Hands-on Exercises for IT Security Education

Xinli Wang
Michigan Technological University
Houghton, MI 49931, USA
xinlwang@mtu.edu

Yan Bai
University of Washington Tacoma
Tacoma, WA 98402, USA
yanb@uw.edu

Guy C. Hembroff
Michigan Technological University
Houghton, MI 49931, USA
hembroff@mtu.edu

ABSTRACT

We have developed a collection of instructional hands-on lab assignments that can be used to help teach security courses or courses with a security component in information technology (IT). Our labs cover a wide spectrum of principles, ideas and technologies along with well-developed open source tools. Lab descriptions are publicly accessible from our web page. All of the labs have been tested in a virtual environment and utilized in our security courses. Feedback from the students has been positive. In this paper, we will present the lab design, topics covered in our labs, lab environments and student evaluation results. We will share our experience in transferring advanced technology to IT security education and lessons learned from this practice.

Categories and Subject Descriptors

K.3.2 [**Computers and Education**]: Computer and Information Science Education—*Information Systems Education*

General Terms

Security

Keywords

IT Security; Hands-on Exercise; Instructional Laboratory; Education

1. INTRODUCTION

It has been widely acknowledged by students, educators and researchers that the benefits of hands-on exercises in the education of computing security are threefold [5, 1, 20, 15, 6, 7, 26, 25]: 1) They expose students to real-world challenges. 2) Hands-on activities help students consolidate knowledge and gain in-depth understanding of the material presented in class lectures. 3) These exercises help students to be well prepared for their careers in industry. However, when we searched for well-designed hands-on laboratories for our IT security courses seven years ago, there were not many publicly available. Although we did locate some individual labs through personal communications, Internet searches and publications [34, 3], there were three difficulties in using them in our classes. Firstly, the coverage of security principles was quite narrow. Many topics in computer and network security were not covered. Secondly, the lab environments varied. Students had to spend a large amount of time to learn how to use and set up the environments. Some tools could not be located. Some software packages did not work correctly for integration. Thirdly, instructions were incomplete. Instructors spent much time on troubleshooting configurations or integration to prepare a lab assignment.

Motivated by the need for education-oriented, coherently-designed and well-supported hands-on exercises for undergraduate education of IT security, we started our journey to develop them several years ago. Our primary objective was to design, develop, implement and publicly deliver a suite of hands-on labs that would cover a wide spectrum of principles, ideas and technologies along with well-developed tools that would be essential for undergraduate education of IT security. By public dissemination through the Internet, these materials can be used by other colleges and universities to enhance security components in IT education and alleviate the workload of instructors in preparing and delivering IT security courses. This project is named as ITSEED by our group which stands for IT SEcurity EDucation.

With recent funding from the National Science Foundation (NSF), we have developed and tested twelve labs. These labs are available from our web page [32]. We have been using these labs in our security courses over the last two years. Lab descriptions have been refined according to the feedback from students and instructors. Overall feedback from students is highly positive. In this paper, we will present the lab design, topics covered by the labs and evaluation results with the hope to share our experiences in teaching IT security courses.

2. LAB DESIGN

To accommodate the diversity in IT practice and education, our labs are developed with a layered and modular design as shown in Figure 1. A lab description consists of four layers: *Goals and Objectives*, *Technologies*, *Tools* and *Effects*. The layers of Technologies and Tools have a modular structure due to their nature of diversity.

In the beginning of a lab, the goals and objectives for each lab are specified. Relevant principles, basic concepts and knowledge are described briefly. Technologies which will be

SIGITE'15, September 30–October 3, 2015, Chicago, IL, USA.

DOI: http://dx.doi.org/10.1145/2808006.2808023.

demonstrated in the lab come in the second layer. There are typically multiple technologies that can be employed to demonstrate the same principle. Only widely accepted technologies are explained. Then, the software tools that will be used in the lab are introduced in the Tool layer. Instructions for installation, configuration and integration are included when needed. Finally, the effects of the lab are tested and examined by students through observations.

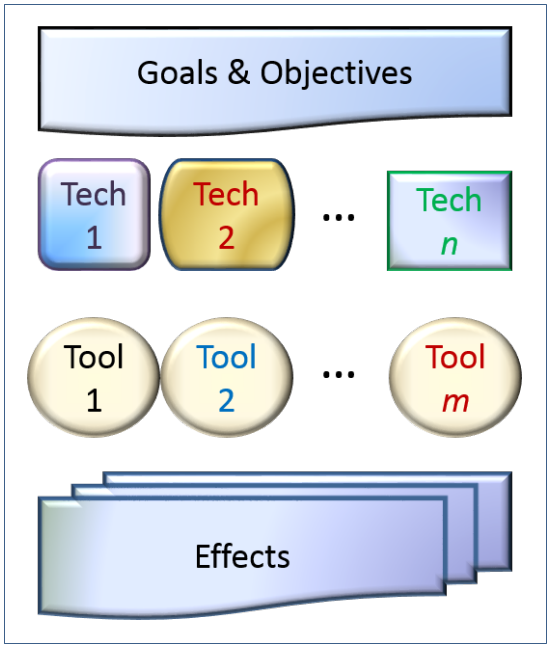

Figure 1: Layered and modular lab design

The layered design improves the labs' robustness, permitting a lab to be tailored for different purposes. By knowing the goals and objectives of a lab, instructors can determine whether they want to use this lab in their class. Descriptions of principles and concepts aid students in focusing on the topics, while actively performing the tasks in a lab.

The modular design allows instructors to easily revise a lab to fit their own condition. Some colleges may have certain software products installed, whereas others may not. The layers of Technologies and Tools create a workspace where components function in a manner of "plug-and-play". Adding or removing some of the modules (tools or technologies) will not affect the goals and objectives of a lab.

Different tools may generate varied effects although the same technology is employed. Varied integration of technologies and tools will typically produce distinct results. By observing and examining these effects, students will be able to discover the difference and motivated to adopt an integration that better fits their needs. In addition to obtaining useful skills, these activities introduce big ideas to students [24] and cultivate innovation in the hands-on activities.

Another advantage of the layered and modular design is the support of sustainability in a rapid-growth era of IT technologies in the sense that the developed labs can be used for a relatively longer time [14]. When advanced technology comes up or a new software tool is released, the obsolete one can be replaced with a new version without affecting the whole structure of a lab.

3. LAB TOPICS

Some of our labs are designed to help students learn how to use security tools such as Snort [28]. Some are developed to help students gain a better understanding of the material presented in class lectures such as cryptographic algorithms. While others are a combination of both. We provide a brief description of the topics covered by our labs in this section. Detailed descriptions can be found from our web page [32].

3.1 Computer Security

Labs on computer security are constructed to demonstrate security features built in a single computing system. This system can be configured as a workstation or a server. Additional computers may be needed for testing purposes. We have developed five labs in this class. The topics include:

- **Evasion and Defense:** This lab demonstrates how an attacker, breaking into a system, can hide the events of this breach and create files which are not visible with commonly used technologies. These "hidden" files can be used to collect data or launch executables when triggered. Techniques for evasion such as Windows Alternate Data Streams (ADS) [2] and removal of shell history and log entries are demonstrated. As a defense against these evasion techniques, tools are introduced to discover hidden ADS files in Windows NT file systems. Instructions are given to configure a Linux *syslog* system to store auditing data on a remote server.

- **Capability and System Hardening:** The educational objective of this lab is to help students gain a better understanding of system hardening principles as well as learn basic skills for system hardening. This is accomplished by introducing essential hardening techniques and observing their effects, which is so called "learning by discovering". Examples include the utilization of Linux capability to grant programs minimum privileges instead of using the Set-UID feature. Other hardening techniques include detection of user accounts with an empty password, group-writable and world-writable files, world-writable directories that have sticky bits set, unauthorized SUID/SGID system executables and sniffing programs. Configuration parameters for defending against buffer-overflow attacks are also examined.

- **Password Cracking:** A common question in IT security is why we need a strong password which is difficult to memorize. This lab is to show that a short or simple password can be easily predicted by an attacker. Tools for dumping password files such as fgdump [13] and password cracking such as John the Ripper [23] are used to implement this lab.

- **Introduction to SELinux:** The Security Enhanced Linux (SELinux) [18] provides an opportunity to limit the damage of a broken process. In the first lab on SELinux, basic concepts and knowledge are introduced. After completion of this lab, students are expected to learn how to discover the current SELinux status in a system, use SELinux commands to accomplish primary administration tasks, and gain a better understanding of the targeted policy running in a system.

- **SELinux Policy:** One of the annoying things in utilizing SELinux is the large volume of access denies that are generated by SELinux policies. In the second lab on SELinux, we lead students to discover why some accesses are denied and where these events are logged. If these denies are not expected, how to fix them. Techniques include discovery and correction of conflicts in security contexts, creation and integration of policy modules and utilization of Boolean variables. We use an Apache web server as a real-world example to demonstrate the benefits and best practices for hardening a web server by using SELinux.

3.2 Network Security

Labs in this class are designed to demonstrate the technologies to secure and monitor network traffics. Relevant tools are introduced. Five labs have been developed on network security. The topics include:

- **Introduction to SNORT:** Snort [28] is a well known and commonly used intrusion detection system. Skills to install and configure a Snort system are introduced. Students will also learn how to write Snort rules and test their effects.

- **Penetration Test:** This lab is designed to help students gain hands-on experience on how a hacker gains accesses to a system. Tools included in Kali Linux [16] are introduced to break into a Linux system through the Internet.

- **PKI Setup with OpenSSL:** The Public Key Infrastructure (PKI) is a widely used technology for distributing public-key certificates authentically. The relevant concepts and knowledge are presented in class lectures. This lab is designed to help students gain a better understanding of PKI: how it works and how to use it. In addition, students will also learn how to build a PKI when needed. Software packages included in OpenSSL [21] are used to establish Certificate Authorities (CAs) and generate certificates for application servers.

  An HTTPS web server is set up by using a software package (*s_server*) included in OpenSSL to observe and test proper configurations of a server certificate. For example, in order for a client to verify a certificate presented by a server, the certificate of the CA, which issued the certificate to the server, must be loaded to the web browser of the client. The certificate of the HTTPS server must be located correctly. Students will observe various errors generated by mismatched subject names on a server certificate and a misplaced server certificate.

- **VPN and Kerberos Policy:** As a network authentication protocol, the Kerberos [19, 17] is presented in a class lecture. The first part of this lab is designed for the students to gain hands-on experience with the configuration of Kerberos policy in an enterprise environment of Windows Active Directory networks. Students are required to study the settings in Kerberos policy and make changes to at least one setting. In addition, students will need to justify why they need to make such changes.

  VPN (Virtual Private Network) is the most widely used technology for secure communications over the public Internet. The second part of this lab is constructed to help students gain hands-on experience with VPN setup. OpenVPN [22] is used to set up point-to-point VPN connections among Linux systems. To set up a VPN connection, students need to generate certificates for the server and clients and allocate these certificates correctly.

- **Spoofing and Man-in-the-Middle Attack:** On a Local Area Network (LAN), a computing device is addressed by its Media Access Control (MAC) address. However, a message is routed on the Internet with an Internet Protocol (IP) address. The map from an IP address to a MAC address is accomplished by the Address Resolution Protocol (ARP) on a LAN. This lab is constructed to demonstrate how an attacker can perform an ARP spoofing and use the poisoned ARP cache to conduct Man-in-the-Middle (MitM) attacks. Ettercap [12] is used to implement this lab.

3.3 Cryptography

It is not easy to teach cryptography in an IT security course due to the lack of mathematical background knowledge of students and time constraints for each topic. Beyond the math, we would want IT students to gain a better understanding of existing cryptosystems such as RC4, DES, 3DES, AES and RSA [29]. We have developed one cryptography lab to demonstrate various cryptosystems, their requirements to work correctly, their encryption and decryption speeds and the differences between operation modes of block ciphers. The padding scheme of PKCS#5 standard is also introduced. Tools included in OpenSSL [21] are used to perform the activities in this lab. Students found that the examination and observation on the effects of different operation modes was very interesting and helpful for them to understand the material presented in class lectures.

3.4 Application Security

A lab has been developed on the topic of web server security. This lab is designed for students to gain first-hand experience on Apache web server basics and advanced configurations as well as web server security. Another component is to investigate Internet routing characteristics using the tool *pchar* which is a built in command in Linux systems to gain routing information of a packet.

First, basic instructions are given to set up a web server using Apache. Then the *.htaccess* file is introduced to control the access to the web page. Instructions are given for several scenarios. Finally, the technology of virtual host, including port-based virtual hosting and name-based hosting, is introduced. Overall, this lab covers the basic techniques to secure a web server and the contents of a web page.

4. LAB ENVIRONMENT AND SOFTWARE

In order for other universities and colleges to use our labs, we keep the following two aspects in mind when we develop our labs:

- **Low-cost and consistent environment:** All of our labs are tested and implemented in a virtual environment. This virtual environment can be set up by using

a commercial cloud computing or virtualization system such as VMware vCloud or Lab Manager [33]. Virtualbox [31] and VMware Player are other options in the domain of free software. Operating systems include Windows servers and workstations and Linux systems.

- **Open source software:** All of the tools we use to implement the labs are from the domain of open source. They are either introduced in the list of top 100 network security tools [27], on their own web pages, or included in the distribution of Kali Linux [16]. No commercial products are used in the labs.

5. EVALUATION

Our labs have been used in IT security courses at junior and senior levels. When students have finished a lab assignment, we encourage students to fill and submit a survey questionnaire voluntarily to evaluate the lab. In total, we have collected 105 responses from the students during the period of 2014 spring to 2015 spring. All of the responses are used for analysis in this study. Due to space limit, we present sample statistics (figures) only in this section. A complete set of the statistics with figures are available on request.

5.1 Metrics and Methodology

We design a questionnaire of 12 questions that can be classified into two categories: the *efficiency* of the lab and the *effectiveness* of the lab.

Efficiency measures the extent to which time and effort are well used to finish a lab. It is quantified by the following metrics:

- the level a student is prepared for a lab;
- the clearness of the instructions in a lab;
- the clearness of the materials in a lab;
- the level of difficulty of a lab;
- the level of student's interest in a lab;
- the approximate time spent for a lab.

Effectiveness measures the outcomes of learning as a result of finishing a lab assignment. It is quantified by the following metrics:

- the level students agree on hand-on experience with the tools introduced in a lab;
- the level students agree on the value of a lab as a part of this course;
- the level students agree on gaining more interests in this class as a result of finishing a lab;
- the level students agree on achieving the learning objectives of a lab.

At the bottom, a question, such as "Comments you would like to add", is given to look for additional feedback and comments from students.

Multiple choices were provided with each question. Most of the questions asked the level students agree on about the lab. The levels included *Strongly disagree*, *Disagree*, *Neutral*, *Agree* and *Strongly agree*. Self-explainable options were provided to other questions.

5.2 Efficiency

More than 75% of the surveyed students responded positively to the questions of efficiency measures. For example, 76% of the surveyed students agreed or strongly agreed that the lab instructions were clear (Fig. 2).

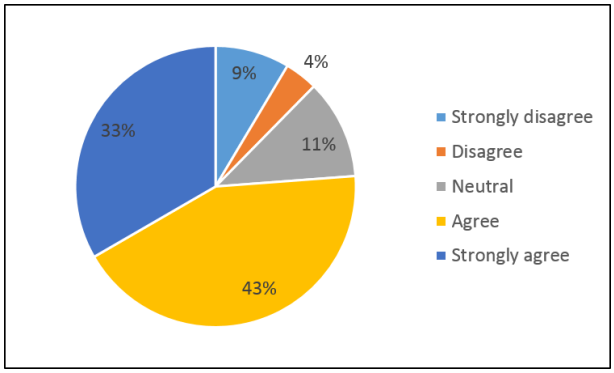

Figure 2: The lab instructions are clear.

As shown in Figure 3, 60% of the students were able to finish a lab within two hours and another 24% were able to finish within four hours. Only 5% of the students spent more than seven hours to finish a lab due to various issues, such as virtual environment, Linux commands and configuration problems.

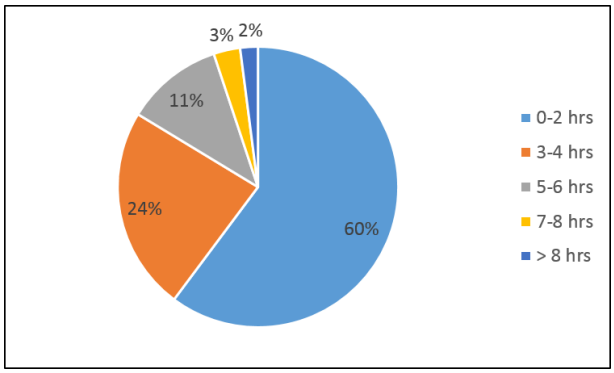

Figure 3: Approximate time spent for this lab

Similar statistics were obtained from other survey questions, which include: more than 75% of the surveyed students agreed or strongly agreed that they were well prepared for a lab; more than 78% strongly agreed or agreed that they understood the material covered in a lab; 67% thought that the level of difficulty of a lab was average with 17% of being difficult and 11% of being easy; and 83% of the students had high or very high interest in a lab.

5.3 Effectiveness

Overall, more than 85% of surveyed students agreed or strongly agreed that the labs were effective to help them learn. For example, 86% of students thought that the time they spent for a lab was worthwhile (Fig. 4).

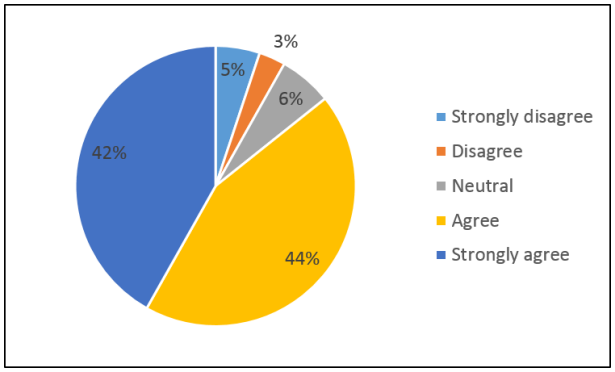

Figure 4: The time I spent for this lab is worthwhile.

Similarly, over 91% of the students agreed or strongly agreed that the lab was a valuable part of this course (Fig. 5). Only 5% did not think so.

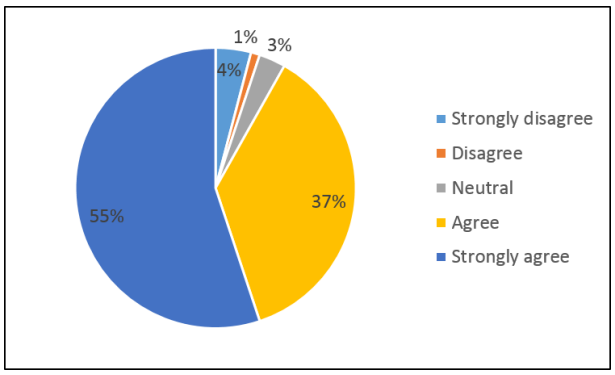

Figure 5: The lab is a valuable part of this course.

Responses to other questions in this category are all positive. For example, over 83% of the students were more interested in the class after finishing a lab. More than 91% of the students gained hands-on experience with the tool introduced in a lab.

5.4 Student's Comments and Lessons

We have observed several issues from the student's comments on conducting these hands-on activities, including:

- **Lab environment:** Some students use their own laptops to set up a virtual environment. This could be rough due to the capacity of the computer and storage, the student's familiarity with the software and the version of the Linux distribution the students use. We recommend to set up a unique environment and test the labs in this environment to improve the efficiency of the hands-on activities.
- **Programming skills and Linux commands:** Some students may not have the required programming skills. Some others may not know the right Linux commands. Instructors can tailor each lab assignment to fit to the audience.
- **Troubleshooting and time management skills:** Some students wanted to finish a lab in a short time, but ended up with a much longer time due to various issues. For example, students worked in the day time and wanted to finish a lab in night when they were tired. A simple typo could result in few hours of debugging. To avoid these issues, an instructor can encourage the students to ask for help when they are stuck or take a break while they are struggling.

6. RELATED WORK

A collection of hands-on guides to some IT security tools has been presented by Boyle [3]. A good lab manual for IT security has been published by Whitman *et al.* [34, 35] which focuses on Windows and Linux security.

A number of projects have been supported by NSF to develop security labs for education in Computer Science (CS). The SEED project [8] has been supported by three NSF grants. In the last decade, they have developed 29 security labs and published a number of papers [11, 10, 9]. Their labs can be categorized into three classes: 1) Vulnerability and attack exercises are designed to illustrate vulnerabilities in detail. 2) Design and implementation labs help students understand the knowledge to develop a secure system. 3) Exploration exercises encourage students to explore existing security functionality. Some of the labs can be used for IT security courses with revisions. A collection of modules, projects and lab assignments has been developed by the SWEET project [4] for secure web development. Security components, including modules and labs, are "injected" to existing courses at Towson University [30].

7. CONCLUSION

We have developed twelve hands-on labs that can be used to enhance the security component in undergraduate IT education. Tools in the domain of open sources are used to implement these labs. The labs have been tested in virtual environments and used in security courses at junior and senior levels. Lab descriptions can be downloaded from our web page [32] and hard copies are available on request.

These labs have been evaluated by students and the data shows that they are efficient and effective to help them gain hands-on experience with introduced software tools and a better understanding of the material presented in class lectures. A majority of the surveyed students developed a greater interest in the course and acknowledged that the labs were a valuable part of the course.

Acknowledgment

We would like to thank the three anonymous reviewers for this paper. This work is supported by the National Science Foundations (NSF) TUES grants (Award#: 1140310 and Award#: 1140308).

8. REFERENCES

[1] R. Abler, D. Contis, J. Grizzard, and H. Owen. Georgia Tech information security center hands-on network security laboratory. *IEEE Transactions on Education*, 49(1):82–87, 2006.

[2] L. Abrams. Windows alternate data streams. online, April 2012. `http://www.bleepingcomputer.com/tutorials/windows-alternate-data-streams/`. last retrieved on February 28, 2015.

[3] R. Boyle. *Applied Information Security: A Hands-On Guide to Information Security Software*. Prentice Hall, Upper Saddle River, NJ 07458, USA, new edition edition, July 2009.

[4] L.-C. Chen. Secure web development teaching modules. online, June 2011. `http://csis.pace.edu/~lchen/sweet/`, last retrieved on February 28, 2015.

[5] P. J. Denning. The field of programmers myth. *Communications of the ACM*, 47(7):15–20, 2004.

[6] P. J. Denning and P. A. Freeman. The profession of IT computing's paradigm. *Communications of the ACM*, 52(12):28–30, 2009.

[7] D. Dobrilovic, V. Brtka, I. Berkovic, and B. Odadzic. Evaluation of the virtual network laboratory exercises using a method based on the rough set theory. *Computer Applications in Engineering Education*, pages 1–11, 2009.

[8] W. Du. Seed: Developing instructional laboratories for computer security education. online, March 2011. `http://www.cis.syr.edu/~wedu/seed/`. last retrieved on February 28, 2015.

[9] W. Du. Seed: Hands-on lab exercises for computer security education. *Security Privacy, IEEE*, 9(5):70–73, Sept 2011.

[10] W. Du, K. Jayaraman, and N. B. Gaubatz. Enhancing security education with hands-on laboratory exercises. In *5th Annual Symposium on Information Assurance (ASIA'10)*, pages 56–61, June 2010.

[11] W. Du and R. Wang. SEED: A suite of instructional laboratories for computer security education. *Journal on Educational Resources in Computing (JERIC)*, 8(1):1–24, 2008.

[12] Ettercap. Ettercap home page. online. `http://ettercap.github.io/ettercap/`, last retrieved on February 28, 2015.

[13] fizzgig. fgdump: A tool for mass password auditing of windows systems. online, Semptemberl 2008. `http://foofus.net/goons/fizzgig/fgdump/`. last retrieved on February 28, 2015.

[14] S. Goel, D. Pon, P. Bloniarz, R. Bangert-Drowns, G. Berg, V. Delio, L. Iwan, T. Hurbanek, S. P. Schuman, J. Gangolly, A. Baykal, and J. Hobbs. Innovative model for information assurance curriculum: A teaching hospital. *Journal on Educational Resources in Computing (JERIC)*, 6(3):1–15, 2006.

[15] C. Heien, R. Massengale, and N. Wu. Building a network testbed for Internet security research. *Journal of Computing Sciences in Colleges*, 23(4):73–79, 2008.

[16] Kali Linux. Project home page. online. `https://www.kali.org/`. last retrieved on February 28, 2015.

[17] J. T. Kohl, B. C. Neuman, and T. Y. T'so. The evolution of the kerberos authentication system. In F. Brazier and D. Johansen, editors, *Distributed Open Systems*, pages 78–94. IEEE Computer Society, Los Alamitos, CA, USA, 1994. Available at `http://web.mit.edu/kerberos/www/papers.html` Last retrieved March 11, 2010.

[18] National Security Agency. SELinux documentation. online, May 2012. `https://www.nsa.gov/research/selinux/docs.shtml`. last retrieved on February 28, 2015.

[19] B. C. Neuman and T. Ts'o. Kerberos: an authentication service for computer networks. *IEEE Communications Magazine*, 32(9):33–38, 1994.

[20] M. O'Leary. A laboratory based capstone course in computer security for undergraduates. *ACM SIGCSE Bulletin*, 38(1):2–6, 2006.

[21] OpenSSL. Home page of the OpenSSL project. online. `http://www.openssl.org/`, last retrieved on February 28, 2015.

[22] OpenVPN. Your private path to access network resources and services securely. online, March 2010. `http://openvpn.net/`. last retrieved on February 28, 2015.

[23] Openwall. John the Ripper password cracker. online, March 2010. `http://www.openwall.com/john/`. last retrieved on February 28, 2015.

[24] S. Papert. What's the big idea? Toward a pedagogy of idea power. *IBM Systems Journal*, 39(3-4):720–729, 2000.

[25] D. Schweitzer and J. Boleng. Designing web labs for teaching security concepts. *Journal of Computing Sciences in Colleges*, 25(2):39–45, 2009.

[26] D. Schweitzer and W. Brown. Using visualization to teach security. *Journal of Computing Sciences in Colleges*, 24(5):143–150, 2009.

[27] SecTools.Org. Top 125 network security tools. online. `http://sectools.org/`. last retrieved on February 28, 2015.

[28] SNORT. Snort home page. online. `http://www.snort.org/`, last retrieved on February 28, 2015.

[29] W. Stallings and L. Brown. *Computer Security: Principles and Practice*. Pearson Prentice Hall Press, Upper Saddle River, NJ 07458, USA, 3rd edition, 2015.

[30] B. Taylor, S. Kaza, and E. Hawthorne. Security injections @ towson. online. `http://cis1.towson.edu/~cssecinj/`, last retrieved on February 28, 2015.

[31] VirutalBox. Home page of the project. online, l. `https://www.virtualbox.org/`. last retrieved on February 28, 2015.

[32] X. Wang, Y. Bai, and G. C. Hembroff. ITSEED: Active-learning laboratory experiments for IT SEcurity EDucation. online, June 2013. `http://www.ece.mtu.edu/~xinlwang/itseed/labs.html`. last retrieved on February 28, 2015.

[33] X. Wang, G. C. Hembroff, and R. Yedica. Using vmware vcenter lab manager in undergraduate education for system administration and network security. In *Proceedings of the 2010 ACM Conference on Information Technology Education*, SIGITE '10, pages 43–52, New York, NY, USA, 2010. ACM.

[34] M. E. Whitman and H. J. Mattord. *Hands-on Information Security Manual*. Course Technology Press, Boston, MA, United States, 3rd edition, 2008.

[35] M. E. Whitman, H. J. Mattord, and A. Green. *Hands-on Information Security Lab Manual*. Course Technology, Cengage Learning, Boston, MA, United States, 4th edition, December 2013.