Teaching Information Security with Virtual Laboratories

Dinghao Wu, John Fulmer and Shannon Johnson

Abstract With rapid advances in the online education and computer virtualization technology, laboratories leveraging virtual machines, or Virtual Hands-on Laboratories, have become one of the key education resources in many fields. In the College of Information Sciences and Technology at the Pennsylvania State University, we have developed a set of virtual hands-on laboratories and used them in many technology courses. These virtual labs are collaboratively developed by instructors, instructional technologists, system administrators, student learning assistants, and interns. They have become an integrated component of many courses taught in our college to enhance hands-on learning experience in the information sciences. In this article, we introduce our teaching experience on using virtual hands-on laboratories in an introductory information security course as part of the curriculum of the Security and Risk Analysis major. Our teaching experiences show that the hands-on virtual labs are quite effective in learning, especially on connecting theory to practice. We expect they will continue to play a critical role in our curriculum as online education becomes indispensable.

Introduction

As education becomes more globalized and the computer virtualization technology becomes more mature, laboratories leveraging virtual machines, or *Virtual Hands-on Laboratories*, have become more and more critical for both online and resident education. Students need access through the Internet from anywhere at anytime. Educational organizations and educators need to lower the cost of laboratory setup, which in turn

D. Wu (✉) · J. Fulmer · S. Johnson
College of Information Sciences and Technology,
Pennsylvania State University, University Park, PA 16802, USA
e-mail: dwu@ist.psu.edu

J. Fulmer
e-mail: jfulmer@ist.psu.edu

S. Johnson
e-mail: sjohnson@ist.psu.edu

J. M. Carroll (ed.), *Innovative Practices in Teaching Information Sciences and Technology,*
DOI 10.1007/978-3-319-03656-4_16, © Springer International Publishing Switzerland 2014

enables more accessible laboratory resources. Virtual labs can also mitigate security concerns in using practical security attack and defense tools for educational purpose.

In the College of Information Sciences and Technology (IST) at the Pennsylvania State University (PSU), we have developed a set of virtual hands-on laboratories and used them in many technology courses. These virtual labs are collaboratively developed by instructors, instructional technologists, system administrators, student learning assistants, and interns. There are several models adopted in developing these labs. An instructor can initiate a lab topic and work with instructional technologists and system administrators to develop the lab. We also have dedicated instructional technologists who can initiate and design labs with interns and learning assistants, and then instructors can select certain labs for each course.

Most of the virtual hands-on labs developed and used are in the network and security courses. The virtual hands-on labs developed include password cracking, computer vulnerability scan and patch, cryptography, penetration testing (pentesting), network security monitoring, intrusion detection, Cisco router setup, DNS, Cisco DHCP, Cisco Router ACL Configuration, Wireshark, SNMP, locating network vulnerabilities, Snort (X-mas tree scan), website hacking, Cross Site Scripting (XXS Internet Banking), Linux hardening, etc. Typically each class selects three to six labs from the list that is designed for the course. We have managed that different classes take mostly different virtual labs so that students do not work on the same lab from different classes.

These virtual labs have become an integrated component of many courses taught in IST to enhance hands-on learning experience in the information sciences. In this article, we introduce our teaching experience on using virtual hands-on laboratories in an introductory information security course, SRA 221, as part of the curriculum of the Security and Risk Analysis major. Our teaching experiences show that the hands-on virtual labs are quite effective in learning, especially on connecting theory to practice. We expect they will continue to play a critical role in our curriculum as online education becomes indispensable.

Virtual Hands-On Laboratories

Many courses in our college have adopted virtual environments for the hands-on lab portion. Students log into a virtual environment with assigned virtual computers for them to use. The virtual machines have been set up using the college's software, hardware and special network setups. Each student, or each team in some cases, is assigned one or several virtual machines located at PSU for each hands-on lab. Students can use PSU computers or their home computers to view the virtual computer desktop. These virtual machines are not connected to the rest of the PSU network or Internet for security purposes. What is required for students to complete the virtual labs includes a computer, an internet connection on the computer, a web browser, and the VMware vSphere Client software.

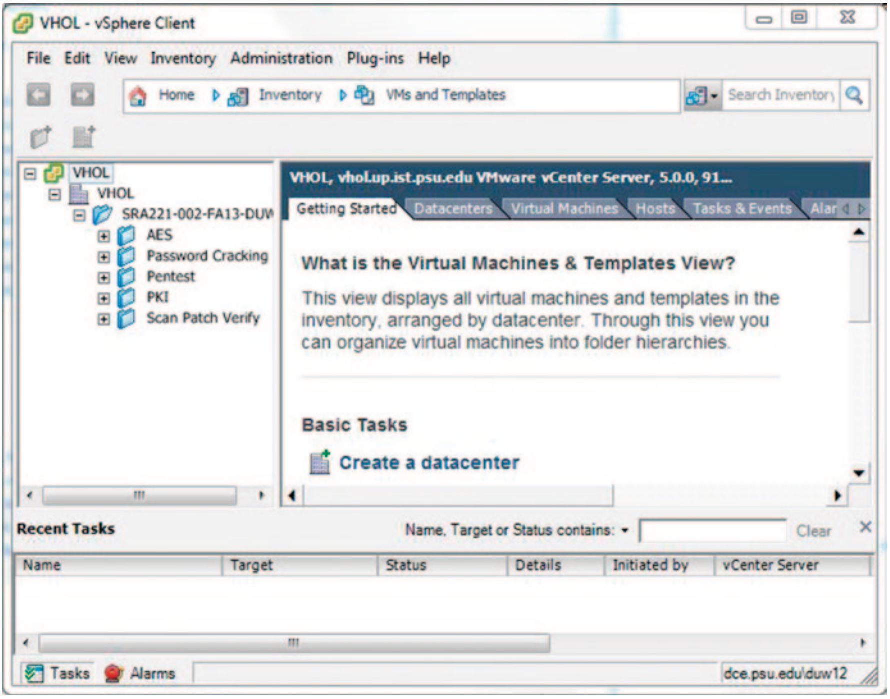
We have implemented the Virtual Hands-on Laboratories in VHOL, a system to provide students with a safe environment to do security-related labs. Each student only sees their own labs, organized into folders named for each course and section, lab name, and student, with permissions set on the folders. One of the key considerations of security lab configuration is isolation. Not only that in some cases the students need root access on their virtual machines, but also that the instructor and the college need to consider criminal liability of potential inappropriate use of security tools involved in the lab.

Our VHOL infrastructure is build with the VMware vSphere software. Physically, the system is a cluster of 17 Dell PowerEdge R610 machines, with varying amounts of resources on each machine, all connected to two networks, namely, the production internet-accessible network and a private internal network. The private internal network handles communication between the virtual machines (VMs), so that labs do not conflict with each other, and are not restricted to running on a single physical server. The physical server run the VMware vSphere 5.0 software, and are managed by the VMware vCenter, which runs as a virtual server in a separate cluster.

VMware allows the administrator to configure a virtual network for each class with no connection to any other classes' virtual network or the outside world. The administrator then creates an account for each of the students in VMware. Permissions are granted for each student to access a folder within the class network. The accounts for the instructor and teaching assistant are granted access to all of the student folders and the instructor folder. Then the administrator deploys the machines appropriate for the individual lab to each student's and instructor's folder. The students access their virtual machines through the vSphere Client software. We were able to deploy the classes and labs more efficiently by writing Powershell scripts to automate all of the steps. In a typical semester, the college would deploy between 3,000 and 5,000 individual virtual machines. Many of the security labs would require two virtual machines, an attacker (usually BackTrack 5 Linux) and a victim (usually Microsoft Windows XP SP1).

In the current incarnation, the labs are suggested by instructors, then created by the Office of Information Technology. The Office of Instructional Design tests them and writes instructions on how to complete the lab, before the lab is presented to the originating instructor. We have built around 20 different lab templates, some of which are used multiple times to teach different concepts. Once a lab template is built and tested, the instructor gives an estimate on when the lab should be deployed for students to use. The deployment process is handled by a script that creates the lab and student folders, grants permissions, clones the template to each student, starts the individual VMs, and changes the computer name and IP address. We also set the VMs to be reset back to a fresh version if they are powered off. This allows students to revert back to a fresh deployment if they are having trouble completing the lab and want to start over.

The types of virtual machines required for the labs depend on the practical applications of theories that we teach in the class. We primarily use open-source systems and applications, e.g., the latest versions of Ubuntu loaded with the latest versions of the appropriate industry standard open-source software. For example, our network security labs use Wireshark and Snort; the forensics labs use DD, Autopsy,



What is the Virtual Machines & Templates View?

This view displays all virtual machines and templates in the inventory, arranged by datacenter. Through this view you can organize virtual machines into folder hierarchies.

Fig. 1 Virtual laboratories

Santoku, and the Sleuthkit; the hacking labs use the tools available on BackTrack; and the cryptology labs use PGP, Cryptool, and even an Enigma Machine simulator. The use of industry standard tools improves the utility of the labs and the marketability of the students trained.

A screenshot of the deployed virtual hands-on labs for the Security and Risk Analysis (SRA) 221 class is shown in Fig. 1. After a student or a team logs into the virtual network, the student will find a list of labs available under their user names on the left. By clicking on a specific lab, the virtual machines deployed for the lab are shown. Then the student can "power on" the virtual machine, log into it, and start the lab following a lab manual. The student can exit the virtual network without shutting down the virtual machines, which allows the lab to be completed with multiple sessions at different locations and time. Students can also "power" off a virtual machine to role back its configuration to the initial setting, which gives the students a way to start over the ongoing lab.

These hands-on labs are usually intended to be team-based. Each team typically consists of two to three students who collaborate to complete the laboratory assignment tasks and write up a lab report for grading. Some labs require collaborations across teams. For example, the public key crypto lab requires a student from one team to send an electronically signed and encrypted message to a student in a different team who will decrypt the message and verify the electronic signature.

The lab assignments often require students to take screenshots of key steps that they complete and include them in the lab reports for grading. Typically a lab has a number of specific requirements. A sample grading rubric is shown below (from the *Scan, Patch, and Verify* virtual lab document).

> Credit for each section of your report is as follows:
>
> Section I: Introduction (10%)
> Section II: Three tasks (60%)
>   Task 1: You are the Attacker.
>   Task 2: You are now the Victim
>   Task 3: Updating Virus Definitions.
> Section III: Vulnerability Paragraph (20%)
> Section IV: Experiment Log (10%)
>
> Note: This is a team project.

SRA 221: Information Security

Security and Risk Analysis (SRA) 221 is an introductory course on the fundamentals of information security. Students learn the principles and practices of information security, security architectures and models, aspects and methods of information security such as access control, threat models, attacks and defenses, systems security, cryptography, software security, network and web security, worms and viruses, and other Internet secure applications. The assignments include several hands-on virtual laboratories and a term paper. The course incorporates collaborative learning experiences with team-based assignments. Emphases are placed on developing and practicing skills through application of the concepts, theories, and technologies introduced in the course.

This is a three-credit course, with 150 min class time each week. It serves as the second technical course in the SRA major. The objectives of the course are for the students to understand the fundamental concepts and issues of information security, to study the mainstream information security technologies, to learn how real world enterprises are attacked, and to apply information security knowledge to defend against a set of widely known security attacks.

SRA 221 covers security fundamentals, classic cryptography, symmetric key cryptography, public key cryptography, hash functions, random numbers, secret sharing, information hiding, authentication, access control, security protocols, software security, malware, and web security. It also includes a hands-on demo on buffer overflow.

The course includes six virtual labs that allow students to get hands-on experience on password cracking and authentication, computer vulnerability scan and patch, cryptography, penetration testing, and network security monitoring (intrusion detection). We describe below these labs briefly on their setup, objectives, and our teaching experiences.

Password Cracking Lab

The first lab in SRA 221 is Password Cracking. The purpose of this lab is to learn the fundamentals of password storing, encrypting, and cracking, as well as obtain some experience in password cracking and recovery. Students use a security tool called Cain to crack passwords in a captured SAM file.

Password cracking software uses a variety of approaches, including intelligent guessing, dictionary attacks, and automation that tries every possible combination of characters. In general, there are three common types of attacks: brute force attack, dictionary attack, and cryptanalytic attack.

A brute force attack is a very powerful form of attack, though it may often take a long time to work, depending on the complexity of the password. The attack tries all possible combinations of numbers and letters, hashes them and compares the hashed results against the hashed passwords. Passwords that are composed of random letters numbers and characters can be cracked by this type of attack.

A simple dictionary attack is usually the fastest way to break into a machine. A dictionary file, a text file full of dictionary words, is loaded into a cracking application, which is run against user accounts located by the application. The power of dictionary attacks comes from understanding the ways in which most people vary names and dictionary words when attempting to create a password. Cracking tools can often detect “clever” ways of manipulating words to hide their origin. Often a list of common rules is applied to each word in the dictionary. For example, typical rules might include alternating upper and lowercase letters; spelling the word forward and then backward, and then fuse the two results (e.g., cannac); and adding a number to the beginning or end of each word.

Another well-known form of attack is the cryptanalytic attack. Cryptanalysis is the study of methods to obtain plaintext by analyzing the cryptographic algorithm used to encrypt it. An example of cryptanalytic attack in password cracking is a time-memory trade-off where the basic idea is to use a large table of pre-computed hash values to save time to compute the hash values. An advanced attacker can create the pre-computed hash table in a specially organized way (e.g., rainbow table) for a better performance.

In this lab, students perform three tasks:

Task 1: Brute force attack,
Task 2: Dictionary attack, and
Task 3: Cryptanalytic attack.

After students complete each task, they need to report the results in a table. A sample report table is shown in Table 1 (adopted from the *Password Cracking* virtual lab document).

Table 1 A sample reporting table

Dictionary Attack Results

Dictionary:____ **Elapsed Time:____**

User ID	**Plain Password**
kmiller	
smacman	
gkoch	
mjones	
tgriffin	
rklatt	
nboyle	
esmith	
jcollins	
aharris	
hdell	
fmoore	

In this lab, students are expected to apply the concepts and theories introduced in the lectures with practical password cracking tools. The concepts touched by this lab include brute-force attacks (or exhaustive search), shortcut attacks, dictionary attacks, rainbow table, and hashing. By completing this lab, students will be able to strengthen and reinforce their understanding of the subjects by connecting theory to practice with hands-on experience.

Preset passwords should not be stored in clear text in computers, but we have to compare the password typed in by users to the preset password. The hashing solution to this dilemma is taught in the lectures, while this lab enables students to apply hashing to experience the solution and test its security in a real setting.

Students are warned beforehand that they should be cautious on applying the password cracking tools outside of the virtual network. It is often a surprise to the SRA 221 students, most of whom are sophomores, that how easy password cracking can be performed and how vulnerable passwords are. It is one of the important lessons that the students realize the importance of information security and how insecure the current information systems are. This lab achieves this goal by allowing students to crack passwords to gain first hand experience.

Scan, Patch, and Verify Lab

The objectives of this lab are to scan a machine for vulnerabilities, attack the machine with a specific vulnerability found, patch the victim machine to defend the attack, and rescan to verify the vulnerability is patched.

Each team has two virtual machines, one called attacker, and the other victim. Three tasks are performed:

Task 1: You are the Attacker.
Task 2: You are now the Victim.
Task 3: Updating Virus Definitions.

In Task 1, the team attacks the victim machine. First, the team scans the victim machine for vulnerabilities using Nessus, and then attacks the victim machine on a specific vulnerability found. In Task 2, the team is on the victim side and attempts to defend the attack. The team updates and patches the vulnerability. In Task 3, the team updates the virus definitions on the victim machine.

In this lab, students are required to scan for vulnerabilities and patch the vulnerabilities found with a published software patch. By conducting such practice, students realize the vulnerability and insecurity of the current information systems that our society relies on. They also gain the first-hand experience on the current industry practice on software reliability and security patching.

Crypto Labs

We have two crypt labs, one on symmetric key crypto and the other on public key crypto. Our first crypto lab uses the Advanced Encryption Standard (AES) Cipher, the current standard for data encryption. In cryptography, AES is a symmetric-key cipher. The standard comprises three block ciphers, AES-128, AES-192, and AES-256, adopted from a larger collection originally published as Rijndael. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192, and 256 bits, respectively. The AES ciphers have been analyzed extensively and are now used worldwide, as was the case with its predecessor, the Data Encryption Standard (DES). AES is the first publicly accessible and open cipher approved by NSA for top secret information.

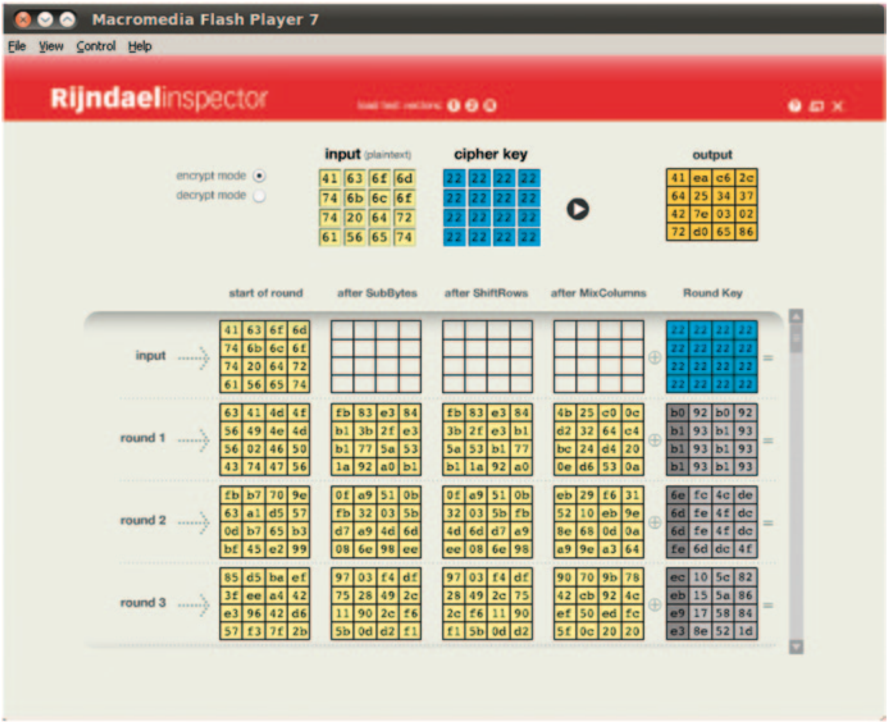
The objectives of this lab are to get basic understanding of the modern cryptography using AES. At the completion of this lab, students should be able to encrypt and decrypt messages using a hex editor and then the AES algorithm, and to capture secret messages using Wireshark from unsuspecting classmates or teams.

The configuration of this lab consists of a Linux virtual machine, the Empathy instant messaging software for sending messages to classmates, Bless Hex Editor as the text to hex converter, Rijndael inspector for AES encryption of the hex code, and Wireshark for capturing other classmate's transmission of secret messages.

Students perform the following tasks:

Task 1: Log on to VMware and watch an animation
Task 2: Create an instant messaging account and encrypt a message.
Task 3: Convert a text message to hex (Bless Hex Editor).
Task 4: Encrypt message with key (Rijndael inspector). A screenshot of the Rijndael inspector is shown in Fig. 2. Students can see the progress as they perform encryption and decryption on the messages.
Task 5: Begin packet capture in Wireshark.
Task 6: Send the encrypted message and key to classmate.
Task 7: Receive the encrypted message and key from classmate and decrypt.
Task 8: Search Wireshark for yet another classmate's message and key.

After complete the lab, students submit a lab report with screenshots showing the completion of each task. The grading rubric specifies that the eight tasks consists



input (plaintext)	cipher key	output

41 63 6f 6d
74 6b 6c 6f
74 20 64 72
61 56 65 74
	22 22 22 22
	22 22 22 22
	22 22 22 22
	22 22 22 22
		41 ea c6 2c
		64 25 34 37
		42 7e 03 02
		72 d0 65 86

	start of round	after SubBytes	after ShiftRows	after MixColumns	Round Key
input	41 63 6f 6d 74 6b 6c 6f 74 20 64 72 61 56 65 74				22 22 22 22 22 22 22 22 22 22 22 22 22 22 22 22
round 1	63 41 4d 4f 56 49 4e 4d 56 02 46 50 43 74 47 56	fb 83 e3 84 b1 3b 2f e3 b1 77 5a 53 1a 92 a0 b1	fb 83 e3 84 3b 2f e3 b1 5a 53 b1 77 b1 1a 92 a0	4b 25 c0 0c d2 32 64 c4 bc 24 d4 20 0e d6 53 0a	b0 92 b0 92 b1 93 b1 93 b1 93 b1 93 b1 93 b1 93
round 2	fb b7 70 9e 63 a1 d5 57 0d b7 65 b3 bf 45 e2 99	0f a9 51 0b fb 32 03 5b d7 a9 4d 6d 08 6e 98 ee	0f a9 51 0b 32 03 5b fb 4d 6d d7 a9 ee 08 6e 98	eb 29 f6 31 52 10 eb 9e 8e 68 0d 0a a9 9e a3 64	6e fc 4c de 6d fe 4f dc 6d fe 4f dc fe 6d dc 4f
round 3	85 d5 ba ef 3f ee a4 42 e3 96 42 d6 57 f3 7f 2b	97 03 f4 df 75 28 49 2c 11 90 2c f6 5b 0d d2 f1	97 03 f4 df 28 49 2c 75 2c f6 11 90 f1 5b 0d d2	90 70 9b 78 42 cb 92 4c ef 50 ed fc 5f 0c 20 20	ec 10 5c 82 eb 15 5a 86 e9 17 58 84 e3 8e 52 1d

Fig. 2 The AES crypto lab

of 80 % of the total credits, and an introduction of the lab report and experiment log consists of the other 20 %.

Our second crypto lab is on the public key system. Public Key Cryptography is an asymmetric crypto system. In such a system, everyone has a pair of keys. One is made public and the other is kept secret. The information encrypted by the public key can only be decrypted by the secret key, and information encrypted (or in a better word, *signed*) by the secret key can only be decrypted (or *verified*) by the public key. Public key crypto solves the key distribution problem and is very useful in e-commence due to its application on electronic signatures.

The lab document uses the following example to illustrate the use of public key crypto:

> Alice writes a love note to Bob, adds a signature to the bottom of the letter she encrypts with her secret Key, then encrypts the whole message with Bob's public key and sends it. Now Bob is the only person that can decrypt the message. Neither Alice nor Eve can decrypt the message. Only Bob can decrypt it with his secret key. He can also positively identify the author as Alice by decrypting [verifying] the signature with Alice's public key. Even if Eve can intercept a key, the only keys she can intercept are unable to decrypt anything

The lab document also has a graphical representation of the above example, for students to better understand the use of public key crypto system. This graphical illustration is shown in Fig. 3. Note that the public key crypto part actually contains steps to verify signatures of the senders, to show the important applications of the public key crypto in e-commerce.

Modern cryptography is very tedious and math-oriented, which presents a challenge on teaching and the virtual hands-on lab design. On one hand, we would like students to link the theories and math learned from the class to the practice. On the other hand, it is hard to present cryptography results to students in an intuitive and sensible way. In our symmetric key (AES) and public key crypto labs, we especially focus on the applications of modern cryptography to get better understanding of the theories learned in class. We let students to simulate secret messaging with symmetric key crypto, and signature verification with public key crypto which is of critical importance to e-commerce.

Pentesting

A penetration testing, or pentesting, is a testing method by simulating an attack on a computer or network system. The objective of this lab is to penetrate an open port and attack a vulnerable victim machine. The lab is configured with a BackTrack 4 virtual machine (a Linux distribution for penetration testing), Nmap (for finding a vulnerable port), and Metasploit (for executing an attack).

In this lab, students will learn to use several popular security tool including Nmap, Metasploit, and tools included in BackTrack 4 for penetration testing such as password cracking tools John the Ripper and Ophcrack. We do not cover in detail on penetration testing in lectures. Since pentesting is a very important practical technique used in today's security testing, this lab is particularly important in exposing students to this field.

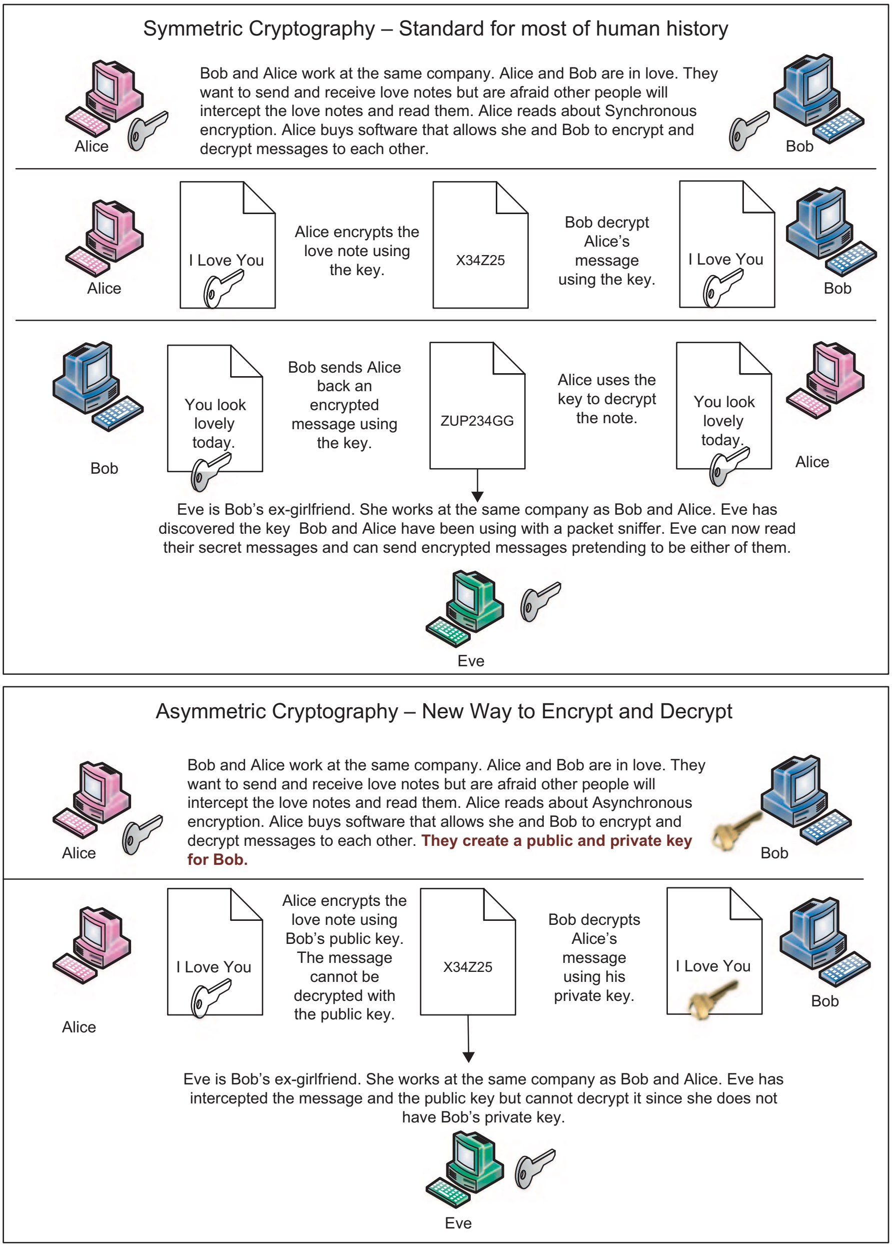
Network Security Monitoring

The network security monitoring lab is configured with a Windows virtual machine and two Intrusion Detection System (IDS) tools, BASE and Wireshark. This lab is set up as shown in Fig. 4. Between the Internet and the server we set up a firewall where intrusion detection and monitoring tasks are performed.

In this lab, students will complete the following tasks:

Task 1: Locate a system vulnerability using an IDS tool.
Task 2: Distinguish between intrusion alerts and typical traffic alerts.
Task 3: Analyze attack packets for additional information.

By completing this lab, students get hands-on experience on network monitoring and intrusion detection systems, and are bettered prepared for the subsequent course on network security (IST 451) in our curriculum.



Symmetric Cryptography – Standard for most of human history

Bob and Alice work at the same company. Alice and Bob are in love. They want to send and receive love notes but are afraid other people will intercept the love notes and read them. Alice reads about Synchronous encryption. Alice buys software that allows she and Bob to encrypt and decrypt messages to each other.

Alice encrypts the love note using the key.

Bob decrypt Alice's message using the key.

Bob sends Alice back an encrypted message using the key.

Alice uses the key to decrypt the note.

Eve is Bob's ex-girlfriend. She works at the same company as Bob and Alice. Eve has discovered the key Bob and Alice have been using with a packet sniffer. Eve can now read their secret messages and can send encrypted messages pretending to be either of them.

Asymmetric Cryptography – New Way to Encrypt and Decrypt

Bob and Alice work at the same company. Alice and Bob are in love. They want to send and receive love notes but are afraid other people will intercept the love notes and read them. Alice reads about Asynchronous encryption. Alice buys software that allows she and Bob to encrypt and decrypt messages to each other. **They create a public and private key for Bob.**

Alice encrypts the love note using Bob's public key. The message cannot be decrypted with the public key.

Bob decrypts Alice's message using his private key.

Eve is Bob's ex-girlfriend. She works at the same company as Bob and Alice. Eve has intercepted the message and the public key but cannot decrypt it since she does not have Bob's private key.

Fig. 3 Alice and Bob communicate with crypto

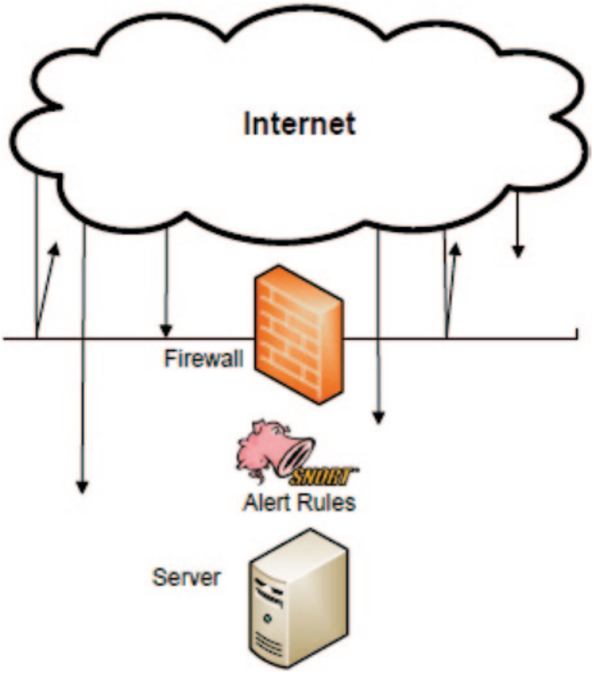


Fig. 4 Network security monitoring lab

Experiences and Discussion

Nestler et al. (Nestler et al. 2006) use a structured framework that consists of four questions to enhance hands-on understanding of security principles:

1. How do networks work?
2. How are networks vulnerable and what are the threats?
3. How do we prevent harm?
4. How do we detect and respond to attacks?

The first question is the subject of another course (IST 220, Networking and Telecommunications) in our curriculum. The hands-on labs we described here cover the other three questions on vulnerabilities, threats, prevention, and detection. Whitman, Mattord, and Shackleford (Whitman et al. 2006) also contains many hands-on information security labs.

In the virtual hands-on lab design and development, we actively seek supporting from learning theory and practice. More specifically, we have been exploring *active learning strategies* (Bonwell and Eison 1991) in the curriculum design and teaching. We strive for the students to learn at a deep level with long-term understanding, enabling problem-solving in new contexts. In particular, a *Problem-Based Learning* approach (Albanese and Mitchell 1993; Woods 1994; Boud and Felleti 1991; Barrows 1985; Duch et al. 2001) is taken in many virtual hands-on lab design, to allow students to work together in teams to explore topics and issues beyond traditional textbook approaches to learning. Our experiences show that problem-based approaches can result in substantially better learning than traditional teaching methods.

As we mentioned in the lab descriptions, there are many practical security tools used in the virtual labs, including Wireshark (an open-source packet analyzer), BackTrack 4 virtual machine (a Linux distribution for penetration testing), Nmap ("Network Mapper" or security scanner, an open-source utility for network discovery and security auditing), Metasploit (a penetration testing software that can help verify vulnerabilities and manage security assessments), Nessus (a security vulnerability scanner), BASE (Basic Analysis and Security Engine, an intrusion detection tool), Rijndael inspector (an open-source cryptography learning software), Cain (a password cracking tool), John the Ripper (a password cracking tool), and Ophcrack (a password cracking tool).

Without hands-on labs, it is quite difficult to cover these practical tools in the classroom. By practicing these production-strength security tools, students touch many concepts and theories introduced in the lectures. For example, the theory of public key cryptography is based on some number theory background. Students practice the theory by applying the public key crypto tools to both encryption-decryption and electronic signature verification. This helps students realize the important applications of public key crypto theory on both cryptography to ensure secrecy of communication and e-commerce with digital signature verification.

Another example is password hashing. As passwords should not be stored in clear text in computers, hashed passwords are stored. When a user types in a password, its hash value is computed and compared to the stored hash value. With the password cracking tools used in the lab, students practice the hashing process and are visually exposed to the hashing procedures and results, which enhances their learning experience and connects the abstract theory and concepts to practice.

Virtual labs also eliminate expensive hardware, software, space, and other facilities needed to set up physical labs. Students can complete the labs from anywhere at anytime. The responses from the students who took courses with virtual hands-on labs are quite positive. A student commented in the course evaluation of SRA 221 (the Spring 2013 semester), "the virtual labs … allowed us to apply security concepts [with] popular industry tools." The virtual labs allow students to connect the concepts and theories learned from the lectures to practical skills and get hands-on experience on security tools used in practice.

Acknowledgement The virtual hands-on labs are jointly developed by instructors, instructional technologists, system administrators, student learning assistants, and interns, and currently maintained by the Office of Information Technology in the College of Information Sciences and Technology. We are grateful to the editor Jack Carroll for many helpful comments and suggestions.

References

Albanese, M. A., & Mitchell, S. (1993). Problem-based learning: A review of literature on its outcomes and implementation issues. *Academic Medicine, 68*(1), 52–81

Barrows, H. S. (1985). *How to design a problem-based curriculum for the preclinical years*. New York: Springer.

Bonwell, C. C., & Eison, J. A. (1991). *Active learning: creating excitement in the Classroom*. Washington DC: ASHE-ERIC Higher Education Report No. 1.

Boud, D., & Felleti, G. (1991). *The challenge of problem-based learning*. London: Kogan.

Duch, B. J., Groh, S. E., & Allen, D. E. (Eds). (2001). *The power of problem-based learning: A practical "how to" for teaching undergraduate courses in any discipline*. Stylus Publishing.

Nestler, V. J., Conklin, W. A., White, G. B., & Hirsch, M. P. (2006). *Computer security lab manual. Information assurance & security series*. McGraw-Hill.

Whitman, M. E., Mattord, H. J., & Shackleford, D. M. (2006). *Hands-on information security lab manual. Course technology*. CENGAGE Learning.

Woods, D. R. (1994). *Problem-based learning: how to gain the most from PBL*. Hamilton: Donald R. Woods Publisher.