained.
- As we hear of more and more security breaches in recent years, tool-oriented labs may facilitate the education/training of students and give them a repertoire of techniques, preparing them to confront challenging security issues in industry.
- Lab exercises using Linux-based security tools are suitable for undergraduate and graduate-level security classes.
- The labs can be extended to implement other more advanced concepts in the security field.
- The labs can be easily incorporated into classes that are taught online[31]. For instance, students can download the appropriate exercise from the class web page, and accomplish the tasks in the lab on his or her own computing resources.
- Similarly, when there are not enough resources to have a separate laboratory environment, the labs can be accomplished by the students using their own computing resources.
- Programming assignments provide students with an excellent opportunity to digest security-related the concepts. However, they usually focus too much on one aspect of the problem rather than overall picture of the particular topic of interest. Thus, the labs discussed here can aid in the completion of programming assignments or other forms of homework.

6. Conclusion

In this paper, we have discussed the development of undergraduate network security laboratory exercises using open source security tools which are freely available with many of the Linux distributions. Many of the lecture topics of network security classes in undergraduate or graduate standing can be complemented with many of these labs. They can facilitate teaching and learning of more difficult security topics and help both students and instructors. Moreover, as the labs are designed using open-source security tools, costs are minimized.

Several other institutions, projects, and class-based competitions[26, 27, 28,] exist that aim to teach security concepts with focused, hands-on experience. It is believed that these efforts and our tool-oriented labs together with future improvements in open-source software will help to fill the gap between the theory and the hands-on experience.

Finally, simple or advanced laboratory exercises targeting different security topics can be created with open source security tools. Thus, our future work includes creating additional tool-oriented security lab exercises. Also, their effectiveness supporting student learning will be measured and evaluated, including the students' feedback.

Bibliography

1. A. S. Uluagac, C. P. Lee, R. A. Beyah, and J. A. Copeland, "Designing Secure Protocols for Wireless Sensor Networks," Proceedings of the 3rd International Conference on Wireless Algorithms, Systems and Applications (WASA 2008), Dallas, Texas, October 2008.
2. Internet Crime Complaint Center, "2007 Internet Crime Report", Prepared by Federal Bureau of Investigation, The National White Collar Crime Center, and Bureau of Justice Assistance
3. http://distrowatch.com, Accessed January 2009
4. A. S. Uluagac and D. Williams, Building Hardware-based Low-Cost Experimental DSP Learning Modules, Proceedings of the American Society for Engineering Education (ASEE), Annual Conference of Composition and Exhibition, Pittsburgh, PA, June 2008
5. William Stallings, "Network Security Essentials: Applications and Standards," Prentice Hall, 3rd Edition
6. The GNU Privacy Guard (GnuPG), http://www.gnupg.org/
7. Wireshark, http://www.wireshark.org/,Accessed January 2009
8. OpenSSH, http://www.openssh.com/, Accessed January 2009
9. OpenSSL, http://www.openssl.org/, Accessed January 2009
10. Nmap, http://nmap.org/, Accessed January 2009
11. Nessus, http://www.nessus.org/, Accessed January 2009
12. JohnTheRipper(JtR) Password cracker, http://www.openwall.com/john/, Accessed January 2009
13. Snort, http://www.snort.org/, Accessed January 2009
14. Tony Howlett, "Open Source Security Tools, A Practical Guide to Security Applications,"Prentice Hall, 2005
15. William Stallings, Lawrie Brown,"Computer Security: Prenciples and Practice," Prentice Hall, 1st Edition
16. OpenLDAP, http://www.openldap.org/, Accessed January 2009
17. OpenVPN, http://openvpn.net/, Accessed January 2009
18. Openswan, http://www.openswan.org/, Accessed January 2009
19. ipsec, http://ipsec-tools.sourceforge.net/, Accessed January 2009
20. netfilter/iptables project homepage, http://www.netfilter.org/, Accessed January 2009
21. Open Source Tripwire® software, http://sourceforge.net/projects/tripwire/, Accessed January 2009
22. Clam AntiVirus http://www.clamav.net/, Accessed January 2009
23. Kismet, http://www.kismetwireless.net/, Accessed January 2009
24. Kaufman, Perlman, and Speciner, "Network Security: Private Communication in a Public World," Prentice Hall, 2nd Ed.
25. Behrouz A. Forouzan, "Cryptography and Network Security," McGraw-Hill, 1st Ed.

26. NetSecLab, ECE 6612 Network Security Class at Georgia Institute of Technology, http://www.csc.gatech.edu/copeland/jac/6612/, Accessed January 2009
27. UCSB Homepage "UCSB Capture The Flag", http://www.cs.ucsb.edu/~vigna/CTF/, Accessed January 2009
28. Abler, R., Contis, D., Grizzard, J., Owen, H., “Georgia Tech Information Security Center “Hands-On” Network Security Laboratory”, IEEE Transactions on Education, vol.49, no.1, pp. 82-87, Feb. 2006