Proceedings of the 14th Colloquium for Information Systems Security Education
Baltimore Marriott Inner Harbor
Baltimore, Maryland June 7 - 9, 2010

Laboratory Exercises for Wireless Network Attacks and Defenses

Xiaohong Yuan, David Matthews, Omari Wright, Jinsheng Xu and Huiming Yu, *North Carolina A&T State University*

Abstract - With the increase of information security programs and curricula, a number of laboratory experiments or exercises, laboratory-based courseware or courses have been developed for information security education. While most of the existing laboratory exercises/experiments focus on security issues in a wired network, this paper describes a series of laboratory exercises we've developed for demonstrating wireless network attacks and defenses using common open source tools. These laboratory exercises demonstrate the following concepts or methods: wardriving, eavesdropping, WEP key cracking/decryption, Man in the Middle, ARP cache poisoning, MAC spoofing and defense techniques of some of the attacks. The goal of these exercises is to help students learn through exploring, or playing with the real system. These laboratory exercises can be used in an introductory network security or computer networks class. Our classroom experiences of teaching these laboratory exercises are also reported in this paper. Our future work includes exploring more ways to conduct these laboratory exercises effectively, and design comprehensive laboratory exercises that require the students to combine various wireless network attack and defense techniques.

Index terms - wireless network security, network laboratory exercise, WEP, ARP cache poisoning, MAC spoofing

I. INTRODUCTION

With the increase of information security programs and curricula in recent years, the importance of hands-on laboratory components in information security education has been recognized. A number of laboratory experiments or exercises, laboratory-based courseware or courses have been developed for information security education.

Different approaches have been used for hands-on security courses or laboratory exercises. A popular approach is the "cyberwar" exercises. For example, Wagner and Wudi [1] describe a cyberwar laboratory exercise in which students work in teams to secure a computer system and then try to gain access to other systems on the network. Hill et al. [2] describe the use of an isolated network laboratory, in which students work in black or gold teams. Black teams attempt to break into other black team computers or attack the gold team. The gold team operates servers and attempts to defend their servers and role play administrators.

Xiaohong Yuan, Department of Computer Science, North Carolina A&T State Univ., 1601 East Market St., Greensboro, NC 27411.Email: xhyuan@ncat.edu, phone: (336)3347245.

Some laboratory exercises avoid placing students in the role of attacker. Brustoloin [3] describes a sequence of laboratory experiments on network security that cast students successively in the roles of computer user, programmer, and system administrator. In each experiment, the instructor demonstrates an attack, and the students then learn how to use open-source defense tools appropriate for the role they are playing and the attack at hand. O'Leary [4] describes a laboratory-based capstone course that focus on defensive and administrative tools with the goal of teaching potential security officers. Wagner and Phillips [5] present seven modules of hands-on laboratory components taught at their computer security training workshop. One of the modules is a cyberwar exercise, in which the participants harden their systems and are subjected to a series of common attacks by the instructors. There is no attack component in this exercise. In these approaches, students analyze systems, discover their vulnerabilities, exploit the vulnerabilities, and use security tools to enhance the security of a system.

Some approaches let students design and implement systems with security components [6, 7]. Others let students use various security tools so students can learn through playing with a system [8, 9]. Du and Wang [10, 11, 12, 13] have developed a general laboratory environment consisting of Minix and Linux, and a list of labs that cover a wide spectrum of principles. The labs are divided into three classes: (1) The design and implementation labs in which students apply security principles, concepts, and ideas to build security systems in a lab environment. (2) The exploration labs in which students learn by "touching" and "interacting with" the key components of a security system. (3) The vulnerability labs in which students identify vulnerabilities, develop attacks to exploit vulnerabilities, fix the vulnerabilities and defend against the attacks.

While most of the existing laboratory exercises/experiments focus on security issues in a wired network, this paper describes a series of laboratory exercises we've developed for demonstrating wireless network attacks and defenses. These laboratory exercises demonstrate the following concepts or methods: wardriving, eavesdropping, WEP Key Cracking/ Decryption, Man in the Middle, ARP cache poisoning, MAC spoofing and defense techniques of some of the attacks. These laboratory exercises let students use common open source tools to launch attacks or defend the

system. The goal of these exercises is to help students learn through exploring, or playing with the real system. These laboratory exercises can be used in an introductory network security or computer networks class. They are suitable for students with weak programming skills.

These laboratory exercises can be carried out by using laptop computers with Windows XP and wireless capabilities, and routers. The students can be paired up and each pair is given a router to carry out the laboratory exercises. This makes laboratory setup very convenient since the students do not need to go to a special laboratory. Laptop computers with the required software installed and routers can be brought to a regular classroom to carry out these laboratory exercises on wireless network security. Since most computer science or information security students today have their own laptop computers, and routers are very inexpensive, students can also carry out these laboratory exercises using their own laptops and routers outside the classroom. Because of this it may also be feasible to use the laboratory exercises in an online computer security class.

The rest of the paper is organized as follows: In the next three sections, we describe three different types of wireless network attacks and defenses, and their corresponding laboratory exercises. Section V discusses our experiences teaching the laboratory exercises in the classroom. Section VI concludes the paper.

II. WEP CRACKING

Wired Equivalent Privacy (WEP) is part of the IEEE 802.11 standard to provide protection to wireless local area networks. WEP uses the stream cipher RC4 for confidentiality and the CRC-32 checksum for integrity. WEP uses a 64-bit (or 128-bit) encryption key. The key is composed of a 24-bit initialization vector (IV) and a 40-bit (or 104-bit) WEP key. Due to the vulnerabilities of WEP, WEP key can be easily cracked using open source tools. WEP was depreciated in 2004 and was replaced by WPA. WEP exists today to allow legacy devices to communicate with Access Points (APs) [14, 15].

The laboratory exercise on WEP cracking involves wardriving, eavesdropping, cracking the WEP key and using the WEP key to decrypt captured data.

A. Wardriving and Eavesdropping

Wardriving is the process of discovering APs while driving around a city or elsewhere with a portable computer. Open source software such as Cain and Abel [16], KisMAC [17], Kismet [18] and Wireshark [19] can be used to perform wardriving. Wardriving collects and logs such information as the Service Set Identifiers (SSIDs) of the wireless networks, the security protocol used (e.g., WEP, WPA, etc.), the AP's MAC address and a list of clients currently connected to it along with their MAC addresses.

During eavesdropping, the hacker uses software to configure his/her network card onto a certain channel and collects all radio signals within the range of the network card. To monitor radio signals the hacker needs a capture card, a hardware device devoted entirely to collecting data such as AirPcap [20] or a network card configured to promiscuous/monitor mode. The software that can be used to capture data includes Kismet, Wireshark and Airodump (which is part of the Aircrack suite [21]). The data captured by using these tools is usually saved in a file (called a capture file) with the extension .cap. The data in the capture file can be used in a number of attacks later.

B. WEP Key Cracking and Decryption

When sufficient amount of data packets are collected through eavesdropping, the hacker can proceed to crack the WEP key. The IVs in the data packets are needed for WEP key cracking. The number of captured IVs needed to crack the key depends on the AP's key size. For a 64-bit encryption, about 250,000 to 500,000 IVs are needed depending on the complexity of the key. The key consists of ten digits with values of A-F and 0-9. It is possible to crack a key with less IVs but the cracking process will take more time. To crack a 128-bit encryption, about 500,000 to 1 million IVs are needed. Programs capable of cracking a WEP key include KisMAC, Aircrack, and Cain and Abel. Once the key is cracked, the hacker can use the key to decrypt the network traffic using a tool such as Airdecap-ng (part of the Aircrack suite).

ARP (address resolution protocol) request replay attack is often used to generate large quantities of IVs rapidly for WEP key cracking. The attacker uses a program to listen for ARP requests on the network. When an ARP request is captured it is retransmitted back to the network. The computer with the IP address in the ARP request will then send an ARP response to the network. Every time the computer sends an ARP response, it generates a new IV which is captured by the attacker. The attacker repeatedly sends out ARP requests until large quantities of IVs are captured. These IVs are then used to crack WEP key. Cain and Abel can be used to launch ARP Request Replay attack.

C. Laboratory Exercise on WEP Key Cracking

This laboratory exercise demonstrates how WEP key can be cracked, and how the WEP key can be used to decrypt intercepted data using open source tools. It includes four phases: (1) Preparing the environment; (2) Wardriving and eavesdropping; (3) Cracking the key; and (4)

Decrypting network traffic. The required hardware (HW) and software (SW) for this laboratory exercise is listed in Table 1.

Table 1. Required HW and SW for WEP cracking

Hardware	Router (to act as an isolated AP) AirPcap wireless packet capture card Ethernet cable (used to configure router) A laptop computer with Windows XP Internet connection (optional for download of files)
Software	Cain and Abel v4.9.14 Aircrack-ng suite

1. Preparing the Environment

In this phase we configure the key of the access point to allow for a quick cracking process, ensure that AirPcap adapter is functioning, and install software Cain and Abel, and Aircrack-ng. For easy cracking, we select 64-bit encryption and a WEP key of 1111111111.

2. Wardriving and Eavesdropping

In this phase, the information on the AP is first collected through wardriving. Then packets from/to the AP are collected through eavesdropping. ARP request replay attack is used to collect a large amount of IVs in a short period of time. Without using ARP request replay attack, it may take several days to collect sufficient IVs for WEP key cracking, which is not practical for a lab environment. Cain and Abel is used for wardriving and ARP requrest replay attack. Airodump-ng, which is part of the Aircrack-ng suite, is used for eavesdropping.

The procedure for this laboratory exercise is listed below:

(1) Start Cain and Abel and select "Passive Scan". The AirPcap adapter will start to scan various channels to discover APs.
(2) Stop Passive Scan and select the channel the target AP is running on, so the AirPcap will collect packets from/to the target AP.
(3) Check "ARP Requests" to enable ARP request replay attack. The AirPcap adapter will capture ARP requests and then repeatedly resend them to the AP in order to greatly increase the amount of IVs generated. Start Passive Scan again to start the IV generation process.
(4) Start Airodump to begin collecting IVs. For quick key cracking, it is advisable to collect at least 150,000 IVs. This process could take approximately 30 to 90 minutes.
(5) When a sufficient amount of IVs are collected, close Airodump and stop capturing packets. The IVs will be saved to the Capture.ivs file (e.g., C:\Aircrack\aircrack-ng\bin\Capture.ivs if Aircrack was extracted to this path).

3. Cracking the Key

In this step, Aircrack-ng and the captured IV file from the previous step (Capture.ivs) are used to crack the WEP key. Run Aircrack-ng_GUI.exe, and select the file Capture.ivs that contains the captured IVs. Select 64-bit encryption key for the key size then click Launch to start cracking the key. The key will be displayed.

4. Decrypting Network Traffic

In this step, the cracked WEP key is used to decrypt intercepted network traffic into plaintext. Airdecap-ng in the Aircrack-ng suite is used to decrypt the intercepted network traffic. The procedure for this phase is described below:

(1) Run Airodump-ng.exe to start collecting data packets.
(2) Open a web browser to generate network traffic on the AP for decryption.
(3) Close Airodump and stop capturing packets.
(4) Run Aircrack-ng_GUI.exe and select the Airdecap-ng tab. Configure Airdecap-ng to decrypt the captured network traffic file. Enter the WEP key in hex (1111111111), and click Launch to start the decryption process. A command prompt will open and the information of the decryption process will be displayed. The information includes the total number of packets read, the total number of WEP/WPA packets, the number of plaintext data packets and the number of decrypted WEP/WPA packets.
(5) Open the decryption file Network-dec.cap in WordPad to see the network traffic in plaintext. Some encrypted text will still be present. These are beacons and ping replies.

III. ARP CACHE POISONING ATTACK AND DEFENSE

Man in the Middle (MITM) is a category of confidentiality attack. The hacker intercepts packets intended for the AP, and then forwards the packets to the AP or vice versa. The MITM attacker may also modify the data before forwarding it to the destination.

ARP Cache Poisoning is one specific implementation of MITM. ARP is a TCP/IP protocol used to convert an IP address into a physical address, or MAC address. A host wishing to obtain a MAC address broadcasts an ARP request onto the network. The host on the network that has the address in the request then replies with its MAC address. In ARP cache poisoning, the attacker poisons the ARP cache table of a host by sending fake ARP replies to the network. The fake ARP reply maps the attacker computer's MAC address with the victim host's IP

address. The hacker then receives all data destined for the victim host.

XArp [22] is a Windows application intended for users and administrators to detect ARP cache poisoning. Arpwatch [23] is an open source software tool for Linux/Unix environment for monitoring ARP traffic. It generates a log of observed pairing of IP addresses with MAC addresses along with a timestamp when the pairing appeared on the network. It also reports certain changes via email. ArpWatch is most commonly used for detecting ARP cache poisoning. ARP-Guard [24, 25] is a product that protects against unauthorized devices and internal attacks, including ARP cache poisoning attacks.

One way to prevent ARP cache poisoning in a small network is to use static ARP routing, that is, the IP/MAC bindings in the cache can not be changed. One can use the "arp –s" command under the command line interface, or use the tool ARP Freeze [26] to ensure static routing. Dynamic ARP Inspection (DAI) is a feature on some high-end Cisco switches. This feature allows the switch to drop ARP packets with invalid IP/MAC bindings [27] . ArpON [28] is an open source tool for Linux/Unix environment that detects and blocks ARP cache poisoning attacks. ArpOn is based on both the static ARP inspection, and the dynamic ARP inspection methods.

A. Laboratory Exercise on ARP Cache Poisoning and Defense

This laboratory exercise demonstrates how ARP cache poisoning attack can be conducted using Cain & Abel, how password stealing and phishing can be conducted through ARP cache poisoning, how XArp is used to detect ARP cache poisoning attack, and how ARP Freeze is used to prevent ARP cache poisoning attack. This exercise includes two phases: (1) ARP cache poisoning attack and detection; (2) ARP cache poisoning prevention. The required hardware and software for this exercise are listed in Table 2.

Table 2. Required HW and SW for ARP cache poisoning and defense

Hardware	Router (to act as an isolated AP) two laptop computers with Windows XP and wireless capabilities Internet connection
Software	Cain and Abel v4.9.14 ARP Freeze XArp

In this exercise, two laptops are connected to the router wirelessly. One laptop is the attacker's computer, the other one is the victim's computer. Cain & Abel is installed on the attacker computer. More victim computers can be added and connected to the router to conduct this laboratory exercise since Cain & Abel can poison the connections between a router and multiple hosts.

1. ARP Cache Poisoning Attack and Detection

The procedure of this phase is described below:

(1) Run XArp on the **victim's** computer, use the default settings.
(2) Run Cain & Abel on the **attacker's** computer. Configure the Cain & Abel Sniffer so that the wireless card of the attacker computer is selected, and "Don't use promiscuous mode" is checked.
(3) Activate Cain & Abel Sniffer function to discover clients connected to the AP (by clicking on the ARP Poison Routing icon). Press the blue cross icon and select "All Hosts in my subnet" in the MAC Address Scanner dialog box. Two IP addresses should be displayed. One is the router/gateway, the other is the victim's computer.
(4) Click on the APR (ARP Poison Routing) tab at the bottom. Click on the top field and then click on the blue cross icon to open the New ARP Poison Routing dialog.
(5) The first IP address listed to the left should be the router, click on that IP address and then on the right hand side, click on the IP address listed (this is the victim's computer). This means the attacker would intercept the traffic between the selected victim and the router. Click on OK. (see Figure 1). Now the connection between the router and the victim is poisoned. Wait a few moments until the status becomes "Poisoning".
(6) On the **victim's** computer, XArp will display an alert window on the lower right hand corner of the screen, informing the user that ARP cache poisoning attack has occurred.
(7) Open the Command Line prompt on the **victim's** computer and type "arp –a". The user should see an entry that has the IP address of the router and the MAC address of the attacker in the ARP cache.

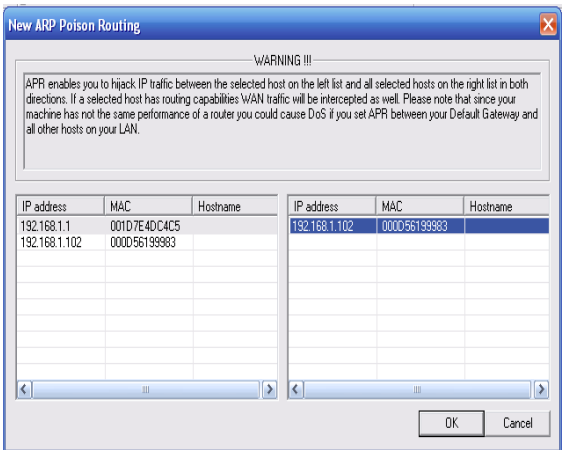

Figure 1. ARP Poison Routing window

The following steps demonstrate how passwords can be stolen as a result of ARP cache poisoning.

(8) On the **victim's** computer, open a web browser and type in the router's address http://<router's IP> (i.e., http://192.168.1.1). Log into the configuration page.
(9) On the attacker's computer, click the Passwords tab at the bottom. Select the HTTP option on the left. The username/password information used by the victim will be shown in the list (Figure 2). This process illustrates how to intercept HTTP passwords but the same procedure can be followed for many different protocols.

The following steps demonstrate how phishing can be conducted as a result of ARP cache poisoning.

(10) On the left panel, the attacker click on "APR-DNS"
(11) Right click on the panel and click on "Add to list". The DNS Spoofer for APR dialog appears (Figure 3).
(12) Under "DNS name requested" type in the URL of the website you want to redirect from (e. g. www.yahoo.com).
(13) Under "IP address to rewrite in response packets", type in the IP address of the fake site (i.e., the IP address of google.com). You can click on the "Resolve" button to find the IP address of a URL. Click on OK.
(14) Now the **victim** opens a browser and goes to www.yahoo.com and actually sees the google.com web page.

2. ARP Cache Poisoning Prevention

To continue from the above exercise, the following steps can be followed to demonstrate the ARP cache poisoning prevention method using static ARP routing.

(15) Close Cain & Abel on the **attacker's** computer. Type "arp –d" on the **victim's** computer to delete the poisoned cache entry.
(16) Open a browser window on the **victim's** computer and go to the router configuration page by typing in the IP address of the router. This will update the victim's ARP cache with the router's IP address again.
(17) Open ARP Freeze on the **victim's** computer. ARP Freeze will scan the current ARP cache and for each entry will ask if you want that entry to become static or not. Click Yes for the entry that has the router (IP gateway) IP address. For all other entries, click No.
(18) Open the command line window again and type "arp –a" to view the ARP cache on the **victim's** computer.
(19) Repeat steps (2) – (5) on the **attacker's** computer to conduct the ARP cache poisoning process again.
(20) Open the command line window and type "arp –a" to view the ARP cache on the **victim's** computer again. Notice that the ARP entry for the router is unchanged. Although Cain and Abel says it's poisoning, the victim was not poisoned and therefore the attack was unsuccessful.

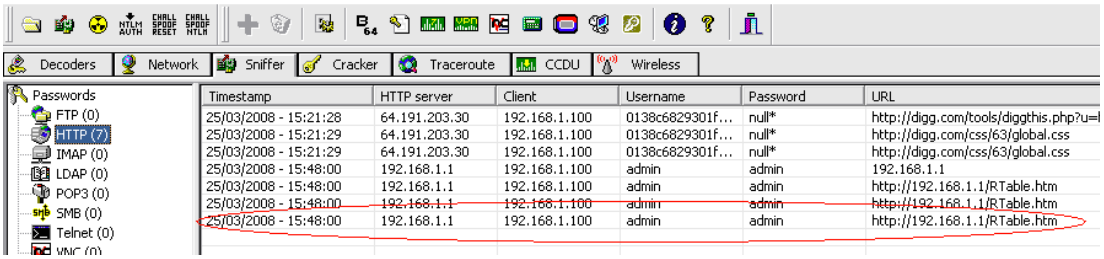

Figure 2. Recovered password in Cain and Abel

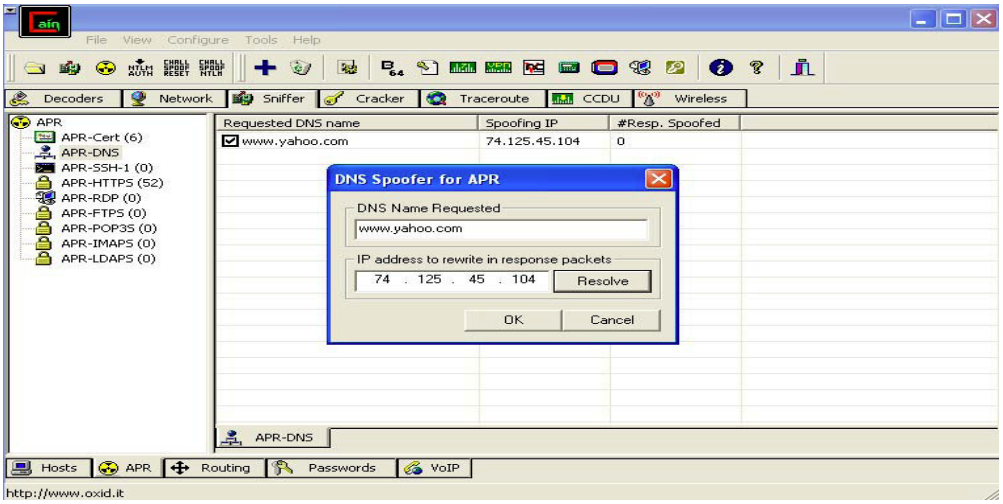

Figure 3. APR-DNS window for redirecting website

IV. MAC SPOOFING AND DEFENSE

MAC spoofing refers to altering the MAC address on a NIC (network interface controller) card. Each network card is shipped from the factory with a unique MAC address. MAC address can be spoofed through hardware or software. The tools that allow MAC spoofing are Mac Makeup and AirJack [29, 30]. A hacker can use MAC spoofing to takeover another computer's identity to enter a target network as an authorized user. To do this, the hacker either disconnects the authorized computer by sending disassociation frames to it or waiting for it to disconnect to assume its identity. MAC spoofing can be used in a Denial of Service (DoS) attack, for example, in authentication or association flood DoS attacks. In an authentication flood DoS attack, the hacker sends association request frames consisting of random MAC addresses in an attempt to flood the AP with login requests.

MAC spoofing can be detected by using reverse ARP. Reverse ARP returns the IP address of a MAC address. If multiple IP addresses have the same MAC address, then a host may be impersonated [31]. ARP based MAC/IP filtering uses a static ARP table to compare the MAC addresses of incoming IP packets to see if they match the stored MAC addresses in the static ARP table. This will help weed out the unauthorized MAC addresses [31]. Sticky ARP is a security feature on the Cisco 7600 series routers to prevent MAC spoofing by ensuring that ARP entries do not get overridden [32].

A. Laboratory exercise on MAC Spoofing and Detection

This laboratory exercise demonstrates how MAC spoofing is conducted through MAC Makeup. To detect MAC spoofing, we use Cain & Abel to scan the network. This method assumes that all the computers have been on before and after the detection of MAC spoofing. The required hardware and software are listed in Table 3.

Table 3. Require HW and SW for MAC Spoofing and Detection

Hardware	Router (to act as an isolated AP) three laptop computer with Windows XP and wireless capabilities
Software	Cain and Abel v4.9.14 MacMakeup

The laboratory setup consists of three computers connected to a wireless router. One computer is the attacker's, one is the victim's and one is the administrator's. The administrator's computer has Cain and Abel installed and the attacker's computer has MAC Makeup installed. The procedure of the laboratory exercise is as below:

(1) On the **attacker's** computer, open the Command Window and type "ipconfig/all". Scroll down to the wireless adapter and write down the name of the adapter and the original MAC (physical) address of the attacker so it can be changed back. Repeat the same thing for the **victim's** computer that is to be spoofed.
(2) Start Cain & Abel on the **administrator's** computer. Configure the Sniffer so the wireless card of the attacker computer is selected, and "Don't use promiscuous mode" is checked.
(3) Click on the sniffer button. Press the blue cross icon to discover clients connected to the AP. There should be 2 computers plus the router listed.
(4) Close Cain and Abel.
(5) Open MAC Makeup on the **attacker's** computer
(6) Click the drop down box and choose the wireless adapter that you wrote down earlier in step 1.
(7) In the MAC address section, enter in the MAC address of the victim's computers. This should be entered without any punctuation.
(8) Ensure that "Auto NIC on/off" is checked and click the Change button. The wireless adapter will shutdown and restart with the victim's MAC address.
(9) Open Cain and Abel again and run another scan on the **administrator's** computer
(10) This time only 2 entries are listed. Since it is assumed that no computer has been turned off, this sudden disappearance of a computer may indicate a MAC spoofing attack.

V. CLASSROOM EXPERIENCES

The above laboratory exercises have been used in several classes in the past. In the Computer Networks class in Spring 2008 semester, the students were able to carry out the following exercises: Wardriving, WEP Key Cracking, WEP Decryption, ARP Cache Poisoning and MAC spoofing attack. Due to the cost of AirPcap card, eavesdropping using AirPcap is only conducted by the instructor in the class as a demo. The students were given a questionnaire survey to assess their overall satisfaction with the laboratories and get their feedback. The student survey results are listed in Table 4. The survey results show that the students had very positive experiences with the laboratory exercises.

The ARP cache poisoning attack and defense, and MAC spoofing attack and defense laboratory exercises were taught in the Network Security class in the Fall 2009 semester. The students were given a questionnaire survey to assess their overall satisfaction with the laboratories and get their feedback. The student survey results are listed in Table 5. Overall the students' feedback was positive.

Table 4. Student Survey Results in Spring 2008 (number of students = 12)

Question	Response
1. Do you enjoy using the tool?	58% strongly agree 42% agree
2. Do you think the lab is easy to follow and straightforward?	8% strongly agree 67% agree 25% neither agreed or disagreed
3. Do you feel you understand the concept better after performing the lab?	25% strongly agreed 58% agree 17% neither agreed or disagreed
4. How likely are you to recommend this tool to others?	50% definitely 33% probably 17% not sure
5. Would you like to see more of these labs (or similar labs) in your courses?	58% strongly agree 42% agree
6. How much do you think you learned from these labs	42% I learned a lot 33% I learned quite a few things 25% I leaned something

Table 5. Student Survey Results in Fall 2009 (number of students = 23)

Question	Responses
1. Did you enjoy the labs?	17% strongly agree 48% agree 26% neither agree or disagree 09% disagree
2. Do you think the labs are easy to follow and straightforward?	13% strongly agree 48% agree 26% neither agree or disagree 09% disagree 04% strongly disagree
3. Do you feel you understand the concepts better after performing the labs?	13% strongly agree 48% agree 30% neither agree or disagree 09% strongly disagree
4. How likely are you to recommend the labs to others?	30% definitely will 30% probably will 5% not sure 05% probably will not
5. Would you like to see these labs (or similar labs) used in your network security classes?	30% strongly agree 52% agree 18% neither agree or disagree
6. Laboratory exercises made me aware of contemporary security threats and what I need to do to counter them.	13% strongly agree 57% agree 19% neither agree or disagree 11% disagree
7. Laboratory exercises helped me to learn how to apply security principles and tools in practice.	17% strongly agree 48% agree 31% neither agree or disagree 04% disagree
8. I believe practical experience in network security is a sought-after skill in the job market.	43% strongly agree 39% agree 04% neither agree or disagree 14% disagree

During both semesters, the laboratory exercises were conducted in a regular classroom using regular class time. Department laptop computers and routers were brought to the classroom, so the students do not need to go to a special lab. This gives flexibility and convenience to the teacher and the students. However, there is the problem of setting up the laboratory ahead of class. If the classroom is used by other classes before the class in which the laboratory exercise is conducted, the instructor does not have enough time to set up the laboratory environment before the class starts (the instructor only has about 10 minutes). Setting up the environment will take class time. Usually the instructor also gives a brief introduction to the laboratory exercise, then the students start the laboratory exercises. For a 50-minute class, there is not enough time for the students to complete the laboratory exercises or comprehend the material covered in the exercises in the class. There is also the problem of not having enough computers for a large class. In Fall 2009, only 10 department computers were available, yet there were 30 students in the class. The students had to work in a team of 3-5 students. This affected the effectiveness of the laboratory exercises. Since most students have their own laptop computers, it may be feasible and more effective to ask the students to conduct these laboratory exercises using their own computers, and assign these laboratory exercises as a group assignment. The student groups are asked to demonstrate the laboratory procedure through classroom presentation. We plan to try this method in the near future. We also plan to design a comprehensive scenario, in which the students combine the techniques in the individual laboratory exercises to achieve an objective. The students have to decide what attacks to launch to achieve the objective, which may be breaking into the system or protecting the system.

VI. CONCLUSION

In recent years, the importance of hands-on experiences in information security education has been realized. This paper describes a series of laboratory exercises we've developed for demonstrating wireless network attacks and defenses. These laboratory exercises demonstrate the following concepts or methods: wardriving, eavesdropping, WEP key cracking/decryption, Man in the Middle, ARP cache poisoning, MAC spoofing and defense techniques for ARP cache poisoning and MAC spoofing. These laboratory exercises let students use common open source tools to launch attacks or defend the system.

These laboratory exercises were conducted in normal classroom with department laptops and routers brought to the classroom. Our classroom experiences show that these laboratory exercises have been well received by the students. We also pointed out the problem of setting up the laboratory environment in order to conduct these laboratory exercises in regular classroom, and the difficulty of managing class time. In the future, we will explore more ways to conduct these laboratory exercises effectively, and design comprehensive laboratory exercises that require the students to combine various wireless network attack and defense techniques.

VII. ACKNOWLEDGEMENT

This work was partially supported by National Science Foundation under the award numbers DUE-0723491 and DUE – 0737304, and by Department of Education under grant P120A090049.

VIII. REFERENCES

[1] Wagner, P. J. and Wudi, J. M. "Designing and implementing a cyberwar laboratory exercise for a computer security course," *Proceedings of SIGCSE'04 - the 35th Technical Symposium on Computer Science Education*, 2004, 402 – 406.
[2] Hill, J. M. D., Carver, C. A., Humphries, J. W. and Pooch U. W. "Using an isolated network laboratory to teach advanced networks and security," *Proceedings of SIGCSE'01 - the 32ndTtechnical Symposium on Computer Science Education*, 2001, 36 – 40.
[3] Brustoloni, J. C. "Laboratory experiments for network security instruction," *ACM Journal on Educational Resources in Computing*, Vol. 6, No. 4, 2006.
[4] O'Leary, M. "A laboratory based capstone course in computer security for undergraduates," *Proceedings of SIGCSE'06 - the 37th technical symposium on Computer science education*, 2006, 2-6.
[5] Wagner, P. J. And Phillips, A. T. "A portable computer security workshop," *ACM Journal on Educational Resources in Computing*, Vol. 6, No. 4, 2006.
[6] Irvine, C. E., Levin, T. E., Nguyen, T. D. and Dinolt, G. W. "The trusted computing exemplar project." *Proceedings of the EEEE Systems Man and Cybernetics Information Assurance Workshop (SMC'04).* West Point, NY, 109-115.
[7] Mitchener, W. G. and Vahdat, A. "A chat room assignment for teaching network security." *Proceedings of SIGCSE'01 - the 32nd Technical Symposium on Computer Science Education*, 2001, ACM press, Charlotte, NC, 31-35.
[8] Ross, K. 2005. CS393/682: Network Security. http://isis.poly.edu/courses/cs393-s2005/.
[9] Bhagyavati, "Laboratory exercises in online information assurance courses," *ACM Journal on Educational Resources in Computing*, Vol. 6, No. 4, 2006.
[10] Du, Wenliang. "Devloping an instructional operating system for computer security education," In *7th Colloquium for Information Systems Security Education (CISSE)*. June 3-5, 2003, Washington DC. 2003.
[11] Du, W., Shang, M., and Xu, Hai. "A novel approach for computer security education using Minix instructional operating system." In *Computer & Security*, Volume 25, Issue 3, 2006. Pages 190-200.
[12] Du, W., Teng, Z., and Wang, R. "SEED: A suite of instructional laboratories for computer security education." In *Proceedings of the SIGCSE Technical Symposium on Computer Science Education*. March 7-10, 2007, Covington, Kentucky, USA.
[13] Wenliang Du and Ronghua Wang, SEED: A Suite of Instructional Laboratories for Computer Security **Education** (Extended Version). In *The ACM Journal on Educational Resources in Computing (JERIC)*, Volume 8, Issue 1, March 2008.
[14] WEP (Wired Equivalency Privacy). http://www.networkworld.com/details/715.html
[15] Attacks on the WEP protocol, http://eprint.iacr.org/2007/471
[16] Cain and Abel v4.9.14. http://www.oxid.it/cain.html.
[17] KisMAC, http://KisMAC.macpirate.ch
[18] Kismet. http://www.kismetwireless.net
[19] Wireshark, http://www.Wireshark.org.
[20] AirPcap, CACE Technologies, http://www.cacetech.com/products/airpcap_family.htm
[21] Aircrack-ng, http://www.aircrack-ng.org/doku.php.
[22] XArp, http://www.chrismc.de/development/xarp/what_is_xarp.html
[23] ARP-Watch, ftp://ftp.ee.lbl.gov/arpwatch.tar.gz
[24]Abad, C.L., Bonilla, R.I. "An Analysis on the Schemes for Detecting and Preventing ARP Cache Poisoning Attacks", *Proceedings of ICDCSW'07 - the 27th International Conference on Distributed Computing Systems Workshops*, 2007 14 Jul. 2009
[25] ARP-Guard, http://www.arp-guard.com
[26] ARP Freeze, http://www.irongeek.com/i.php?page=security/arpfreeze-static-arp-poisoning
[27] Security Features on Switches < Dynamic ARP Inspection (DAI) >, http://www.ciscopress.com/articles/article.asp?p=1181682&seqNum=8
[28] ArpON. http://arpon.sourceforge.net/documentation.html
[29] Mac Makeup, http://www.gorlani.com/publicprj/macmakeup/macmakeup.asp.
[30] Airjack, http://sourceforge.net/projects/airjack.
[31] MAC Spoofing – An Introduction, http://www.giac.org/certified_professionals/practicals/gsec/3199.php
[32]Configuring Sticky ARP, http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SXF/native/configuration/guide/dos.html#wp1139323